APLICACIONES CIENTÍFICAS Y TECNOLÓGICAS DE LAS CIENCIAS COMPUTACIONALES

María del Carmen Santiago Díaz

APLICACIONES CIENTÍFICAS Y TECNOLÓGICAS DE LAS CIENCIAS COMPUTACIONALES

APLICACIONES CIENTÍFICAS Y TECNOLÓGICAS DE LAS CIENCIAS COMPUTACIONALES

María del Carmen Santiago Díaz
Gustavo Trinidad Rubín Linares
Ana Claudia Zenteno Vázquez
Judith Pérez Marcial
(Editores**)**

María del Carmen Santiago Díaz
(Coordinador)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Judith Pérez Marcial
(editores BUAP)

María del Carmen Santiago Díaz
(coordinador BUAP)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Judith Pérez Marcial, Yeiny Romero Hernandez, José Luis Pérez Rendón, Nicolás Quiroz Hernández, Héctor David Ramírez Hernández, Gregorio Trinidad Garcia, María De Lourdes Sandoval Solís, Rogelio González Velázquez, Jose Luis Hernández Ameca, Armando Espíndola Pozos, María Blanca del Carmen Bermudez Juárez, José Martín Estrada Analco, Luz Del Carmen Reyes Garcés, Luis Enrique Colmenares Guillén, Elsa Chavira Martínez, Pedro García Juárez, Nelva Betzabel Espinoza Hernández, Roberto Contreras Juárez, Graciano Cruz Almanza, Maya Carrillo Ruiz, Bárbara Emma Sánchez Rinza, Beatriz Beltrán Martínez, Gabriel Juárez Díaz, José Andrés Vázquez Flores, Ivo Humberto Pineda Torres, Miguel Ángel Peña Azpiri, Hermes Moreno Álvarez, Raúl Antonio Aguilar Vera, Julio Cesar Diaz Mendoza, Francisco Marroquín Gutiérrez, Jéssica Nayeli López Espejel, Eden Belouadah
(revisores)

Primera edición: 2020
ISBN: 978-607-8728-35-0

Montiel & Soriano Editores S.A. de C.V.
15 sur 1103-6 Col. Santiago Puebla, Pue.

Contenido

Prefacio

El avance vertiginoso de la ciencia y la tecnología han generado un gran abanico de soluciones a diversos problemas, sin embargo, en nuestra sociedad nos encontramos inmersos en un sistema que ya no brinda el soporte para una población que ha crecido de forma tan acelerada. Por ello es imprescindible resolver problemas propios de una sociedad en constante crecimiento, a fin de generar mejores condiciones de vida a nuestra población. Aunque hay identificados a nivel nacional y local los problemas que requieren solución inmediata, casi en cualquier área se requiere realizar innovaciones tecnológicas, algunas muy sofisticadas y complejas y otras no tanto, pero finalmente innovaciones, es decir, aplicar ideas y conceptos para solucionar problemas utilizando las ciencias computacionales, que aunque en muchos casos no se requiere una solución que implique años de investigación, si se requiere que la solución esté plenamente enfocada a un problema en particular. Muchas de estas soluciones en general no están a la vista sin embargo, resolver problemas ambientales, de vialidad, de producción de alimentos, etc., representan en sí aplicar la tecnología de forma innovadora ya que aunque en nuestro país tenemos un enorme retraso tecnológico, lo que no se quiere en universidades e institutos de investigación es tener este retraso en la aplicación de la tecnología que se genera, por ello se cuenta con diversos programas internacionales, nacionales y locales para apoyar la innovación tecnológica. En nuestra universidad como en muchas otras en México y en el mundo se cuentan con programas específicos de innovación tecnológica para que laboratorios de investigación generen y apliquen tecnología propia. Pero estos esfuerzos no son suficientes, se requiere de una concientización colectiva para que desde el aula se socialice la necesidad de resolver toda clase de problemas aplicando el conocimiento impartido en clases, y no se debe esperar a una asignatura o catedra de emprendedurismo o innovación, sino desde cualquier tópico que se aborde en ciencias e ingeniería, ya que además de abstraer del mundo real el problema, se deben plantear las diversas alternativas de solución, las implicaciones tecnológicas para llevarlas a cabo, la importancia de la implementación, pero sobre todo, resolver el problema quizá por etapas o versiones hasta llegar a la solución óptima.

María del Carmen Santiago Díaz
Gustavo Trinidad Rubín Linares

Memory Aid Assistant using Deep Learning and Cloud Computing Services

Erick Quezada Mosqueda, Ivo Humberto Pineda Torres, María C. Pérez de Celis Herrero

Benemérita Universidad Autónoma de Puebla, Ciudad Universitaria, Puebla 72570, México
{erick.quezadamos, ivopinedatorres}@gmail.com

Abstract. Memory loss problems are caused by a large variety of factors, and in some cases, they can become a serious and progressive issue that drastically affects the quality of life of the people suffering them. Although there are some systems in the market that aim to assist patients with this conditions, the majority of these are focused on monitoring health conditions, preventing accidents or identifying the symptoms of the disease, rather than assisting the patients to accomplish their daily activities. Therefore, the goal of this work is to present the development of a virtual assistant for people with mild memory impairments, that interacts with the users and that helps them by remembering details of past events. The proposed assistant is composed of a deep learning model that enables it to achieve state-of-the-art performance and it is deployed using cloud computing services as infrastructure.

Keywords: Memory-Aid Assistant, Natural Language Processing, Deep Learning, Cloud Computing.

1 Introduction

Amnesia is the unusual loss of memories that can be events, information or experiences. Usually people with amnesia can identify themselves but have trouble withholding new information. Although memory loss can be temporary, it can also be permanent and become an impediment to carrying out daily activities. Aging can cause certain memory problems, in addition to a slight decrease in other reasoning skills and the ability to learn new information; however, these changes should not be drastic and prevent people from living an independent and productive life [1].

Most people with amnesia have short-term memory problems, meaning they have difficulty retaining new information. The memory loss by itself does not affect the intelligence, reasoning, personality or judgment of the affected person, in fact, people who suffer from it can usually understand written or spoken words, learn new skills and it is even possible that they understand the fact that they have a memory problem [1].

Memory loss can also be a sign of the dementia syndrome, which not only affects memory but also brings with it other cognitive problems such as changes on the thought process, language and behavior of the individuals who suffer from it. Although dementia does not only affect the elder people, aging is one of the greatest risk factors, which

is why it tends to occur more frequently in this type population. Some other risk factors for contracting dementia are hypertension, diabetes, use of tobacco and alcohol, obesity, physical inactivity, low educational level and depression.

The goal of this work is to present the development a virtual assistant adapted for people who suffer from mild memory impairments. Virtual assistants have been gaining popularity in the past couple of years due the variety of tasks that they can solve and the intuitive interaction with the users that they achieve. Therefore, section 2 is dedicated to be a short introduction to task-oriented dialog systems, as well some important concepts of these systems.

In order to achieve its main functionality, the proposed assistant uses artificial intelligence to enhance its capabilities, specifically, using state-of-the-art deep learning models to do natural language processing tasks. Section 3 is dedicated to give the reader a general overview of natural language processing and how this field of research is integrated with deep learning to solve challenging tasks. In addition, a quick summary of the latest improvement in deep natural language processing is given, as well as an explanation of the deep learning model used during the development process.

In section 4, the design concepts and architecture of the virtual assistant are presented. This section is dedicated to present the proposed interaction between the memory assistant and the end users, as well as a detailed description of the components that were employed for its deployment, mainly, the training and deployment process of the deep learning model and the cloud computing services that were used.

Finally, we will present some of the future work that is needed in order to have a fully functional prototype as well as some closing thoughts on the current development of the project.

2 Task-oriented dialog systems

A task-oriented dialog system is a computational system that is capable of communicating with the humans via text, voice or other media. They are also known as conversational agents and they are currently being used for solving a variety of tasks, such as sending text messages, playing songs, making phone calls or doing online shopping. Some commercial examples of these kind of dialog systems are “Assistant” from Google, “Alexa” from Amazon and “Siri” from Apple.

In general, a task-oriented dialog system is able to resolve a number of tasks for which it was previously programmed. Therefore, the first thing that a conversational agent needs to learn is to recognize, which task the user is requesting, this is known as *intent classification*. As a more formal definition, an intent classification task consist of, given a statement by some end-user, the system must identify which of its predefined scenarios is trying to execute the user [2].

Along with intent classification, a task-oriented dialog system should be able to resolve a *slot filling* task (also referred as *slot tagging*). In this task, once the intent has been properly identified, the dialog system needs to gather from the user all the information needed to fulfill such intent [2]. As an example, if the users ask the dialog system to play a song on some device, the system will classify the intent as “play music”, but it still needs to know more information like the name of the song to play or the name of

the artist that interprets it. In this scenario, the conversational agent needs to communicate back to the user asking for the missing information.
Building all the components needed to deploy a functional conversational agent is a complex task; fortunately, there are many platforms available to develop these systems. For the conversational agent proposed in this work, we decided to use Google's Dialogflow, which is a powerful framework that requires only a few example sentences to train an agent to correctly identify an intent. Dialogflow also offers an intuitive and very complete user interface for developers, as well as many possible built-in integrations in order to place the trained agent on a production environment.

3 Deep natural language processing

3.1 Natural language processing

Goldberg & Hirst [3] describe Natural Language Processing or NLP as "a collective term referring to automatic computational processing of human languages. This includes both algorithms that take human-produced text as input, and algorithms that produce natural looking text as outputs". NLP is a very broad area of research since it cover any computational manipulation of natural language, which can be a simple task such a counting the frequency in which certain words appear in a text, or a much more complex task such as identifying the sentiment associated with some sentence (sentiment analysis) [4].
Natural Language Processing is a challenging field of research, since the words in a text have very complex non-linear relationships with each other, which makes it very difficult to model and capture their information correctly. In addition, each language has its own grammar, syntax and vocabulary, which mean that both technological advances and available resources are linked to specific languages [5].
Some common NLP task are tokenization, named entity recognition, part-of-speech tagging, sentence classification, question answering and machine translation. There are three main approaches for solving NLP tasks: using rule-based methods, using machine learning techniques and using deep learning; the last being the one that has recently obtained high performance across multiple NLP tasks.

3.2 Deep learning

Deep Learning is a subfield of artificial intelligence and machine learning, composed of algorithms vaguely inspired by the structure and behavior of the human brain, where data representations are sought through a multi-layered structure of algorithms known as Artificial Neural Networks or ANN. Similar to machine learning, the training of a deep learning model is an iterative process, for which it is necessary to have a large amount of data related to the problem to be solved, and divide it into training and test set.

Artificial neural networks are computational systems that, although they are not considered mathematical models of their biological counterparts, they do have two main

similarities. First, the building blocks of both networks are simple computational devices called "neurons" that are highly interconnected; second, the connections between neurons determine the behavior of the entire network [6].
There are different types of deep learning models depending on the architecture of the neural network; one of the most common types is known as Feed Forward Neural Network (FFNN). This kind of network has an input layer where the initial conditions of the input data are represented; an output layer with representations of the categories or output values of the model; and one or more layers between the input and the output layer, known as "hidden layers". These hidden layers allow the model to solve non-linear problems by obtaining a greater number of representations of the data, and by applying a non-linear function, known as "activation function", to the input data.
Besides the FFNN there are other common types of deep learning architectures used to solve different kinds of tasks. For example, the Convolutional Neural Networks or CNN are widely used in computer vision, and the Recurrent Neural Networks or RNN are commonly used for time series analysis.

3.3 Attention mechanism and transformer networks

Recurrent neural networks and LSTM networks (a special type of RNN), were the bases of more complex models in the field of deep learning and natural language processing. An example of such models is the *seq2seq* model, which demonstrated a very effective approach to solving machine translation and language modeling tasks.
The seq2seq model is composed of an encoder-decoder architecture, where the function of the encoder is to compress the information of the input sequence into a fixed-dimension vector called *context vector*, which will save (in theory) a summary of the entire input sequence and will serve as the input of the decoder. The function of the decoder is to produce an output sequence based on the information in the context vector [7]. Examples of the use of the seq2seq model are to translate a sentence from one language to another or to create a chatbot that responds to an input sentence.
Thanks to their ability to process text sequences, the seq2seq models presented a great advance in the field of deep natural language processing; however, they presented problems when dealing with very long input sequences. That is why, with the purpose from helping these models to memorize the context of long input sequences of text, the concept of *attention* was proposed.
The attention mechanism was design as a method to improve the results in machine translation, and was initially introduced by Bahdanau, Cho & Bengio [8]; and Luong, Pham & Manning [9]. The attention mechanism proposes that, in an encoder-decoder architecture such as seq2seq, weights are used between the context vector and each of the input values, rather than using a single fixed-length context vector. These weights vary depending on the element of the sequence that the decoder is trying to generate and are an indicator of which elements of the input are more important to generate such output.

3.4 BERT model

Deep learning has proven to be a powerful tool for solving many different tasks and for NLP was no exception. BERT [10], which stands for Bidirectional Representations

from Transformers, is a language representation model introduced by Google in 2018 that uses a deep architecture using Transformer networks that obtained state-of-the-art results in eleven NLP tasks at the time it was presented.
BERT is an important milestone in the NLP filed because it doesn't only achieves amazing results in different NLP tasks, it also attempts to resolve two big issues of the Deep NLP models: the lack of available data in languages others than English and the fact that different models where needed to train a deep learning model for different task.
In the same way as its predecessor, ELMo [11], BERT uses context-dependent word representations with the improvement that this representations are bidirectional, this means that the model takes into account both the left and the right contexts in order to make a word representation.
The BERT framework has two steps: pre-training and fine-tuning. The pre-training step consists in training the model with unlabeled data using two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). A BERT model cannot be trained as the standard left-to-right and right-to-left models given its multi-layered bidirectional architecture that would allow it to trivially predict a target word, therefore in the MLM Task the model masks a percentage of the inputs tokens at random and tries to predict what word corresponds to those masked tokens.
Some important NLP tasks such as Question Answering require that the model understands some correlations between sentences and this is achieved in the BERT model via the NSP Task. In this task the model takes in every training example two sentences A and B and predicts whether or not the sentence B follows sentence A. During this task, 50% of the time the sentence B is indeed following sentence A and the other 50% of the time is a random sentence from the training corpus.
Thanks to the self-attention mechanism of the Transformer networks, it is possible to use the same model architecture (apart from the output layers) for both steps: pre-training and fine-tuning. The fine-tuning step consist in swapping the pre-trained inputs and outputs for the task-specific ones and then fine-tune the weights from all the parameters of the model. Task-specific details can be found in the original BERTs paper.

3.5 ALBERT model

ALBERT stands for "A Lite BERT", and as its name implies is a BERT based model but with an architecture that has significantly fewer parameters and that offers better performance than the original BERT.

Similar to BERT, ALBERT has different configurations and according to the original paper "an ALBERT configuration similar to BERT-large has 18x fewer parameters and can be trained about 1.7x faster" [12]. As a comparison a BERT-large model has 334M parameters while an ALBERT-large model has only 18M parameters.

In order to achieve this parameter reduction without significantly hurting performance the ALBERT model incorporates two parameter reduction techniques: factorized embedding parameterization and cross-layer parameter sharing.
The ALBERT model also changes the NSP loss from the original BERT to a sentence-order prediction (SOP) loss, and as a result, the model improves consistently its performance in the downstream tasks. A full comparison between ALBERT and other BERT-based models (including BERT itself) can be found in the original paper.

4 Design and architecture of the memory-aid assistant

In this section, we present the details of the architecture and design of the proposed memory-aid assistant for people with mild memory impairments. Although the idea of retrieving someone's memories is very complex, we propose a simple, yet powerful idea to help the people that suffer from memory loss problems the possibility to retrieve information from past events via a conversational agent and a question answering system.
The goal of the virtual assistant is to help the users to remember events that happened previously in their life, but cannot be remembered properly; therefore, we propose the following interaction (illustrated in Fig. 1):

1. The user or a caregiver asks the assistant to register an event by narrating it to the assistant.
2. The assistant will identify the intention of the user and will confirm the registration of the event.
3. When the user forgets some detail about a past event, he or she can ask the conversational agent a question, which will proceed to read all the previously stored events in its memory and then try to answer the question of the user based on the stored information.
4. A response will be presented to the user, which will be either the answer to the question or a message indicating that the assistant does not have enough information to answer that question.

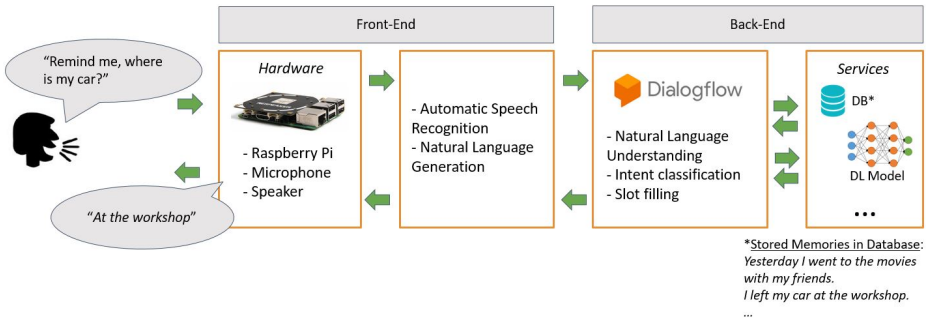


Fig. 1. Proposed interaction between the user and the virtual assistant. The user ask a question related to a past event that was previously registered, and the conversational agent emit a correct response.

In order to fulfill the previous scenario the conversational agent needs to be able to access at least two services: a database to store and fetch the past events of the user and

a Question Answering (QA) system able to solve this task with the best possible performance.

The QA Task is one of the different NLP Tasks in which deep learning models have excelled, in particular, the BERT model was able to achieve state-of-the-art results using the SQuAD (Stanford Question Answering) dataset [13] as a benchmark; therefore, we decided to use a this type of model to achieve this task. In addition, in order to maintain the conversation with the end-user fluid and make it feel more natural, the conversational agent needs to be able to make fast predictions. In order to achieve this requirement, while maintaining state of the art results in the QA task, the use of the ALBERT (A Lite BERT) model was proposed.

Likewise to BERT, ALBERT has both a pre-training step where it learns the language modeling task and a fine-tuning step where it is trained in a specific downstream task. For the English language, a pre-trained ALBERT model in English can be downloaded from the official project's Github repository; therefore, we can focus on the fine-tuning step. In the original paper, to solve the QA task, ALBERT was trained with the SQuAD 1.1 and 2.0 datasets.

Both dataset are composed of paragraphs containing information of different topics extracted from Wikipedia and questions related to those paragraphs. The difference between both datasets is that the 2.0 version includes more than 50,000 question with no possible answers. The purpose of introducing these types of questions is that the model learns to identify when it does not have enough information to answer and abstains to give a response, rather than giving a random answer [14]. The ALBERT model used for this work was fine-tuned using the deep learning library Tensorflow and was trained with the SQuAD 2.0 Dataset.

Since we decided to feed the model with a structure similar to the one presented on the SQuAD 2.0 dataset, no further modifications to the dataset were needed. The structure of the queries that the conversational agent will process have two parts; first, a paragraph, populated with the events stored in the database; second, the question that we need the model to answer. It is important to notice that the version 2.0 of the SQuAD Dataset is being used, because the user may ask questions where the information has not been properly registered previously, and in that case, we expect the model to return an empty answer. In this scenario, the conversational agent will recognize that it does not have enough information to respond and then will ask the user to talk to a caregiver.

Once the model was fine-tuned with the SQuAD 2.0 dataset, some optimizations needed to be done to ALBERT's original code in order to integrate it with the assistant. First we needed to change how ALBERT reads its inputs and writes its output predictions. Its original Tensorflow code is based on reading an input file and output its results into an output file, therefore we modified the code for it to accept inputs as strings and write its output to variables instead of .txt files.

The second change was to make the necessary modifications for ALBERT to serve as a REST API. For this implementation, we used Google's App Engine, a cloud computing service used for hosting applications. App Engine is part of the Google Cloud Platform and enable the developers to quickly deploy applications without investing time in managing a correct infrastructure and configuring the network, thus the developers can focus only on the code correct execution. App Engine also offers automatic scaling, therefore is ideal for taking an app to a production environment.

In order to enable our ALBERT model to be served in App Engine as a REST API we used Flask, a python micro web development framework. The App Engine official documentation indicates that we can use Flask to develop our API and once the code runs in App Engine it will scale properly according to the app needs.
Regarding the database of the project, a NoSQL database was proposed given the need to execute fast queries and the fact that NoSQL databases have great synergy with applications communicating in JSON format. Since Google's Dialogflow is the platform where the agent is being developed, we decided to use, along with other Google Cloud Services, the Google's Firebase platform to fulfill the needs of the assistant. Therefore, the chosen database for the project was Google's Firestore, a flexible and scalable NoSQL database that is part of the Firebase development platform and that has an easy integration with our Dialogflow project.
At this point of the development, the conversational agent is composed of three different systems (Dialogflow, a Firestore database and the ALBERT model) that currently work independently from each other but that need to be connected in order to fulfill the user's requests. In order to achieve this task we use another Google Cloud service called Cloud Functions, this cloud service is also part of the Firebase platform and allow us to run code in a cloud server enabling us to create complex applications by interconnecting other existing Google Cloud services as well as third-party services
A diagram showing the proposed architecture and the back-end components of the assistant can be seen in Fig. 2.

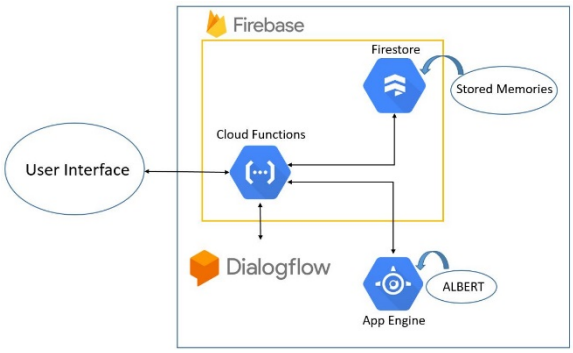


Fig. 2. Details of the back-end architecture of the virtual assistant.

It is worth noticing that in this work only the back-end of the conversational agent is being presented, and even though the proposed way of interacting with the end-users is detailed as future work, it is possible to verify the behavior of the assistant thanks the web demo integration that offers the Dialogflow platform. Some testing results are shown in Fig. 3, where one scenario of interaction between the memory assistant and the end users is being presented. Just like in the case of all the commercial and custom virtual assistants, more scenarios can be programmed in order to have more powerful and complete agent.

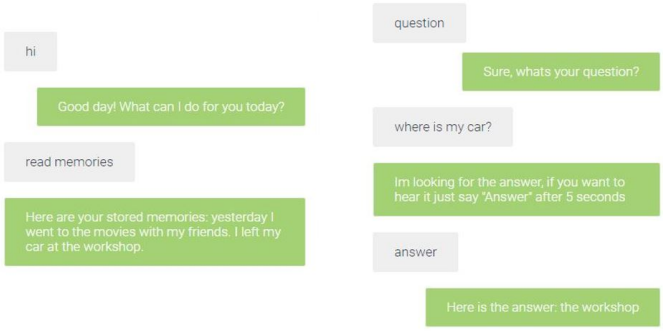


Fig. 3. Example of a conversation with the virtual assistant using the Web Demo Integration from Dialogflow.

5 Conclusions and future work

In this work, we were able to integrate state of the art techniques in order to make a functional and robust virtual assistant for people with mild memory impairments. The recent advances in deep NLP allowed us to create a high performance question answering system that, with some modifications and adjustments specific to our application, serves as core element of the virtual assistant and enables a way to help the users to retrieve information of past event that they may have forgotten.

During the development process of this work the availability of different cloud computing services in the market took an important role, given that we were able to focus more on the development side of the project allowing the cloud service provider to handle all the infrastructure and configuration of the servers.

Given that the majority of end-users of the assistant are most likely to be elder people, we propose as future work the use voice commands as user interface to communicate with the assistant, as well as receiving voice-based responses.

Google's Dialogflow, offers a lot of quick integration options for text based platforms like Slack and Facebook Messenger, it also offers a Web Demo to show how would the assistant would look and behave in a website. In order to use voice integration Dialogflow offers a quick integration to Google Assistant, this offers the opportunity to extend Google Assistant functionalities with the new ones that your custom agent provides, making your assistant available in all devices that support Google Assistant. Even though this is a quick and effective integration, we want to simplify the interaction of the assistant as well as the amount of intents available since we are only interested in those related to memory issues; therefore, we want to use a custom hardware to interact with the assistant.

Our hardware proposal is to use a Raspberry Pi with a Respeaker 4-Mic Array, which is a low cost microphone array used to develop custom voice assistants, as well a mini-speaker to hear the assistant responses. In order to achieve the communication to the back-end a speech recognition module is needed to convert the voice commands of the user to text messages. A text to speech system is also needed in order to produce voice

responses to the end-users. Google Cloud Services as well as other cloud service provides like AWS offer both of this modules and can be as well integrated to the project.

References

1. Mayo Clinic: Amnesia. *Mayo Clinic*. https://www.mayoclinic.org/diseases-conditions/amnesia/symptoms-causes/syc-20353360 (2017). Acceded on April 10, 2020
2. Potapenko, A.; Zimovnov, A.; Kozlova, A.; Zobnin, A.; Yudin, S.: Natural language processing. *Coursera*. https://www.coursera.org/learn/language-processing (2017). Acceded on January 18, 2020
3. Goldberg, Y.: Introduction. Hirst, G. (Ed): *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, pp. 1-4 (2017)
4. Bird, S.; Klein, E.; Loper, E.: Language Processing and Python. Steele, J.: *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., pp. 1-10 (2009)
5. Ganegedara, T.: Introduction to Natural Language Processing. Pohlman, F.; Atitkar, R.; Nelson, C.; Rai, B.; Jacob, T. (Eds): *Natural Language Processing with TensorFlow: Teach language to machines using Python's deep learning library*. Packt Publishing Ltd., pp. 1-15 (2018)
6. Park, Y. S.; Lek, S.: Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling. *Developments in environmental modelling* (Vol. 28). Elsevier, pp. pp. 123-140. (2016)
7. Sutskever, I.; Vinyals, O.; Le, Q.V.: Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, pp. 3104-3112 (2014)
8. Bahdanau, D.; Cho, K.; Bengio, Y.: Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014)
9. Luong, M.T.; Pham, H.; Manning, C. D.: Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015)
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018)
11. Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L.: Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018)
12. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019)
13. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016)
14. Rajpurkar, P.; Jia, R.; Liang, P.: Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018)

Object Detection with Cameras vs LIDAR Sensors in Autonomous Vehicles

Fernando Rojas Ramos, Ivo Humberto Pineda, Mario Rossainz Lopez

[1] Benemérita Universidad Autónoma de Puebla, Ciudad Universitaria, Puebla 72570, México
[1{]toshk1515, ivopinedatorres, mrossainzl}@gmail.com

Abstract. Research centers and companies dedicated to the development of autonomous vehicles are opting for two trends: using only cameras as a vision system or using LIDAR sensors plus cameras. Tesla has a fully camera vision system and Waymo has a cameras and LIDAR system. The difference of the LIDAR sensor could prevent accidents and save human lives in the future.
Therefore, the main contribution of this work is the design of a methodology based on the comparison of detection efficiency for vision devices (Cameras and LIDAR). Applying measurement parameters provided by Neural Networks and models evaluation metrics in machine learning; it has to be concluded if it's necessary to use LIDAR sensors in the development of autonomous cars.

Keywords: Autonomous Vehicles, Vision System, Neural Networks, Evaluation metrics, Machine Learning.

1 Introduction

As one of the most exciting engineering projects in the modern world, autonomous driving has been an aspiration for many researchers and engineers from generation to generation. This is a goal that could fundamentally redefine the future of human society and everyone's everyday life. Once autonomous driving is in full maturity, we will witness a transformation of public transport, infrastructure and appearance of our cities that contribute to a better standard of living.
One of the greatest expectations of using artificial intelligence in the medium term is the development of autonomous cars that improve the mobility of intense traffic in cities, to avoid car accidents and to reduce pollution. The world hopes to exploit autonomous driving to reduce traffic accidents caused by driver errors, to save time to reach our destination and take advantage of time by executing other productive activity, as well as to save parking spaces, especially in urban areas.

Car accidents are the eighth leading cause of death worldwide, with 95% being caused by human error; the expectation is that the automation of transport represents a significant reduction in the number of occurrences and mainly of victims.

Nowadays many current image based object detectors using convolutional neural networks exhibit excellent performance on existing datasets such as KITTI [6]. This data set will be used to carry out the study with images and real data from LIDAR sensors.

Many companies and research centers dedicated to the development of the autonomy of cars are divided into using only cameras and radars or LIDAR sensors in fusion with cameras and radars. In order to opt for cars with a complete vision system, this work reviews both devices by comparing their detection results using artificial intelligence tools.

Tesla has been heavily relying on vision and going against LIDAR sensors. Simultaneously, all the other companies use LIDAR and seem to dismiss other options. The most apparent reason for Tesla has taken a different route is the cost. The cost of placing a single LIDAR device on a car is somewhere around $10,000. Google with its Waymo project has been able to slightly decrease the number by introducing mass production. However, the cost is still rather significant.

The purpose is to compare the performance in the detection of objects separately in order to determine if the LIDAR sensors are essential in the development of vision prototypes. It is proposed to implement and design a CNN-type neural network architecture model based on camera images and LIDAR sensor data.

The main contributions of this work is:

The development of a result evaluation-based methodology to compare the detection efficiency of both devices (Cameras and LIDAR). According to the measurement parameters provided by neural networks and model evaluation metrics in machine learning.

This paper is organized as follows: Section 2 gives a brief analysis of the most related papers. Section 3 provides an overview of the CNN model used in this work. Section 4 proposes methodology to analyze the performance of the built detectors. Finally, the conclusions and future work in Section 5.

2 Related works

There are a variety of studies aimed at the autonomy of automobiles and, in particular, at devices that function as a means of vision-in this case LIDAR sensors and Cameras.

The studies related to this project are focused on examining the detection of objects separately, that is, the results are based on the detection of objects using only cameras [8][9][10][11] or LIDAR sensors [14][15][16][17]. In this project, the two trends are covered to carry out a methodology for comparing the results. This serves to determine which device provides the best detection.

Manuel Herzog and Klaus Dietmayer [1], propose in their work the detection of objects with LIDAR sensors in driverless cars and by applying the strategy of training a model using data from different types of LIDAR sensors. In comparison to the authors Simon Chadwick, Will Maddern, and Paul Newman [3] their study consists of the detection of objects with cameras and the calculation of distance using radar. Their work uses the KITTI dataset, which has different types of data from sensors. It is also important to mention the use of Deep learning for the detection of objects. Deep convolutional neural networks (CNN) represent the ultimate in object detection and have proven to work well in a variety of different scenarios. Nonetheless, they also struggled to detect with precision small objects.

The focus of Jyothish K.James'work is [5] the LIDAR sensors. The performance of the LIDAR sensors has become significantly important to guarantee their reliability

and, therefore, to guarantee the vehicle's safety. United Nations underestimate them with the performance of the sensor to detect the objects reliably. Nevertheless, in extreme weather conditions, performance is heavily affected by tampering with the sensor face plate. They focused on classifying different types of contamination using deep learning; they trained a deep neural network (DNN) following a multiple view concept for the generation of training and test data.
Experiments have been conducted in which the front plate of a LIDAR sensor has been artificially contaminated with aiming at the verification of the efficiency of the sensor to do its job with the same results.

Di Feng [4]: He tries to summarize systematically the methodologies in his article and to discuss the challenges for deep multimodal object detection and semantic segmentation in autonomous driving. To this end, they first provided an overview of sensors embedded in test vehicles, open datasets, and basic information for object detection and semantic segmentation in autonomous driving research.

The technological trend in using sensor fusion for the development of autonomous cars is of great interest and study for the coming years because it involves the integration of all data from radars, LIDAR, cameras, GPS, etc.
Wangs' work [2] is based on the use of stereo cameras for the detection of objects, these images have the advantage of showing more information than the images of monocular cameras; they use depth in image scenes to locate how far away the detected object is as with LIDAR sensors. LIDARs are more expensive than stereo cameras for cost reasons. With these images they are able to simulate the operation of the LIDAR sensors. However, they also prove that the fusion of the images from the cameras plus the point cloud of the LIDARs obtain better detection results.
The article proposes as future work to compare the processing times of the data from both detection devices.

3 Model Deep Neural Network

The U-Net was developed by Olaf Ronneberger [7] for the segmentation of biomedical images. The architecture contains two paths. The first path is the shrink path (also called the encoder) that is used to capture the context in the image. The encoder is just a traditional stack of maximum grouping and convolutional layers. The second path is the symmetric expansion path (also called a decoder) that is used to allow precise localization by transposed convolutions. So it is a fully convolutional end-to-end network (FCN), that is, it only contains convolutional layers and does not contain any dense layers due to which it can accept images of any size. This project analyzes the segmentation, location and detection of objects with very effective results.

The architecture of the U-net convolutional network is described below (see Fig. 1).

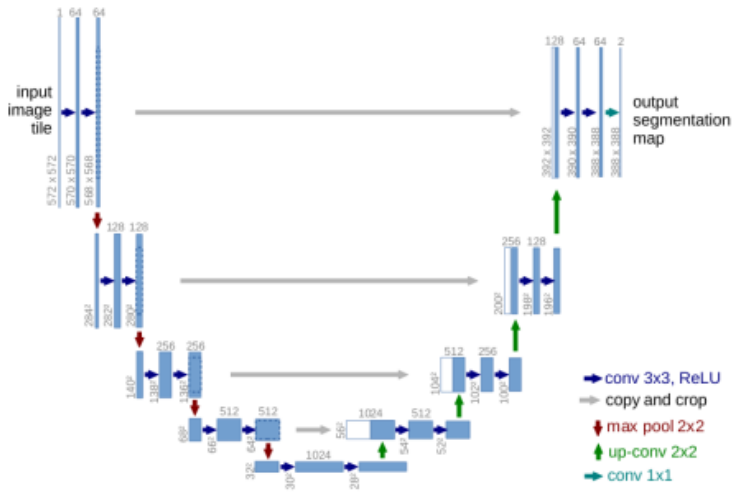


Fig. 4. U-net Convolutional Neural Network.

- Convolution and grouping operations shrink the image, that is, convert a high-resolution image to a low-resolution one
- The Max Pooling operation helps to understand "WHAT" is in the image by increasing the receptive field. However, it tends to lose the information of "WHERE" the objects are.
- In semantic segmentation, it is not only important to know “WHAT” is present in the image, but it is also important to know “WHERE” it is present. Therefore, we need a way to scale up the image from low resolution to high resolution that will help us restore the “WHERE” information.
- Transposed convolution is the most preferred option for sampling, which basically learns the parameters through reverse propagation to convert a low resolution image to a high resolution one.

4 Proposed Methodology

In order to solve the problem of object detection using deep learning, the methodology shown in Fig. 2 is proposed. A convolutional neural network called U-net is used for the segmentation of labeled objects for their subsequent detection in a bounding box.

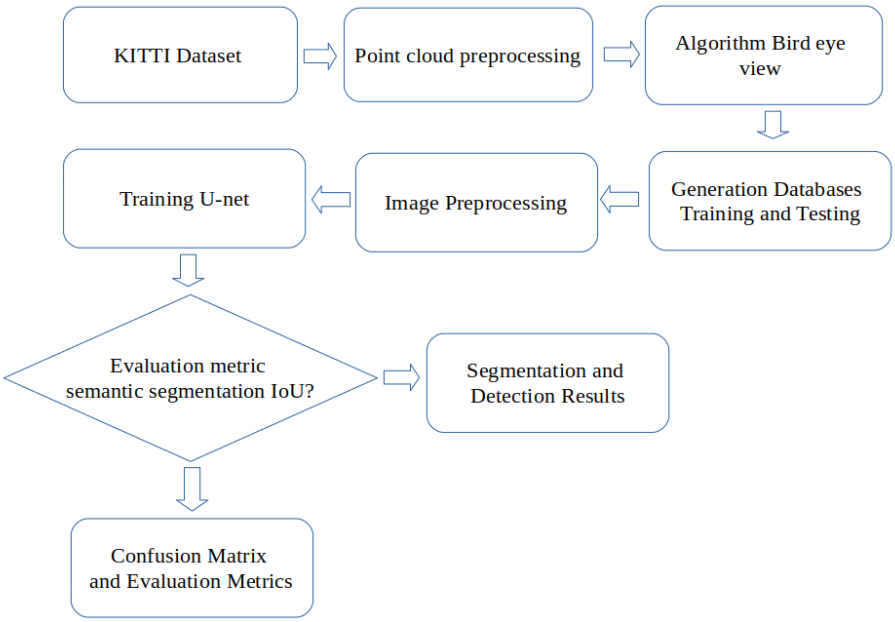


Fig. 2. Proposed methodology for comparing results between Camera vs LIDAR.

4.1 KITTI Dataset

KITTI [6] is a dataset available to carry out the study with real data of a prototype autonomous vehicle. The vehicle is equipped with four cameras: 1 stereo pair of color cameras and 1 stereo pair of grayscale cameras. The color and grayscale cameras are mounted close to each other (~6 cm), the baseline of both stereo decks is approximately 54 cm. This configuration allows you to obtain information in both color and grayscale from the left and right camera. While color cameras (obviously) come with color information, grayscale camera images have higher contrast and slightly less noise.

All cameras are clocked at approximately 10 Hz relative to the Velodyne laser scanner. The trigger is mounted in such a way that the images from the camera roughly coincide with Velodyne lasers facing forward (in the driving direction).

All camera images are provided as lossless compressed and rectified png sequences. Native image resolution is 1382x512 pixels and slightly lower after rectification.

The classes available in the KITTI dataset are 8 and the number of instances are the following, Car = 28614, cyclist = 612, Misc = 959, Pedestrian = 4448, Person sitting = 220, Tram = 504, Truck = 1079, Van = 2900.

4.2 Point cloud preprocessing

In this section, a brief description of the two approaches to work with the point cloud of the LIDAR sensors in order to train the U-net neural network with other types of data not coming from a camera.

Point clouds are a collection of points that represent a 3D shape or feature. Each point has its own set of X, Y and Z coordinates and in some cases additional attributes.

Nowadays, object detection systems can be divided into two main categories [19]. The first ones are the geometric based, which retrieve the obstacles using geometric and morphological operations on the 3D points. The seconds are the deep learning based, which process the 3D points, or an elaboration of the 3D point cloud, with deep learning techniques to retrieve a set of obstacles.

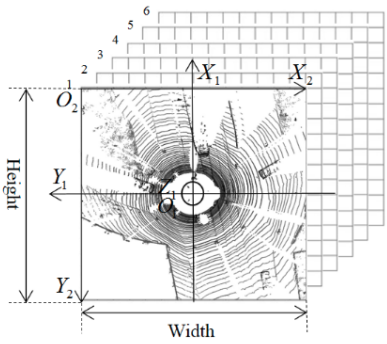
This work is focused on the second approximation: projection based methods implement a single or multi-view projection of a 3D point cloud, resulting in a 2D grid, which is then processed to find object clusters with the desired confidence. Afterwards, this grid is processed by is then processed by a 2D Convolution neural network.

4.3 Bird eye view algorithm

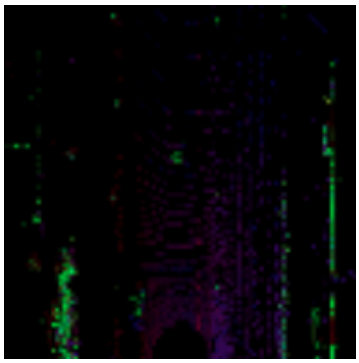
The perspective transform that interests us is a bird's-eye view transform [18] that let's us view a lane from above. Aside from creating a bird's eye view representation of an image, a perspective transform can also be used for all kinds of different viewpoints.

Channel feature extraction [19]: the 3D point-cloud provided by the LIDAR is projected in a BEV image Fig. 3b with pre-determined width, height and grid cell size. To avoid loss of information during the projection of 3D point-cloud into a 2D image, 6 additional channels shown in Fig. 3a are stacked together the new pattern to recover information about the peak and the medium values of height, intensity and distribution of the collapsed points for each cell. Moreover, binary information concerning the effective occupancy of each grid are included.

The image on the right side represents the bird eye view transformation of a sweep of the point cloud available in the KITTI dataset for the generation of the images corresponding to training and testing database.



(a) Bird eye view of point cloud with 6 channels features.



(b) Sample of bird eye view images of real point cloud data from a real Velodyne 3D LIDAR from KITTI dataset.

Fig. 3. Perspective Bird Eye View.

4.4 Generation Databases Training and Testing

This KITTI dataset is totally unbalanced in instances per class. The database was generated with a single class "Car" and the number of instances are as follows: Car = 28614.
The dimensions of the camera images is (375, 1242, 3) and image bird eye view from LIDAR is (480, 480, 6).

This is to evaluate and compare the performance of the two U-net neural network models, in accordance with their segmentation and detection predictions. The following databases are proposed only with the Car class, the training set is composed of 7381 images and 100 images for the testing. The KITTI dataset has a test set similar in number of images to the training set; however, it does not have the calibration files to perform the perspective transformation of the LIDAR sensor point cloud. It is for this reason the distribution of the databases for this project see Table 1.

Table 1. Training and Testing databases for the Car class with images from Camera.

Databases	Number of classes	Number of images	Number of Car instances
Training Dataset 1	1	7381	28460
Testing Dataset 1	1	100	154

The bird eye view projection is limited by the resolution of the LIDAR sensor, in this work the following dimensions are taken side range -25m to 25m and forward range 0 to 40m, that is, for the objects in an instant of time included with the image and the labels objects further away at that distance do not fall within the range of the LIDAR sensor. If the bird eye view projection is generated with larger dimensions, object masks are generated that to not exit information from the point cloud of the LIDAR sensors, which affects the learning period of the deep neural network generating over fitting and therefore bad predictions. Training and testing database for LIDAR sensor data see Table 2.

Table 2. Training and Testing databases for the Car class with data from LIDAR.

Databases LIDAR	Number of classes	Number of images	Number of Car instances
Training Dataset 1	1	7381	22687
Testing Dataset 1	1	100	129

4.5 Image preprocessing

In this phase, the preprocessing of digital images to provide them with different attributes with data augmentation [12][13]. In the real world scenario, it's possible that the existing datasets are taken under a limited conditions. This is why data Augmentation is applied to simulate different conditions and generate a random variety images.

Popular augmentation techniques [22]: lightening the image to increase their clarity and avoiding dark images; decreasing the contrast to reduce the total color range of the image; applying a Gaussian filter to eliminate noise and details of the texture; flipping

images horizontally and vertically; rotating it at right angles will preserve the image size; scaled outward or inward; the Method of resizing the section is popularly known as random cropping; translation just involves moving the image along the X or Y direction (or both).

4.6 Training U-net

In this phase of the work, the two U-net convolutional neural network models are trained, the first with images from the cameras and the second with previously processed data from the LIDAR sensors. In this phase, the preprocessing of digital images to provide them with different attributes with data augmentation [12][13].
Applying state of the art research and based on the proposed methodology, the U-net neural network model resembles the project to the neural networks that Tesla uses, its cars equipped with hardware and software installed to achieve total autonomy. Its purpose is to train deep neural networks in problems ranging from perception to control. Its camera-based neural network models analyze raw images to perform semantic segmentation, object detection and monocular depth estimation, with panoramic view (bird's eye view) taking videos from all cameras to show road layout, static infrastructure and 3D objects directly in the top-down view.

During the neural networks training, they are learning to predict where the labeled objects are located on the image with their corresponding segmentation see Fig. 4.

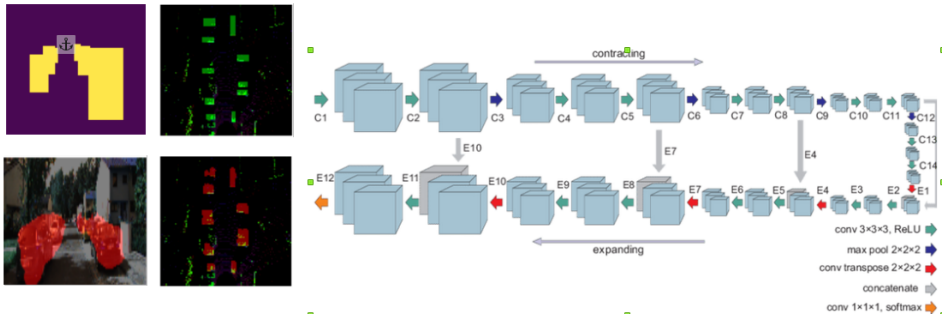


Fig. 4. The architecture of the Deep Convolutional Neural Network U-net shows at its input the initial masks and real images, with this set the neural network is trained in order to predict the targets with the least possible error.

4.7 Evaluation metric semantic segmentation IoU

The performance during the training process of the convolution neural network U-net is measured to determine the combination of features and the segmentation and detection algorithm that presents the best results. This phase consists in applying evaluation metric such as: Jaccard Index [20]. To do this, several experiments are carried out with the databases above described.

The intersection over the Union (Jaccard Index) is a measure of the magnitude of the overlap between two bounding boxes (or, in the most general case, two objects). Calculate the size of the overlap between two objects, divided by the total area of the two objects combined (see Fig. 5).

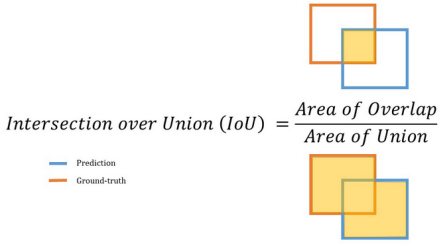


Fig. 5. Intersection over union

4.8 Segmentation and Detection Results

According to the design of the methodology, the necessary phases have been implemented to obtain the results of segmentation and detection in images of the cameras Fig. 6 and images from the data of the LIDAR sensors Fig. 7 in this phase the experiments are carried out successfully with the Car class which has the highest number of labels and therefore presents balance with respect to the rest of the classes available in the KITTI dataset.

The processes described in the methodology have been run on a computer with a GeFORCE GTX 1050 graphics card and 24 GB of RAM necessary for the process of generating the images with bird eye view from the point cloud of the LIDAR sensors; this takes extra time compared to the images from cameras. This phase of the methodology is of primary interest-working with other types of sensor data to train a neural network.

Objects evaluated with the IoU segmentation metric that were correctly evaluated for a particular class are delimited by a rectangle that represents the object's location within the scene. The name of the sample is indicated; it has a delimited area with a confidence, xmin, xmax, ymin, ymax and the name of the class "Car".

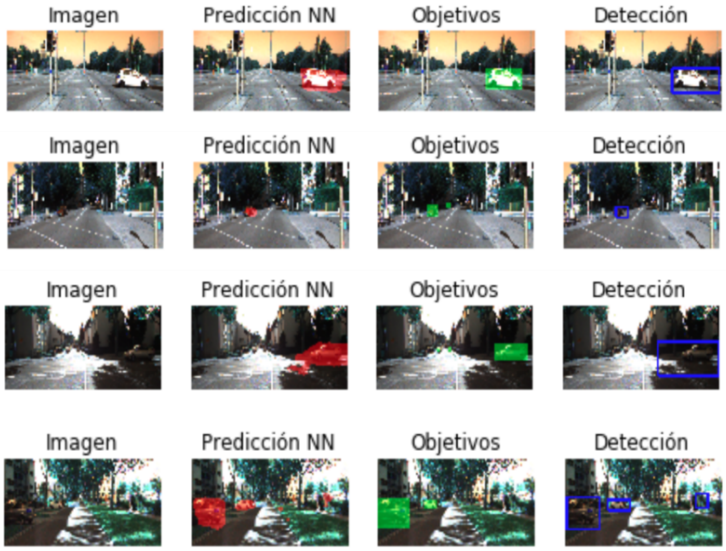


Fig. 6. Segmentation and detection of the "Car" class with Camera.

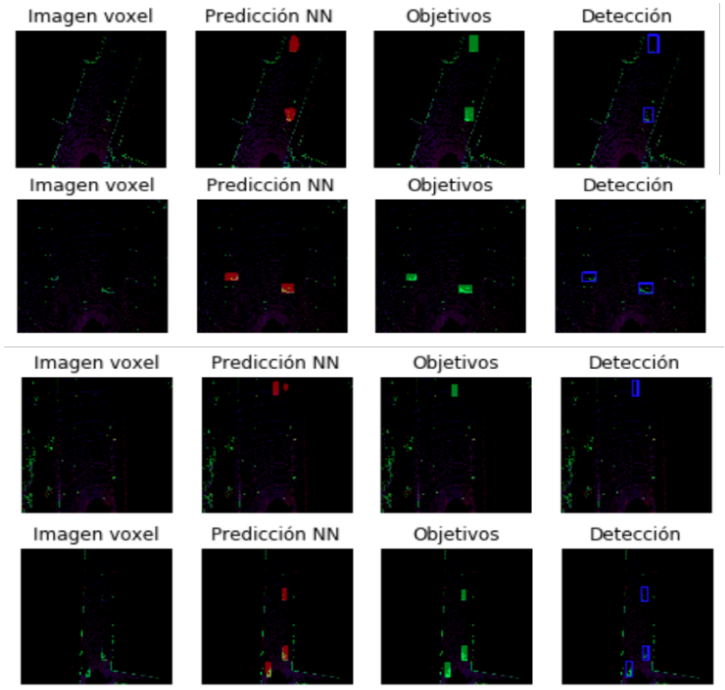


Fig. 7. Segmentation and detection of the "Car" class with LIDAR.

4.9 Confusion Matrix and Evaluation Metrics

Here are three metrics used as reference; these are precision, recall, and measure F1, which assess the balance between precision and recall [21].

$$presicion = \frac{TruePositives}{TruePositives + FalsesPositives}$$

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$F1measure = 2 \times \frac{presicion \times recall}{presicion + recall}$$

Based on the confusion matrix, the evaluation metrics corresponding to the predictions of the two deep neural networks are calculated.
The following Table 3 shows the evaluation metrics with respect to the predictions made by deep neural networks separately with the same training and test data set. It is important to mention that at this point the only class that is being evaluated is "Car", as described it is the class with the largest number in instances, unlike the other classes available in the dataset.

Table 3. Evaluation metrics.

Database	# TP	# FN	# TN	# FP	Precision	Recall	F1 measure
Cámara	113	47	1	4	96.50%	70.60%	81.50%
LIDAR	105	17	1	19	84.00%	86.00%	85.00%

The F1 metric depicts a better result with the data from the LIDAR sensors. In brief, this occurs due to the pre-processing of the point cloud, which helps to reduce too much noise and points that surround the objects of interest. Which does not happen with the images of the cameras, due to the great variety of colors and shapes that are part of the visual environment of the image. The results shown by the LIDAR are good and are equal to the images, however, they can be improved with another type of neural network architecture, and for example that only performs detection.

5 Conclusions and future work

In this work, was possible to integrate advanced computer vision algorithms using deep neural networks, whose results were positive in recent studies conducted in autonomous prototype vehicles.

The results of the U-net network trained with data from the LIDAR sensors, its prediction efficiency is is as optimal as that of the images, in addition to the advantages offered by the LIDAR sensor such as deep estimation, location and mapping is possible with this work get good results.

The prediction times for the process of testing the neural network trained with LIDAR data responds with greater speed compared to the image network. This data is extremely important in response to the detection of objects in an autonomous vehicles.

The depth estimation and the creation of navigation maps of the LIDAR sensors help to detect objects; this could prevent and avoid accidents such as those that have occurred with the automatic pilot of Tesla cars. It is concluded that it is essential to use and mount LIDAR sensors in autonomous cars.

Also, the development of LIDAR sensors with signals of better resolution and quality will directly contribute to a better detection of objects in the future and will come at a more accessible price; a factor that is most desired by automotive companies.

As a future work, it is proposed to update the proposed methodology by testing with different deep neural network architectures and using stereo or monocular vision with deep estimation vs LIDAR. It is proposed to analyze the possibility of instrumenting a car with a LIDAR sensor and cameras to generate own dataset on roads in Mexico.

References

1. Herzog, Manuel; Dietmayer, Klaus. Training a Fast Object Detector for LiDAR Range Images Using Labeled Data from Sensors with Higher Resolution. En 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019. p. 2707-2713.
2. Wang, Yan, et al. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.p. 8445-8453 (2019).
3. Chadwick, Simon; Maddetn, Will; Newman, Paul. Distant vehicle detection using radar and vision. En 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019. p. 8311-8317.
4. Feng, Di, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems, 2020.
5. James, Jyothish K., et al. Classification of LIDAR sensor contaminations with deep neural networks. En Proceedings of the Computer Science in Cars Symposium (CSCS). p. 8.(2018)
6. Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3354–3361. IEEE (2012). http://www.cvlibs.net/datasets/kitti/
7. Olaf Ronneberger, Philipp Fischer, and Thomas Brox: U-Net: Convolutional Networks for Biomedical Image Segmentation.Computer Science Department and BIOSS Centre for Biological Signalling Studies,University of Freiburg, Germany, arXiv:1505.04597v1[cs.CV] (2015)
8. Ren, Shaoqing, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. En Advances in neural information processing systems. p. 91-99. (2015)
9. He, Kaiming, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence. vol. 37, no 9, p. 1904-1916.(2015)
10. Redmon, Joseph, et al. You only look once: Unified, real-time object detection. En Proceedings of the IEEE conference on computer vision and pattern recognition. p. 779-788.(2016)
11. Liu, Wei, et al. Ssd: Single shot multibox detector. En European conference on computer vision. Springer, Cham. p. 21-37. (2016)

12. Tian, Yuchi, et al. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. En Proceedings of the 40th international conference on software engineering. p. 303-314. (2018).
13. Du, Shuyang; Guo, Haoli; Simpson, Andrew. Self-driving car steering angle prediction based on image recognition. arXiv preprint arXiv:1912.05440, (2019)
14. Retterath, James E.; LAUMEYER, Robert A. Methods and apparatus for object detection and identification in a multiple detector lidar array. U.S. Patent No 9,360,554, 7 Jun. 2016.
15. Zhou, Yin; Tuzel, Oncel. Voxelnet: End-to-end learning for point cloud based 3d object detection. En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. p. 4490-4499.(2018)
16. Ali, Waleed, et al. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. En Proceedings of the European Conference on Computer Vision (ECCV). p. 0-0.(2018).
17. Beltrán, Jorge, et al. Birdnet: a 3d object detection framework from lidar information. En 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, p. 3517-3523. (2018)
18. Lee, Kuan-Hui, et al. End-to-end Birds-eye-view Flow Estimation for Autonomous Driving. arXiv preprint arXiv:2008.01179, (2020).
19. Yang, Guidong, et al. LiDAR point-cloud processing based on projection methods: a comparison. arXiv preprint arXiv:2008.00706, (2020).
20. Real, Raimundo; Vargas, Juan M. The probabilistic basis of Jaccard's index of similarity. Systematic biology, 1996, vol. 45, no 3, p. 380-385.
21. Powers, David Martin. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2011.
22. Perez, Luis; Wang, Jason. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621, 2017.
23. Oksuz, Kemal, et al. Imbalance problems in object detection: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
24. Dumolin, Vincent; VISIN, Francesco. A guide to convolution arithmetic for deep learning, 2016. arXiv preprint arXiv:1603.07285, 2016.
25. Bojarski, Mariusz, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.
26. Murugan, P. Feed forward and backward run in deep convolution neural network. 2017.
27. Noh, Hyeonwoo; Hong, Seunghoon; Han, Bohyung. Learning deconvolution network for semantic segmentation. En Proceedings of the IEEE international conference on computer vision. 2015. p. 1520-1528.
28. Rusinkiewicz, Szymon; Levoy, Marc. Efficient variants of the ICP algorithm. En Proceedings third international conference on 3-D digital imaging and modeling. IEEE, 2001. p. 145-152.
29. Deakin, Rodney E. 3-D coordinate transformations. Surveying and land information systems, 1998, vol. 58, no 4, p. 223-234.
30. Chen, Chenyi, et al. Deepdriving: Learning affordance for direct perception in autonomous driving. En Proceedings of the IEEE International Conference on Computer Vision. 2015. p. 2722-2730.

Árbol de Decisión como Composición de Objetos Paralelos para la Tipificación de Secuencias ADN del Virus de la Hepatitis-C

Mario Rossainz-López, Sarahi Zúñiga-Herrera, Iván Olmos-Pineda, Ivo Pineda-Torres
Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 sur, San Manuel, Puebla, Puebla, 72000, México
{rossainz, iolmos, ipineda}@cs.buap.mx, zarahi.suhe@gmail.com

Resumen. Se muestra una solución paralela para resolver el problema de clasificación de ADN mediante la tipificación de secuencias (STP). La solución propuesta hace uso de las Composiciones Paralelas de Alto Nivel (CPANs) para encontrar regiones conservadas de secuencias ADN que ayuden a clasificar los distintos tipos del virus de la hepatitis C usando un árbol de decisión, entropía de Shannon y ganancia de información. Se muestra la utilidad de la propuesta paralela con datos reales de un laboratorio clínico de la ciudad de Puebla y se registra el buen rendimiento obtenido en la velocidad de sus ejecuciones y escalabilidad del speedup en cuanto al número de cores utilizados en la ejecución del CPAN. Se concluye que la propuesta ayudará a los laboratoristas que están limitados a observar y determinar de manera manual las regiones adecuadas para resolver el problema de clasificación, a resolverlo de una manera más rápida y automatizada.

Palabras Clave: STP, CPANS, Árboles de decisión, Objetos Paralelos, Hepatitis-C

1 Introducción

La identificación de genes es un área importante para entender el genoma de una especie una vez que éste ha sido secuencializado y el ADN contiene toda la información genética de los organismos vivos. Una secuencia de ADN se compone de un alfabeto que se forma con las letras de las cuatro bases nitrogenadas que lo componen: adenina, timina, guanina y citosina. En el área de la genómica esto toma importancia ya que se pueden usar diferentes tipos de pruebas de secuencia que logren identificar, por ejemplo, agentes infecciosos presentes en una muestra de sangre tomada de un paciente, para tareas de diagnóstico. En ese sentido, uno de los problemas más frecuentes es la clasificación de una secuencia dada de ADN y determinar la clase o subclase a la que pertenece (STP) [1]. Toma gran importancia aquellos problemas de clasificación que tienen una alta tasa de variabilidad como lo es el caso de estudio que se aborda en este trabajo que es el del virus de la hepatitis C. Este virus tiene una alta variabilidad genética y hasta el día de hoy se han descubierto siete tipos asociados con diferentes comportamientos en el huésped como enfermedades hepáticas crónicas, cirrosis y trasplantes de hígados [2]. Para determinar el tipo o subtipo del virus de la Hepatitis C,

es necesario utilizar una prueba de diagnóstico molecular que permita al médico guiarse sobre el tratamiento para el paciente. La Reacción en cadena de la Polimerasa (PCR) es una de ellas y actualmente representa la prueba más rápida y eficaz utilizada en investigación y laboratorios clínicos en el mundo [2]. El proceso PCR amplifica los segmentos cortos de una molécula de ADN más larga utilizando un procedimiento llamado "Ciclo de Amplificación" [3]. Cada ciclo PCR duplica la cantidad de secuencia objetivo o amplicón en la reacción, de modo que, por ejemplo, 20 ciclos multiplican el amplicón por un factor de más de un millón en cuestión de horas. [4]. La prueba PCR es una técnica con alta sensibilidad, reproducibilidad y eficiencia, que genera resultados confiables en poco tiempo y fácil de analizar [5,6]. Pero, el diseño de una prueba de diagnóstico por PCR puede ser muy complejo si el número de secuencias a clasificar es grande y variable. Es ahí en donde el tiempo que transcurre desde el inicio de la prueba hasta que se obtienen resultados para emitir el diagnóstico puede sufrir un retraso considerable y una propuesta computacional paralela puede compensar el tiempo de ejecución de la prueba. El éxito de una prueba PCR depende en gran medida de un diseño apropiado de los cebadores que se utilizan en ella para maximizar su eficiencia y especificidad [7]. De forma general el diseño de los cebadores consta de 4 fases: Fase-1: la obtención de una base de datos (ViRP) con las secuencias genéticas objetivo, Fase-2: procesar la base de datos con alguna herramienta de cómputo para localizar las regiones homólogas y conservadas, Fase-3: identificar los cebadores que garanticen especificidad y sensibilidad a los criterios químicos y Fase-4: validar los propuestos por las reacciones en el procedimiento. La tercera fase es la más costosa de todo el diseño en términos de tiempo y recursos económicos, sobre todo para las pruebas en las que se desea realizar una clasificación de secuencias como la mencionada en el caso de la Hepatitis-C. Por el contrario, la fase-2 es la más simple de llevar a cabo gracias a la gran cantidad de herramientas de cómputo que existen para ello (Jalview, Strap, ClustalX or Clustal Omega, entre otras [8]). Sin embargo, para la primera fase no existen herramientas de cómputo; los investigadores están limitados a observar las secuencias y determinar manualmente cuál es la región correcta para resolver el problema de clasificación. Es aquí donde lo planteado en este trabajo puede ser utilidad para ayudar a decidir al investigador si las secuencias de ADN no clasificadas pertenecen o no a una clase particular del virus de la Hepatitis tipo C. En [9] se propone una solución paralela usando Composiciones de Objetos Paralelos o CPANs, al reconocimiento de secuencias de ADN haciendo uso de algoritmos de aprendizaje automático como el uso de redes neuronales convolucionales donde se utiliza un patrón de comunicación tipo PipeLine para la construcción de la Red Neuronal, sin embargo si acotamos el problema de la clasificación al diagnóstico clínico donde se extrae una muestra de un paciente que contiene material genético del cual se desconoce la secuencia de nucleótidos, esa propuesta de solución no es factible porque en el aprendizaje automático se requiere conocer dicha secuencia. De ahí la importancia de contar con algoritmos que resuelvan el problema de clasificación mediante métodos de diagnóstico clínico como el mencionado PCR generando una heurística de solución incorporando como parte de ella una propuesta de paralelización vía CPANs que proporcione un buen rendimiento y aceleración en la obtención de dichas secuencias para su clasificación, respecto de su contraparte secuencial que es la manera tradicional que se viene llevando a cabo como parte de todo un proceso que es lento y complejo. La propuesta consiste en diseñar e implementar una Composición

Paralela de Alto Nivel (CPAN) que mediante el patrón de comunicación de procesos de Árboles de Decisión de aridad-n automatice el problema de clasificación de secuencias ADN que los investigadores llevan a cabo manualmente dentro del diseño de los cebadores para una PCR.

2 Composición Paralela de Alto Nivel (CPAN)

Se utiliza la Programación Paralela Estructurada y la Programación Orientada a Objetos para construir los llamados Objetos Paralelos y con ellos las Composiciones Paralelas de Alto Nivel o CPANs. Dado que las aplicaciones paralelas generalmente siguen patrones predeterminados de ejecución, que son rara vez arbitrarios y no estructurados en su lógica se propone el diseño, implementación y uso de los CPANS como patrones paralelos de comunicación bien definidos y lógicamente estructurados que, una vez identificados en términos de sus componentes y de su esquema de comunicación, pueden llevarse a la práctica como constructos añadidos a un lenguaje de programación orientado a objetos y estar disponibles como abstracciones de alto nivel en las aplicaciones del usuario, dentro de un entorno o ambiente de programación que también se proporciona [10,11]. Un CPAN es la composición de un conjunto de objetos paralelos de tres tipos: Un objeto manager que representa al CPAN en sí mismo y hace de él una abstracción encapsulada que oculta su estructura interna. El manager controla las referencias de un conjunto de objetos (un objeto denominado Collector y varios objetos denominados Stage), que representan los componentes del CPAN y cuya ejecución se lleva a cabo en paralelo y debe ser coordinada por el propio manager. Los objetos Stage son los encargados de encapsular una interfaz tipo cliente-servidor que se establece entre el manager y los objetos esclavos (objetos pasivos que contienen el algoritmo secuencial de la solución de un problema). Y un objeto Collector que es un objeto encargado de almacenar en paralelo los resultados que le lleguen de los objetos stage que tenga conectados [12] (ver fig 1). Un CPAN cuenta con las siguientes propiedades que se pueden estudiar a detalle en [12]: Modo asíncrono y futuro asíncrono de comunicación entre los objetos paralelos del CPAN [13], objetos con paralelismo interno, disponibilidad de mecanismos de sincronización; paralelismo máximo (MaxPar), exclusión mutua (MuTex) y sincronización (Sync) del tipo Productor-Consumidor, disponibilidad de control de tipos genéricos, transparencia en la distribución de aplicaciones paralelas y rendimiento satisfactorio: programabilidad, portabilidad y performance.

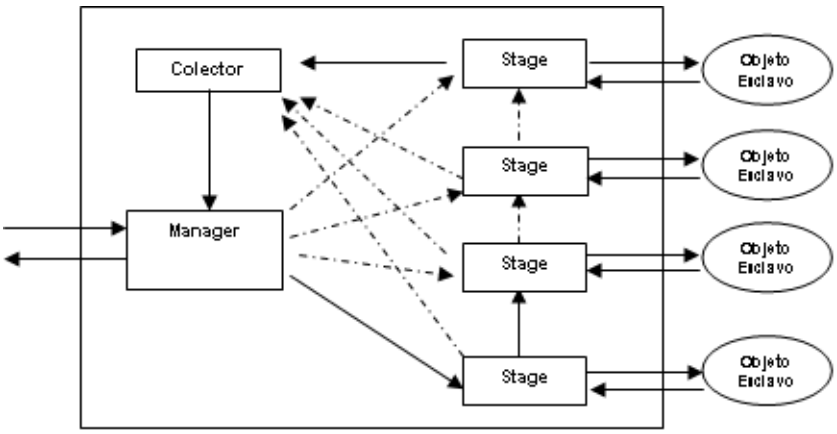


Fig. 1. Modelo abstracto de un CPAN

3 El Patrón de Comunicación de Procesos tipo Árbol como un CPAN

Como parte de una biblioteca de clases que el programador puede utilizar dentro de sus aplicaciones para generar y/o utilizar diversos patrones de comunicación genéricos y particulares como CPANS, se encuentran aquellos que representan árboles de procesos. La Fig.2 muestra el modelo particularizado por simplicidad a árboles binarios. Este es un modelo abstracto que requiere ser adaptado al problema que se está tratando de resolver mediante el uso de las propiedades del paradigma de la orientación a objetos tales como la herencia o el polimorfismo. Para consultar los detalles de su implementación ver [12,13]. El árbol binario que se forma dentro del CPAN en la fig.2, está representado por los objetos paralelos o procesos "stage" como nodos de dicho árbol y a los cuales se les asocia el o los algoritmos secuenciales respectivos que resuelven el problema a través de los llamados Objetos Esclavo (OE). El nodo raíz del árbol recibe como entrada un problema completo que se divide en dos partes, una se envía al nodo hijo izquierdo, la otra se envía al nodo que representa al hijo derecho (Fig.2). Se repite recursivamente el proceso de división hasta llegar a los niveles más bajos del árbol hasta que, transcurrido un cierto tiempo, todos los nodos hoja reciben como entrada un subproblema de su nodo padre, entonces lo resuelven y le devuelven las soluciones. Cualquier nodo padre en el árbol obtendrá dos soluciones parciales de sus nodos hijos y las combinará para proporcionar una única solución que será la salida del nodo padre (técnica de diseño algorítmica divide y vencerás). Finalmente, el nodo raíz proporcionará como salida la solución completa del problema inicial al proceso Manager en el CPAN. A diferencia de otros modelos de CPAN como los CPAN-farms o los CPAN-pipeline, donde los objetos esclavos pueden ser predeterminados fuera del modelo CPAN, sólo un objeto esclavo es predefinido de forma estática en el CPAN-Tree y asociado al primer stage del árbol en este modelo, el de la Fig.2.

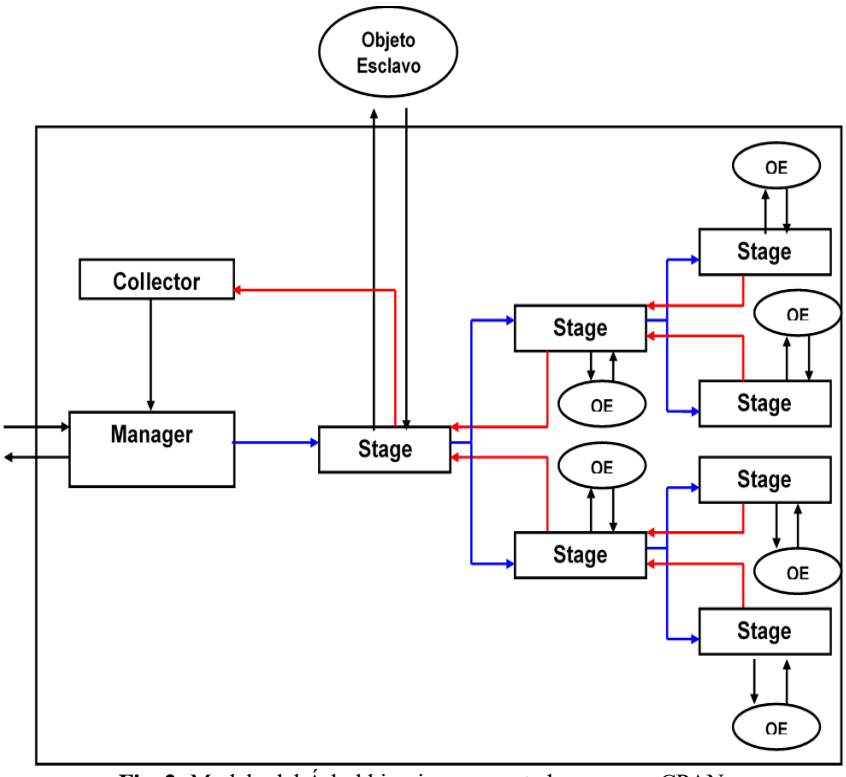


Fig. 2. Modelo del Árbol binario representado como un CPAN

Los siguientes objetos esclavos serán creados internamente por los propios stages de forma dinámica, pues los niveles del árbol dependen del problema a resolver y no se conoce a priori el número de nodos (procesos) que pueda tener el árbol, ni tampoco su nivel de profundidad. No obstante, al igual que en los otros modelos CPAN, los procesos Manager y stage_i son instancias de clases concretas de la biblioteca de clases en donde se encuentra su implementación. El proeso Collector es también una instancia de la clase que lo instancia.

El modelo de CPAN-Tree ha sido utilizado y adaptado a patrones de trees particulares para la solución de problemas diversos como: problemas de ordenación, búsqueda y optimización, problemas de integración numérica, problemas NP-Completos como el del Agente Viajero, problemas de simulación de movimientos como el de las partículas y su atracción en el espacio, o simulación del movimiento y atracción de los planetas en el sistema solar y recientemente en problemas referentes a la construcción de GNOMAS construyendo secuencias de ADN o de clasificación como lo muestra el presente artículo.

3.1. Solución de problemas relacionados utilizando el patrón de comunicación tipo Árbol como CPAN

Actualmente se tiene un estado del arte en el uso del patrón de comunicación de procesos tipo Árbol como CPANs. A continuación, se listan los problemas que se han resuelto utilizando y particularizando éste CPAN a situaciones específicas:

- Una Biblioteca de clases de Objetos Paralelos para implementar Patrones de Comunicación usando CPANs: En este trabajo se desarrolló un método de programación basado en Composiciones Paralelas de Alto Nivel o CPANs y se implementaron los CPANS PipeLine, Farm y TreeDV que forman parte de una librería de clases propuesta para el desarrollo de Objetos Paralelos
- Una propuesta metodológica para generar un particionamiento de la técnica de diseño algorítmica Divide y Vencerás como un CPAN. Esta propuesta se utilizó para generar un CPAN representado como un árbol binario en paralelo y resolver con ello el problema de la ordenación paralelizando el algoritmo Quicksort.
- Uso del CPAN Branch & Bound para la solución del Problema del Agente Viajero: Se implementó un Árbol Binario como CPAN para utilizarlo como un patrón de comunicación entre procesos que representara la técnica de Divide y Vencerás y con el resolver el problema NP-Completo del Agente Viajero
- Simulación Paralela del Problema M-Body usando el CPAN Quad-Tree. Se implementó una simulación en paralelo del Problema N-body utilizando el patrón de comunicación entre procesos de un árbol de aridad-N bajo el Modelo CPAN. De forma particular se resolvió el problema del movimiento en el espacio de N-partículas para determinar los efectos de las fuerzas de atracción entre ellas según la ley electrostática de Coloumb.

4 Árboles de Decisión

Existen muchas técnicas de clasificación que se utilizan frecuentemente en la solución de diferentes problemas dentro de la Bioinformática (que se define como la unión de la Informática con la Biología y cuyo objetivo es procesar el gran volumen de datos biológicos que se han generado en los últimos años para su posterior utilización en distintas áreas como la biotecnología por citar alguna) [14]. Estas técnicas de clasificación surgen como técnicas de aprendizaje automatizado necesarias en la descripción de dominios biológicos como lo son las secuencias y clasificación de cadenas ADN. Destacan por su utilidad los árboles de decisión como una herramienta alternativa para la predicción y clasificación de grandes cantidades de datos que es utilizada ampliamente en la inteligencia artificial [15]. Un árbol de decisión es una estructura formada por un conjunto de nodos, hojas y ramas que representa un modelo de predicción cuyo objetivo es el aprendizaje inductivo a partir de observaciones y construcciones lógicas [15]. El nodo raíz del árbol es el atributo a partir del cual se inicia el proceso de clasificación; los nodos internos corresponden a cada una de las preguntas acerca del atributo en particular del problema. Cada posible respuesta se representa mediante un nodo hijo. Las ramas que salen de cada uno de estos nodos se encuentran etiquetadas con los posibles valores del atributo. Los nodos hoja

corresponden a una decisión, la cual coincide con una de las variables clase del problema a resolver [15]. En un algoritmo de generación de árboles de decisión se distinguen 3 etapas:

- La etapa de Selección: Se genera el nodo raíz a partir de un atributo de prueba y se crean nodos de entre el conjunto de entrenamiento inicial que se divide en cada uno de ellos y que tienen la posibilidad de ser ramificados, dependiendo, la forma de elección directamente de la estrategia de clasificación que se decida utilizar en el algoritmo;
- La etapa de Ramificación: se construyen los posibles nodos hijos del nodo seleccionado en el paso anterior. De esta manera se va formando el árbol de decisión que representa el espacio de clasificación en cada momento.
- La etapa final: se obtienen los nodos hojas con la clasificación final y que ya no puede ser ramificado. Se recorre el árbol desde el nodo raíz hasta una hoja. El camino a seguir en el árbol lo determinan las decisiones tomadas en cada nodo interno, de acuerdo con el atributo de prueba presente en él.

5 Automatización de la clasificación de secuencias ADN del virus de la Hepatitis-C mediante árboles de decisión como CPAN

En base a los elementos descritos en la sección 2, se muestra la definición del CPAN-DecisionTree como parte integrada de una propuesta de solución (entropía, ganancia de información y árbol de decisión) en el problema a resolver: En un conjunto de secuencias o instancias *Gw* de ADN queremos ubicar aquellos atributos Ai que brindan más información y se consideran los mejores atributos para resolver el problema de clasificación de las siete clases *Cy* del virus de la hepatitis C que existen actualmente. Estos atributos deben pertenecer a una región conservada y considerar los criterios que favorecen la realización de una prueba de diagnóstico molecular por PCR. En el contexto de la clasificación, la calidad de un atributo Ai tiene que ver con su capacidad para separar las instancias Gw, entre las diferentes clases posibles. Si existe una relación directa entre los valores de los atributos y las posibles clases, significa que el atributo es bueno para clasificar. La calidad de un atributo tiene que ver con qué clases se pueden separar cada vez que instanciamos ese atributo Ai. Las clases están bien separadas cuando cada subgrupo generado por la división del producto es homogéneo, es decir: en cada subgrupo todas las instancias Gw pertenecen a la misma clase una vez que instanciamos el atributo. Para medir la homogeneidad utilizamos la entropía de Shannon. Calculamos la ganancia de información que permite cuantificar la información proporcionada por un atributo Ai con respecto al problema de clasificación haciendo la diferencia entre la entropía de Shannon antes con la entropía de Shannon después de conocer el valor del atributo Ai. Esto nos da una representación S del conjunto de instancias Gw, donde cada instancia pertenece a una clase Cy. A cada posición o nucleótido de un conjunto de secuencias alineadas S se le asignó el nombre del atributo Ai donde i indica la posición del nucleótido. Para comprender lo anterior, en la Tabla 1 se muestra un ejemplo asociado con los valores. Al aplicar la ganancia de información para cada uno de los atributos del conjunto S se observa que el atributo A_3 se evalúa con la mayor ganancia de información y se divide en tres subconjuntos del

conjunto S. El primero es $S_C{}^3$, donde sus instancias pertenecen a las clases C_2 y C_3. El segundo es $S_A{}^3$ con todas sus instancias pertenecientes a la clase C_4. El último subconjunto es el $S_T{}^3$ donde todas las instancias pertenecen a la clase C_1 (los detalles de las fórmulas matemáticas de la Entropía de Shannon y Ganancia de Información se pueden encontrar en [16]). Si dos instancias no tienen el mismo valor para cada atributo y pertenecen a diferentes clases, los atributos son adecuados para llevar a cabo la clasificación, como se puede ver en la Tabla 1, el subgrupo $S_C{}^3$ los atributos A_1 y A_2 permiten clasificar los elementos correctamente para las clases C_2 y C_3.

Con este ejemplo particular, se muestra que es posible clasificar las secuencias de ADN utilizando los conceptos de entropía y ganancia de información. Para este caso se selecciona el atributo A_3 con mayor I_G (Ganancia de Información), este atributo permite discriminar rápidamente las clases C_1 y C_4. Al calcular nuevamente I_G de todos los atributos, se obtiene que tanto A_1 como A_2 permiten discriminar las clases C_2 y C_3. Esto se obtiene generando un árbol de decisión donde cada vértice tiene un máximo de 4 valores posibles (la Fig.3 muestra una parte del Árbol de Decisión en cuestión). Dicha estructura se crea haciendo uso del CPAN-DecisionTree (Fig.4) que se implementa al definir una instancia concreta (y particularizada al problema que se estudia en este trabajo) del CPAN-Tree genérico de la Fig.2., sección 3.

Tabla 1. Representación de S conjunto de instancias Gw, donde cada instancia

Cy	**Gw**	**A 1**	**A 2**	**A 3**	**A 4**	**A 5**
C_1	G_1	A	T	T	A	T
C_1	G_2	A	T	T	C	T
C_2	G_3	A	G	C	A	C
C_2	G_4	A	G	C	G	C
C_2	G_5	A	G	C	A	T
C_2	G_6	A	G	C	G	T
C_3	G_7	G	T	C	T	C
C_3	G_8	G	T	C	T	C
C_3	G_9	G	T	C	A	G
C_3	G_{10}	G	T	C	T	G
C_4	G_{11}	A	G	A	A	C
C_4	G_{12}	A	G	A	G	C

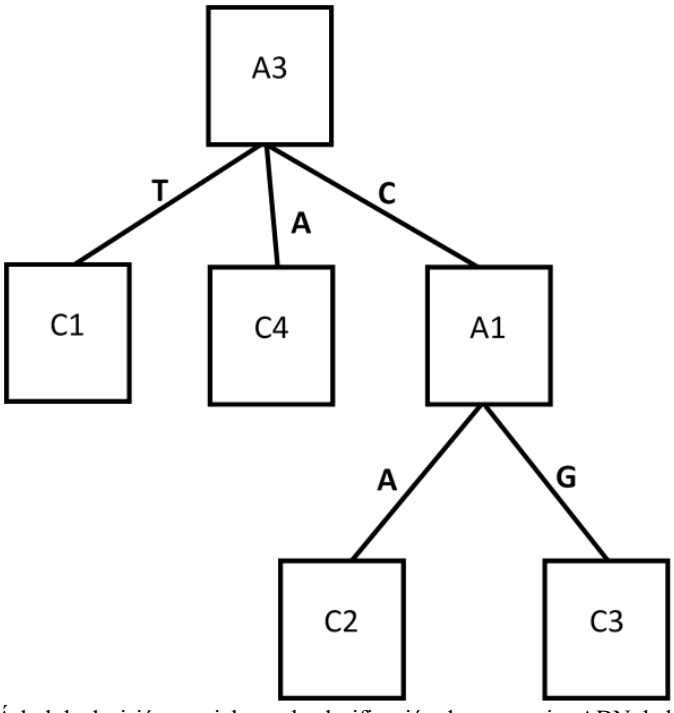


Fig. 3. Árbol de decisión parcial para la clasificación de secuencias ADN de los datos de la Tabla 1

En la Fig.4, el CPAN-DecisionTree recibe a través de su proceso Manager la base de datos o repositorio con las instancias de secuencias de ADN del virus de la Hepatitis-C y pasa la información al primer proceso Stage que representa la raíz de árbol de decisión el cual tiene asociado un objeto esclavo con los algoritmos y modelos matemáticos que llevan a cabo la entropía de Shannon y la ganancia de información para así obtener un atributo considerado como "mejor atributo" e ir resolviendo el problema de clasificación o bien ramificar para ir generando más nodos que puedan proporcionarnos los mejores atributos los cuales serán enviados al proceso Collector quien los ira recibiendo y formará el conjunto solución de los mejores atributos del problema de clasificación para que, una vez recibidos todos los atributos, éstos sean pasados al proceso Manager quien los entregará al usuario como resultado final del proceso.

La ejecución de los procesos u objetos activos del CPAN-DecisionTree se lleva a cabo en paralelo con las políticas de restricción, de sincronización y comunicación de procesos, que se explicaron en la sección 2 y cuyo detalle se puede encontrar en [17, 18].

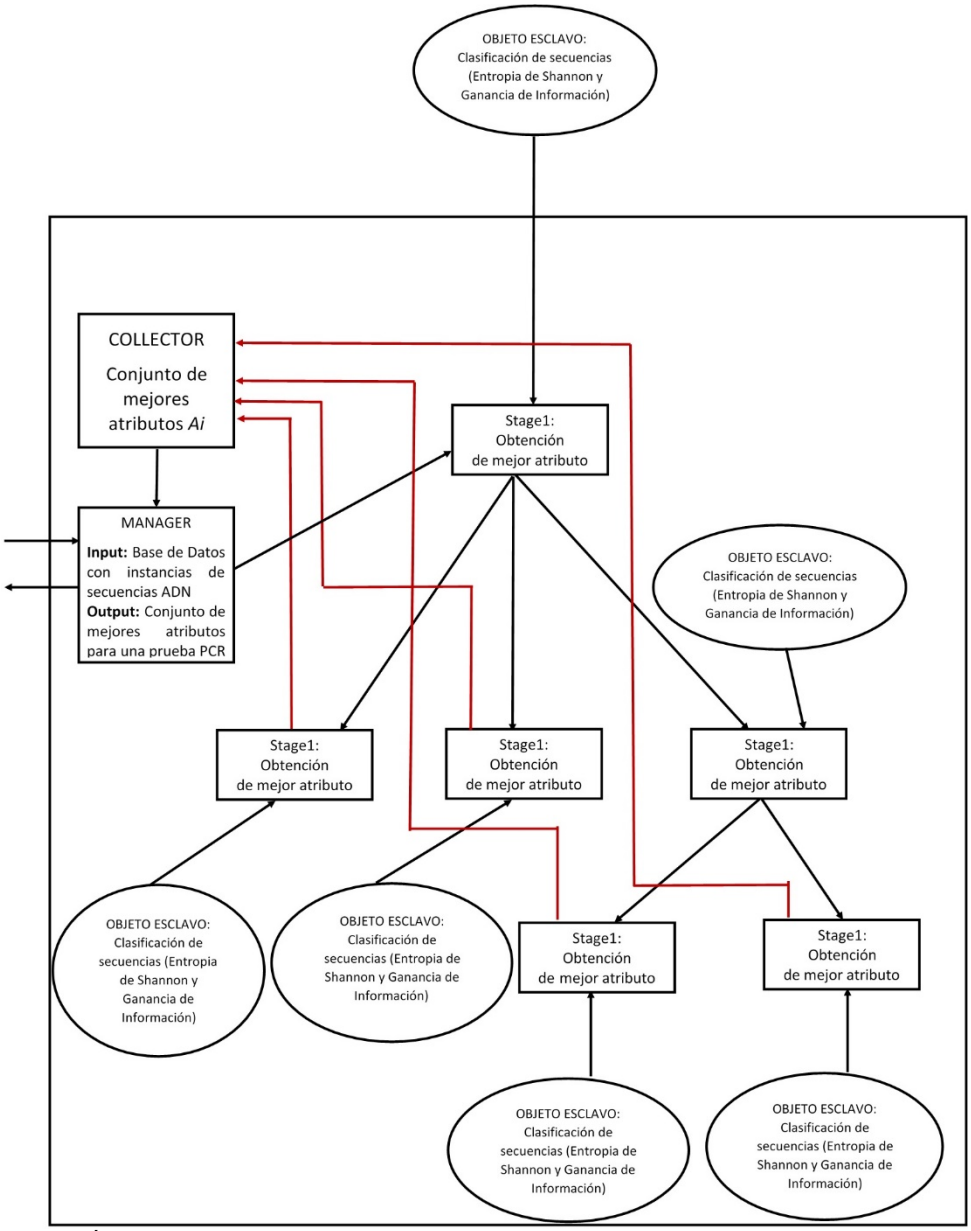


Fig. 4. Árbol de decisión parcial representado como CPAN para el problema de clasificación de secuencias ADN

El procedimiento de generación dinámica del árbol de decisión dentro del CPAN es el siguiente:

1. Una vez que el Manager recibe la base de datos con las secuencias ADN como información de entrada, se crea el nodo raíz inicial del árbol asociándole el objeto esclavo correspondiente para la ejecución del procedimiento de

clasificación que incluye la Entropía de Shannon y la ganancia de información y la información de entrada es recibida por este nodo raíz.

2. Si todas las instancias tienen el atributo objetivo que pertenece a la misma etiqueta C entonces el árbol de decisión será único un nodo raíz con la etiqueta C.
3. Si los atributos están vacíos, entonces el árbol de decisión estará formado por un único nodo raíz que tendrá como solución la etiqueta más común del atributo objetivo en las instancias, si no, se elige de la lista de atributos aquel que mejor clasifique las instancias y éste será un atributo de decisión de un nodo raíz denotado como *Ai*.
4. Para cada valor posible v_i del atributo *Ai*,
    a. Añadir una nueva rama a partir de ese nodo raíz, hacer que *Ai* sea v_i y regresar al punto 3 volver a realizarlo.
    b. Dadas las instancias *vi* como el subconjunto de instancias con el valor *vi* para el atributo *Ai*; si las instancias *vi* es el vacío entonces debajo de esta rama, agregar un nuevo nodo hoja con el valor más común de atributo objetivo en las instancias y enviarlo al proceso *Collector* como el "mejor atributo", sino regresar recursivamente al punto 1.

El análisis anterior indica que los conceptos de entropía y ganancia de información y árbol de decisión permiten clasificar las secuencias de ADN y pueden considerarse como buenos criterios que favorezcan el diseño de cebadores para una prueba de diagnóstico PCR en particular para el virus de la Hepatitis-C y sus distintas clases.

6 Resultados de Escalabilidad, Aceleración y Rendimiento

La ejecución del CPAN-DecisionTree propuesto (ver Fig. 4) se llevó a cabo en una computadora servidor con procesador Intel-CORE-i5 de 2.30 GHz y 8 cores, una memoria RAM de 125 Gbytes y un sistema operativo Linux de 64 bits. El análisis de la ejecución se muestra en la Tabla 2 y Fig.5, donde podemos ver que los tiempos de ejecución del CPAN disminuyen conforme se incrementan el número de cores que se utilizan en la solución del problema de clasificación de secuencias de cadenas ADN del virus de la Hepatitis-C con los datos de la Tabla 1. En la fig.6 y la Tabla 2, se muestra la escalabilidad del Speedup encontrado del CPAN-DecisionTree usando de 3 a 8 cores en su ejecución obteniendo una buena aceleración a medida que se incrementa el número de cores y comparándola con la cota superior que arroja la Ley de Amdahl como aceleración máxima.

Tabla 2. Medidas de rendimiento del CPAN-DecisionTree (Árbol de Decisión) para la clasificación de secuencias ADN del virus de la Hepatitis-C del ejemplo de la Tabla 1.

	CPU SEQ	***CPU 3***	***CPU 4***	***CPU 5***	***CPU 6***	***CPU 7***	***CPU 8***
Run Time (min)	15.55	12.77	9.21	8.78	6.70	4.90	4.79
SpeedUp	1	1.22	1.69	1.77	2.32	3.17	3.25
Amdahl	1	2.50	3.08	3.57	4.00	4.38	4.71

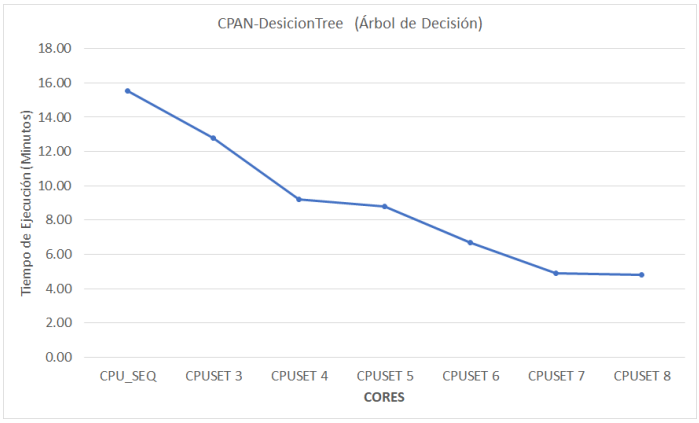


Fig.5. Comparación del tiempo de ejecución secuencial, respecto del tiempo de ejecución paralelo del CPAN-DecisionTree con 3, 4, 5, 6, 7 y 8 cores para la clasificación de secuencias ADN del virus de la Hepatitis-C del ejemplo de la Tabla 1.

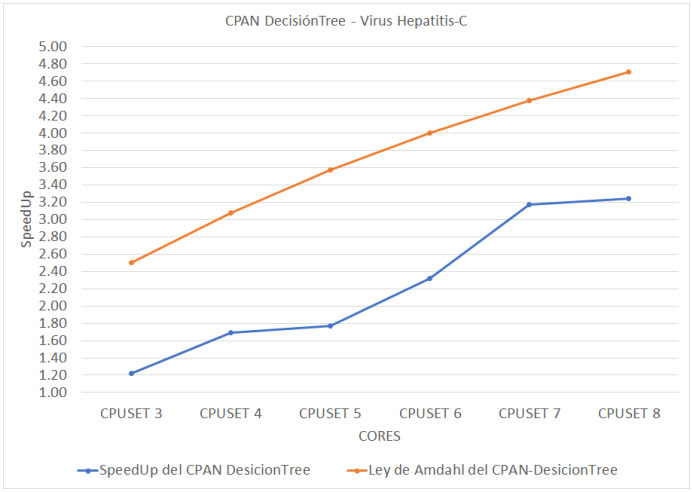


Fig.6. Escalabilidad de la magnitud del SpeedUp encontrado para el CPAN-DecisionTree para 3,4,5,6,7 y 8 cores en el problema de clasificación de secuencias ADN del virus de la Hepatitis-C del ejemplo de la Tabla 1.

7 Conclusiones

Ésta propuesta representa una ayuda útil y adecuada a los laboratoristas que diseñan los cebadores para resolver problemas de tipificación de secuencias (STP); en particular para discriminar entre los diferentes tipos de virus de la hepatitis C, por ser uno de los problemas más frecuentes en la clasificación de secuencias ADN con una alta tasa de variabilidad genética que es lo que dificulta la clasificación. Dichos laboratoristas actualmente están limitados a observar y determinar de forma manual las secuencias y

las regiones adecuadas para resolver el problema de clasificación. Esta propuesta paralela ayudará a llevar a cabo el procedimiento de la clasificación de secuencias de una manera más rápida y automatizada, no sólo para la discriminación de los distintos tipos de virus de la Hepatitis-C sino en general para cualquier tipificación de secuencias ADN donde se requiera llevar a cabo una clasificación. Se mostró entonces una solución automatizada, programando los conceptos de Entropía de Shannon, ganancia de información y árbol de decisión bajo una propuesta paralela diseñando el CPAN-DecisionTree como una Composición de Objetos Paralelos particular al problema planteado. Se utilizó CPAN-DecisionTree en el problema de clasificación de secuencias ADN del virus de la Hepatitis-C con datos reales de un laboratorio clínico particular de la ciudad de Puebla, México (ver Tabla 1) y se obtuvo un conjunto de los mejores atributos del conjunto de secuencias ADN que favorecieron la realización de una prueba de diagnóstico molecular por PCR (ver sección 5). Por otro lado, la ejecución paralela del CPAN-DecisionTree mostró un buen rendimiento al comparar su aceleración con respecto a su ejecución secuencial de los datos mostrados en la Tabla 2 y graficados en la Fig.5. El rendimiento obtenido en la velocidad de sus ejecuciones y la escalabilidad del speedup en comparación con la ley de Amdahl también arrojaron buenos resultados (ver Fig.6).

Referencias

1. Eisele, J.G. Roldán, C. Y. C., Galindo M.O., Gil M.P., (2004). Usefulness of solution algorithms of the traveling salesman problem in the typing of biological sequences in a clinical laboratory setting. In: 14th IEEE International Conference on Electronics, Communications and Computers CONIELECOMP, pp. 264–269.
2. Messina J.P., Humphreys I., Flaxman A., Brown A., Cooke G.S., Pybus O.G., Barnes E., (2015). Global distribution and prevalence of hepatitis C virus genotypes. Hepatology 61(1), 77–87.
3. Sharkey D., (1994). Antibodies as thermolabile switches: High temperature triggering for the polymerase chain reaction. Biotechnology 12, 506–509.
4. Eckert K.A., Kunkel T.A., (1991). DNA polymerase fidelity and the polymerase chain reaction. PCR Methods, pp.1, 17–24.
5. Mas E., Poza J., Ciriza J., Zaragoza P., Osta R., Rodellar C. (2016). Fundamento de la Reacción en Cadena de la Polimerasa (PCR). AquaTIC 15, pp. 70–78.
6. Tamay de Dios L., Ibarra C., Velasquillo C., (2013). Fundamentos de la reacción en cadena de la polimerasa (PCR) y de la PCR en tiempo real. Investigación en discapacidad 2(2), pp. 70–78.
7. Abd-Elsalam K.A., (2003). Bioinformatic tools and guideline for PCR primer design. African Journal of biotechnology 2(5), pp. 91–95.
8. Blanca J., Cañizares J., Ziarsolo P., Sánchez Matarredona D., "Bioinformatic & Genomic Group at COMAV Institute", Valencia, España. Recuperado de: http://bioinf.comav.upv.es/courses/intro_bioinf/multiple.html.
9. Rossainz M., Zúñiga S., Capel M, Pineda I., (2019). Representation of a convolutional neuronal network as a high-level parallel composition applied to the recognition of DNA sequences, EMSS 2019, Lisboa Portugal, pp 1-8.
10. Rossainz M., Capel M., (2017). Design and implementation of communication patterns using parallel objects. Especial edition, Int. J. Simulation and Process Modelling, Vol. 12, No. 1.

11. Danelutto M and Torquati M., (2014). Loop parallelism: a new skeleton perspective on data parallel patterns, in Proc. Of Intl. Euromicro PDP 2014. Parallel Distributed and Network-based Processing, Torino, Italy.
12. Rossainz M., Capel M., (2014). Approach class library of high-level parallel compositions to implements communication patterns using structured parallel programming. 26TH European Modeling & Simulation Symposium. Bordeaux, France.
13. Rossainz M., Capel M., (2008). A Parallel Programming Methodology using Communication Patterns named CPANS or Composition of Parallel Object. 20TH European Modeling & Simulation Symposium.Campora S. Giovanni. Italy.
14. Diaz Barrios H., Aleman Rivas Y., Cabrera Hernández L., et-al, (2015). "Machine Learning algorithms for Splice Sites classification in genomic sequences", Revista Cubana de Ciencias Informáticas, Volumen 9, Número 4. Recuperado de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992015000400012.
15. Barrientos Martínez R.E., Cruz Ramírez N., Acosta Meza H.G., et-al, (2009). "Árboles de Decisión como herramienta en el diagnóstico médico", Revista Médica de la Universidad Veracruzana, México. Volumen 9, Número 2, Recuperado de https://www.uv.mx/rm/num_anteriores/revmedica_vol9_num2/articulos/arboles.pdf.
16. Ebeling W., Frommel C., (1998). Entropy and predictability of information carriers. Biosystems 46(1-2), pp. 47–55.
17. Collins A.J., (2011). Automatically Optimising Parallel Skeletons, MSc thesis in Computer Science, School of Informatics University of Edinburgh, UK.
18. Ernsting S. and Kuchen H. (2012). Algorithmic skeletons for multi-core, multi-GPU systems and clusters, Int. J. of High-Performance Computing and Networking, Vol. 7, No. 2, pp.129–138.

Algoritmo genético para la localización de motivos de secuencias de ADN

Marcela Rivera Martínez, Luis René Marcial Castillo, Lourdes Sandoval Solís, Yesenia Guadalupe Romero Sánchez

Benemérita Universidad Autónoma de Puebla, Puebla, México
{marcela.rivera,luis.marcial,Lourdes.sandoval}@correo.buap.mx,
yesenia.romero.lcc@gmail.com

Resumen. El problema de búsqueda de motivos es de gran importancia en biología molecular, los motivos se generan a partir de alineamientos múltiples de secuencias de ADN. La relevancia de la localización es encontrar motivos mutados que conduzcan a la falta de producción de alguna proteína y puedan causar una enfermedad. El problema está clasificado como NP-completo. En este trabajo, se presenta un algoritmo genético para encontrar motivos en secuencias de ADN utilizando problemas sintéticos.
Palabras clave: Localización de Motivos, ADN, algoritmos genéticos, NP-Completo.

1. Introducción

Las secuencias tanto nucleotídicas como de proteínas contienen patrones o motivos que se han preservado a través de procesos evolutivos. A partir de dichas secuencias, se pueden inferir características importantes para la estructura o la función de moléculas [4, 6].

El problema de búsqueda de motivos es de gran importancia para la biología molecular, y sirve para el descubrimiento de secuencias reguladoras o promotores. El descubrimiento de motivos ayuda a identificar motivos mutados, responsables de la falta de producción de alguna proteína que pueda ocasionar alguna enfermedad [12].

Un motivo es una secuencia que se repite concierta frecuencia en regiones determinadas del ácido desoxirribonucleico (ADN). El problema de localización de motivos consiste en que, dado un conjunto de secuencias desalineadas de ADN, y dada la longitud del motivo, se deben localizar todas las ocurrencias que contienen las secuencias de entrada con la longitud estimada. No es necesario que todas las secuencias tengan ocurrencias del motivo y algunas de ellas podrán tener más de una ocurrencia. La localización de motivos es muy importante ya que por ejemplo puede revelar los factores de transcripción que controlan la expresión del gen en un ser vivo [15], o bien para distinguir entre células sanas y cancerígenas o para apoyar a los investigadores en el campo de la genética o biología [8].

La localización de motivos se encuentra clasificado como un problema NP-Completo [9], es decir, no se conoce un algoritmo de tiempo polinomial que pueda encontrar una

solución exacta, por lo tanto, se recurre a otras alternativas, como por ejemplo las heurísticas. Se han propuesto diferentes formas de abordar el problema de localización de motivos en las secuencias genéticas de ADN entre ellas están: la búsqueda exhaustiva [2], enfoque algorítmico de conteo de palabras [14], algoritmos meméticos [3], algoritmos inspirados en la naturaleza y métodos probabilísticos [8], finalmente también se encuentran algoritmos secuenciales en sistemas distribuidos [11]. En este trabajo se propone resolver el problema de localización de motivos usando un algoritmo genético secuencial con la finalidad de medir su desempeño en problemas sintéticos para posteriormente ser utilizado en problemas reales.

La estructura del documento es la siguiente: en la sección dos se describen en general los algoritmos genéticos y se presenta el esquema general del algoritmo, en la sección tres se proporciona una descripción del problema a resolver explicando detalladamente los elementos involucrados, en la sección cuatro se detallan los pasos del algoritmo para resolver el problema de localización de motivos, la sección cinco muestra las pruebas y resultados obtenidos en los problemas sintéticos, las conclusiones y el trabajo futuro están plasmados en el capítulo 6, finalmente se listan las referencias bibliográficas utilizadas en el desarrollo de este trabajo.

2. Algoritmo genético

Los algoritmos genéticos fueron introducidos por John Holland a finales de los 60´s inspirándose en el proceso observado en la evolución natural de los seres vivos. Son algoritmos de búsqueda basados en la mecánica de la selección natural y en la genética. Estos combinan la supervivencia de los individuos más aptos entre las cadenas de estructuras con un intercambio aleatorio para formar un algoritmo de búsqueda [7, 10].

Los algoritmos genéticos se basan en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes (unidad básica de codificación de cada uno de los atributos de un ser vivo) de un individuo, y que los atributos más deseables, es decir, los que le permiten a un individuo adaptarse mejor a su entorno, se transmiten a sus descendientes, cuando éste se reproduce sexualmente [5]. Los algoritmos genéticos están determinados por tres características fundamentales: selección de parejas, cruza y mutación. En cada problema de debe construir una función conocida como función de aptitud que es usada para evaluar a cada uno de los individuos y distinguir a los más aptos.

Los algoritmos genéticos en sus inicios representaban los problemas de manera binaria, con ceros y unos, actualmente la representación puede ser entera, real, entre otras de acuerdo al problema a tratar, de igual manera los operadores de un algoritmo genético se eligen de acuerdo al problema a resolver, razones por las cuales los algoritmos genéticos siguen siendo utilizados hoy en día.

3. Definición del problema

Se proponen artefactos para el modelado de contenido y modelo navegacional, como a continuación se detalla: El ácido desoxirribonucleico (ADN) es el material genético de todos los organismos celulares y de casi todos los virus. El ADN lleva la información necesaria para dirigir la síntesis de proteínas y la replicación. Cada molécula de ADN está constituida por dos cadenas o bandas formadas por un elevado número de compuestos químicos llamados nucleótidos. Estas cadenas forman una especie de escalera retorcida que se llama doble hélice. Cada nucleótido está formado por tres unidades: una molécula de azúcar llamada desoxirribosa, un grupo fosfato y uno de cuatro posibles compuestos nitrogenados llamados bases, las cuales son: Adenina (a), Guanina (g), Tiamina (t) y Citosina (c).

La doble hélice ilustrada en la figura 1 [18], es la descripción de la estructura de una molécula de ADN. Una molécula de ADN consiste en dos cadenas que serpentean, una alrededor de la otra, como una escalera de caracol. Cada cadena tiene una espina dorsal en la cual se alternan un azúcar y un grupo fosfato. A cada azúcar se une una de las cuatro bases: adenina, citosina, guanina o tiamina. Las dos cadenas se mantienen unidas por enlaces entre las bases nitrogenadas, adenina formando enlaces con la tiamina, y citosina con la guanina. La doble hélice se ha convertido en el icono para muchos, y rodea muchas discusiones acerca del pasado y el futuro de la ciencia. Es realmente una estructura impresionante. No se puede mirar a la doble hélice por mucho tiempo sin tener una especie de respeto hacia la elegancia de esta molécula de ADN en que se guarda la información genética. La forma de doble hélice es, básicamente, la manera en que todas las formas de vida están conectadas entre sí, porque todos utilizan esta misma estructura para la transmisión de esa información. se trata del increíble descubrimiento de Watson y Crick en 1953, pero se mantendrá probablemente en la historia como uno de los momentos científicos más importantes de todos los tiempos [20].

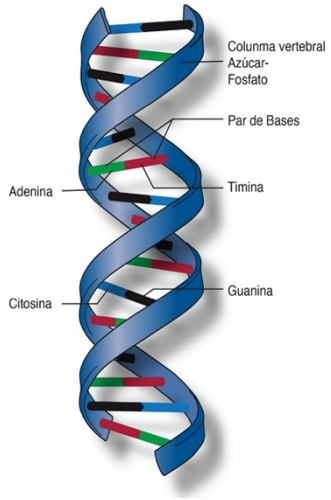


Fig. 1. La doble hélice.

El ácido ribonucleico (ARN) es una molécula similar a la de ADN. A diferencia del ADN, el ARN es de cadena sencilla. Una hebra de ARN tiene un eje constituido por un

azúcar y grupos de fosfato de forma alterna. Unidos a cada azúcar se encuentra una de las cuatro bases adenina, uracilo (u), citosina o guanina. Hay diferentes tipos de ARN en la célula: ARN mensajero (ARNm), ARN ribosomal (ARNr) y ARN de transferencia (ARNt). Más recientemente, se han encontrado algunos ARN de pequeño tamaño que están involucrados en la regulación de la expresión génica [21].

La adenina tiene la propiedad de que, cuando se encuentra en la doble hélice, siempre está formando pareja con la tiamina de la hebra opuesta. La adenina también está presente en otras partes de la célula, no sólo en el ADN o el ARN. También forma parte de la molécula adenosina trifosfato, la fuente energética de la célula por excelencia. Por lo tanto, la adenina juega un doble papel en la célula: sirve para construir ADN y ARN, y también se utiliza para almacenar energía en la célula [19].

La citosina es uno de los cuatro componentes básicos del ADN y el ARN, y es uno de los cuatro nucleótidos que son parte del código genético. La citosina tiene la propiedad única de unirse en la doble hélice frente a la guanina, uno de los otros nucleótidos. La citosina tiene otra propiedad interesante que no posee ninguno de los otros nucleótidos, y es que muy a menudo en la célula, la citosina puede tener un producto químico adicional ligado a ella; un grupo metilo. Esta metilación del ADN en citosinas se cree que ayuda a regular los genes, a activarse y desactivarse [22].

La guanina se aparea con la citosina en la doble hélice, por lo que verán pares “gc”; una en una hebra y la otra en la otra hebra. Los pares “cg” crean uniones más fuertes que los pares “at”, por lo que tramos largos de “cg” dan lugar a hebras más firmes que hélices con regiones “at”. Dentro de la molécula de ADN, las bases de tiamina se encuentran en una línea que forma enlaces químicos con las bases de adenina en la cadena opuesta [23].

La tiamina es uno de los componentes básicos del ADN. Es uno de los cuatro nucleótidos que se unen para hacer la larga secuencia que se encuentran en el ADN, de c, a, g y t. En la doble hélice, la tiamina se aparea con la adenina; el nucleótido a [25].

Una mutación es un cambio en la secuencia del ADN. Las mutaciones pueden ser el resultado de errores en la copia del ADN durante la división celular, la exposición a radiaciones ionizantes o a sustancias químicas denominadas mutágenos, o infección por virus. Las mutaciones de la línea germinal se producen en los óvulos y el esperma y puede transmitirse a la descendencia, mientras que las mutaciones somáticas se producen en las células del cuerpo y no se pasan a los hijos. Las mutaciones en realidad simplemente pueden ser un error cometido al copiar una secuencia de ADN. Algunas de ellas forman parte del ruido de fondo, ya que el proceso de replicación del ADN no es perfecto, de lo cual debemos estar contentos o no existiría la evolución. Pero una mutación también puede ser inducida por cosas como la radiación o por sustancias cancerígenas, de forma que puede aumentar el riesgo de padecer cáncer o defectos congénitos. En el fondo es bastante simple, no es más que una falta de ortografía inducida de la secuencia de ADN. Eso es una mutación [24].

Un dominio de unión al ADN es un dominio proteico independientemente plegado que contiene al menos un motivo estructural que reconoce hebras de ADN dobles o simples. Un dominio de unión al ADN puede reconocer una secuencia específica de ADN (una secuencia de reconocimiento) o tener una afinidad general hacia el ADN. Algunos dominios de unión al ADN pueden incluir ácidos nucleicos en su estructura [17].

El problema de búsqueda de motivos consiste en identificar pequeños sitios conservados en el ADN. La dificultad de este problema es que es un NP-Completo,

dado que se deben encontrar todos los posibles motivos en una secuencia de ADN la cual está conformada en base a cuatro elementos {a, g, c, t}, es decir, se tienen 4l posibilidades, donde l es la longitud del motivo. El problema de búsqueda de motivos se puede definir de la siguiente manera: Dado un conjunto de N secuencias cada una de longitud T y dada la longitud del motivo l, encontrar todas las ocurrencias del motivo de tamaño l que se encuentran implantadas en las N secuencias.

Existen diferentes métodos para resolver esta problemática, en general para resolver este problema existen cuatro grandes grupos de métodos: los enumerativos, de optimización determinística, de optimización probabilística y algoritmos inspirados en la naturaleza.

Los métodos enumerativos buscan exhaustivamente todos los posibles motivos en el espacio de búsqueda, esto para un modelo descriptivo específico, entre ellos se encuentran los métodos basados en diccionarios, los cuales buscan todas las ocurrencias de n secuencias y calculan cuál es la que tiene una mayor presencia.

Los métodos basados en optimización determinística son aquellos que pueden optimizar simultáneamente matrices de posiciones de peso para describir determinados motivos, entre ellos encontramos a MEME (Multiple Expectation Maximization for Motif Elicitation), el cual busca en los motivos repetidos dentro de una base de datos de genes el mejor resultado y luego itera hasta lograr convergencia, generalmente el algoritmo MEME se queda atrapado en óptimos locales [1].

Los métodos de optimización probabilística utilizan una búsqueda estocástica de entrada, luego iteran varias veces y cada secuencia se evalúa con el modelo inicial. En cada iteración, el algoritmo decide de forma probabilística si agrega o elimina un sitio, al terminar la iteración se ajustan las probabilidades y se calcula de nuevo el modelo. Un ejemplo representativo de este tipo de métodos es Gibbs Sampler [9, 15].

Los algoritmos inspirados en la naturaleza se han presentado como una alternativa de resolver problemas complejos y dinámicos usando un tiempo y costo razonable. Estos algoritmos simulan el comportamiento de insectos u otros animales al resolver sus problemas propios, la ventaja de estos algoritmos es que tratan de escapar de los óptimos locales, los algoritmos más comunes de este campo son: colonia de abejas, colonia de luciérnagas, colonia de hormigas y PSO (Particle Swarm Optimization) [8].

Computacionalmente el problema de alinear secuencias se define como sigue: Dados un alfabeto $\sum$={a,c,g,t}, un conjunto de secuencias de entrada sobre el alfabeto {s1, ..., sk}cada una de longitud T y un motivo P con longitud l, el objetivo es encontrar todas las ocurrencias del motivo de tamaño l que se encuentran implantadas en las k secuencias. Dado un motivo, éste puede coincidir con la secuencia en diversas partes, sin embargo, la coincidencia debe ser consecutiva.

Ejemplo. Sean el alfabeto Σ={a, c, g, t}, las secuencias s1=acgtga, s2=acgaga, s3=aacgta, s4=gctacg, y la longitud del motivo igual a 3. Entonces, usando fuerza bruta se encontraría la solución del problema de la siguiente manera: la subcadena acg de s1 aparece en s2, s3 y s4 dando un total de 4/4 =1 (la división entre 4 es porque existen 4 secuencias), la subcadena cgt de s1 aparece 1, 0, 1, 0 respectivamente en las secuencias s1, s2, s3 y s4 dando un total de 2/4 = 0.5,la subcadena gtg de s1 aparece 1, 0, 0, 0 en las secuencias s1, s2, s3 y s4dando un total de 1/4 = 0.25, la subcadena tga de s1 aparece 1, 0, 0, 0 en las secuencias s1, s2, s3 y s4dando un total de 1/4 = 0.25. La subcadena acg de s2 aparece en s1, s3 y s4 dando un total de 4/4 =1, la subcadena cga de s2 aparece 0, 1, 0, 0 respectivamente en las secuencias s1, s2, s3 y s4 dando un total de 1/4 = 0.25,

la subcadena gag de s2 aparece 0, 1, 0, 0 en las secuencias s1, s2, s3 y s4 dando un total de 1/4 = 0.25, la subcadena aga de s2 aparece 0, 1, 0, 0 en las secuencias s1, s2, s3 y s4 dando un total de 1/4 = 0.25. La subcadena aacde s3 aparece 0, 0, 1, 0 en s1, s2, s3 y s4 dando un total de 1/4 = 0.25, la subcadena acg de s3 aparece 1, 1, 1, 1 respectivamente en las secuencias s1, s2, s3 y s4 dando un total de 4/4 = 1.0,la subcaden acgt de s3 aparece 1, 0, 1, 0 en las secuencias s1, s2, s3 y s4 dando un total de 2/4 = 0.5, la subcadena gta de s3 aparece 0, 0, 1, 0 en las secuencias s1, s2, s3 y s4 dando un total de 1/4 = 0.25. Finalmente, la subcadena gctde s4 aparece 0, 0, 0, 1en s1, s2, s3 y s4 dando un total de 1/4 = 0.25, la subcadena cta de s4 aparece 0, 0, 0, 1 respectivamente en las secuencias s1, s2, s3 y s4 dando un total de 1/4 = 0.25,la subcadena tac de s4 aparece 0, 0, 0, 1 en las secuencias s1, s2, s3 y s4 dando un total de 1/4 = 0.25, la subcadena acg de s4 aparece 1, 1, 1, 1 en las secuencias s1, s2, s3 y s4 dando un total de 4/4 = 1.0. Por lo tanto, el valor máximo es 1 y el motivo es acg que se encuentra implantada en todas las secuencias y sería la solución del problema planteado.

4. Algoritmo para resolver el problema de localización de motivos en secuencias de ADN

La forma de calcular si el motivo se encuentra dentro de la secuencia es de la siguiente forma:

$$FS(S_m, P_n) = \sum_{i=1}^{l} emparejamientos(S_{mi}, P_{ni})/\, k \qquad (1)$$

donde:

$$emparejamiento(S_{mi}, P_{ni}) = \begin{cases} 1 & \text{si } S_{mi} = P_{ni} \\ 0 & \text{si } S_{mi} \neq P_{ni} \end{cases}$$

En el cual *m* es el índice de las secuencias que puede tomar valores de *1* a *k, n* es el índice de los patrones del motivo, *l* es la longitud del motivo, *i* es la posición dentro del motivo que puede tomar valores de *1* a *l*, finalmente, *k* es la cantidad de secuencias.

La propuesta de este trabajo, para encontrar motivos en secuencias de ADN sigue los pasos del esquema general de un algoritmo genético. A continuación, se detallan cada uno de los puntos del algoritmo genético [7].

Definir parámetros y función de aptitud. Los parámetros utilizados son los símbolos para los cuatro posibles compuestos nitrogenados: adenina, citosina, guanina, y tiamina. La función de aptitud que permite clasificar a los individuos según su aptitud es dada por la fórmula 1.

Representación de parámetros. Los parámetros se representan con los símbolos “a”, “c”, “g”, y “t”.

Generar la población inicial. El algoritmo acepta como entrada un conjunto de cadenas o secuencias *k* (donde *k* debe ser de la forma 2*i* con *i* mayor que 2 y entera) cada una de una misma longitud j, la población inicial se genera tomando elementos del alfabeto de manera aleatoria, donde cada cadena será de longitud *l*, es decir, de la longitud de la cual se desean encontrar motivos. Ejemplo: Dado *l* =4, se pueden generar elementos de la población inicial como PobInicial = {acg, cgg, agt, tac}, lo cual

significa que se tienen 4 motivos de longitud 3 y los elementos de esos motivos son tomados del alfabeto Σ={a, c, g, t}.

Evaluar cada individuo en base a la función aptitud. Cada uno de los elementos de la población inicial se evalúan usando la fórmula 1.

Seleccionar parejas. La forma de seleccionar parejas que se utilizó fue la elitista, la cual consiste en elegir los mejores individuos de la población, de acuerdo a su función de aptitud dada por (1). Es decir, los individuos evaluados son ordenados en base a su aptitud, de los más aptos a los menos aptos obteniendo el orden consecutivo 1, 2, 3, 4, 5, 6,… Las parejas seleccionadas son el individuo 1 con el individuo 2, el individuo 3 con el individuo 4, el individuo 4 con el individuo 6, etcétera. En la implementación computacional sólo la mitad de los individuos tienen pareja siendo estos los más aptos y se les permite reproducirse.

Realizar la reproducción. Para el desarrollo del problema se considera la reproducción o cruza por medio de dos puntos esto quiere decir que se toman dos puntos distintos en cada padre, y se intercambian los alelos entre los padres para generar el mismo número de individuos, tomando en cuenta los valores antes del punto de cruza del primer padre, los valores medios del segundo padre y los valores finales del primer padre, para la generación de uno de los hijos, para la generación del otro hijo se toman en cuenta los valores antes del punto de cruza del segundo padre, los valores medios del primer padre y los valores finales del segundo padre. Ejemplo: Dado la pareja, padre 1 = “agtcatdg” y padre 2 = “gtactact” ambos de longitud 8, se generan aleatoriamente dos valores enteros entre 1 y 7 supóngase que se obtienen los números3 y 5, por lo que el primer hijo se obtiene concatenando los símbolos de las posiciones 1 a 3 del padre 1, seguido de los símbolos de las posiciones 4 a 7 del padre 2 y finalizando con la concatenación de los símbolos de las posiciones 8 a 8 del padre 1, por lo que el hijo 1 sería la cadena “agtctacg”. El segundo hijo se obtiene concatenando los símbolos de las posiciones 1 a 3 del padre 2, seguido de los símbolos de las posiciones 4 a 7 del padre 1 y finalizando con la concatenación de los símbolos de las posiciones 8 a 8 del padre 2, por lo que el hijo 2 sería la cadena “gtacatdt”. Al finalizar la reproducción se obtiene la misma cantidad de hijos que lo que se tiene de padres.

Realizar la mutación. La mutación que se implementa en este trabajo es una mutación por reemplazamiento, donde se elige de manera aleatoria una posición del motivo y se remplaza por algún elemento aleatorio del alfabeto. Sólo a un porcentaje de los motivos se les aplica la mutación. Ejemplo: Dado el motivo “agtctcag”, de longitud 8, se genera aleatoriamente un valor entero entre 1 y 8supóngase que se obtiene el número 2, por lo que, se reemplaza el símbolo “g” por algún elemento del alfabeto que es seleccionado de modo aleatorio, en este caso, se supone que es el símbolo “c”, por lo que, el nuevo motivo mutado será la cadena “actctag”. La mutación sólo es aplicable a los hijos que se obtienen de la reproducción.

¿Convergencia? Existen varias formas de salir de un algoritmo genético: por tiempo, por cantidad de generaciones, por población sin cambio, etcétera. En este trabajo, el criterio de convergencia usado es en base a la cantidad de generaciones preestablecidas por el usuario. Si no se ha cumplido con el criterio de salida del algoritmo genético, se renueva la nueva población que se usará en la siguiente generación. En la implementación computacional de este trabajo, se hace de la manera siguiente: se conjuntan los padres, hijos e individuos que no pudieron reproducirse, se evalúan los hijos usando la fórmula de la ecuación (1) para obtener su aptitud. Se ordenan padres,

hijos y los que no pudieron reproducirse según su aptitud y se desecha el tercio de los individuos menos aptos, de esta forma no se incrementa el tamaño de la población de una generación a otra, en la siguiente generación se ejecutan los pasos: seleccionar parejas, realizar la reproducción y realizar mutación.

5. Pruebas y resultados

Los resultados obtenidos bajo la implementación en Octave [16], se reportan en la tabla 1 es donde: la primera columna contiene la cantidad de secuencias que el programa recibe como entrada, la longitud de dichas secuencias está colocado en la segunda columna, en la tercera columna se reporta la longitud de los motivos a encontrar y finalmente la cuarta columna muestra la cantidad de motivos encontrados por el algoritmo genético programado. Todos los problemas de prueba se corrieron con 50 generaciones y un porcentaje de mutación de 0.6.

Tabla 1. Resultados de las pruebas.

Cantidad de secuencias k =	**Longitud de las secuencias l =**	**Longitud de los motivos a encontrar**	**Cantidad de motivos válidos encontrados**
6	8	3	2
4	11	3	2
5	9	4	2
6	12	4	2

Como puede observarse de la tabla 1, en todos los casos la cantidad de motivos válidos que se encontraron fue de dos, la cantidad de motivos válidos en todos los problemas prueba era de tres, por lo cual se considera que funciona bastante bien la propuesta del algoritmo genético para este tipo de problemas.

6. Conclusiones y trabajo futuro

Los resultados obtenidos (tabla 1), muestran que la propuesta de este trabajo localiza motivos con una buena aproximación a los motivos reales de una secuencia.
Como trabajo futuro, se probará la propuesta con problemas reales tomados de bases de datos de secuencias de ADN. Además, se pretende realizar diversas pruebas con el fin de lograr una buena calibración de los parámetros usados en el algoritmo genético con el fin de obtener mejores resultados y comparar los resultados obtenidos con algunas otras técnicas para localizar motivos. Finalmente se pretende realizar una implementación paralela del algoritmo genético.

6 Referencias

1. Bailey, T. L., Elkan, C.: Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Biopolymers, Proceeding International First Conference Intelligent Systems for Molecular Biology, pp. 28-36 (1994)
2. Brazma, A., Jonassen, I., Gilbert, D.: Approaches to the automatic discovery of patterns in biosequences, Journal of Computational Biology, Vol. 5, pp. 279-305 (1998)
3. Caldonazzo, J. M., Kashiwabara, K., Sanches, A. S.: Sequence motif finder using memetic algorithm, BMC Bioinformatics, pp. 1-13 (2018)
4. Carvalho, A., Freitas, A., Oliveira, A., Sagot, M.: An efficient algorithm for the identification of structured motifs in dna promoter sequences. IEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 3, pp. 126-140 (2006)
5. Coello, C. A.: Introducción a los Algoritmos Genéticos, Soluciones Avanzadas, Tecnologías de Información y Estrategias de Negocios, VoL. 17, pp. 5-11 (1995)
6. Datta, S., Patel, M., Patel, D., Singh, U.: Distinct DNA Sequence Preference for Histone Occupancy in Primary and Transformed Cells, Cancer Informatics, Vol. 18: 1–15 (2019)
7. Haup, R. L., Haup, S. E.: Practical Genetic Algorithms, John Wiley & Sons, Inc (1998)
8. Hashim, F. A., Mabrouk, M.S., Al-Atabany, W.: Review of Different Sequence Motif Finding Algorithms, Avicenna Journal of Medical Biotechnology, Vol. 11, No. 2, pp. 130-14 (2019)
9. Karaoglu, N., Maurer, S., Manderick, B.: GAMOT: An efficient genetic algorithm for finding challenging motifs in dna sequences. Proceedings of 3rd Annual Stallite Workshop on Regulatory Genomics, pp. 4-12 (2006)
10. Malik A.: A Study of Genetic Algorithm and Crossover Techniques, International Journal of Computer Science and Mobile Computing, Vol.8 Issue.3, pp. 335-344 (2019)
11. Sarumia, O.A., Leungb, C.K, Adetunmbi, A. O.: Spark-based data analytics of sequence motifs in large omics data, Elsiever Procedia Computer Science, Vol. 126, pp. 596–605 (2018)
12. Strachan, T., Andrew P. R.: Human molecular genetics 2, BIOS Scientific Publishers. (1999)
13. Thompson, W., Rouchka, E. C., Lawrence C.: E.: Gibbs Recursive Sampler: Finding Transcription Factor Binding Sities, Nucleic Acids Research, Vol. 31, pp. 3580-3585 (2003)
14. Tompa, M.: An Exact Method for Finding Short Motifs in Sequences, with Application to the Ribosome Binding Site Problem. Proceedings International Conference Intelligent System Molecular Biology, pp. 262-271 (1999)
15. Train, N. T., Huang, Ch.: MODSIDE: a motif discovery pipeline and similarity detector, BMC Genomics, Vol 19, pp 1-9 (2018)
16. Sin autor: Disponible en Web: https://www.gnu.org/software/octave/, Consultado26, 08 (2020)
17. Sin autor: Disponible en Web: https://es.wikipedia.org/wiki/ Dominio_de_unión_al_ADN, Consultado 26, 08 (2020)

18. Collins, F. S.: Disponible en Web: https://www.genome.gov/sites/default/files/tg/es/illustration/Doble_he__lice.jpg, Consultado 26, 08 (2020)
19. Brody, L. C.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/Adenina, Consultado 26, 08 (2020)
20. Austin C. P.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/ADN-acido-Desoxirribonucleico, Consultado 26, 08 (2020)
21. Biesecker, L. G.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/ARN, Consultado 26, 08 (2020)
22. Brody, L. C.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/Citosina, Consultado 26, 08 (2020)
23. Brody, L. C.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/Guanina, Consultado 26, 08 (2020)
24. Collins, F. S.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/Mutacion, Consultado 26, 08 (2020).
25. Brody, L. C.: Disponible en Web: https://www.genome.gov/es/genetics-glossary/Timina, Consultado 26, 08 (2020)

Comparación de Obras Literarias

Luis E. Morales-Márquez, Maya Carrillo-Ruiz, José L. Hernández-Ameca
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Ciudad Universitaria. Blvd. Valsequillo y Av. San Claudio, Col. Jardines de San Manuel, Puebla, México. CP 72570.
luise.morales@viep.com.mx, maya.carrillo@correo.buap.mx, joseluis.hdzameca@correo.buap.mx

Resumen. La Real Academia Española define el término *comparar* como: "Fijar la atención en dos o más objetos para descubrir sus relaciones o estimar sus diferencias o su semejanza". Este trabajo de investigación considera obras literarias para evaluar sus semejanzas y diferencias con la finalidad de determinar si aportan o no información similar. Se utilizó el tamaño del texto, cantidad de tokens, cantidad de tipos, colocaciones, concordancias, revisión de contextos similares, riqueza del vocabulario y clasificación por géneros como criterios de comparación. Para ejemplificar la utilidad de los criterios mencionados se compararon dos obras y se encontró que sólo tenían parecido al encontrarse dentro del mismo par de géneros, así que no pudo decirse que aportaran conocimiento similar, lo que coincide con el juicio experto.

Palabras Clave: Comparación de Obras Literarias, Riqueza del Vocabulario, Colocaciones, Concordancias.

1 Introducción

La Real Academia Española define el término comparar como: "Fijar la atención en dos o más objetos para descubrir sus relaciones o estimar sus diferencias o su semejanza". En general, una comparación arroja dos grupos de resultados: todos aquellos elementos que son iguales o altamente parecidos y aquellos elementos de los escritos que difieren en su totalidad o en un grado muy alto.
La forma lógica para identificar semejanzas y diferencias entre dos objetos puede ser establecida con los siguientes pasos:

1. Determinar el conjunto de características más relevantes que componen a cada objeto de estudio y encontrar aquellas que compartan, éstas serán los criterios de comparación.
2. La entidad evaluadora es una persona que elige los criterios que se consideren adecuados para el propósito de la comparación.
3. Para cada uno de los criterios de comparación, determinar el valor o categoría a la que pertenece el contenido de los objetos comparados.

4. Para cada uno de los criterios de comparación, la experiencia y conocimiento de la entidad evaluadora, determinará el grado de similitud entre los objetos bajo estudio.
5. La entidad evaluadora determinará con base en el conjunto de criterios, el resultado de la comparación y emitirá una conclusión que satisfaga su propósito.

Las obras literarias comúnmente son objeto de comparación, por ejemplo, para determinar si un texto es valioso por presentar conocimiento que otros textos no aportan. Este trabajo considera dos obras literarias y evaluar sus semejanzas y diferencias.
Los criterios de evaluación se seleccionaron por la información que arrojan en cuanto a vocabulario, género y uso de las palabras. Dichos criterios han sido utilizados en diversos trabajos. En [1] los autores buscan establecer una relación entre la riqueza léxica y la utilización de colocaciones con el dominio del italiano por parte de estudiantes de niveles L1 y L2 comparándolos con hablantes nativos. En [2] los autores utilizan diferentes parámetros como la longitud del texto, el tamaño del vocabulario, riqueza del vocabulario para investigar si realmente el lenguaje utilizado en las obras de teatro de Shekespeare era notoriamente más rico que el de sus contemporáneos, llegando a la conclusión de que no fue así. En [3] los autores analizan la obra del Fantasma de la Opera utilizando la proporción Token-Tipo estandarizada, entropía y taza de repetición relativa para concluir que el lenguaje utilizado es coloquial y con pequeños cambios. El presente trabajo se organiza de la siguiente manera: en la sección 2 se presentan los criterios de comparación utilizados, en la sección 3 el método empleado, en la 4 los resultados obtenidos y finalmente en la 5 las conclusiones y trabajo futuro.

2 Criterios de comparación

Los textos de cultura popular seleccionados para la investigación fueron: *The Book of Genesis* (conocido por ser el primer libro del antiguo testamento de las Sagradas Escrituras) y *Monty Python and the Holy Grail* (escrito por Graham Chapman originalmente publicado en 1975) a los que se llamará *Genesis* y *Holy*, respectivamente. Estas obras fueron seleccionadas por ser parte de los recursos disponibles en la biblioteca de NLTK y su alcance dentro de la cultura popular. Además, estas obras se leyeron previamente para conocer su contenido y temática, esto permitiría tener la certeza de que los resultados obtenidos coincidirían con los resultados esperados. De esta manera estaríamos en posibilidades de establecer si los criterios propuestos y el método descrito eran de utilidad para comparar obras y dar conclusiones confiables, aun cuando no se tuviese conocimiento del contenido del escrito. Ambos textos elegidos, se encuentran escritos en inglés y han sido procesados en el mismo idioma, al ser textos relativamente cortos se analizaron en su totalidad con el objetivo de obtener una comparación lo más precisa posible e ilustrar las diferencias encontradas. Es importante mencionar que en este trabajo se propone un conjunto de criterios de comparación de obras literarias y se describe un conjunto de pasos para aplicarlos. Estos

criterios proporcionan valores numéricos de mediciones tanto léxicas como contextuales de las obras. Los criterios seleccionados para la comparación fueron:

Tamaño del texto: Normalmente, la primera característica que resalta en una obra es la extensión de la misma, no sería correcto evaluar este criterio fijándose en atributos como número de páginas o capítulos pues no todos los escritos están estandarizados bajo el mismo formato. Entonces para medirá la extensión de la obra se consideró el número de palabras que la componen, al cual llamaremos *Tamaño del texto*.

Tokens: Para refinar los elementos discriminativos, fue necesario eliminar las palabras vacías (palabras sin significado como artículos, pronombres, preposiciones, etc.), posteriormente se eliminaron los elementos no alfanuméricos, obteniendo así un conjunto de palabras útiles que llamaremos *Tokens*.

Tipos: Cuando se obtienen todos los tokens y se eliminan sus repeticiones(vocabulario), a cada uno de ellos se le considera un *Tipo*.

Colocaciones: Una colocación es una secuencia de palabras (normalmente un bigrama) donde ninguna de las palabras que la conforman puede ser sustituida por un sinónimo o palabra que tenga el mismo sentido, por ejemplo, *vino tinto* no puede ser sustituido por *vino rojizo* [4], como seres humanos tenemos la capacidad de entender el significado de esta última expresión, sin embargo, resulta tema de investigación en procesamiento de lenguaje natural. Las colocaciones en un libro pueden ayudar a determinar si este es sencillo de entender y sugerir ligeramente que contextos maneja dicha obra. Para obtener las colocaciones no se eliminan las palabras vacías.

Concordancia: La concordancia dentro de un texto permite buscar las ocurrencias de una palabra dentro del texto y arrojar un segmento de la línea que la incluye, esto deja al lector observar con qué sentido se utiliza dicha palabra, esto permite ilustrar la diversidad de usos de un término en distintos textos [4].

Contextos similares: A partir de un token seleccionado para la búsqueda de concordancias, se pueden obtener frases de la obra que se refieren al token en cuestión, permitiendo evaluar si puede tener connotaciones positivas o negativas y otros contextos que puedan estar relacionados [4]. Esta búsqueda generalmente se hace siempre con el mismo token en las obras comparadas.

Riqueza del vocabulario: El estudio de la riqueza del vocabulario ha sido de alto impacto en la investigación literaria, tanto para definir estilos como para comprender la cantidad y complejidad de las palabras utilizadas en un determinado texto [5], este indicador puede resultar de gran importancia para algunos investigadores pues es natural pensar que una mayor riqueza del vocabulario implicaría que se podría extraer mayor conocimiento de una obra

Existen varios métodos para obtener la riqueza del vocabulario, para esta investigación se ha utilizado un método clásico: la *Proporción Token-Tipo Estandarizada* o STTR *(standardised type/token ratio*), este cálculo determina la proporción de tokens distintos usados por cada determinado número de palabras, por ejemplo, si un texto está formado por dos mil palabas, se toman grupos de mil palabras y se cuenta cuantas de ellas son distintas, el resultado de cada grupo es promediado para obtener un porcentaje. El tamaño de la muestra N es determinado por la persona que ejecutará el análisis, el N

recomendado para obras largas es de mil elementos, sin embargo, para propósitos de este trabajo se ha utilizado N con valor de 100 al estar evaluando obras cortas y con el objetivo de mejorar la precisión del cálculo, es importante aclarar que este valor N es el que justifica la parte estandarizada del indicador [5].
Puesto que STTR puede definirse como el promedio simple de un conjunto de porcentajes, se ha calculado igualmente un indicador más descriptivo conocido como Lambda (λ) definido por:

$$\lambda = \frac{\sum_{i=1}^{V-1} \sqrt{(f_i - f_{i+1})^2 + 1}}{N} \quad (1)$$

Donde N es el tamaño del texto, f_i son las frecuencias absolutas ordenadas para $(i = 1,2,\dots,V)$ y V es el tamaño del vocabulario [5].

Clasificación por géneros: Cada obra está siempre etiquetada bajo uno o más géneros literarios y casi siempre esta etiqueta está definida por el autor, sin embargo, algunas partes del texto pueden incluir temas que podrían bien catalogarse dentro de otro género, esto sugiere que de cada obra puede obtenerse información de diversas temáticas que puede pasar desapercibida por estar englobadas dentro del género de la obra completa. Se seleccionaron las categorías del Brown Corpus para este ejercicio por su diversidad de géneros, sus casi 60 años de vigencia y por estar alimentado por más de 500 fuentes distintas de información [4]. Las categorías del Brown Corpus son las siguientes:

Adventure, belles_lettres, editorial, fiction, government, hobbies, humor, learned, lore, mystery, news, religion, reviews, romance, science_fiction.

La clasificación de géneros requiere evaluar un conjunto de tokens que serán buscados en dichas categorías, con base en el número de apariciones de los mismos, se determina el género al que pertenece la obra. En este trabajo se consideraron los cien tokens con mayor frecuencia de aparición en cada obra, a los que se nombran *palabras populares*.

3 Método

Inicialmente las obras literarias se almacenaron dentro de un arreglo compuesto de todas las cadenas separadas por espacios y saltos de línea, posteriormente se eliminaron de la lista todas las palabras que no tienen significado y los signos de puntuación, considerando la lista de palabras vacías proporcionada por NLTK para obtener los tokens.
Resulta evidente que no todos los tokens son únicos, pues es imposible escribir sin repetir palabras, entonces este grupo no puede ser utilizado como una única medida de variedad de lenguaje, consideramos entonces convertir estos tokens en tipos usando una técnica de conteo de los tokens repetidos, la lista final es un arreglo ordenado con el número de incidencias dentro del texto para cada tipo.

Finalmente se obtuvieron los dos criterios numéricos de comparación, es decir, la riqueza del vocabulario y la clasificación por géneros. La riqueza del vocabulario es un porcentaje calculado mediante una especie de varianza estandarizada que ocupa la cantidad de tipos, mientras que el STTR es una proporción más simple que solo divide la cantidad de tipos por el tamaño del texto. Por otro lado, en la clasificación por géneros se seleccionan las palabras con mayor incidencia en la obra, el número de palabras a elegir es determinado por la entidad evaluadora. Seleccionadas las palabras, estas se buscan dentro de un banco de palabras que asocia a cada palabra una etiqueta que establecen la categoría de la misma. Finalmente se genera una estadística descriptiva breve.

El procesamiento anterior supone una pérdida total del contexto y sentido del conocimiento de la obra pues el resultado es un conjunto de palabras sin ninguna relación entre ellas. Para establecer contextos, concordancias y colocaciones se utilizó la obra original sin ningún procesamiento previo. Las colocaciones se obtuvieron mediante un método que compara cada bigrama posible dentro del texto contra una biblioteca de colocaciones existentes dentro del idioma, la lectura de cada bigrama obtenido, ayuda a la construcción del contexto en que se desarrolla la obra.

La obra original, sin alteraciones, se utilizó para establecer los criterios de concordancias y contexto para un token específico. En general cualquier token es útil como argumento de búsqueda. Aunque estos criterios están relacionados, no producen los mismos resultados y no aportan el mismo tipo de conocimiento. Las concordancias indican el significado que el token tiene dentro del texto y el contexto retorna un segmento de la línea que contiene al token y queda como tarea de la entidad evaluadora establecer cuál es el entorno en el que se está utilizando la palabra.

4 Resultados

El procesamiento se llevó a cabo utilizando la biblioteca NLTK 3.5 de Python 3.8, y la tecnología de WordSmith Tools.

Se extrajeron todas las palabras de cada obra, el total de elementos incluyendo puntuación y palabras vacías se muestra en la Tabla 1.

Tabla 1. Palabras por Obra literaria.

Obra	Número de palabras
Génesis	44 564
Holy	16 967

Una vez obtenido el tamaño del texto, se eliminaron las palabras vacías y puntuación el número de tokens obtenidos se muestra en la Tabla 2.

Tabla 2. Tokens para cada obra literaria

Obra	Tokens
Génesis	18 335
Holy	6 692

Como se mencionó anteriormente, un criterio importante es el número de palabras diferentes que componen a cada obra literaria, mismo que se muestra en la Tabla 3. Puede notarse que el vocabulario de las obras esta menos distante en comparación a las palabras y los tokens obtenidos.

Tabla 3. Tipos por cada obra literaria

Obra	Tipos
Génesis	2495
Holy	1667

En la Figura 1, pueden observares las palabras, tokens y tipos para cada obra

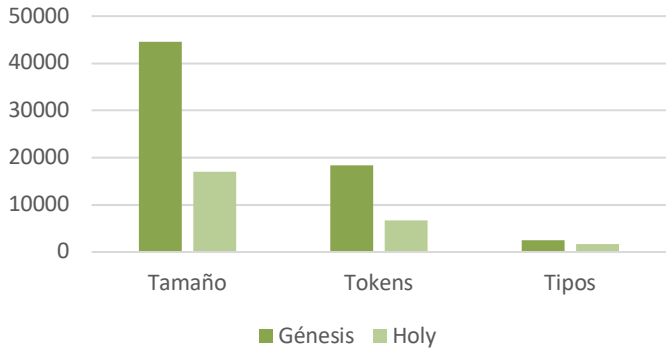


Fig. 1. Comparación del total de cada categoría.

Las colocaciones dentro del español no son tan comunes como en el inglés, este último idioma se ve beneficiado con su uso, pues apoyan a la descripción de entidades, o incluso marcar el ambiente en el cual se desarrolla una historia, a continuación, se presentan las colocaciones halladas en cada obra.

A) Génesis:

said unto; pray thee; thou shalt; thou hast; thy seed; years old; spake unto; thou art; LORD God; every living; God hath; begat sons; seven years; shalt thou; little ones; living creature; creeping thing; savoury meat; thirty years; every beast

B) Holy:

BLACK KNIGHT; clop clop; HEAD KNIGHT; mumble mumble; Holy Grail; squeak squeak; FRENCH GUARD; saw saw; Sir Robin; Run away; CARTOON CHARACTER; King Arthur; Iesu domine; Pie Iesu; DEAD PERSON; Round Table; clap clap; OLD MAN; dramatic chord; dona eis

Los siguientes resultados corresponden a la revisión de concordancia del token *book*, se eligió este token por ser una palabra lo bastante genérica como para poder ser utilizada en variedad de contextos y con diferentes significados lo cual la hace una opción que permite ver las distintas formas en que cada texto puede utilizar un mismo concepto.

A) Génesis:

"… the name of the LORD . This is the book of the generations of Adam . In the…"

B) Holy:

"… my liege . ARTHUR : Consult the Book of Armaments ! BROTHER MAYNARD : Arm…"

Cuando se analizan tanto la concordancia como la revisión de contexto, siempre se utiliza como argumento el mismo token, en síntesis, la concordancia evalúa la forma en la que se utiliza el token y los contextos indican situaciones en las que se utiliza el término seleccionado. A continuación, se muestran los contextos obtenidos del uso del token *book*.

A) Génesis:

"…beginning god face spirit waters good day firmament midst place land…"
"…gathering herb fruit tree days stars life fowl blessed…"

B) Holy:

"…castle knights court snows kingdom back lady decision case bosom…"
"…strength bridge ways name dragon battle ferocity tale keepers…"

Sería lógico pensar que una mayor cantidad de tokens desemboca en una riqueza de vocabulario superior, casi siempre se cumple esta proposición y los resultados obtenidos son un ejemplo de ello. Sin embargo, esta diferencia no es tan significativa como lo sugiere la cantidad de tipos mostrada anteriormente. La Tabla 4 contiene los valores, para la riqueza del vocabulario, determinados con el análisis. Las Figuras 2 y 3 muestran los mismos resultados gráficamente.

Tabla 4. Riqueza del vocabulario según dos formas de evaluación

Obra	STTR	λ
Génesis	0.61 (61%)	1.21
Holy	0.57 (57%)	1.13

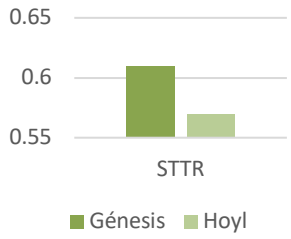


Fig. 2. STTR

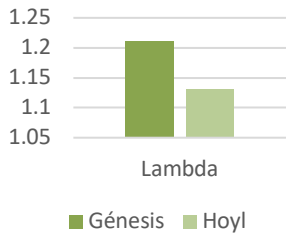


Fig. 3. Lambda

Finalmente se obtuvieron los conjuntos de palabras populares para cada obra. Las frecuencias de dichas palabras se muestran en las Tablas 5 y 6.

A) Génesis.

'unto', 'said', 'thou', 'thy', 'shall', 'thee', 'god', 'lord', 'father', 'land', 'jacob', 'came', 'joseph', 'son', 'sons', 'upon', 'abraham', 'behold', 'man', 'earth', 'went', 'wife', 'years', 'name', 'called', 'ye', 'let', 'us', 'every', 'brother', 'pharaoh', 'also', 'hand', 'pass', 'house', 'took', 'hath', 'brethren', 'saying', 'go', 'isaac', 'come', 'shalt', 'egypt', 'esau', 'day', 'made', 'one', 'give', 'begat', 'men', 'children', 'days', 'hundred', 'brought', 'seed', 'abram', 'saw', 'hast', 'bare', 'two', 'seven', 'laban', 'take', 'daughters', 'cattle', 'well', 'bring', 'make', 'gave', 'pray', 'face', 'eat', 'therefore', 'place', 'blessed', 'daughter', 'old', 'good', 'field', 'sent', 'rachel', 'forth', 'may', 'canaan', 'put', 'city', 'noah', 'servant', 'spake', 'israel', 'thing', 'lived', 'yet', 'found', 'eyes', 'servants', 'away', 'sarah', 'done'

Tabla 5. Frecuencias de tokens

Género	Frecuencia
Aventura	73
Bellas Letras	81
Editorial	73
Ficción	73
Gobierno	61
Pasatiempos	68
Humor	64
Aprendizaje	66
Ciencia	74
Misterio	64
Noticias	71
Religión	78
Reseñas	65
Romance	72
Ciencia Ficción	55

B) Holy.

'arthur', 'oh', 'launcelot', 'knight', 'galahad', 'father', 'sir', 'ni', 'bedevere', 'knights', 'well', 'head', 'ha', 'robin', 'right', 'guard', 'yes', 'villager', 'boom', 'come', 'uh', 'witch', 'clop', 'away', 'grail', 'king', 'one', 'black', 'burn', 'french', 'tim', 'us', 'look', 'singing', 'scene', 'get', 'dead', 'mumble', 'go', 'music', 'squeak', 'herbert', 'got', 'tell', 'hello', 'camelot', 'holy', 'castle', 'dennis', 'soldier', 'must', 'man', 'shall', 'zoot', 'going', 'heh', 'brave', 'run', 'stop', 'three', 'say', 'think', 'see', 'old', 'concorde', 'master', 'could', 'please', 'guests', 'bridgekeeper', 'clang', 'know', 'saw', 'crowd', 'name', 'narrator', 'god', 'cartoon', 'hee', 'back', 'like', 'good', 'shut', 'let', 'um', 'dingo', 'yeah', 'bring', 'make', 'shh', 'clap', 'maynard', 'cart', 'stops', 'woman', 'thank', 'questions', 'customer', 'want', 'officer'

Tabla 6. Frecuencias de tokens

Género	Frecuencia
Aventura	60
Bellas Letras	72
Editorial	57
Ficción	59
Gobierno	45
Pasatiempos	55
Humor	53
Aprendizaje	57
Ciencia	62
Misterio	54
Noticias	55
Religión	53
Reseñas	53
Romance	58
Ciencia Ficción	45

5 Conclusiones y trabajo futuro

Las colocaciones en Genesis sugieren en su mayoría temáticas relacionadas con la fé, creación y seres vivos, por lo cual sería correcto pensar que este texto podría hablar sobre el origen de las especies desde un punto de vista religioso, lo cual es muy aproximado al contenido real (conocido previamente) de dicha obra, mientras que para Holy indican temas de caballeros, mesas redondas, nombres de gobernantes y personajes de caricatura, entonces sería fácil concluir que esta obra se relaciona con humor medieval, resultado nuevamente aproximado a la temática real del texto.
Génesis utiliza el término *book* como una forma abstracta de referirse a un conjunto de historias y conocimientos que se irán juntando a lo largo de la historia del hombre, por

otro lado, la obra cómica utiliza el concepto *book* literalmente como un libro físico con información de armamento.

La obra bíblica regresa contextos sobre lugares de la naturaleza y el texto medieval arroja temáticas propias de acciones de caballeros o fantásticas, por lo que se concluye que los contextos relacionados con el token *book* no tienen relación alguna entre ambos textos.

Se debe notar de la Tabla 4 que Genesis presenta un 4% más de riqueza de vocabulario que Holy considerando STTR, y esta superioridad en riqueza, se ve respaldada por un valor λ mayor. Con la consistencia de estos dos valores, es válido proponer que Genesis tiene una mayor variedad de conceptos a utilizar y posiblemente una menor repetición de su uso. Las frecuencias de aparición de las 100 palabras populares de Genesis sugieren que la obra podría pertenecer a los géneros bellas letras, religión y ciencia, un resultado bastante parecido a las frecuencias de aparición de las 100 palabras populares de Holy que sugieren que la obra podría pertenecer a los géneros bellas letras, ciencia y aventura.

Resulta que estas dos obras comparten dos géneros según el análisis por lo que sostienen una similitud que no depende directamente del contenido incluido, lo que puede ser un indicador para recomendar una u otra obra a lectores que tengan interés en dichos géneros. Siete de los ocho criterios evaluados han arrojado diferencias significativas entre el par de obras, la clasificación de género es la única que ha presentado cierta similitud, sin embargo, una sola coincidencia no basta para poder decir que los textos sean similares o que aporten información similar. El método propuesto puede aplicarse a cualquier par de textos para determinar si una obra particular puede ser de utilidad a un lector que busca una nueva lectura, dando como referencia una obra qua a leído previamente. Cabe mencionar que las limitaciones de las computadoras para entender el lenguaje natural obligan a depender de entidades humanas que establezcan juicios basándose en el sentido común que una máquina no posee.

Se buscará ampliando el número de criterios de evaluación para tener un análisis más detallado del contenido de las obras, como número de capítulos, análisis de temática por cada capítulo, determinar el porcentaje del vocabulario que se refiere a temas particulares como: naturaleza, física, medios de comunicación, animales fantásticos, etc. También sería adecuado calcular un tamaño de muestra distinto de las palabras más populares para mejorar la precisión de las clasificaciones por género y el tamaño de muestra ideal para calcular STTR.

Al incrementar las dimensiones del análisis de obras literarias con las ideas anteriores, resultaría interesante aplicar aprendizaje profundo que permita mejorar los resultados del análisis.

La aportación presentada podría utilizarse para determinar cierta intención de plagio ente obras aun cuando presenten características distintas. También podría ser el inicio de la construcción de un motor de recomendación de obras literarias con base en las características de las obras.

Referencias

1. 1. Benigno, V., Vedder, I.: Lexical richness and collocational competence in second-language writing. IRAL - International Review of Applied Linguistics in Language Teaching. (2016).
2. 2. Labbé, C., Labbé, D.: Was Shakespeare's Vocabulary the Richest?. In: 12th International Conference on Textual Data Statistical Analysis, pp.323-336. ffhal-01002960, (2014).
3. 3. Xiao-fei, W., Dan, H.: Application of Multivariate Methods in the Investigation of Literary Style: The Phantom of the Opera. In: Journal of Literature and Art Studies, Vol. 10, No. 3, pp. 183-192, (2020).
4. 4. Bird S, Klein E, Loper E: Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit, O'Reilly Median. Versión digital: https://www.nltk.org/book/ (2009) (1) Accedido el 26 de Agosto de 2020
5. 5. Fang, Y., Liu, H.: Comparison of vocabulary richness in two translated Hongloumeng. In: Glottometrics, vol. 31, pp.54-75, (2015)
6. 6. Scott M: Type/Token Ratios and the Standardised Type/Token ratio, Help for WordSmith Tools Versión digital: https://lexically.net/downloads/version5/HTML/index.html?type_token_ratio_proc.htm. (2011)

Deep Learning for Fast Identification of Bacterial Strains in Resource Constrained Devices

Rafael Gallardo-García[1], Sofía Jarquín-Rodríguez[2], Beatriz Beltrán-Martínez[1] and Rodolfo Martínez[1]

[1] Faculty of Computer Science, Benemérita Universidad Autónoma de Puebla
[2] Faculty of Chemical Science, Benemérita Universidad Autónoma de Puebla
{rafael.gallardo, ana.jarquin}@alumno.buap.mx
{bbeltran, beetho}@cs.buap.mx

Abstract. This papers presents a proposal of a deep learning based system for classification of bacterial species, this system is able to work in resource constrained devices by fine tuning the state of the art mobile network called MobileNetV2, which drastically reduces the memory footprint without sacrificing accuracy, this is achieved by using depth wise separable convolutions and a novel layer module, called inverted residual with linear bottleneck. This resource constrained approach aims to provide complementary information to the traditional bacterial identification tests by being a suitable option for deployment in mobile or embedded systems. We measured the performance of several architectures over the original Digital Image of Bacterial Species (DIBaS) dataset and an augmented version of it, the best models achieved 95.53% and 94.22% of accuracy, respectively. We provide comparative tables between the original and augmented data sets as well as the results obtained by different configurations of the architecture, we also made publicly available the code and the pre-trained models with higher scores.

Keywords: Bacterial colony, classification, infectious diseases, mobile networks

1 Introduction

Bacteria are small unicellular microorganisms and are found almost everywhere on Earth. Most bacteria aren't harmful for humans, less than 1% of the different species of bacteria make people sick [18].

Due to the impact of the bacteria in the human life, the recognition of bacterial genera or specie is a very common and important task in many areas like medicine, veterinary science, biochemistry, food industry or farming [30]. The traditional laboratory methods for the identification of bacterial strains commonly require an expert with knowledge and experience in the field, most techniques designed for rapid and automated identification of microbiological samples are based of biochemical or modular biology technologies [15,27]. These traditional approaches are well established, however they are expensive and time consuming since they require complex sample preparation [1].

Automatizing the process of bacteria identification is very promising in the field of bioimage informatics. This field has yielded powerful solutions for specific image analysis task such as object detection, motion analysis or morphometric features [26]. Even so, most image analysis algorithms have been developed for very specific tasks or biological assays.

We propose a general solution to the task of bacterial specie recognition trough a deep learning (DL) approach. The implementation aims to reduce the memory footprint and the computational cost in the training and inference phases, to achieve this, we used a pretrained model of MobileNetV2 over the ImageNet [5] dataset. We fine-tuned the model by redefining the last dense layer of the original architecture. Faster and less expensive computational approaches to bacterial strain classification are necessary for deployment in mobile devices or embedded systems.

2 Related works

Holmberg et al. published of the first attempt to classify bacterial strains in an automated way, they used classification trees to extract the most important features and then classified it with artificial neural networks, achieving an accuracy of 76% in a dataset with 5 species of bacteria [12].

Later, in 2001, Liu et al. presented a computer-aided system to extract size and shape measurements of digital images of microorganisms to classify them into their appropriate morphotype [16]. Their work aimed to classify automatically each cell into one of 11 predominant bacterial morphotypes. This classifier had an accuracy of 96% on a set of 1,471 cells and 97% on a test of 4,270 cells.

Trattner et al. proposed an automatic tool to identify microbiological data types with computer vision and statistical modeling techniques [28]. Their methodology provided an objective and robust analysis of visual data, to automatize bacteriophage typing methods.

Some works consist in the identification of just one specie of bacteria. Forero et al. presented a method for automatic identification of *Mycobacterium tuberculosis* by using Gaussian mixture models [9]. In this work, authors use geometrical and color features of the images.

In 2013, Ahmed et al. published a light scatter-based approach to bacterial classification using distributed computing (due to the computational complexity of their feature extractors). They used Zernike and Chebyshev moments and Haralick texture features, then, the best features were selected using Fisher's discriminant. The classifier was a support vector machine with a linear kernel. The classification accuracy varies from 90% to 99% depending on the specie [1].

More recently, in 2017, Zielinski et al. published a deep learning approach to bacterial colony classification. In this work, authors used Convolutional Neural Networks as feature extractors and support vector machines or random forests as classifiers. Their overall highest accuracy over the Digital Image of Bacterial Species dataset was 97.24%, achieved by Fisher vector [21] encoders combined with VGG-M network.

3 Traditional methods for bacterial identification

Bacterial identification systems, both manual and automated, are in widespread use today in clinical microbiology laboratories.
Biochemical and phenotypic tests usually require a time commitment ranging from a few hours to several days and require large amounts of biological material, which can be a major disadvantage for the identification of microorganisms. Molecular methods have been demonstrated to have a complementary value, but they are not practical for routine use due to their high cost and the level of expertise required for their implementation.
To support the idea that automated and computer-based methods are useful to bring faster or complementary information, in this section we present a short review of the traditional methodologies used in the detection of bacterial agents in the clinical laboratory. A summarize of the methods and some examples are presented in Table 1.

3.1 Phenotypic methods

Traditional bacterial phenotypic identification schemes are based on the observable characteristics of bacteria, such as their morphology, development, biochemical and metabolic properties.
Each genus of bacteria has a characteristic protein expression. Although many proteins, including enzymes, are common to most bacteria, a range of unique biochemical pathways define each bacterial genus and the proteins expressed can even differ between species within a genus [29].

Biochemical testing Culture, when feasible, remains the diagnostic method by default, it allows the isolation of the microorganism, its identification and the antimicrobial susceptibility test (AST).
Genus is assigned by a combination of morphological features such as colony size or color, microscopic features such as Gram stain, and rapid biochemical tests such as catalase or oxidase enzyme activity, all are made with simple reagents.
Because of different species within a genus can have different pathogenicities and resistance profiles, identification at the specie level is performed by specific biochemical or serological tests.
These traditional approaches, based on the pure culture of the microorganism, require at least 36–48 hours [2].
Manual systems or multi-test galleries These are isolated cells with lyophilized substrate that are inoculated individually, the test results are expressed numerically and each specie is defined by a numerical code.
Biochemical test kits include tests for carbohydrate fermentation, methyl red, citric acid utilization and hydrogen sulfide production. The most representative are the Minitek identification system with paper substrates, API-20A system with dry powder substrates, PIZYMAN-IDENT rapid enzyme activity assay system and RaPID-ANA systems. The aforementioned microbial biochemistry reaction plate includes 30 biochemical matrices and their related biochemical test indicators, it also includes phosphate

buffered saline (PBS), bacterial turbidity standard tube, and eight identification series [7].

Automated systems The automated microbiology analysis provides results by capturing images with a high-definition digital camera and then analyzing them using its built-in software. Microbiology analyzers can be used for automatic colony counting, inhibition zone measurement to test antibiotic sensitivity test, and measurement of the hemolytic zone [7]. The first automated identification system to become available for clinical laboratories was the Vitek system (bioMerieux, Inc., Durham, NC) [3].

Mass spectrometric methods show promise for rapid identification, particularly Matrix Assisted Laser Desorption/Ionization-Time of Flight (MALDI-TOF) mass spectrometry. This offers the analysis of whole bacterial cultures for unique mass spectra from charged macromolecules by rapid, high-throughput testing with a rapidly growing database [22].

3.2. Molecular methods

Molecular diagnostic methods rely on the analysis of genomic markers corresponding to nucleic acid sequences. The taxonomy and phylogeny of bacteria is based on the sequences of conserved genes, especially those coding for ribosomal ribonucleic acids (rRNA) [29].

Nowadays, several molecular methods are available for microorganism identification, such as polymerase chain reaction (PCR) and related PCR-based methods including random amplification of polymorphic DNA (RAPD), amplified ribosomal DNA restriction analysis (16S-ARDRA) and DNA/DNA hybridization. For successful inclusion in the species, 70% similarity to the consensus sequence based on DNA-DNA hybridization and more than 97% similarity to the consensus sequence of the 16S rRNA genes are required [6].

PCR techniques PCR is a technique used to amplify small and targeted segments of DNA to produce millions of copies of a specific gene fragment. This technique was developed in 1983 by Kary Mullis. Each cycle of PCR have 3 main important steps such as denaturation, alignment of specific primers, annealing and final extension [23].

Real-time PCR Real-time PCR has catalyzed wider acceptance of PCR because it is faster, sensitive and reproducible, while the risk of carryover contamination is minimized. There is an increasing number of chemicals used to detect PCR products as they accumulate within a closed reaction vessel during real-time PCR. These include the non-specific DNA-binding fluorophores and the specific, fluorophore-labeled oligonucleotide probes [17].

This testing method combines PCR chemistry with fluorescent probe detection of amplified product in the same reaction vessel. In general, both PCR and amplified product detection are completed in an hour or less, which is considerably faster than conventional PCR detection methods [8].

Multiplex-PCR Multiplex polymerase chain reaction (Multiplex PCR) is a variant of PCR in which two or more loci are simultaneously amplified in the same reaction. Since its first description in 1988, this method has been successfully applied in many areas of DNA testing, including analysis of deletions, mutations and polymorphisms, or quantitative assays and reverse-transcription PCR [11].

DNA sequencing In the 1980s, a new standard for identifying bacteria began to be developed. In some laboratories, it was shown that phylogenetic relationships of bacteria, and, indeed, all life-forms, could be determined by comparing a stable part of the genetic code [1]. Genes such as; the ribosomal, 18S rRNA, 16S rRNA, 23S rRNA and 16S-23S rRNA internal transcribed sequences, rpoB (encoding β sub-unit of RNA polymerase), groEL (encoding heat-shock protein), gyrB (encoding β sub-unit of DNA gyrase) and recA (involved in the homologous recombination of DNA) are found in almost all bacteria and have been used for identification using conserved sequences as universal primers in PCR [14].

Table 1. Summary of some bacterial detection methods. The table presents advantages and disadvantages of the example systems.

Detection method	System examples	Advantages	Disadvantages
Phenotypic methods			
Culture on microbiological media and identification by biochemical tests	API (bioMérieux) Enterotube (BBL) RapID systems and MicroID (Remel) Biochemical ID systems (Microgen)	Sensitive. Inexpensive.	Lengthy and time-consuming process [25]. Might require 24–48 h [25]. Some species cannot be distinguished by morphology and cultural characteristics [22].
MALDI-TOF MS	Vitek MS Biflex III Autoflex III Microflex LT Biotyper	Fast. Accurate. Less expensive than molecular detection methods. Trained laboratory personnel not required.	High initial cost of the MALDI-TOF equipment [25]. Detection is not direct from clinical samples.
Molecular methods			
Real-time PCR Multiplex-PCR	MagNA Pure LC (Roche Applied Science) BioRobot EZ1 (Qiagen) BI Prism 6100 (Applied Biosystems) NucliSens Extractor (bioMérieux)	Culturing of the sample is not required. Specific, sensitive, rapid, and accurate. Closed-tube system reduces the risk of contamination. Can detect many pathogens simultaneously.	A highly precise thermal cycler is needed. Trained laboratory personnel required for performing the test.
DNA sequencing	Databases available for 16S rRNA: GenBank MicroSeq RDP-II RIDOM	Can identify fastidious and uncultivable microorganisms. 16S rRNA sequencing is the gold standard.	Trained laboratory personnel and powerful interpretation softwares are required. Expensive. Not suitable for routine clinical use. Contamination of a sample by post-amplification products [22].

4 Materials and methods

This section describes the methodologies we used to generate two versions of the dataset (the class distribution of the samples is also presented) and the details about the implementation of the system (transfer learning and fine tuning).

4.1 Digital Images of Bacteria Species dataset

The original version of DIBaS dataset contains a total of 33 species of microorganisms, approximately 20 RGB images (of 2048x1532 pixels) per specie. We remove the set of images of *Candida albicans* colonies as it is considered fungi [19]. The dataset was collected by the Chair of Microbiology of the Jagiellonian University in

Krakow. The samples were stained using the Gramm's method. All images were taken with an Olympus CX31 Upright Biological Microscope and a SC30 camera with a 100 times objective under oil-immersion [30]. The DIBaS dataset is publicly available[1].

Data augmentation We perform some steps to augment the quantity of samples in the dataset, the following list enumerates the steps we have taken:

1. Obtain 10 crops of the original image. Each crop with a size of 224x224 pixels. Four crops from the corners and one from the center, each was flipped horizontally to obtain the other five crops. Add the 10 crops of each image to the augmented dataset.
2. Add the resized versions of the original images (of 2048x1532 pixels) to the augmented set. Each image was resized to 224x224 pixels using Lanczos resampling.
3. Rotate every sample in 90, 180 and 270 degrees. Add the resulted images to the augmented dataset.

Images in both datasets are RGB. For a graphical example of the resulting images, see Figure 1.

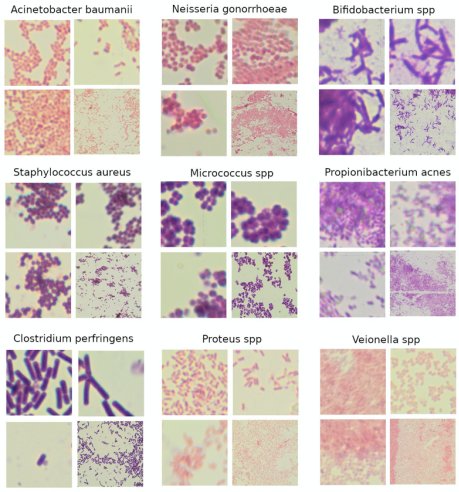


Fig.1. Augmented samples for 9 of the 32 classes in the dataset.

Dataset versions and splits Two different versions of the dataset were used in the experiments: the original version (after *Candida albicans* removal) with a total of 669 samples and an augmented version with a total of 26,524 samples. Each dataset was divided into train, validation and test sets, with 80% of the total samples for the training set and 20% for validation and test sets (10% each). The samples for each set were selected randomly.

Table 2 describes the distribution of the samples per each genera and specie, for both versions of the dataset.

[1]http://misztal.edu.pl/software/databases/dibas/

Table 2. Quantity of examples per each specie in the two versions of the DIBaS dataset.

Genera	Specie	Original Samples	Augmented Samples
Acinetobacter	baumanii	20	712
Actinomyces	israelli	23	860
Bacteroides	fragilis	23	960
Bifidobacterium	spp.	23	896
Clostridium	perfringens	23	868
Enterococcus	faecalis	20	784
	faecium	20	680
Escherichia	coli	20	880
Fusobacterium	spp.	23	924
Lactobacillus	casei	20	856
	crispatus	20	724
	delbrueckii	20	592
	gasseri	20	820
	jehnsenii	20	788
	johnsonii	20	880
	paracasei	20	840
	plantarum	20	784
	reuteri	20	820
	rhamnosus	20	820
	salivarius	20	840
Listeria	monocytogenes	22	936
Micrococcus	spp.	21	600
Neisseria	gonorrhoeae	23	892
Porfyromonas	gingivalis	23	984
Propionibacterium	acnes	23	932
Proteus	spp.	20	880
Pseudomonas	aeruginosa	20	832
Staphylococcus	aureus	20	780
	epidermidis	20	776
	saprophiticus	20	816
Streptococcus	agalactiae	20	848
Veionella	spp.	22	920

4.2 Model architecture

MobileNets are efficient models for mobile and embedded vision applications. These models uses depth-wise separable convolutions to build light weight deep neural networks [13]. MobileNets are capable to perform a wide range of tasks such as object detection, classification, feature extraction, semantic segmentation and large scale geolocalization [24].

In this paper, we used the mobile architecture called MobileNetV2, published by Sandler, M. et al. [24], which is mainly based on the MobileNetV1 [13], but improves the accuracy and efficiency by inserting *linear bottleneck layers* to the separable convolution blocks, they also use an inverted residual block instead the "traditional" residual block [10]. The main contribution of the MobileNetV2's paper is the introduction of a new layer module called: *inverted residual with linear bottleneck*, this module takes

a low-dimensional compressed representation as an input, which is first expanded to a high dimension and then filtered with a lightweight depth-wise convolution, features are subsequently projected back to a low-dimensional representation with a linear convolution, the inverted residual with linear bottleneck module allows to reduce the memory footprint needed in the inference phase by never fully materializing large immediate tensors [24].
We strongly recommend to read the full MobileNetV2's paper to get deeper into the theory and intuition behind this architecture.

4.3 Implementation

The system is written in a Jupyter[2] notebook with the 3.8.2 version of Python and PyTorch [20] in version 1.5.1, full details of the system are well explained in the paper repository[3]. We propose two different implementations; both are fine-tuned versions of the MobileNetV2 architecture. The first is the simplest and aims to achieve good accuracy without scaling the computational power needed. The second aims to achieve a higher accuracy but also increasing the computational cost. The description of both architectures is presented in this section.
Preprocessing Both models use 3 channel (RGB) images of 224x224 pixels as inputs. As the pre-trained models were trained with a specific pre-processing and in order to achieve consistency, the input images were normalized in the same way: the dataset was loaded in a range of 0-1 and then was normalized using a mean of 0.485, 0.456, 0.406 and standard deviation of 0.229, 0.224, 0.255 (for each RGB channel). At the end of the pre-processing, each input to the network is a tensor of shape 3x224x224, where each element is normalized as mentioned previously.
Model with one dense layer This version of the architecture is the result of fine tuning the dense layer of the original MobileNetV2 (which has 1280 inputs and 1000 outputs). The new classifier is composed by a dense layer with 1280 inputs and 32 outputs, which is preceded with a dropout layer with probability of 0.2.
Model with two dense layers To achieve a higher accuracy, we decided to increment the model's capacity by adding a dense layer to the classifier with a Rectified Linear Unit (ReLU) as activation and a dropout layer (with probability of 0.2) for regularization. Table 3 contains the blocks of the new classifier. As we mentioned above, this version is computationally more expensive.

Table 3. Blocks in the new version of the classifier.

Block type	Inputs	Outputs	Probability
Dropout	-	-	0.2
Linear	1280	1280	-
ReLU	-	-	-
Dropout	-	-	0.2
Linear	1280	32	-

[2]https://jupyter.org/
[3] https://github.com/gallardorafael/bacterialidentification

4.4 Transfer learning

To compensate the lack of training examples (given the difficulty of collecting them), transfer learning is a very good option to achieve high accuracy in classification while reducing the training time. Both of our proposed models were trained after being initialized with a pre-trained version of the weights. The pre-trained weights are from a model trained over the ImageNet database [5], the pre-trained model is available in the *model zoo* [4] of PyTorch. As mentioned above, all the implementation details are available in the paper repository.

5 Results

As detailed in the section about the implementation, we proposed two different architectures, which mainly vary in the dense layers at the end of the network. In this section, we present the accuracy scores for both architectures, trained and tested over the two versions of the dataset and varying the quantity of epochs.
After several experiments with different configurations of the hyper parameters, we realized that the best accuracy scores were achieved by models trained with batch size of 32. Also, the increase of the training time with respect to the input size was not worth it and we decided to fix the input shape to 224x224 pixels.
Table 4 summarizes the results of each trained model, with different hyper parameters and for both versions of the dataset. The first column indicates the number of fully connected layers in the classifier block of the network, the second column is for the version of the dataset, the third column shows the number of training epochs and the fourth presents the accuracy score obtained by the model in that row. We determine the number of epochs by analyzing the training and validation loss.

Table 4. Summarize of accuracy scores by training epochs and dataset version.

Dense layers in classifier	Dataset version	Training epochs	Test accuracy
1	Original	10	0.8874
		30	0.8785
		50	**0.9553**
	Augmented	10	0.9375
		15	**0.9382**
		30	0.9229
2	Original	10	0.8642
		30	**0.9232**
		50	0.8839
	Augmented	10	**0.9422**
		15	0.9364
		30	0.9410

Accuracy scores in bold are the best for each model architecture trained over one version of the dataset. The highest accuracy for the original dataset was achieved by the

[4] https://pytorch.org/docs/stable/model zoo.html#module-torch.utils.model_zoo

architecture with just one dense layer in 50 epochs, the highest accuracy for the augmented dataset was reached by the architecture with two dense layers in just 10 epochs. Validation and training loss plots indicates that both architectures begin to overfit after the 10th epoch when training over the augmented dataset.
Some output examples are presented in Figure 2. From left to right, the species are: *Staphylococcus epidermidis*, *Lactobacillus paracasei*, *Neisseria gonorrhoeae* and *Porfyromonas gingivalis*. Top five most probable classes are shown below each sample.

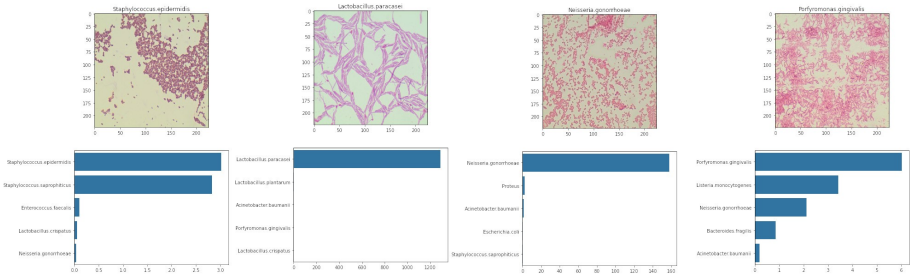


Fig.2. Top-5 most probable species for 4 samples in the test set.

6 Conclusions

The overall highest score (0.9553) was achieved by the architecture with one fully connected layer, trained in 50 epochs over the original dataset. It is important to consider that the test set of the original dataset consists of an average of 2 images per class (in experiments, we extract 10 samples per test image using ten crops), which can reduce the difficulty of the test set. On the other hand, the architecture with two dense layers achieved a 0.9422 over a test set with more than 4000 samples, which indicates that the model had a good generalization, increasing the certainty of the predictions.
For real world applications, we strongly recommend to use the model trained and tested over the augmented dataset. The lack of test samples in the original dataset may cause a high accuracy score, even if the model is not good enough. In a sensitive area such as microbiology, false positives and false negatives are a big problem. To reduce the impact of errors in identification, researchers can rely on probability graphs such as the showed in Figure 2, to see the *n* most probable classes for each sample.
In future work, we could include an analysis of the false positives and false negatives for each class in the dataset, so that researchers can have a measure of certainty about the predictions for each specie. The repository of the paper contains the code and instructions to prepare new species to increment the dataset and re-train the models if needed. A mobile application could be launched, in this app, the users could identify bacterial strains by just connecting the camera of the mobile devices to the microscope.

References

1. Ahmed, W.M., Bayraktar, B., Bhunia, A.K., Hirleman, E.D., Robinson, J.P., Rajwa, B.: Classification of bacterial contamination using image processing and distributed computing. IEEE journal of biomedical and health informatics 17(1), 232–239 (2012)
2. Angeletti, S.: Matrix assisted laser desorption time of flight mass spectrometry (maldi-tof ms) in clinical microbiology. Journal of microbiological methods 138, 20–29 (2017)
3. Carroll, K.C., Patel, R.: Systems for identification of bacteria and fungi. In: Manual of Clinical Microbiology, Eleventh Edition, pp. 29–43. American Society of Microbiology (2015)
4. Clarridge, J.E.: Impact of 16s rrna gene sequence analysis for identification of bacteria on clinical microbiology and infectious diseases. Clinical microbiology reviews 17(4), 840–862 (2004)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Duskov˘ a, M.,´ Sedo, O., K˘ sicov˘ a, K., Zdr´ ahal, Z., Karp´ ´ıskov˘ a, R.: Identification of lacto-´ bacilli isolated from food by genotypic methods and maldi-tof ms. International journal of food microbiology 159(2), 107–114 (2012)
7. Elmer, W., Stephen, D., William, M., Paul, C., Washington, C.: Color atlas and textbook of diagnostic microbiology (1997)
8. Espy, M., Uhl, J., Sloan, L., Buckwalter, S., Jones, M., Vetter, E., Yao, J., Wengenack, N., Rosenblatt, J., Cockerill, F.., et al.: Real-time pcr in clinical microbiology: applications for routine laboratory testing. Clinical microbiology reviews 19(1), 165–256 (2006)
9. Forero, M.G., Cristobal, G., Desco, M.: Automatic identification of mycobacterium tubercu-´ losis by gaussian mixture models. Journal of microscopy 223(2), 120–132 (2006)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Henegariu, O., Heerema, N., Dlouhy, S., Vance, G., Vogt, P.: Multiplex pcr: critical parameters and step-by-step protocol. Biotechniques 23(3), 504–511 (1997)
12. Holmberg, M., Gustafsson, F., Hornsten, E.G., Winquist, F., Nilsson, L.E., Ljung, L., Lund-¨ strom, I.: Bacteria classification based on feature extraction from sensor data. Biotechnology¨ techniques 12(4), 319–324 (1998)
13. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
14. James, G.: Universal bacterial identification by pcr and dna sequencing of 16s rrna gene. In: PCR for clinical microbiology, pp. 209–214. Springer (2010)
15. Jenkins, S.A., Drucker, D., Keaney, M.G., Ganguli, L.A.: Evaluation of the rapid id 32a system for the identification of bacteroides fragilis and related organisms. Journal of applied bacteriology 71(4), 360–365 (1991)
16. Liu, J., Dazzo, F.B., Glagoleva, O., Yu, B., Jain, A.K.: Cmeias: a computer-aided system for the image analysis of bacterial morphotypes in microbial communities. Microbial Ecology 41(3), 173–194 (2001)
17. Mackay, I.M.: Real-time pcr in the microbiology laboratory. Clinical microbiology and infection 10(3), 190–212 (2004)

18. MedlinePlus: Bacterial infections (2020), https://medlineplus.gov/ bacterialinfections.html
19. Nobile, C.J., Johnson, A.D.: Candida albicans biofilms and human disease. Annual review of microbiology 69, 71–92 (2015)
20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. pp. 8026–8037 (2019)
21. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: 2007 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)
22. Pitt, T., Barer, M.: Classification, identification and typing of micro-organisms. Medical Microbiology p. 24 (2012)
23. Rajalakshmi, S.: Different types of pcr techniques and its applications. International Journal of Pharmaceutical, Chemical & Biological Sciences 7(3) (2017)
24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
25. Singhal, N., Kumar, M., Kanaujia, P.K., Virdi, J.S.: Maldi-tof mass spectrometry: an emerging technology for microbial identification and diagnosis. Frontiers in microbiology 6, 791 (2015)
26. Sommer, C., Gerlich, D.W.: Machine learning in cell biology–teaching computers to recognize phenotypes. Journal of cell science 126(24), 5529–5539 (2013)
27. Spratt, D.A.: Significance of bacterial identification by molecular biology methods. Endodontic topics 9(1), 5–14 (2004)
28. Trattner, S., Greenspan, H., Tepper, G., Abboud, S.: Automatic identification of bacterial types using statistical imaging methods. IEEE transactions on medical imaging 23(7), 807–820 (2004)
29. Varadi, L., Luo, J.L., Hibbs, D.E., Perry, J.D., Anderson, R.J., Orenga, S., Groundwater, P.W.: Methods for the detection and identification of pathogenic bacteria: past, present, and future. Chemical Society Reviews 46(16), 4818–4832 (2017)
30. Zielinski, B., Plichta, A., Misztal, K., Spurek, P., Brzychczy-Włoch, M., Ocho´nska, D.: Deep learning approach to bacterial colony classification. PloS one 12(9), e0184554 (2017)

Análisis de ciberacoso con big data México: caso MySQL y MongoDB

Victor G. Morales-Murillo[1], Maria J. Somodevilla-Garcia[1], Concepción Pérez de Celis-Herrero[1], Ivo H. Pineda-Torres [1]

[1] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Edificio CC01, 14 sur y Ave. Sn. Claudio, Fraccionamiento Jardines de Sn. Manuel C.P. 72570 Puebla, México
vg055@hotmail.com, maria.somodevilla@correo.buap.mx, maria.perezdecelis@correo.buap.mx, ivo.pineda@correo.buap.mx

Abstract. En México, el número de víctimas por ciberacoso está creciendo con el uso de las tecnologías de la información y el internet. En este trabajo, se presenta un análisis del ciberacoso en México con *big data*. Se analiza la encuesta de MOCIBA 2017 para generar un modelo de datos E-R e implementar una base de datos en MySQL y en MongoDB. Se realizan 19 consultas para comparar el rendimiento de MySQL y MongoDB, evaluando el procesamiento de datos y los tiempos de ejecución de las consultas realizadas. También se realizan tres consultas a datos geoespaciales en MongoDB. Finalmente, se justifica que MongoDB presenta un mejor rendimiento que MySQL en altos volúmenes de datos por los resultados observados.

Keywords: Ciencia de datos, Big data, MySQL, MongoDB, Ciberacoso.

1 Introducción

El uso de dispositivos electrónicos con internet ha beneficiado a la población con herramientas para la comunicación, sin embargo, estas herramientas también son utilizadas para actividades delictivas como es el ciberacoso porque facilitan la búsqueda y la comunicación de las víctimas a través de computadoras, celulares, redes sociales y aparatos de geo-localización. Además, el ciberacoso tiende a incrementar con el continuo crecimiento del internet [1]. En el 2019, se estimaron 4.388 millones de usuarios en internet que representan el 53% de la población mundial y el ingreso de 1 millón de nuevos usuarios por cada día, en el caso de México el 67% de la población se conectó a internet [2]. En el 2017, aproximadamente 10,3millones de personas vivieron alguna situación de ciberacoso, en donde, las mujeres y los adolescentes fueron los más expuestos a este crimen [3].

El ciberacoso o acoso cibernético es un problema grave y complejo. En el trabajo de Chacón [4] se mencionan que no hay una definición universal para el ciberacoso, pero se considera como una conducta repetitiva de acercamiento, acoso y amenazas a otra persona, usando herramientas de internet u otro instrumento electrónico de comunicación, además, el lenguaje del ciberacoso es mucho más fuerte que el acoso de

la vida real por el posible anonimato del acosador. En Smith et al. [5] se define el ciberacoso como un acto agresivo e intencional llevado a cabo por un grupo o individuo, utilizando formas electrónicas de contacto , repetidamente y con el tiempo contra una víctima que no puede defenderse fácilmente. En Patchin e Hinduja [6] el concepto de ciberbullying es cuando alguien, intencional y repetidamente acosa, maltrata o se burla de otra persona en línea o a través del uso de teléfonos celulares u otros dispositivos electrónicos de forma tal en la que esta no es capaz de responder. En el Módulo sobre Ciberacoso (MOCIBA) 2017 del INEGI [7] se refiere a la situación en que una persona es expuesta, repetidamente y de forma prolongada en el tiempo a acciones negativas con la intención de causar, o tratar de causar, daño o molestias, por parte de una o más personas usando medios electrónicos tales como el celular e Internet. Las principales situaciones de ciberacoso que se presentan a través del teléfono y el internet son por mensajes de texto, llamadas telefónicas, fotografías o video, correos electrónicos, sesiones de chat, programas de mensajería instantánea y páginas web [8]. Además, otras formas de acoso cibernético son el fraude de identidad, robo de datos, la piratería y el daño a los equipos informáticos [1].
Es importante realizar investigaciones sobre el ciberacoso porque es un grave problema que está creciendo con el uso de las TICs (Tecnologías de la Información y Comunicación), En este trabajo se realiza un análisis del ciberacoso en México con *big data* utilizando paradigmas SQL y NOSQL, en particular MYSQL Y MongoDB.
La estructura de este artículo es la siguiente. En la sección 2, se presenta el estado del arte con los trabajos relacionados y algunos antecedentes. En la sección 3, se propone la metodología considerando la implementación de la base de datos en MySQL y en MongoDB describiendo las pruebas realizadas. En la sección 4, se presentan los resultados de las pruebas realizadas, seguidas de las conclusiones en la sección 5.

2 Estado del arte

En el primer trabajo relacionado de Győrödi, Andrada Olah, y Bandici [10] se presenta un estudio comparativo entre las capacidades de uso de una base de datos no relacional con MongoDB y las capacidades de uso de una base de datos relacional con MySQL para el back-end, de una plataforma en línea, que permite a los usuarios publicar y compartir sus diferentes artículos, libros, revistas, etc. La comparación se realiza a través de varias operaciones ejecutadas en paralelo por muchos usuarios. Actualmente una aplicación debe ser accesible 24 horas al día y 7 días de la semana, por lo que es importante implementar una base de datos apropiada con conexiones simultaneas para miles de usuarios. Una base de datos debe manejar un gran volumen de datos, sin embargo, las bases de datos relacionales presentan limitaciones para el manejo de un gran volumen de datos por una parte y por otra cada usuario tiene sus propias necesidades y requisitos, por lo que es posible que dentro de la misma aplicación, se necesite una personalización diferente para cada usuario. Las bases de datos relacionales no permiten una configuración completa, que pueda conformarse según las necesidades específicas de los diferentes usuarios. Estas limitaciones han llevado al desarrollo de bases de datos no relacionales, también conocidas comúnmente como NoSQL La base más adecuada para la aplicación back-end es la no relacional con

MongoDB porque presenta menores tiempos de respuesta con las operaciones de inserción, actualización y eliminación realizadas por diferentes números de usuarios y con diferentes números de productos .

El segundo trabajo relacionado de Soni, Ambavane, Ambre y Maitra [11] se menciona que las bases de datos relacionales han sido la base para las aplicaciones empresariales durante décadas, desde el lanzamiento de MySQL en 1995 han sido un recurso popular y económico. Sin embargo, con la explosión en volumen de datos y la variedad de datos, surgen tecnologías de bases de datos no relacionales para abordar nuevas aplicaciones o remplazar la infraestructura relacional existente. Además, se presenta una comparativa entre MongoDB y MySQL, en donde, MongoDB resulta ser más eficiente que MySQL.

En los dos trabajos relacionados se identifica que el modelo no relacional con la tecnología MongoDB presenta mejores resultados de eficiencia que el modelo relacional con la tecnología MySQL, en ambos trabajos las pruebas se realizan con múltiples usuarios conectados a la base de datos realizando diferentes operaciones de inserción, actualización y eliminación. Solo se realizaron consultas en el primer trabajo relacionado.

3. Metodología

Oracle México menciona que Big data son los datos que contienen una mayor variedad, que se presentan en volúmenes crecientes y a una velocidad superior. Big data está formado por conjuntos de datos voluminosos y complejos, procedentes de nuevas fuentes de datos. Estos conjuntos de datos son tan voluminosos que el software de procesamiento de datos convencional no puede administrarlos. Sin embargo, estos volúmenes masivos de datos pueden utilizarse para abordar problemas que antes no hubiera sido posible solucionar [9].

3.1 Conjunto de datos MOCIBA 2017

Los datos que se utilizaron son del MOCIBA 2017 que presenta la prevalencia del ciberacoso entre las personas de 12 a 59 años de edad (usuarias de Internet en cualquier dispositivo), la situación de ciberacoso vivida y su caracterización. En MOCIBA 2017 se presentan los resultados de la prevalencia de ciberacoso ocurrido los doce meses previos al levantamiento de la encuesta y caracteriza a la población que lo ha experimentado a través de las diferentes situaciones de acoso declaradas. Además, trata de establecer la identidad y sexo de la persona que lo lleva a cabo, la intensidad de ciberacoso y el impacto causado en la víctima. [7].

El cuestionario de MOCIBA 2017 está formado por 12 preguntas con 158 respuestas posibles, algunas repuestas son de opción múltiple y con pre requisitos, de la pregunta 4 a la pregunta 9 se presentan un listado de 10 posibles situaciones de ciberacoso que son: recibir mensajes ofensivos, con insultos o burlas; Recibir llamadas ofensivas, con insultos o burlas; Que una persona publique información personal, fotos o videos (falsos o verdaderos) para dañar; Ser criticado(a) o que se burlen en línea por su apariencia o clase social; Recibir insinuaciones o propuestas de tipo sexual; Que una persona se hiciera pasar por usted para enviar información falsa, insultar o agredir a

otros; Ser contactado(a) por medio de nombres falsos para molestarle o dañarle; Ser vigilado en sus sitios o cuentas en Internet; Ser provocado para que reaccione de forma negativa; Recibió fotos o videos con contenido sexual que le molesto.

3.2 Modelo de datos E-R

El modelo de datos E-R se generó con base en el cuestionario de MOCIBA 2017. En la figura 1, se muestra el modelo de datos E-R, se identificaron 9 entidades Encuesta, Persona, Sexo, Ciudad, Entidad, Nivel, Pregunta, Respuesta y Situación. Estas entidades con sus relaciones, sirven para reducir el número de datos nulos en las respuestas de los encuestados que presenta el conjunto de datos de MOCIBA2017.

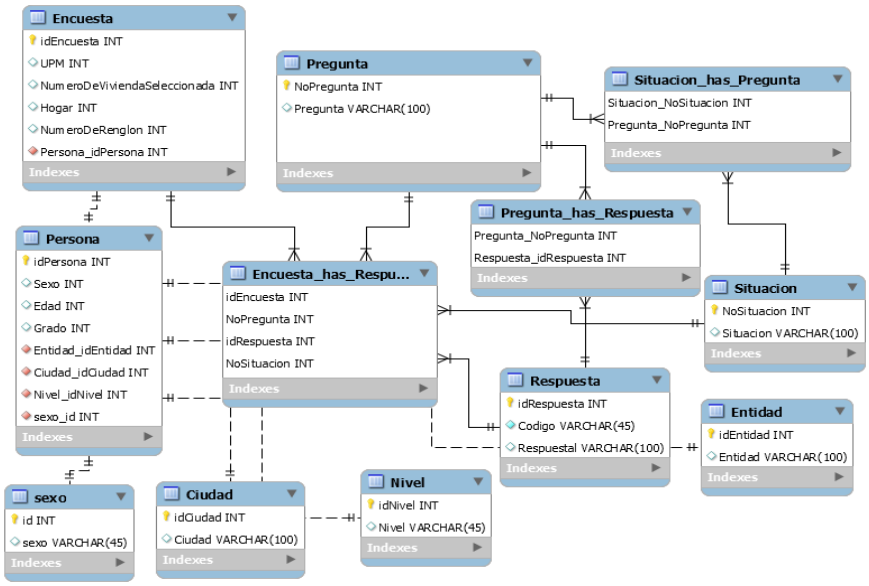
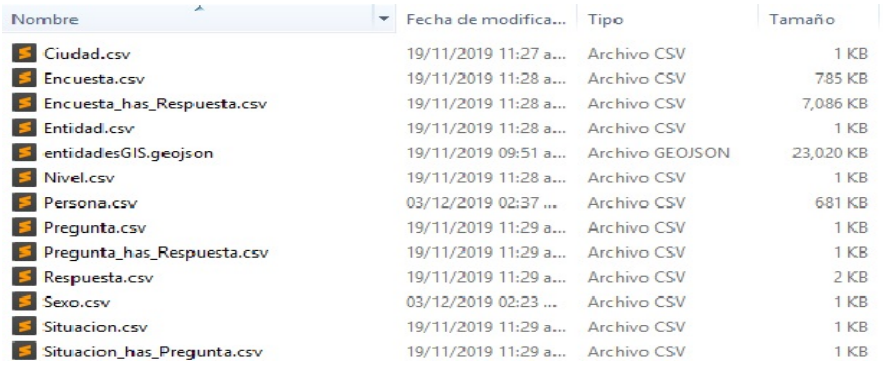


Fig. 1 . Modelo de datos E-R.

3.3 Pre-procesamiento del conjunto de datos

El conjunto de datos de MOCIBA 2017 contiene 5 carpetas llamadas *catalogos, conjunto_de_datos, diccionario_de_datos, metadatos y modelo_entidad_relacion*. De la carpeta *catalogos* se utilizan los archivos ciudad.csv, ent.csv, nivel.csv y sex.csv, de la carpeta *conjunto_de_datos* se utiliza el archivo conjunto_de_datos_mociba_2017.csv, a estos archivos se les realizó un pre-procesamiento de datos para generar los siguientes archivos que se importaran a la base de datos. En la figura 2, se muestran los archivos generados. En el caso de Encuesta_has_Respuesta.csv, el pre-procesamiento se realiza con Python debido a la complejidad del archivo conjunto_de_datos_mociba_2017.csv.



Nombre	Fecha de modifica...	Tipo	Tamaño
Ciudad.csv	19/11/2019 11:27 a...	Archivo CSV	1 KB
Encuesta.csv	19/11/2019 11:28 a...	Archivo CSV	785 KB
Encuesta_has_Respuesta.csv	19/11/2019 11:28 a...	Archivo CSV	7,086 KB
Entidad.csv	19/11/2019 11:28 a...	Archivo CSV	1 KB
entidadesGIS.geojson	19/11/2019 09:51 a...	Archivo GEOJSON	23,020 KB
Nivel.csv	19/11/2019 11:28 a...	Archivo CSV	1 KB
Persona.csv	03/12/2019 02:37 ...	Archivo CSV	681 KB
Pregunta.csv	19/11/2019 11:29 a...	Archivo CSV	1 KB
Pregunta_has_Respuesta.csv	19/11/2019 11:29 a...	Archivo CSV	1 KB
Respuesta.csv	19/11/2019 11:29 a...	Archivo CSV	2 KB
Sexo.csv	03/12/2019 02:23 ...	Archivo CSV	1 KB
Situacion.csv	19/11/2019 11:29 a...	Archivo CSV	1 KB
Situacion_has_Pregunta.csv	19/11/2019 11:29 a...	Archivo CSV	1 KB

Fig. 2 . Archivos generados del pre-procesamiento del conjunto de datos MOCIBA 2017.

En el caso de entidadesGIS.geojson son datos geoespaciales que se obtuvieron del archivo destdv250k_2gw_c.zip del Geoportal del Sistema Nacional de Información sobre Biodiversidad [12], al descomprimir este archivo se muestran 8 archivos llamados destdv250k_2cw con la diferencia de su tipo de archivo dbf, html, prj, shp, shx, xml y png. Se utilizó el sitio web ArcGIS [13] para generar el archivo del tipo geojson, después, se usó la herramienta jq [14] para obtener las *features* representadas como vectores del archivo geojson utilizando el comando jq --compact-output ".features" entidadesp2.geojson > entidadesGIS.geojson.

3.4 Implementación en MySQL

Se utilizó para la implementación de la base de datos relacional WampServer 2.5 con MySQL 5.6.17. Se creó la base de datos mociba17 que contiene 12 tablas como se muestran en la figura 3. Se importaron los archivos obtenidos del pre-procesamiento del conjunto de datos de MOCIBA 2017 correspondiente a cada archivo csv.

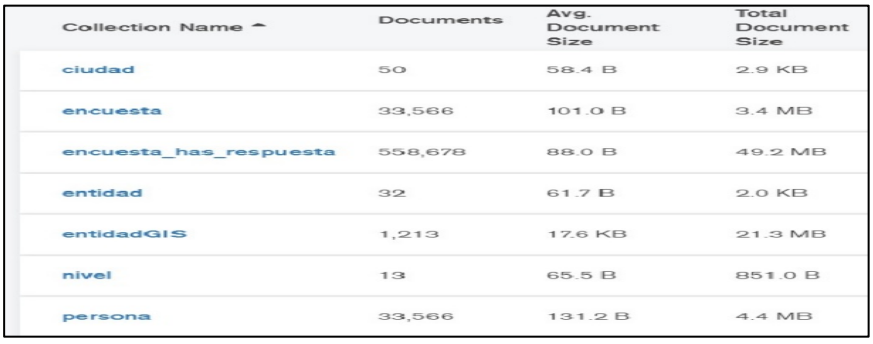
Fig. 3 . Implementación de la base de datos MOCIBA 2017 con MySQL.

3.5 Implementación en MongoDB

La herramienta que se utilizó para la implementación de la base de datos no relacional es MongoDB 4.2.0 y MongoDB Compass Community 3.4.0. está última requerida para realizar las consultas geoespaciales. Se creó una base de datos con el nombre mociba17 que contiene 13 colecciones: *ciudad, encuesta, encuesta_has_respuesta, entidad, nivel, persona, pregunta, pregunta_has_respuesta, respuesta, sexo, situación, situación_has_pregunta y entidadesGIS*. Se importaron los archivos de cada colección correspondiente. En la tabla 1, se muestra la sintaxis de importación y en la figura 4, la implementación de la base de datos MOCIBA 2017 en MongoDB.

Tabla 1. Sintaxis para importar los datos de colecciones en MongoDB.

mongoimport --db mociba17 -c encuesta --type=csv --headerline --file=mociba17/Encuesta.csv
mongoimport --db mociba17 -c encuesta_has_respuesta --type=csv --headerline --file=mociba17/Encuesta_has_Respuesta.csv
mongoimport --db mociba17 -c entidad --type=csv --headerline --file=mociba17/Entidad.csv
mongoimport --db mociba17 -c pregunta --type=csv --headerline --file=mociba17/Pregunta.csv
mongoimport --db mociba17 -c respuesta --type=csv --headerline --file=mociba17/Respuesta.csv
mongoimport --db mociba17 -c situacion --type=csv --headerline --file=mociba17/Situacion.csv
mongoimport --db mociba17 -c situacion_has_pregunta --type=csv --headerline --file=mociba17/Situacion_has_Pregunta.csv
mongoimport --db mociba17 -c entidadGIS --file "mociba17/entidadesGIS.geojson" --type json



Collection Name	Documents	Avg. Document Size	Total Document Size
ciudad	50	58.4 B	2.9 KB
encuesta	33,566	101.0 B	3.4 MB
encuesta_has_respuesta	558,678	88.0 B	49.2 MB
entidad	32	61.7 B	2.0 KB
entidadGIS	1,213	17.6 KB	21.3 MB
nivel	13	65.5 B	851.0 B
persona	33,566	131.2 B	4.4 MB

Fig. 4 . Implementación de la base de datos MOCIBA 2017 con MongoDB.

3.6 Consultas

Se realizaron 19 consultas, las cuales se muestran en la tabla 2. Las consultas 1 a la 9, son consultas de conteo de la encuesta MOCIBA 2017 categorizadas por grupo etario y sexo. Para comparar la eficiencia de procesamiento de datos y tiempos de ejecución entre MySQL y MongoDB, se realizaron dos pruebas, la primera prueba es de la consulta 10 a la 14, donde, se incrementa el número de registros recuperados por cada consulta de forma ascendente. La segunda prueba considera de la consulta 15 a la 19, considerando un número menor de registros recuperados por cada consulta de forma ascendente.

Tabla 2. Consultas.

No	Consulta	Respuesta
1	¿Cuántas personas contestaron la encuesta?	33566
2	¿Cuántas personas que contestaron son menores y mayores de 18 años?	5216-28350
4-5	¿Cuántas mujeres y hombres contestaron la encuesta?	17129-16437
6-7	En el ciberacoso ¿Cuántas mujeres y hombres son acosadoras?	1811-3644
8-9	¿Cuántas mujeres y hombres son víctimas de ciberacoso?	3051- 2772
10-14	Mostrar 100,1000, 10000, 100000 y todas respuestas de las encuestas.	100-558,678
15-19	Mostrar los datos de 100, 10000, 20000 y todos usuarios.	100-33566

Se implementaron 19 consultas en MySQL y MongoDB. Se observa que las consultas más complejas tienden a ser diferentes como es el caso para las consultas 8 y 9, donde, se muestran diferente número de operadores. Por tal motivo, para evaluar el rendimiento de MySQL con MongoDB se utilizaron desde la consulta 8 a la 19 (Tabla 3), estas agrupadas en dos pruebas.

Tabla 3. Consultas implementadas en MySQL y MongoDB.

No	Consulta en MySQL	Consulta en Mongo DB
8	SELECT COUNT(persona.idPersona) FROM persona, encuesta WHERE persona.sexo=2 AND persona.idPersona = encuesta.idPersona AND encuesta.idEncuesta IN (SELECT idEncuesta FROM encuesta_has_respuesta WHERE NoPregunta = 4 AND idRespuesta=1);	db.encuesta_has_respuesta.aggregate([{ $match: { $expr: { $and: [{ $eq: ["$NoPregunta", 4] },{ $eq: ["$idRespuesta", 1] }]}}}, {$lookup: { from: "encuesta", localField: "idEncuesta", foreignField: "idEncuesta", as: "encuesta"}}, {$replaceRoot: { newRoot: { $mergeObjects: [{ $arrayElemAt: ["$encuesta", 0] }, "$$ROOT"] } }}, { $project: { encuesta: 0 } }, {$lookup:{ from: "persona", localField: "idPersona", foreignField: "idPersona", as: "persona"}}, { $replaceRoot: { newRoot: { $mergeObjects: [{ $arrayElemAt: ["$persona", 0] }, "$$ROOT"] } }}, { $project: { persona: 0 } }, { $match : { Sexo : 2 } }, {$group: { _id : "$idEncuesta" }}, {$count:"documentos"}])
9	SELECT COUNT(persona.idPersona) FROM persona, encuesta WHERE persona.sexo=1 AND persona.idPersona = encuesta.idPersona AND encuesta.idEncuesta IN (SELECT idEncuesta FROM encuesta_has_respuesta WHERE NoPregunta = 4 AND idRespuesta=1);	db.encuesta_has_respuesta.aggregate([{ $match: { $expr: { $and: [{ $eq: ["$NoPregunta", 4] },{ $eq: ["$idRespuesta", 1] }]}}}, {$lookup: { from: "encuesta", localField: "idEncuesta", foreignField: "idEncuesta", as: "encuesta"}}, {$replaceRoot: { newRoot: { $mergeObjects: [{ $arrayElemAt: ["$encuesta", 0] }, "$$ROOT"] } }}, { $project: { encuesta: 0 } }, {$lookup:{ from: "persona", localField: "idPersona", foreignField: "idPersona", as: "persona"}}, { $replaceRoot: { newRoot: { $mergeObjects: [{ $arrayElemAt: ["$persona", 0] }, "$$ROOT"] } }}, { $project: { persona: 0 } }, { $match : { Sexo : 1 } }, {$group: { _id : "$idEncuesta" }}, {$count:"documentos"}])

10	SELECT * FROM `encuesta_has_respuesta` WHERE 1 LIMIT 100;	db.encuesta_has_respuesta.find().limit(100)
11	SELECT * FROM `encuesta_has_respuesta` WHERE 1 LIMIT 1000;	db.encuesta_has_respuesta.find().limit(1000)
12	SELECT * FROM `encuesta_has_respuesta` WHERE 1 LIMIT 10000;	db.encuesta_has_respuesta.find().limit(10000)
13	SELECT * FROM `encuesta_has_respuesta` WHERE 1 LIMIT 100000;	db.encuesta_has_respuesta.find().limit(100000)
14	SELECT * FROM `encuesta_has_respuesta`;	db.encuesta_has_respuesta.find( )
15	SELECT * FROM `persona` WHERE 1 LIMIT 100;	db.persona.find( ).limit(100)
16	SELECT * FROM `persona` WHERE 1 LIMIT 1000;	db.persona.find( ).limit(1000)
17	SELECT * FROM `persona` WHERE 1 LIMIT 10000;	db.persona.find( ).limit(10000)
18	SELECT * FROM `persona` WHERE 1 LIMIT 20000;	db.persona.find( ).limit(20000)
19	SELECT * FROM `persona`;	db.persona.find( )

3.7 Consultas con datos geoespaciales

MongoDB soporta consultas con datos geoespaciales, para poder realizar estas consultas se utiliza la colección llamada entidadesGIS que contiene la división política estatal de México y los estados son representados en polígonos. Primero, se genera un índice 2d-esferico (*2dsphere)* con la siguiente sintaxis: *db.entidadGIS.createIndex({ geometry: "2dsphere" }*). Después, ya se pueden realizar consultas con operadores para datos geoespaciales. La primera consulta es para buscar objetos alrededor de las coordenadas de Puebla con una longitud: -98.2019300 y latitud: 19.0433400, dentro de un radio de 10,000 metros. La sintaxis de la consulta es la siguiente: *db.entidadGIS.find({ 'geometry' : { $near : { $geometry: { type: 'Point', coordinates: [-98.2019300, 19.0433400] } , $maxDistance: 10000 }} }).* Recupera dos documentos, uno de Puebla y uno de Tlaxcala. En la segunda consulta se reduce el radio a 100 metros. La sintaxis de la consulta es la siguiente: *db.entidadGIS.find({ 'geometry' : { $near : { $geometry: { type: 'Point', coordinates: [-98.2019300, 19.0433400] } , $maxDistance: 100 }} }).* Recupera un solo documento el de Puebla. En la tercera consulta se aumenta el radio a 50000 metros y agrega un contador de los documentos cercanos, la sintaxis de consulta es la siguiente: *db.entidadGIS.find({ 'geometry' : { $near : { $geometry: { type: 'Point', coordinates: [-98.2019300, 19.0433400] } , $maxDistance: 50000 }} }).count()*. Recupera 4 documentos.

4 Resultados

Los resultados del tiempo de ejecución de las consultas de la tabla 3 se presentan en la tabla 4. En la primera columna se muestra el número de consulta, en la segunda y tercera columna se muestra el tiempo de ejecución de la consulta en cada gestor en segundos.

Tabla 4. Tiempos de ejecución de las consultas implementadas en MySQL y MongoDB.

No. Consulta	T/Seg. MySQL	T/Seg. MongoDB
8	1.23671325	1572.908
9	0.291917	1704.789
10	0.0006485	0.000
11	0.0029445	0.001
12	0.02643975	0.011
13	0.293094	0.120
14	1.541852	0.773
15	0.000796	0.000
16	0.002605	0.001
17	0.02441325	0.011
18	0.0437215	0.024
19	0.09797475	0.035

La primera prueba considera desde la consulta 10 con 100 registros recuperados hasta la consulta 14 con 558,678 registros recuperados, esta prueba sirve para evaluar el procesamiento de datos y el tiempo de ejecución entre MySQL y MongoDB (figura 5). Se observa que el rendimiento de MongoDB es mejor que el de MySQL.
La segunda prueba considera desde la consulta 15 con 100 registros recuperados hasta la consulta 19 con 33,566 registros recuperados, (figura 5). Nuevamente se observa que el rendimiento de MongoDB es mayor.
En general en la tabla 4, se observa que MongoDB presenta un mejor rendimiento de procesamiento de datos y menores tiempos de ejecución de las consultas, en comparación de MySQL. Sin embargo, es importante observar que en las consultas 8 y 9 MySQL presenta un menor tiempo de respuesta que MongoDB debido a la complejidad de las consultas en MongoDB.

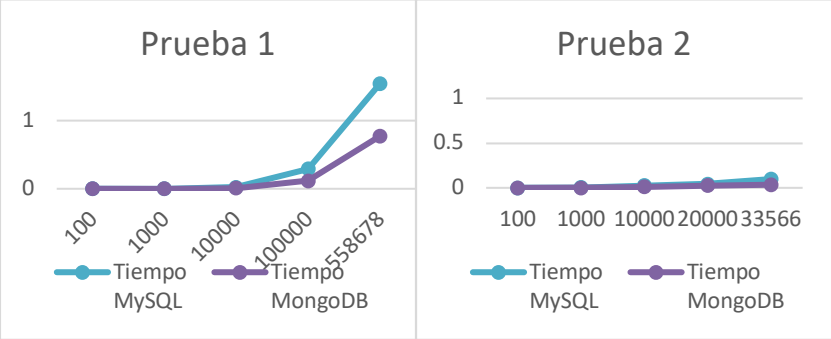


Fig. 5. Prueba 1: Evaluación de las consultas 10-14. Métricas: tiempo y registros recuperados. Prueba 2: Evaluación de las consultas 15-19. Métricas: tiempo y registros recuperados.

5 Conclusiones

En este artículo se realizó un análisis del ciberacoso en México con *bigdata* comparando MySQL con MongoDB. Se analizó la encuesta MOCIBA 2017 para generar el modelo conceptual independiente de implementación en E-R, se pre-procesó el conjunto de datos de MOCIBA2017 para poder importar los datos necesarios a una base de datos en MySQL y MongoDB. En MongoDB se crearon las colecciones y se pre-procesaron datos geoespaciales sobre los estados de México. Se implementaron 19 consultas no espaciales y 3 con datos geoespaciales en MongoDB. En las dos pruebas que se realizaron se observó que MongoDB presentó un mejor rendimiento en el procesamiento de datos y en el tiempo de ejecución de sus consultas, debido a la representación que ofrecen para bigdata los gestores NOSQL.

Referencias

7. Retana, B., Sánchez, R.: Acoso Cibernético: Validación en México del ORI-82. Revista acta de investigación psicológica 5(3), 2097-2111 (2015).
8. We are Social y Hootsuite, https://wearesocial.com/global-digital-report-2019, último acceso 23/11/2019.
9. Así sufren el ciberacoso los mexicanos | CNN, https://cnnespanol.cnn.com/2019/04/12/asi-sufren-el-ciberacoso-los-mexicanos/, último acceso 2019/12/01.
10. Chacón, A.: Una nueva cara de internet: el acoso. Revista científica electrónica de Educación y Comunicación en la Sociedad del Conocimiento 1, 1-10 (2003).
11. Smith, P., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., Tippett, N.: Cyberbullying: its nature and impact in secondary school pupils. Journal of Child Psychiatry 49(4), 376-385 (2008).
12. Patchin, J., Hinduja, S. Measuring cyberbullying: Implications. Research Aggression and Violent Behavior. 23(1), 69- 74 (2015).
13. INEGI | MOCIBA-2017, https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2019/EstSociodemo/MOCIBA-2017.pdf, último acceso 2019/11/23.

14. INEGI | Módulo sobre Ciberacoso 2017 MOCIBA Diseño conceptual, https://www.inegi.org.mx/contenidos/investigacion/ciberacoso/2017/doc/mociba2017_diseno_conceptual.pdf, último acceso 2019/11/23.
15. Oracle México, https://www.oracle.com/mx/big-data/guide/what-is-big-data.html, 2019/11/23.
16. Győrödi, C., Győrödi, R., Andrada Olah, I., Bandici, L.: A Comparative Study Between the Capabilities of MySQl Vs. MongoDB as a Back-End for an Online Platform. International Journal of Advanced Computer Science and Applications 7(11), 73-78 (2016).
17. Soni, S., Ambavane, M., Ambre, S., Maitra, S.: A Comparative Study: MongoDB vs MySQL. International Journal of Scientific & Engineering Research 8(5), 120-123 (2017).
18. Geoportal del Sistema Nacional de Información sobre Biodiversidad. http://www.conabio.gob.mx/informacion/gis/?vns=destdv250k_2cw, last accessed 2019/11/23.
19. ArcGIS, https://www.arcgis.com/home/index.html, last accessed 2019/11/21.
20. Jq, https://stedolan.github.io/jq/, last accessed 2019/11/25.

Sustracción de Fondo en Imágenes 3D utilizando Mezcla de Gaussianas

Miguel Angel Razo Salas[1], Ana Marcela Herrera Navarro[1], Hugo Jiménez Hernández[2]
[1]Facultad de Informática, Universidad Autónoma de Querétaro Campus Juriquilla
Av. de las Ciencias s/n Nuevo Juriquilla 76230 Juriquilla, Qro.
[2]Centro de Ingeniería y Desarrollo Industrial CIDESI
Av. Playa Pie de la Cuesta No. 702 Desarrollo San Pablo Querétaro, Qro.
[1]{mrazosalas21, anaherreranavarro}@gmail.com, [2]hugojh@gmail.com

Resumen. En la actualidad la visión por computadora ha generado mucho interés en diferentes ramas de la ciencia donde se han desarrollado innumerables aplicaciones utilizadas en la vida real dentro de campos como la navegación automatizada, video vigilancia, conteo de personas, reconocimiento facial, etc. Distintas técnicas son aplicadas en dichas aplicaciones, una de ellas es la sustracción de fondo que impacta directamente en el rendimiento de los sistemas, la segmentación de personas entre otras por el simple hecho de estar procesando información que no es útil para el fin al que se pretende llegar. Ahora que la información 3D es un tema de mucha relevancia este problema se ha trasladado a una nueva área en la cual facilita ciertos procesos. Diferentes sensores son utilizados para realizar este tipo de medición, la cámara TOF es uno de ellos. TOF entrega una imagen de profundidades sin necesidad de realizar ningún movimiento mecánico, no trabaja con modelos de textura ni sufre por los cambios lumínicos como las cámaras RGB. En esta investigación se aplica una técnica de sustracción de fondo utilizada ampliamente en imágenes a color a imágenes de profundidad, con el fin de evitar los errores por texturas con las mismas intensidades entre el fondo y los objetos del primer plano, la oclusión entre los mismos objetos, los fallos comunes de los modelos de apariencia con los cambios lumínicos, etc.
Palabras Clave: Sustracción de fondo, TOF, Time of flight, Filtrado.

1. Introducción

La sustracción de fondo y el filtrado de imágenes forma parte importante en el proceso de la detección de objetos en el escenario, estas técnicas son ampliamente utilizadas en aplicaciones como el seguimiento de personas, segmentación de objetos, videoconferencias, control de tráfico, etc. Aunque las imágenes RGB proveen muchas veces la información para la sustracción de fondo, presentan problemas relacionados principalmente con cambios de iluminación, sombras creadas por los objetos en movimiento, el camuflaje de los objetos y el fondo, entre otras. Recientemente la información 3D ha abierto nuevas formas de contrarrestar los problemas que tiene la visión por computadora en conjunto con el procesamiento de imágenes, otorgando información tridimensional del espacio.

Teniendo en cuenta que la velocidad de operación y la precisión de la información 3D influye directamente en el rendimiento de las aplicaciones, existen sensores de profundidad que proveen dicha información del escenario como lo son los scanners 3D que, en dependencia del hardware y de la tecnología de escaneo vienen con sus propias ventajas y costos. Algunos scanners pueden producir información 3D de alta calidad,

pero son demasiado caros y la mayoría de ellos requieren de un experto para su uso. Otro tipo de sensores utilizados para la caracterización de actividades son los sensores Radio Frequency Identification (RFID) con la desventaja que son demasiado intrusivos ya que es necesario ponerlos físicamente sobre el cuerpo al que se pretende detectar y estudiar [1].

Las cámaras tiempo de vuelo o TOF por sus siglas en ingles son una opción atractiva dado que entregan información geométrica del escenario dentro de un mapa 2D (rango y profundidad), el cual puede ser interpretado como una imagen a escala de grises, donde no requieren un gran procesamiento computacional o movilidad de partes mecánicas donde cabe mencionar que su uso es relativamente sencillo en relación con la manipulación de imágenes al momento de obtención de siluetas, volúmenes, pose, mímica, entre otras [2]. Al proveer información de profundidad en tiempo real y a alta velocidad de cuadros por segundo, abre nuevas posibilidades en campos como la reconstrucción tridimensional, realidad aumentada, video vigilancia, navegación robotizada, etc.

En general, los sensores tienen diferentes limitaciones, la señal IR reflejada por la tecnología TOF presenta variaciones en la amplitud y fase causando errores de medición en dependencia al color, reflectividad y estructura geométrica del objeto, por lo tanto, la arquitectura de censando registra ruido (bias) sistemático y no sistemático al momento de la medición. El ruido sistemático son factores internos causados por ejemplo por la baja potencia del láser infrarrojo y los tiempos cortos de integración [3].El ruido no sistemático esta dado por de factores externos los cuales no pueden ser influenciados por el usuario o por el diseño del sistema como la pobre reflexión de los objetos debido a sus texturas, pocos ángulos de reflexión, la multi reflectividad y la luz ambiental; por ejemplo, la precisión de la medida está limitada por la fuerza de la emisión la señal IR, la cual mayormente es menor comparada con la luz del día y por esta razón contamina la señal original, la amplitud de la señal IR reflejada también varia en dependencia del material y el color de la superficie del objeto. De esta manera la señal IR reflejada no es confiable para todo tipo de superficies, por ejemplo, los espejos causan reflexión, mientras que objetos translucidos causan refracción de la señar IR. Por esta razón es importante realizar el filtrado de datos para una tener una medición confiable y al momento de segmentar los objetos de interés no obtener objetos espurios.

La manera más sencilla de separar el fondo del primer plano es tener una imagen del fondo en el escenario en la cual no incluya ningún objeto en movimiento. En algunos de los escenarios esto no es posible debido a que el fondo tenga cambios drásticos debido a la iluminación u objetos que son agregados o eliminados del escenario, etc. Los métodos para la sustracción de fondo se clasifican como modelado básico de fondo, modelado de fondo estadístico, modelado fuzzy de fondo, estimación de fondo [4]. La limitante computacional ha sido la barrera dentro de las aplicaciones de procesamiento de imágenes en tiempo real. Debido a esto muchos de los enfoques que se han propuesto son demasiado lentos para ser prácticos o realizan bien el trabajo, pero con restringidos a escenarios controlados. En la actualidad los nuevos métodos permiten el modelado procesos reales bajo condiciones variantes.

2. Trabajos relacionados

La tecnología TOF tiene ciertas ventajas sobre las imágenes RGB debido a que muchos de las técnicas utilizadas en RGB se basan en modelos de apariencia que tienden a fallar por distintas causas como la iluminación en el escenario, el solapamiento entre los objetos, texturas homogéneas entre el fondo y los objetos, etc. Omar Arif y Wayne Daley [5] realizan una segmentación en función de una minimización de energía donde cada pixel es clasificado como fondo con valor de 0 y primer plano como valor de 1 utilizando un historial de la distancia obtenida en cada frame y el promedio de un modelo de fondo. Liang Wang [6] trabaja con imágenes RGB y de profundidad a la vez, utiliza también el etiquetado binario calculando la probabilidad de que un pixel pertenezca al fondo tanto en la parte de color como en la de profundidad cada que llega un frame nuevo.Faisal Mufti y Robert Mahony [7] realizaron la segmentación de sobras con base a un modelo radiométrico, información proporcionada por una cámara TOF donde obtienen sombras del escenario conociendo o sementando de igual manera el fondo. Hamed Tabkhi [8] hacen uso del algoritmo mezcla de gausseanas donde van modelando el fondo actualizando los valores de media, desviación estándar y peso de las gauseanas por cada pixel en cada frame. Murino V. [9] haciendo uso de un Kinect realizan una segmentación de fondo aplicando primero un umbral a la imagen de profundidades para eliminar los pixeles con ruido sin quitar la parte importante que son los objetos de la imagen, para después realizar la segmentación de los objetos haciendo crecer regiones a partir de los máximos de la imagen si y solo si el vecindario cumple ciertas condiciones. Javier Giacomanntone y Maria Lucia Violini [10] hacen uso de máquinas de soporte vectorial para la sustracción de fondo compuesto por dos partes importantes, la primer parte es el entrenamiento de la máquina de soporte vectorial para modelar el fondo y la segunda es el clasificador linear de fondo y objetos para imágenes TOF.

3. Filtrado de imágenes

El filtrado de imágenes es un proceso importante en cualquier técnica del procesamiento de imágenes o visión por computadora debido a que existen factores externos e internos en los sensores que provocan una mala medición generando ruido en la señal final. Los sensores TOF no son la excepción, por ejemplo, en dependencia al color, reflectividad y estructura geométrica del objeto, la señal IR reflejada muestra variaciones en la amplitud y fase causando errores de medición llamados factores externos ya que no pueden ser influenciados por el usuario o por el diseño del sistema. Por otro lado, los factores internos pueden estar dados por la baja potencia del láser infrarrojo y tiempos cortos de integración. Por esta razón es importante realizar el filtrado de datos para tener una medición confiable.

Es muy común que en imágenes obtenidas mediante sensores TOF existan estos errores de medición, problema que puede parecer sencillo, pero si no se resuelve es difícil de obtener resultados correctos en procesos que puedan o no llevarse a cabo después, para estos errores de medición se aplicó un filtro ampliamente utilizado en el procesamiento de imágenes llamado median filter o sal y pimienta [11], este filtro trabaja recorriendo la imagen completa reemplazando cada valor con el de la media que

tiene el vecindario definido por un elemento estructurante predefinido con el fin de eliminar partículas aisladas que tengan valores igual al mínimo 0 o al máximo que es el 255 en el caso de las imágenes en escala de grises (ver ecuación 1).

$$\delta(x,y) = \begin{cases} 0, & Ruido\ pimienta \\ 255, & Ruido\ sal \end{cases} \quad \textbf{(1)}$$

4. Sustracción de fondo

Una vez eliminados los errores de medición en las imágenes es importante segmentar los objetos en movimiento dentro del escenario con el fin de conocer su posición y límites para procesarlos de alguna manera en algún momento posterior. Para esto se utilizó la técnica mezcla de gaussianas, la cual, consiste en generar entre 3 y 5 distribuciones de probabilidad por razones de complejidad computacional por cada cambio o valor atípico en la imagen (ecuación 2).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \textbf{(2)}$$

Donde $f(x)$ es la función de distribución normal, σ es la desviación estándar y μ es la media de la distribución. De esta manera podemos asumir que dichas distribuciones tienden a modelar el fondo ya que los pixeles que pertenecen al fondo tienden a aparecer con mayor frecuencia en la imagen por ser estático. Es probable que esto no siempre ocurra de esta manera, es decir, puede llegar un valor atípico el cual no esté dentro de las gaussianas creadas, por lo tanto, podemos clasificar dicho valor como movimiento. Por esta razón, es necesario verificar obteniendo la distancia normalizada (ecuación 3), si el valor está dentro del intervalo de confianza establecido, tomando en cuenta que alrededor de un 68% de los valores están a una distancia d de $\sigma < 1$ de la media y el 95% de los valores está a $\sigma < 2$ de la media (μ).

$$d = \frac{(X-\mu)^2}{\sigma^2} \quad \textbf{(3)}$$

Teniendo en cuenta que X es el valor nuevo a procesar. Entonces si el valor pertenece a alguna de las distribuciones se puede asumir que está dentro del fondo y por consiguiente se actualiza μ y σ de dicha distribución con las ecuaciones 4 y 5.

$$\mu_t = P \times \mu_{t-1} + (1-P)(X) \quad \textbf{(4)}$$

$$\sigma^2{}_t = P \times \sigma_{t-1} + (1-P)(X-\mu_{t-1})^2 \quad \textbf{(5)}$$

Donde P es el factor de aprendizaje y μ_{t-1}es el valor de la media anterior. Por consiguiente, se puede decir que el algoritmo va "aprendiendo" de manera automática actualizando los valores de μ y σ conforme van llegando nuevas imágenes.

5. Remoción de objetos espurios

Morfología matemática trabaja con la teoría de conjuntos donde cada objeto en las imágenes puede ser representado por un conjunto, comúnmente es utilizada para investigar la interacción entre la imagen y ciertos elementos estructurales usando operaciones como la erosión y la dilatación. El elemento estructurante usualmente es más pequeño en relación a la imagen. En el caso de las imágenes digitales, es normal utilizar elementos estructurales con diferentes conectividades. La erosión morfológica esta dada por la ecuación 6 y se resume a colocar el elemento estructurante en cualquier parte de la imagen y verificar si esta completamente contenido por el objeto o conjunto dentro de la imagen, si es así el origen del elemento estructural es parte del conjunto a erosionar.

$$\varepsilon(X) = X \ominus \lambda B = \{x \mid \forall s \in S, x + s \in X\} \quad \textbf{(6)}$$

Donde S es el elemento estructurante. La dilatación es una operación extensiva, ya que las estructuras crecen después de aplicarla. Muchas veces es útil agregar una restricción a dicho crecimiento de manera que no rebasen el limite de un contorno predefinido. La dilatación geodésica está dada por la ecuación 7 y se define como el mínimo de dilataciones hasta llegar a una imagen marcador que restringe los límites de los objetos.

$$\delta_g(f) = min(\delta(f), g) \quad \textbf{(7)}$$

Donde $\delta(f)$ es la dilatación ordinaria y g es la imagen de control que restringe dichas dilataciones.

6. Metodología

La sustracción de fondo utilizada en esta investigación consta de tres etapas, en la primera etapa se aplica un proceso de filtrado de imágenes mediante el filtro sal y pimienta, y el promedio de un vecindario al pixel analizado. En una segunda etapa se aplica la sustracción de fondo mediante la técnica mezcla de gaussianas (MoG) y por último se aplica morfología matemática para eliminar partículas sobrantes que no son de nuestro interés. A continuación, se presenta un diagrama de bloques explicando la metodología que se siguió para esta investigación Fig. 1.

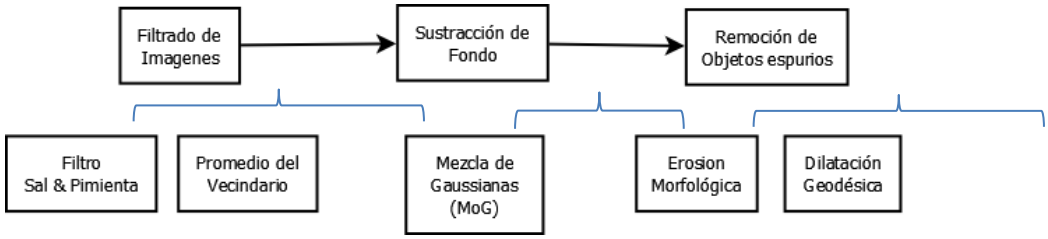


Fig. 1. Diagrama de flujo de la metodología que sigue esta investigación.

Como se mencionó en la introducción la señal reflejada del escenario contiene ruido por diferentes razones y para obtener una buena segmentación es importante eliminar la mayoría de este ruido posible. Se inicio aplicando el filtro sal y pimienta con el fin de eliminar las pequeñas partículas aisladas, que por sus características no son objetos de interés para nosotros. Posterior mente se recorrió la imagen en busca aquellos errores de medición que quedaron para eliminarlos aplicando el promedio con una ventana de *8x8* pixeles para asegurar obtener información suficiente del vecindario y así obtener una imagen filtrada removiendo la mayor parte del ruido que contenía la imagen original véase Fig.2.

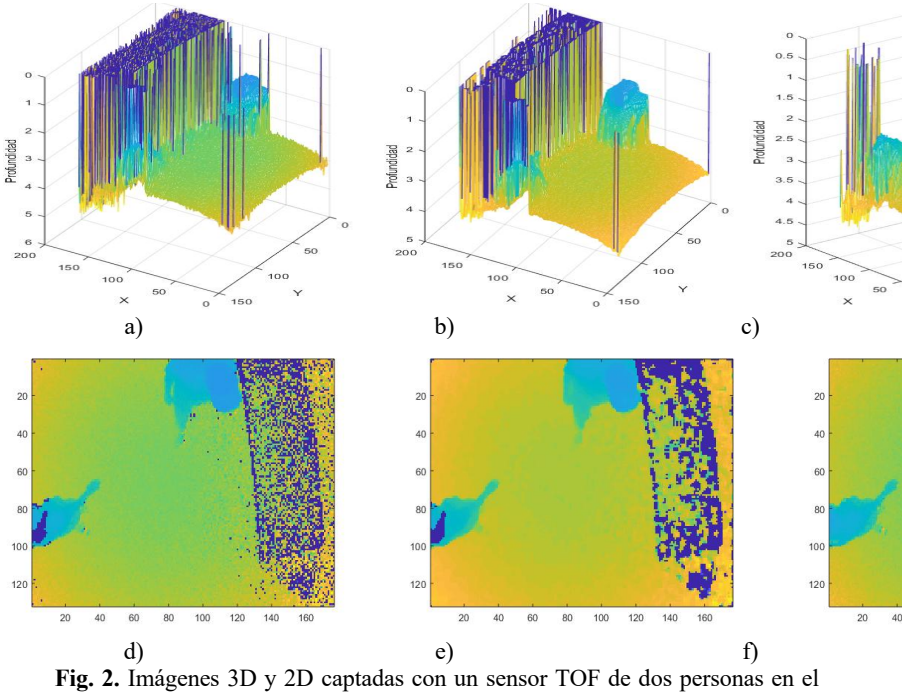


Fig. 2. Imágenes 3D y 2D captadas con un sensor TOF de dos personas en el escenario. a) y d) Imágenes 3D y 2D respectivamente del frame original, b) y e) Imágenes 3D y 2D respectivamente con el filtro sal y pimienta aplicado, c) y f) Imágenes 3D y 2D respectivamente con los errores de medición eliminados.

En la etapa de segmentación se utiliza un modelo de sustracción de fondo para segmentar las personas u objetos en movimiento. Esto se logra con una sustracción de fondo, es decir modelar el fondo para conocer que es movimiento y que parte es fondo con el algoritmo mezcla de gaussianas (MoG) utilizado ampliamente en imágenes a color.

Una vez segmentados los objetos en movimiento aún se muestran partículas que aparecen y desaparecen en el video, debido a las mediciones que no son constantes en objetos del escenario que no reflejan bien el rayo infrarrojo. Para lograr una segmentación adecuada se removieron estas partículas con morfología matemática, primero se utilizó una erosión con un elemento estructurante en forma de disco con un tamaño de 7 pixeles con el fin eliminar todos los objetos menos las personas de la imagen. Una vez aplicada la erosión se puede observar que los objetos sufren cambios, aquí es donde aplicamos la dilatación geodésica a la imagen erosionada tomando como marcador la imagen original sin la erosión y así lograr obtener una imagen completamente segmentada solo con los objetos de interés que son las personas en su forma original sin cambios en su estructura.

7. Resultados

Se grabó un video de 36,000 frames con una resolución de 132 × 176 pixeles para realizar pruebas del algoritmo de sustracción de fondo en donde se muestran dos personas dentro del escenario realizando diversas actividades. La cámara se colocó en el techo para trabajar desde una perspectiva superior de manera que se puedan observar los objetos y personas que entran al escenario con el fin de evitar las oclusiones entre ellos. Se inicio con imágenes en donde no existía movimiento para que el algoritmo fuera ajustando la media y la desviación estándar de las distribuciones de manera paulatina antes de agregar personas al escenario. Al principio la sustracción de fondo no hace efecto debido a que las distribuciones empiezan a aprender que valores de profundidad son fondo y que otros no lo son, por lo tanto, las imágenes de erosión, dilatación geodésica y la imagen 3D tampoco muestran nada. Después de un tiempo y varios frames adelante el algoritmo poco a poco va detectando movimiento entre las variaciones de medición en el suelo y las personas en el video gracias al aprendizaje de los frames anteriores, pero como las variaciones de medición son muy pequeñas en la imagen, la erosión aplicada las elimina y por consiguiente no hay ningún objeto que reconstruir con la dilatación geodésica.

Por último, se presenta una imagen donde el algoritmo ya tiene un buen aprendizaje y en la sustracción de fondo aplicada muestra a dos personas en el escenario en conjunto de las variaciones de medición sobre el suelo, al aplicar la erosión a la imagen vemos que las dos personas perseveran en la imagen para después aplicar la dilatación geodésica sobre esos dos objetos tomando como marcador la imagen de la sustracción de fondo véanse las Fig. 5, 6, 7 y 8.

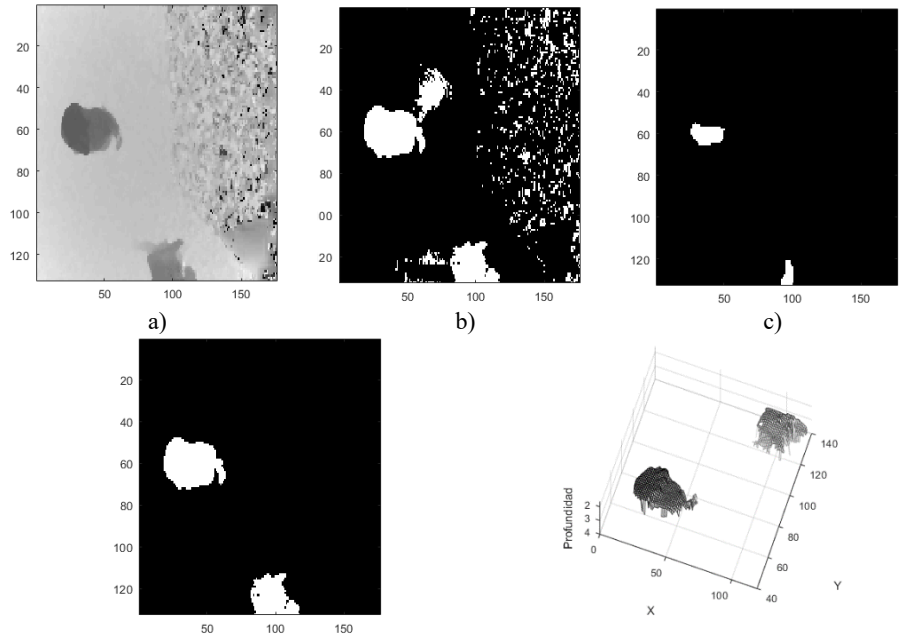


Fig. 5. a) Imagen original frame 3,900, b) Imagen con la sustracción de fondo, c) Imagen después de aplicar la erosión morfológica, d) Imagen de las personas reconstruidas por el filtro de morfológico de dilatación geodésica y por último e) representación de las dos personas en un espacio 3D.

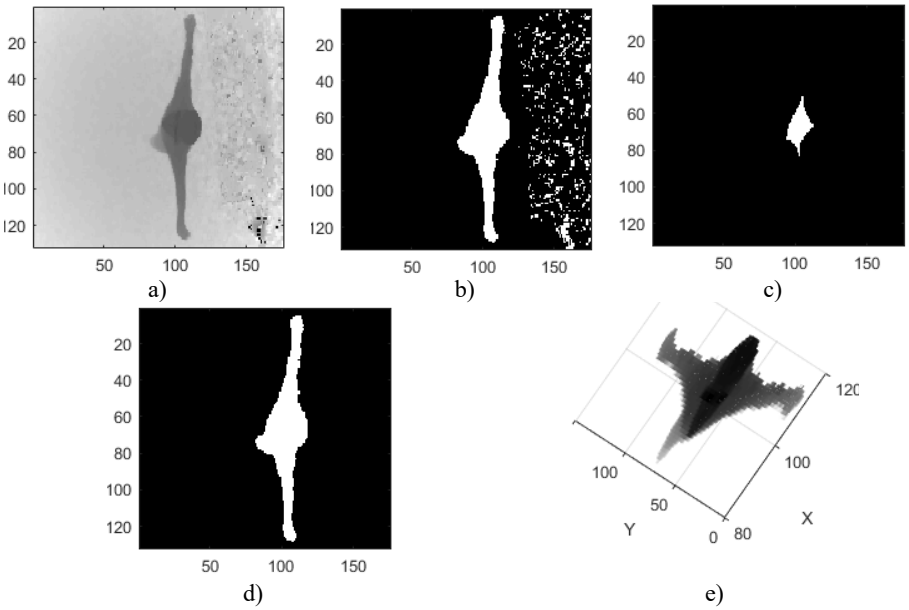


Fig. 6. a) Imagen original frame 5,682, b) Imagen con la sustracción de fondo, c) Imagen después de aplicar la erosión morfológica, d) Imagen de la persona reconstruida por el filtro de morfológico de dilatación geodésica y e) representación de la persona en un espacio 3D.

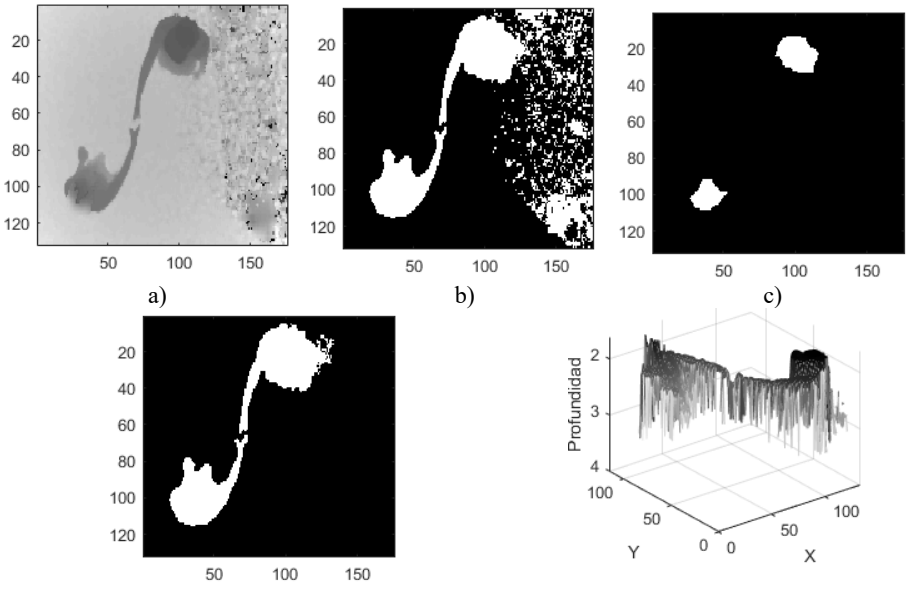
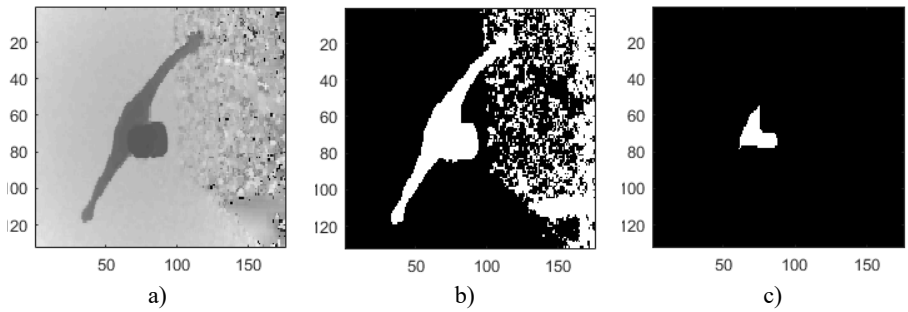
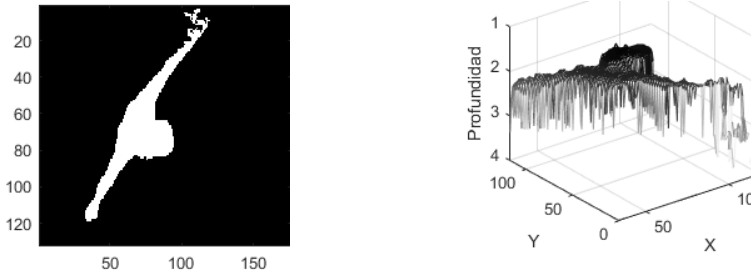


Fig. 7. a) Imagen original frame 13,332, b) Imagen con la sustracción de fondo, c) Imagen después de aplicar la erosión morfológica, d) Imagen de las personas reconstruidas por el filtro de morfológico de dilatación geodésica y e) representación de las dos personas en un espacio 3D.





d) e)

Fig. 8. a) Imagen original frame 20,668, b) Imagen con la sustracción de fondo, c) Imagen después de aplicar la erosión morfológica, d) Imagen de la persona reconstruida por el filtro de morfológico de dilatación geodésica y por último representación de la persona en un espacio 3D.

Es difícil encontrar bases de datos con imágenes de profundidad tomadas con cámaras tof que sirvan como *ground truth* y poder realizar una comparación entre el numero de personas que encontró el algoritmo dentro del escenario y los que en realidad hay, sin embargo, se hizo un etiquetamiento manual de las personas en los frames que se analizaron para corroborar que en realidad las segmentara de una manera correcta.

8. Conclusiones y trabajos futuros

Se logro satisfactoriamente la sustracción de fondo y la segmentación de personas en movimiento dentro imágenes de profundidad captadas por una cámara time of flight mediante el uso de la técnica mezcla de gaussianas, podemos llamarla adaptativa ya que va aprendiendo poco a poco con base al historial de cada pixel. No es necesario un aprendizaje previo con imágenes del fondo sin personas ni objetos en movimiento para poder aplicarlo, pero se recomienda comenzar con este tipo de imágenes donde solo sea el escenario sin movimiento con el fin de inicializar con distribuciones que contemplen solo fondo y a partir de esas empezar a generar las nuevas distribuciones que contendrán valores atípicos a los ya conocidos anteriormente que en su defecto se clasificaran como movimiento.

La sustracción de fondo como se mencionó en la introducción es importante ya que es un paso previo a muchos procesos posteriores como lo es el etiquetamiento de personas u objetos, seguimiento de personas, conteo de objetos, etc.

Como trabajo futuro se pretende utilizar posteriormente este trabajo de sustracción de fondo para obtener la estructura topológica o el esqueleto de las personas mediante erosiones 3D donde el cuerpo humano no está bien definido por la nube de puntos que entrega este tipo de cámara.

Referencias

1. D. Liciotti, E. Frontoni, P. Zingaretti, N. Bellotto, and T. Duckett, “HMM-based Activity Recognition with a Ceiling RGB-D Camera,” *Icpram Proc. 6Th Int. Conf. Pattern Recognit. Appl. Methods*, pp. 567–574, 2017.
2. A. Bevilacqua, L. Di Stefano, and P. Azzari, “People Tracking Using a Time-of-

Flight Depth Sensor," *AVSS '06 Proc. IEEE Int. Conf. Video Signal Based Surveill.*, p. 89, 2006.
3. M. Georgiev and R. Bregovi, "Fixed-Pattern Noise Modeling and Removal in Time-of-Flight Sensing," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 4, pp. 808–820, 2016.
4. T. Bouwmans *et al.*, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey To cite this version : HAL Id : hal-00338206 Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey," 2008.
5. O. Arif, W. Daley, P. Vela, J. Teizer, and J. Stewart, "Visual tracking and segmentation using Time-of-Flight sensor," no. September 2014, pp. 2241–2244, 2010.
6. L. Wang, C. Zhang, R. Yang, and C. Zhang, "TofCut: Towards Robust Real-Time Foreground Extraction Using a Time-of-Flight Camera," *Proc. Fifth Int. Symp. 3D Data Process. Vis. Transm. -- 3DPVT*, pp. 1–8, 2010.
7. F. Mufti and R. Mahony, "Shadow segmentation using time-of-flight cameras," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6978 LNCS, no. PART 1, pp. 78–87, 2011.
8. H. Tabkhi, R. Bushey, and G. Schirner, "Algorithm and architecture co-design of Mixture of Gaussian (MoG) background subtraction for embedded vision," *Conf. Rec. - Asilomar Conf. Signals, Syst. Comput.*, pp. 1815–1820, 2013.
9. V. Murino and E. Puppo, "Image Analysis and Processing – ICIAP 2015: 18th International Conference Genoa, Italy, September 7–11, 2015 Proceedings, Part II," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9280, pp. 56–65, 2015.
10. J. Giacomantone, M. L. Violini, and L. Lorenti, "Background Subtraction for Time of Flight Imaging," *J. Comput. Sci. Technol.*, vol. 17, no. 02, p. e18, 2017.
11. J. Azzeh, B. Zahran, and Z. Alqadi, "Salt and Pepper Noise: Effects and Removal," *JOIV Int. J. Informatics Vis.*, vol. 2, no. 4, p. 252, 2018.

Sistemas de reconocimiento automático de la Lengua de Señas Mexicana (LSM): Revisión Sistemática

Diana-Margarita Córdova-Esparza, Kenneth Mejía-Pérez, Nayeli Perales-Soto, Jaime-Rodrigo González-Rodríguez
Universidad Autónoma de Querétaro, Facultad de Informática, Av. de las Ciencias S/N, Campus Juriquilla, Querétaro, Qro. México, C.P. 76230
diana_mce@hotmail.com, ickennethmp@gmail.com, naye.ps@outlook.com, , jrodrigogr98@gmail.com

Resumen. En este documento, se presenta una revisión bibliográfica de sistemas existentes en la literatura para el reconocimiento automático de la Lengua de Señas Mexicana (LSM) que hacen uso de dispositivos electrónicos y algoritmos de Visión por Computadora e Inteligencia Artificial. Se analizaron dieciséis artículos de investigación mediante tablas comparativas donde específicamente se destaca información referente al modo de adquisición de las señas, tipos de señas y su representación, la técnica de aprendizaje automático utilizada y la tasa de reconocimiento obtenida. Entre los principales resultados de la revisión indican que la mayoría de los sistemas de reconocimiento de la LSM, analizan señas estáticas capturadas mediante cámaras convencionales y que usan como clasificador las redes neuronales artificiales.

Palabras Clave: Lengua de Señas Mexicana, Aprendizaje Automático, Reconocimiento de Patrones.

1 Introducción

1.1 Lengua de Señas Mexicana

La Lengua de Señas Mexicana (LSM) es uno de los principales medios de comunicación que utilizan las personas con problemas auditivos y la comunidad sorda en México. La LSM no solo consiste en un conjunto de signos gestuales articulados con las manos, también hace uso de signos no manuales que utilizan expresiones faciales, posturas corporales y mirada intencional para transmitir un significado semántico.
La LSM está compuesta de la dactilología referente a la representación manual de cada letra del abecedario y los ideogramas que describen una palabra con una o varias configuraciones de la mano [1].
Las señas se clasifican en estáticas y dinámicas como se muestran en la Fig. 1, y pueden representarse con una sola mano o haciendo uso de ambas manos.

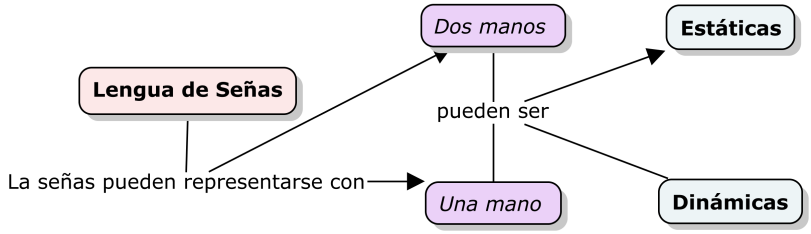


Fig. 1. Clasificación de la Lengua de Señas Mexicana.

1.2 Sistemas de reconocimiento automático de la Lengua de Señas Mexicana

Existen en la literatura diversos sistemas para el reconocimiento automático de la LSM que hacen uso de guantes, cámaras convencionales, sensores de profundidad (RGB-D), y otro tipo de dispositivos electrónicos que permiten recibir y transmitir información mediante el movimiento de las manos. Las principales de técnicas aprendizaje automático que utilizan estos sistemas se muestran en la Fig. 2.

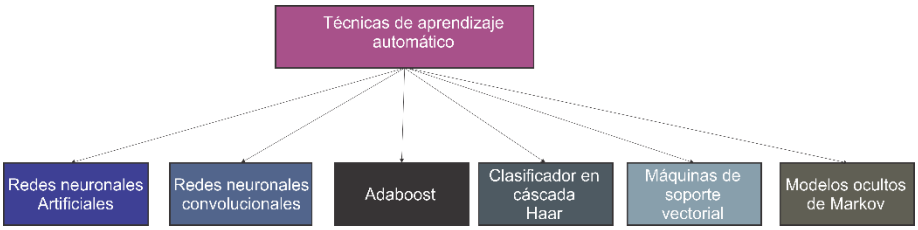


Fig. 2. Técnicas de aprendizaje automático utilizadas para el reconocimiento de la LSM.

2 Metodología

En esta sección se describen los sistemas existentes en la literatura para el reconocimiento de la Lengua de Señas Mexicana. Se realizó una búsqueda de artículos en las bases de datos especializadas mediante las palabras clave en español: Lengua de Señas Mexicana, Reconocimiento Automático, LSM. También se utilizaron palabras clave en inglés: *Mexican Sign Language Recognition* para ampliar la investigación.

La búsqueda se realizó considerando los siguientes criterios:

Criterios de inclusión:

- Artículos publicados durante los últimos cinco años de 2015 a 2020, con el objetivo de tener una revisión de la literatura actualizada.
- Artículos que en la metodología aplicaran técnicas de reconocimiento automático.
- Artículos en español e inglés.

Criterios de exclusión:

- Artículos publicados antes de 2015.
- Artículos en idiomas diferentes al español e inglés.
- Artículos de sistemas de reconocimiento de señas de otros países.

Para clasificar en categorías los artículos seleccionados, se identificaron los dispositivos de adquisición de cada sistema requeridos para llevar a cabo el reconocimiento de la Lengua de Señas Mexicana, obteniendo las cuatro categorías que se describen en las secciones 2.1 a 2.4. Para el análisis de los artículos se consideró el tipo de seña y su representación, además de identificar y comparar las técnicas de aprendizaje automático utilizadas y la tasa de reconocimiento.

2.1 Guantes traductores de la Lengua de Señas Mexicana

La recopilación de los trabajos cuyo objetivo es la implementación de un guante para la identificación de la Lengua de Señas Mexicana se encuentran en la Tabla 1. En el trabajo de Ruvalcaba et al. [2] se desarrolló un software en el IDE de Arduino para procesar la información proveniente de los distintos sensores del guante al momento de ejecutar 18 palabras básicas y letras del alfabeto. Para comprobar la efectividad en el reconocimiento de las señas, se lleva a cabo un proceso de verificación, el cual consiste en realizar 6 veces cada seña para poderla comparar con todos los datos de los sensores almacenados, de esta manera se delimitan los porcentajes de error en la ejecución de la seña y se amplía la capacidad de reconocimiento sin importar las pequeñas variaciones que se puedan presentar. Con este intervalo de error de ejecución definido individualmente en cada seña, al momento de solicitar el reconocimiento de un movimiento, se determina si los valores que se están recibiendo se encuentran dentro de los límites establecidos y, por lo tanto, en alguna de las categorías de clasificación, emitiendo una retroalimentación con el resultado obtenido mediante voz y texto.
La propuesta desarrollada por González et al. [3] cuenta con dos interfaces en LabVIEW, la primera realiza la adquisición de datos de los sensores en cada dedo del guante, los cuales son procesados fuera de línea para realizar el trabajo de clasificación de cada una de las letras del alfabeto. Para esta tarea se probaron distintos algoritmos de aprendizaje automático como lo son: J48, SMO y perceptrón multicapa, observando que el tiempo de análisis de cada uno de ellos es inversamente proporcional, siendo el perceptrón el que tiene un mayor tiempo de análisis y una precisión del 97%. La segunda interfaz se enfoca en el usuario quien realiza la seña para ser identificada y recibe una retroalimentación por escrito para corroborar la correcta ejecución de la seña.

Tabla 1. Guantes traductores de la Lengua de Señas Mexicana

CATEGORIA: GUANTES TRADUCTORES							
Autor, referencia y año	**Modo de adquisición**	**Una/ dos manos**	**Estático/ dinámico**	**Tipo de seña**	**Técnica utilizada**	**Base de datos**	**Tasa de reconocimiento**
Ruvalcaba et al. [2] (2018)	Guante (sensores)	Dos manos	Ambas	Letras/ Palabras	Comparación de la muestra con los parámetros almacenados en la base de datos	Propia	No se especifica
Saldaña-González et al. [3] (2018)	Guante (sensores)	Una mano	Ambas	Letras	J48 SMO Redes neuronales (Perceptrón multicapa)	Propia	J48: 89.04% SMO: 92.60% Perceptrón: 97%

2.2 Reconocimiento de la LSM utilizando *leap motion*

En la Tabla 2 se presentan los trabajos que hacen uso del sensor *leap motion*. En el trabajo de Cuecuecha et al. [4] desarrollaron una red neuronal perceptrón multicapa con función de activación sigmoidea. Para la clasificación usaron aprendizaje supervisado con un conjunto de datos que consta de 100 muestras (20 para cada vocal), tomadas de cuatro autores del proyecto y un intérprete traductor experto en lenguaje de señas. La mitad de estas muestras fueran utilizadas para el aprendizaje y la otra mitad para la etapa de validación. En el artículo presentado por Estrivero-Chavez et al. [5] para la etapa de entrenamiento utilizaron la herramienta *Classification Learner App* de MATLAB que consta de diferentes tipos de clasificadores. Se obtuvieron 20 muestras por cada una de las 27 señas a clasificar, con un total de 1080 archivos que incluyen dos clases: 540 corresponden a los datos de la posición de la mano y los 540 restantes a los datos de la rotación de la mano. El sistema es capaz de identificar la señal capturada mediante el Leap Motion, y se concluye que el SVM CUBIC es quien da un mejor rendimiento en precisión y estabilidad con respecto a los demás clasificadores.

Tabla 2. Sistemas que utilizan *leap motion* para el reconocimiento de la LSM

CATEGORÍA: LEAP MOTION							
Autor, referencia y año	**Modo de adquisición**	**Una/ dos manos**	**Estático/ dinámico**	**Tipo de seña**	**Técnica utilizada**	**Base de datos**	**Tasa de reconocimiento**
Cuecuecha et al. [4] (2018)	Leap Motion Sensor	Una mano	Estático	Letras	Redes Neuronales (Perceptrón multicapa)	Propia	100%
Estrivero-Chavez et al. [5] (2019)	Leap Motion Sensor	Dos manos	Estático	Letras/ palabras	Máquinas de Soporte Vectorial (SVM)	Propia	99.8%

2.3 Reconocimiento de la LSM mediante cámaras

En esta sección se describen brevemente los artículos que se encuentran en la Tabla 3, en referencia al uso de cámaras para el reconocimiento de la LSM. En el trabajo desarrollado por Solís et al. [6] se presenta un sistema que permite el reconocimiento de señas estáticas de la LSM, para realizar el análisis de las imágenes y extraer la información relevante de cada seña se utilizaron los Momentos de Jacobi-Fourier (JFMs, por sus siglas en inglés) y se hizo la clasificación mediante redes neuronales artificiales (ANN, por sus siglas en inglés). Además, se realizó una base de datos con imágenes adquiridas en un ambiente controlado para entrenar el clasificador perceptrón multicapa y se probó con el software WEKA, obteniendo una tasa de reconocimiento del 95%. Los autores Cervantes et al. [7] desarrollaron un método que analiza secuencias de video, de estas secuencias se extrajeron 743 características empleando un algoritmo genético. Recopilaron una base de datos con 5478 fragmentos de video para 249 palabras en LMS, cada palabra descrita por 22 personas. Para la clasificación de las señas se utilizó una máquina de soporte vectorial (SVM, por sus siglas en inglés) basándose en las características obtenidas del algoritmo genético. En [8], los autores Martínez et al. presentan un sistema para identificar la LSM aplicando técnicas básicas de procesamiento de imágenes, el desarrollo del proyecto se realizó en el lenguaje Java, incluyendo una interfaz gráfica para el usuario final. Para utilizar el sistema se captura una imagen mediante la cámara, esta imagen se procesa y se compara para determinar su similitud entre un conjunto de señas previamente registradas.

En el trabajo presentado por Solís et al. [9], se implementó un sistema para la detección de la LSM, que utiliza como descriptor los momentos normalizados invariantes a traslación y escala; y para el reconocimiento de patrones una red neuronal artificial. Esta red tiene 42 neuronas de entrada y 21 de salida, las pruebas fueron realizadas en un entorno totalmente controlado, con un fondo verde y cuatro paneles LED para reducir la sombra. Además, se desarrolló una interfaz gráfica en MATLAB para ejecutar el procesamiento. En [10], los autores Pérez et al. describen una metodología para el reconocimiento de objetos en una imagen, aplicando para ello diversos algoritmos. Para la segmentación de imágenes se utilizó el algoritmo de clustering difuso Fuzzy C-Means, para la extracción de características los momentos de Hu y para la clasificación redes neuronales artificiales. Para el entrenamiento del clasificador se utilizan los 7 momentos Hu en un arreglo vectorial de 7 columnas. La red neuronal solo usa una capa oculta, que contiene 177 neuronas y cada una de ellas tiene una función de activación sigmoidea, el vector de salida de la red es de 21 elementos que representa cada letra del alfabeto de la LSM.

En el trabajo desarrollado por Mancilla et al. [11], se realiza un análisis del desempeño de redes neuronales multicapa y una máquina de soporte vectorial (SVM) como clasificador para el reconocimiento de señas de la LSM. Se consideraron tres algoritmos para la extracción de características: Momentos de Hu, momentos de Zernike e histogramas de gradientes orientados (HOG, por sus siglas en inglés). Para entrenar y probar la red neuronal se creó una base de datos que consta de 6300 imágenes en total, que corresponden a 300 imágenes de cada una de las 21 señas estáticas, de estos datos el 70% se utilizó para el entrenamiento y el 30% para prueba, la programación se realizó en Python 2.7.

En el artículo de Martínez-Seis [12], se propone un sistema de reconocimiento de la LSM utilizando un smartphone para identificar 27 letras que incluyen señas estáticas y dinámicas. Se usaron las técnicas Canny, Camshift para el reconocimiento de bordes y trayectorias de señas en movimiento, para la clasificación de las señas se utilizaron redes neuronales convolucionales mediante la librería de Tensor Flow.

Tabla 3. Sistemas que utilizan cámaras para el reconocimiento de la LSM

CATEGORÍA: CÁMARAS CONVENCIONALES						
Autor, referencia y año	**Modo de adquisición**	**Una/ dos manos**	**Estático/ dinámico**	**Tipo de seña**	**Técnica utilizada**	**Tasa de reconocimiento**
Solis et al. [6] (2015)	Cámara	Una mano	Estático	Letras	Preprocesamiento: Momentos de Jacobi-Fourier Clasificador: Redes neuronales	95%
Cervantes et al. [7] (2016)	Cámara	Dos manos	Estático y dinámico	Palabras	Obtención de características: algoritmo genético. Clasificador: Máquina de soporte vectorial (SVM)	97%
Martínez et al. [8] (2016)	Cámara	Una mano	Estático	Letras	Técnicas básicas de procesamiento de imágenes. Identificación mediante la comparación de imágenes.	75%
Solís et al. [9] (2016)	Cámara	Dos manos	Estático	Letras	Obtención de características mediante momentos normalizados Clasificación: Redes Neuronales artificiales (ANN por sus siglas en inglés)	97%
Pérez et al. [10] (2017)	Cámara	Una mano	Estático	Letras	Segmentación: Fuzzy C-Means (FCM). Extracción de características: Momentos de Hu. Clasificación: Redes neuronales artificiales (ANN).	91%

Mancilla et al. [11] (2019)	Cámara	Una mano	Estático	Letras	Extracción de características: Momentos de Hu, Momentos de Zernike e Histogramas de gradientes orientados (HOG). Clasificación: Máquina de Soporte Vectorial (SVM) Redes Neuronales Multicapa.	98.7%
Martínez-Seis et al. [12] (2019)	Cámara (smartphone)	Una mano	Estático y Dinámico	Letras	Descripción: Canny y Camshift para señas dinámicas. Clasificación: Redes neuronales convolucionales	92%

2.4 Reconocimiento de la LSM utilizando cámaras RGB-D

En esta sección se describen los artículos correspondientes a la Tabla 4 que utilizan cámaras de profundidad para el reconocimiento de la LSM. En el trabajo desarrollado por Galicia et al. [13] se propone un sistema en tiempo real para el reconocimiento de las vocales: A, E, I, O, U y las letras L y B de la LSM. Se utilizó el algoritmo de bosque aleatorio para la extracción de características y para el reconocimiento de las señas se implementó una red neuronal de tres capas, obteniendo una precisión del sistema de 76.19%. En [14], Sosa-Jiménez et al. desarrollaron un sistema de visión cognitivo bimodal, como datos de entrada para el reconocimiento se tienen imágenes 2D y gestos 3D. A las imágenes 2D se les aplicó un pre-procesamiento para obtener los momentos geométricos que junto con las coordenadas en 3D se utilizan para entrenar los Modelos Ocultos de Markov (HMM) que permiten la clasificación de cada seña. En la propuesta presentada por García-Bautista et al. [15] se implementó un sistema en tiempo real que permite el reconocimiento de 20 palabras de la LSM, clasificadas en seis categorías semánticas: Saludos, preguntas, familia, pronombres, lugares y otras. Mediante el sensor Kinect v1 se adquirieron las imágenes de profundidad y se utilizó el SDK para realizar el seguimiento de las coordenadas 3D de las articulaciones del cuerpo. Se recolectaron 700 muestras de 20 palabras expresadas por 35 personas que utilizan la LSM. Se aplicó el algoritmo de Deformación Dinámica del Tiempo (DTW, por sus siglas en inglés) para interpretar los gestos de las manos y se utilizó el método de validación cruzada K-Fold para probar la precisión del sistema. Los autores Jiménez et al., [16] desarrollaron un sistema de reconocimiento alfanumérico de señas de la LSM. Crearon una base de datos con 10 categorías alfanuméricas (cinco letras: A, B, C, D y E; y cinco números: 1, 2, 3 4 y 5) cada una con 100 muestras. El 80% de los datos fueron usados para entrenamiento y el 20% para pruebas. Para el preprocesamiento de las imágenes se utilizaron técnicas de morfología matemática y se realizó la extracción

de características tipo Haar 3D de las imágenes de profundidad. La clasificación de las señas se hizo mediante el algoritmo AdaBoost (versión paralelizada) obteniendo una eficiencia del 95%. En [17], los autores Martínez-Gutiérrez et al., desarrollaron un software para la clasificación de señas estáticas mediante el uso de una cámara RGB-D. El software funciona con JAVA y el SDK de la cámara que permite capturar 22 puntos de la mano en coordenadas 3D. La información capturada se almacena en archivos CSV y se utiliza para el entrenamiento del clasificador que consiste en una red neuronal perceptrón multicapa y el algoritmo de retropropagación, los resultados obtenidos con la red indican una precisión del 80.1%.

Tabla 4. Sistemas que utilizan cámaras RGB-D para el reconocimiento de la LSM

CATEGORÍA: CÁMARAS RGB-D							
Autor, referencia y año	**Modo de adquisición**	**Una/ dos manos**	**Estático/ dinámico**	**Tipo de seña**	**Tiempo real**	**Técnica utilizada**	**Tasa de reconocimiento**
Galicia et al. [13] (2015)	Kinect	Ambas	Estático	Letras	Si	Extracción de características: Bosque aleatorio Clasificador: Redes neuronales	76.19%
Sosa-Jiménez et al. [14] (2017)	Kinect	Ambas	Dinámico	Palabras/ frases	Si	Preprocesamiento de las imágenes: Filtro de color Binarización Extracción de contorno Clasificador: Modelos Ocultos de Markov	Especificidad 80% Sensibilidad 86%
García-Bautista et al. [15] (2017)	Kinect	Ambas	Dinámico	Palabras	Si	Deformación dinámica del tiempo (DTW, por sus siglas en inglés)	98.57%
Jiménez, J. et al. [16] (2017)	Kinect	Una	Estático	Letras y números	No	Extracción de características tipo Haar 3D. Clasificador: Adaboost	95%
Martínez-Gutiérrez M. et al. [17] (2019)	Intel Realsense f200	Una	Estático	Letras y palabras	No	Obtención de las coordenadas 3D de la mano. Clasificador: Red neuronal perceptrón multicapa	80.11%

3 Discusión

En este trabajo de investigación se analizaron 16 artículos en relación a sistemas de reconocimiento automático de la Lengua de Señas Mexicana. Los cuales se clasificaron debido a sus características en cuatro categorías: 1) Guantes traductores, 2) Leap Motion, 3) Cámaras convencionales y 4) Cámaras RGB-D.

Los artículos analizados en la Tabla 1 corresponden a la implementación de guantes traductores para el reconocimiento de la Lengua de Señas Mexicana, en el trabajo de González et al. [3] se demostró que la técnica que da una mejor precisión es el perceptrón multicapa con una precisión del 97%, en comparación con el algoritmo de Optimización Secuencial Mínima (SMO, por sus siglas en inglés) que obtuvo una precisión de 92.6%. Sin embargo, en Ruvalcaba et al. [2] no se especifica un método para evaluar la precisión del sistema propuesto, debido a que los valores de los sensores se ajustan a cada seña para pueda ser identificada, de esta manera los márgenes de aceptación siempre deben preestablecerse. Sin embargo, una ventaja que se puede resaltar es que el sistema propuesto se aplica para señales que utilizan ambas manos, abriendo la posibilidad de identificar tanto letras como palabras. También cabe mencionar que el sistema puede funcionar de manera inalámbrica e independiente, gracias a los componentes que contiene, permitiendo una retroalimentación e identificación de la seña sin necesidad de hardware adicional, como lo es una computadora. Por otro lado, en Saldaña-González et al. [3] se necesita un equipo de cómputo con el software LabVIEW para poder acceder a las interfaces de funcionamiento del sistema propuesto.

En los trabajos revisados en la Tabla 2 correspondientes al uso del sensor Leap Motion, se puede observar una tasa de reconocimiento entre el 99% y 100%. Sin embargo, en el caso de Cuecuecha et al. [4], uno de los motivos por los cuales se obtuvo un porcentaje alto se debe a que solo se realiza la identificación de las vocales de la LSM, siendo un número reducido de categorías a clasificar y limitándose a señales estáticas que utilizan una sola mano. En cambio, en el trabajo de Estrivero-Chavez et al. [5] la clasificación es de 17 letras (incluidas las vocales) y 10 palabras, pero todas son identificadas de manera estática sin importar que sean señas que utilicen una o dos manos o que impliquen movimiento. Se puede destacar que dentro de este mismo trabajo se realizó la comparación de los resultados obtenidos con otros clasificadores utilizados en trabajos similares, demostrando que las Maquinas de Soporte Vectorial tienen una mayor precisión.

En relación a los artículos analizados en la Tabla 3 cuyos sistemas hacen uso de cámaras convencionales, se puede observar que para un reconocimiento adecuado es necesario tener un entorno controlado en el que existan las condiciones apropiadas para la adquisición de las imágenes. En el sistema descrito en el artículo de Solís et al. [6], utilizan un fondo de color sólido (verde) y una iluminación mediante cuatro paneles led para lograr un contraste entre la mano y el fondo, tratando de eliminar sombras al momento de capturar las imágenes, obteniendo una tasa de reconocimiento del 93%.

En el trabajo de Martinez et al. [8], utilizan también un fondo oscuro para adquirir las imágenes y se apoyan de técnicas de procesamiento de imágenes para delimitar el área y para extraer características adecuadas. Sin embargo, solo se obtuvo una tasa de reconocimiento de señas del 75%. Por otro lado, en Cervantes et al. [7], utilizan un ambiente totalmente controlado para la adquisición de las imágenes, las personas deben usar ropa negra y la escena debe tener un fondo negro para facilitar la segmentación y la extracción de características que se usan para entrenar el clasificador que consiste en una Máquina de Soporte Vectorial, logrando un reconocimiento promedio del 97%. Los autores Pérez et al. [10] también requieren de condiciones controladas para la captura de las imágenes. Sin embargo, utilizan la técnica Fuzzy C-Means (FCM) para la segmentación de las imágenes y aplican los momentos de Hu para la extracción de características, logrando un reconocimiento del 96% de las señas mediante el uso de redes neuronales. Por otra parte, los autores Solis et al. [6] utilizan un fondo sólido color blanco para separar con mayor facilidad las manos del fondo y utilizan los Momentos de Jacobi Fourier (JFMs) para la extracción de las características, obteniendo un 95% de precisión mediante el uso de redes neuronales artificiales. En contraste, los autores Mancilla et al. [11] no realizan la segmentación y extracción de características de imágenes adquiridas en entornos controlados. Sin embargo, utilizan tres descriptores para la extracción de características los cuales son: Momentos de Hu, Momentos de Zernike e Histogramas de gradientes orientados (HOG) y dos clasificadores que son: SVM y redes neuronales, obteniendo una tasa de reconocimiento del 98.7%. Finalmente, en el trabajo de Martínez-Seis et al. [12], se utiliza para el reconocimiento de las señas un clasificador basado en redes neuronales convolucionales. Se realizan experimentos con diferentes fondos y condiciones de luz similares, concluyendo que al tener un fondo blanco incrementa la tasa reconocimiento al 93.1%.

De acuerdo a los artículos revisados en la Tabla 4, referentes al uso de cámaras RGB-D se puede observar que una de las principales ventajas que ofrecen es la facilidad de realizar el reconocimiento de gestos sin necesidad de colocar dispositivos externos en el usuario, logrando una comunicación más natural. Además, estos trabajos aprovechan la funcionalidad del SDK de las cámaras RGB-D, el cual permite conocer la posición del cuerpo de las personas. En el caso de los trabajos de Sosa-Jiménez et al. [14] y García-Bautista et al. [15] se reconocieron en tiempo real palabras con movimiento que utilizan una o ambas manos, y se exploraron otras técnicas de clasificación diferentes a las redes neuronales como en el caso de [14] que usa Modelos Ocultos de Markov obteniendo resultados de especificidad y sensibilidad del 80% y 86%, respectivamente. Mientras que en [15] se utiliza como clasificador el algoritmo DTW obteniendo una tasa de reconocimiento del 98.57%. En [13], también se clasifican señas que utilizan ambas manos. Sin embargo, son señas estáticas y solo obtuvieron el 76.19% de reconocimiento mediante el uso de redes neuronales. En [16] y [17] los sistemas no son en tiempo real y solo clasifican señas estáticas que usan una sola mano, en el caso de [16] utiliza un clasificador Adaboost obteniendo un 95% de precisión. Mientras que [15] aplica una red neuronal multicapa logrando una de tasa de reconocimiento del 80.11%.

4 Conclusiones y trabajo futuro

En este trabajo de investigación se analizaron dieciséis artículos referentes al reconocimiento de la Lengua de Señas Mexicana (LSM), publicados en un periodo de cinco años de 2015 a 2020. Para el análisis se consideraron características como: el modo de adquisición de los datos, el tipo de seña (estática o dinámica) y su representación (con una mano o dos manos), uso de bases de datos, tipo de procesamiento, la técnica de clasificación y la tasa de reconocimiento.

De acuerdo con la revisión de la literatura, se puede observar que es limitado el número de trabajos que implementan guantes traductores de la LMS, para esta categoría se analizaron dos artículos que permiten reconocer señas de tipo estáticas y dinámicas. Al igual que los guantes, el número de trabajos que utilizan el dispositivo *leap motion* es reducido, solo se encontraron dos artículos que permiten reconocer señas estáticas de la LSM.

En referencia a los sistemas de reconocimiento que usan cámaras convencionales, se puede destacar que trabajan en entornos controlados y hacen uso de diferentes descriptores para mejorar el rendimiento del clasificador. De los siete artículos analizados en esta categoría, la mayoría reconocen señas estáticas y utilizan como clasificador redes neuronales artificiales.

Los sistemas de reconocimiento que utilizan cámaras RGB-D poseen la funcionalidad de trabajar con imágenes tridimensionales que facilitan el reconocimiento de señas dinámicas que se representan con una o dos manos, además este tipo de sistemas se consideran como no invasivos debido a que el usuario no requiere colocarse dispositivos externos. En esta categoría también se puede señalar que para la clasificación se exploran otras técnicas de aprendizaje automático como lo son: los modelos ocultos de Markov, el algoritmo DTW y el clasificador Adaboost y no solo el uso de redes neuronales artificiales y máquinas de soporte vectorial (SVM) como se indica en las otras categorías.

Finalmente, se puede concluir que el desarrollo de sistemas de reconocimiento automático de la Lengua de Señas Mexicana, continúa siendo un tema de investigación que tiene como principal objetivo facilitar la comunicación de personas con discapacidades y/o dificultades auditivas en México, haciendo uso de herramientas tecnológicas y algoritmos inteligentes que se encuentran en desarrollo y en mejora continua, tratando de eliminar las limitaciones existentes.

Referencias

1. Serafín, M. E., & González Pérez, R.: Manos con voz. Diccionario de Lengua de Señas Mexicana. Libre acceso. Consejo Nacional para Prevenir la Discriminación. México (2011).
2. Ruvalcaba, D. G., Ruvalcaba, M., Orozco, J. P., López, R., & Cañedo, C. E.: Prototipo de guantes traductores de la lengua de señas mexicana para personas con discapacidad auditiva y del habla. In *Memorias del Congreso Nacional de Ingeniería Biomédica*. Vol. 5, No. 1, pp. 350-353 (2018).

3. Saldaña González, G., Cerezo Sánchez, J., Bustillo Díaz, M. M., & Ata Pérez, A.: Recognition and classification of sign language for spanish. *Computación y Sistemas*, *22*(1), 271-277 (2018).
4. Cuecuecha, E., Martínez, J., Mendez, D., Saucedo, A. Zambrano A., Barreto A., Bautista, B., & Ayala, S.: Sistema de reconocimiento de vocales de la Lengua de Señas Mexicana. *Pistas Educativas*, vol. 39, No. 128 (2018).
5. Estrivero-Chavez, C. R., Contreras-Terán, M. A., Miranda-Hernández, J. A., Cárdenas-Cornejo, J. J., Ibarra-Manzano, M. A., & Almanza-Ojeda, D. L.: Toward a Mexican Sign Language System using Human Computer Interface. *In 2019 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pp. 13-17, IEEE (2019).
6. Solís, F., Toxqui, C., & Martínez, D.: Mexican sign language recognition using jacobi-fourier moments. *Engineering*, 7(10), 700 (2015).
7. Cervantes, J., García-Lamont, F., Rodríguez-Mazahua, L., Rendon, A. Y., & Chau, A. L.: Recognition of Mexican sign language from frames in video sequences. In *International Conference on Intelligent Computing*, pp. 353-362. Springer, Cham (2016).
8. Martínez, M., Rojano-Cáceres, J., Bárcenas I., & Juárez, F.: Identificación de lengua de señas mediante técnicas de procesamiento de imágenes. Res. Comput. Sci., 128, 121-129 (2016).
9. Solís, F., Martínez, D., & Espinoza, O.: Automatic mexican sign language recognition using normalized moments and artificial neural networks. *Engineering*, 8(10), 733-740, (2016).
10. Pérez, L. M., Rosales, A. J., Gallegos, F. J., & Barba, A. V.: LSM static signs recognition using image processing. In *2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1-5. IEEE (2017).
11. Mancilla, E., Vázquez, O., Arguijo, P., Meléndez, R., & Vázquez, A.: Traducción del lenguaje de señas usando visión por computadora. *Research in Computing Science*, 148, 79-89 (2019).
12. Martinez-Seis, B., Pichardo-Lagunas, O., Rodríguez-Aguilar, E., & Saucedo-Díaz, E. R.: Identification of Static and Dynamic Signs of the Mexican Sign Language Alphabet for Smartphones using Deep Learning and Image Processing. *Research in Computing Science*, 148, 199-211 (2019).
13. Galicia, R., Carranza, O., Jiménez, E. D., & Rivera, G. E.: Mexican sign language recognition using movement sensor. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pp. 573-578, IEEE (2015).
14. Sosa-Jiménez, C. O., Ríos-Figueroa, H. V., Rechy-Ramírez, E. J., Marin-Hernandez, A., & González-Cosío, A. L. S.: Real-time mexican sign language recognition. In *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pp. 1-6, IEEE (2017).
15. García-Bautista, G., Trujillo-Romero, F., & Caballero-Morales, S. O.: Mexican sign language recognition using kinect and data time warping algorithm. In *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)* pp. 1-5. IEEE (2017).
16. Jiménez, J., Martin, A., Uc, V., & Espinosa, A.: Mexican sign language alphanumerical gestures recognition using 3D Haar-like features. *IEEE Latin America Transactions*, 15(10), 2000-2005 (2017).
17. Martínez-Gutiérrez, M. E., Rojano-Cáceres, J. R., Benítez-Guerrero, E., & Sánchez-Barrera, H. E. Data Acquisition Software for Sign Language Recognition. *Research in Computing Science*, 148, 205-211 (2019).

Implementación de GRASP en Java para la búsqueda de soluciones del SQAP

Rogelio Gonález-Velázquez[1], Erika Granillo-Martínez[2], M. Beatriz Bernábe-Loranca[1], Jairo E. Powell-González[1]

[1] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Prol. 14 sur Esq. Av. Sn. Claudio, C.P 72590, Puebla, México
[1]{rogelio.gzzvzz,beatriz.bernabe}@gmail.com, [1]jairoe.powell@viep.com.mx
[2] Facultad de Administración, Benemérita Universidad Autónoma de Puebla, Av. Sn. Claudio, C.P 72590, Puebla, México
[2]erika.granillo76@gmail.com

Resumen. El Problema de Asignación Cuadrática (QAP por sus siglas en inglés) consiste en encontrar una asignación óptima de *n* instalaciones a *n* localidades de tal manera que se minimice el costo de interacción entre las instalaciones. El QAP es NP-hard. En este artículo se presenta una aplicación del QAP que consiste en modelar un sistema de tráfico introduciendo una matriz de transición de probabilidad para transformar el QAP en el Problema de Asignación Cuadrática Estocástico (SQAP por sus siglas en inglés). Este artículo tiene por objetivo buscar soluciones para el SQAP por medio de la implementación de la metaheurística Procedimiento de Búsqueda Voraz, Aleatorizada y Adaptativa (GRASP por sus siglas en inglés), en el lenguaje de programación JAVA. Se ejecuta el programa para un conjunto de instancias de prueba y se muestran los resultados obtenidos por la aplicación de dos estrategias de búsqueda que hacen cuatro combinaciones en las fases de construcción y las fases de post-procesamiento de la metaheurística. Ante la necesidad de ofrecer soluciones a problemas de logística de la clase NP-hard se presenta una herramienta de aproximación a las soluciones óptimas.

Palabras Clave: Metaheuristica, GRASP, Asignación, Estocástica, NP-hard.

1 Introducción

El Problema de Asignación Cuadrática QAP es un problema de optimización discreta en particular de optimización combinatoria y consiste en encontrar una asignación óptima de *n* instalaciones a *n* localidades con el propósito de minimizar el costo de transporte, dadas dos matrices, una matriz de distancias entre las localidades y otra con el flujo de materiales entre las instalaciones. Nótese que cada instalación sólo puede ser asignado a una localidad y cada localidad sólo puede aceptar una instalación, es decir es una relación uno a uno. Las matrices de distancia y de flujo de materiales son simétricas. Lo que se requiere es que las instalaciones que tienen mayor flujo de materiales estén lo más cercanas posible con el fin de minimizar el costo de transporte. Las soluciones del QAP planteados como un problema de optimización combinatoria son permutaciones observe en la figura 1 la solución de esa asignación es 2413 que

representa la asignación la instalación 2 a la ciudad A, la 4 a la ciudad B, la 1 a la ciudad C y la 3 a la ciudad D.

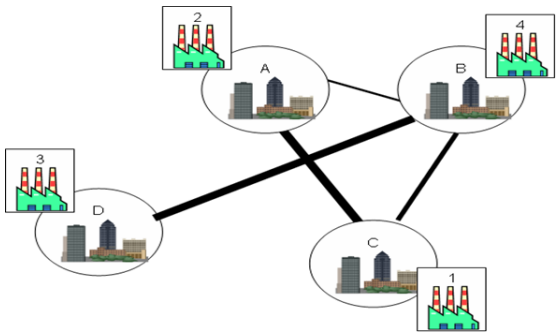


Fig. 1 Esquema del QAP para n=4.

Por otro lado Loila et al [1] menciona que el QAP es uno de los problemas más difíciles de la clase NP-hard y que la motivación para su estudio es la cantidad de aplicaciones que se pueden encontrar en áreas logística en investigación de operaciones y análisis combinatorio de datos en ciencias de la computación, para reforzar la importancia teórica del estudio del QAP adicionalmente en [1] se afirma que los siguientes problemas se pueden plantear como el QAP son: Traveling Salemaman Problem, Bin Packing, Maximal Clique, Isomorphism and Graph Partiionig y en el artículo [1] *A survey for the quadratic assignment problem* se presenta un análisis de 362 publicaciones donde el 95% están directamente relacionadas con el QAP.

El QAP fue propuesto por Koopmans y Beckmann en 1957 [12] como modelo matemático para la asignación de actividades económicas. En 1976 Shani y González probaron que QAP es un problema NP-hard [9] y demuestran que para los problemas de esta clase no existe un algoritmo exacto que los pueda resolver en un tiempo de orden polinómico, he aquí otra de las razones de interés en estudiar los problemas NP-hard.

Hasta hoy sólo se han encontrado soluciones óptimas usando métodos exactos para instancias de tamaño menores que 30 [11]. El QAP aparece en muchas aplicaciones, tales como el diseño de teclados de computadora, la programación de manufactura, el diseño de terminales en aeropuertos y procesos de comunicaciones, entre otras. El algoritmo exacto más popular para resolver el QAP es el de ramificación y acotamiento con algunas variantes, el primer algoritmo exacto de ramificación y acotamiento en paralelo fue propuesto por Roucairol [11], sin embargo en los últimos años se ha propuesto su solución mediante diferentes técnicas de aproximación llamadas metahurísticas para el QAP [2], tales como Colonia de Hormigas (ACO), Redes neuronales Artificiales (NN), Algoritmos Genéticos (GA), Búsqueda dispersa (SS), Recocido Simulado (SA), Búsqueda Tabu (TS) [6] y GRASP [3] entre otras. Pardalos y Pitsoulis [5] implementaron GRASP en paralelo para el QAP.

Los procedimientos metaheurísticos son una clase de métodos aproximación, diseñados para resolver problemas difíciles de optimización combinatoria, donde los

heurísticos clásicos no son ni efectivos ni eficientes mucho menos los métodos exactos [10].

Las metaheuristicas tienen la virtud de obtener soluciones cercanas a la óptima en un tiempo razonable de computo con respecto al al tamaño del problema uso moderado de los recursos de computo

En este artículo se presenta el diseño de una metaheuristiaca GRASP para el QAP y aplicado al SQAP y las instancias de prueba se tomaron de la página web QAPLIB.

GRASP es un procedimiento iterativo en donde cada paso consiste en una primera fase de construcción y otra fase de mejora. En la fase de construcción se aplica un procedimiento heurístico constructivo para obtener una buena solución inicial. Esta solución se mejora en la segunda fase mediante un algoritmo de búsqueda local. La mejor de todas las soluciones examinadas se tomas como resultado final [3,4].
Los resúmenes y análisis de la eficiencia de GRASP en una gran variedad de problemas se pueden encontrar en [13].

El artículo está organizado de la manera siguiente, en la sección 2 se describen los dos modelos matemáticos para el QAP, en la sección 3 se explica el diseño de la meta-heurística GRASP para el QAP, en la sección 4 se describe la transformación del QAP al SQAP, en la sección 5 se muestran los resultados obtenidos y finalmente en la sección 6 se dan las conclusiones y trabajo futuro.

2 Modelos matemáticos del QAP

El QAP tiene dos modelos para su planteamiento, como problema de optimización combinatoria es la siguiente:

Sean $\aleph$={1,2,..., n} y F=(f_{ij}) y D=(d_{kl}) dos matrices cuadradas de *n* x *n*; se trata de encontrar la asignación de *n* instalaciones a *n* localidades, es decir una permutación $p \in \Pi_{\aleph}$ que minimice.

$$\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}\, d_{p(i)p(j)} \qquad (1)$$

donde $\Pi_{\aleph}$ es el conjunto de todas las permutaciones del conjunto $\aleph$. f_{ij} representa el flujo de materiales de la planta *i* a la planta *j* y d_{kl} es la distancia de la ciudad *k* a la ciudad *l*.

El QAP también se puede plantear como un problema de programación entera binaria de la siguiente forma

$$Min\ f = tr(FXDX^t) \qquad (2)$$

Sujeto a:

$$\sum_{i=1}^{n} x_{ik} = 1\ (k = 1,2 \dots, n) \qquad (3)$$

$$\sum_{k=1}^{n} x_{ik} = 1\ (i = 1,1 \dots, n) \qquad (4)$$

$$x_{ik} = \begin{Bmatrix} 1\ si\ i\ se\ asigna\ a\ k \\ 0\ en\ otro\ caso \end{Bmatrix}$$

$$(i, k = 1,2 \ldots, n) \quad (5)$$

Donde *X* es una matriz de variables binarias y tr es la traza de la matriz, con las matrices *F* y *D* simétricas. Para nuestra aplicación la matriz de flujo *F* es una matriz no simétrica estocástica y *D* es la matriz de distancias la cual sigue siendo simétrica, con lo cual la función objetivo del problema pierde la estructura de traza de programación entera anterior, pero conserva la estructura de optimización combinatoria.

3 GRASP para el QAP

El diseño de este GRASP ha sido empleado por algunos investigadores para resolver el QAP para diferentes instancias Li, Resende y Pardalos [3]. El pseudocódigo se muestra en la figura 1.

3.1 Fase de construcción

Etapa 1. Las dos asignaciones iniciales se hacen simultáneamente, específicamente diremos que el recurso *i* es asignado a la localidad *k* y el recurso *j* es asignado a la localidad *l*, cuando su costo correspondiente a este par de asignaciones es (f_{ij}) (d_{kl}). Sean α, β, (0 < , α, β, < 1) los parámetros que restringen la lista de candidatos, F = (f_{ij}) y D = (d_{kl}) las matrices simétricas *n* x *n* con ceros en la diagonal.

Se procede a listar estas entradas de las distancias y los flujos en orden creciente y decreciente respectivamente.

Ahora tenemos dos listas ordenadas; usaremos el parámetro β para restringir ambas listas, por lo cual se cortan hasta el elemento [β*m*]. Se genera una nueva lista de costos multiplicando las distancias por los flujos en el orden correspondiente, así, se tiene la nueva lista, la cual se ordena en forma creciente, ahora usamos el parámetro α para obtener la lista restringida y definitiva de los candidatos (LRC), de la cual sólo se tomarán los primeros [αβ*m*] elementos y se elegirá aleatoriamente un elemento $(f_{ij})(d_{kl})$ que representa un costo de hacer un par de asignaciones (*i, k*) y (*j, l*) , es decir, tenemos dos componentes de la solución, que para simplificar será escrita como permutación, donde la componente *k*-ésima y la componente *l*-ésima son colocadas. Con esto concluye la primera etapa de la fase de construcción. En el algoritmo 1 se mustra el procedimiento de la construcción de una solución con el greedy.

Algoritmo 1. Pseudocódigo de la fase construcción

```
Procedure ConstructSolucionGreedyRandomizeAdaptative
1 Solucion = {};
2       While (solucion incompleta) do
3              CrearLCR ();
4              s = ElementoSeleccionadoAleatoriamente(LCR)
5              Solucion = Solucion  ∪  {s};
6             AdaptativoAvido(s,LCR);
7       end {while}
```

```
8 Return (SolucionCompleta)
End {ConstructSolucionGreedyRandomizeAdaptative}
```

Etapa 2. En esta etapa lo que se pretende es completar la solución inicial calculando las *n*-2 asignaciones restantes, mediante un procedimiento ávido que produce una a una las asignaciones que tienen el costo mínimo con respecto a las asignaciones ya existentes y que en caso de empate se romperá aleatoriamente y apoyándose en una componente adaptativa que se encarga de actualizar la solución a medida que esta se va construyendo.

Sea: Γ el conjunto de asignaciones que está en construcción. La etapa 2 inicia con $|\Gamma| = 2$ a consecuencia de los resultados de la etapa1.

$$Sea \qquad C_{ij} = \sum_{(j,l)\in\Gamma} f_{ij}\, d_{kl}$$

el costo de asignar a planta i a la localidad k con respecto a las asignaciones ya existentes. Seleccionamos de las parejas (i, k) no asignadas la que tenga el costo mínimo C_{ik} , en esto consiste el procedimiento ávido.

En esta parte también hay una lista restringida de candidatos, se ordenan los C_{ik} en forma creciente y se toma aletoriamente uno de los primeros $[\alpha z]$, donde z es la cantidad de parejas aun no asignadas.

La componente adaptativa de GRASP tiene como función actualizar el conjunto Γ adicionando nuevas parejas asignadas, es decir $\Gamma = \Gamma \cup \{(i, k)\}$, al finalizar esta etapa concluye también la primera fase, se ha construido una solución contenida en el conjunto Γ ordenando las primeras componentes de las parejas, tomamos las segundas componentes para formar la permutación equivalente a la solución. En resumen, se tiene una solución de buena calidad para arrancar la segunda fase. La etapa 2.

3.2 Fase de mejoramiento

Empezamos la segunda fase o fase de mejoramiento tomando como solución inicial la solución emanada de la primera fase empleando algún procedimiento de búsqueda local, cuando finalice este procedimiento se tendrá una solución óptima local, que pudiera ser también óptimo global, el algoritmo 2 nos muestra el pseudocódigo de la búsqueda local.

```
Algoritmo 2. Pseudocódigo para la fase de búsqueda local
Procedure local (p, V(p), s)
1       While s no sea óptima local do
2            Encontrar una mejor solución t ∈ V (s);
3            Sea s = t;
4       End;{ While }
5      Return (s como optima local)
End local;
```

4 Modelo SQAP

El SQAP es una aplicación del QAP propuesta por Wu-Ju Li y Macgregor Smit [7], esta transformación se realiza por medio de modelar un sistema de tráfico de personas en un centro comercial, las personas que llegan al centro comercial se incorporan a un pasillo central (área de circulación del flujo de clientes) también conocido como nodo Steiner, van caminado en dirección al local de su preferencia con una probabilidad de ir al local *i* al salir se dirigen al local j con otra probabilidad de elección este, comportamiento del sistema de tráfico se modela por medio de una línea de espera con tasa de llegada, tasa de salida con los parámetros definidos a continuación, pero antes establecemos que el objetivo es la asignación de los servicios a los locales del centro comercial de tal manera que se planea la disposición de los negocios de tal manera que el tráfico se ágil y evite aglomeraciones.

Para la formulación matemática en general del SQAP como en [7]consideremos la notación siguiente:

v = velocidad de traslado de los clientes.

d_{ij}= distancia entre los nodos (o locales) *i* al *j*

t_{ij} = tiempo de servicio al cliente que deja el nodo *i* para ir al nodo *j* dado por d_{ij}/v.

p_{ij} =la probabilidad de que un cliente salga del nodo *i* al nodo *j*.

λ_j= la tasa de llegada de un cliente al nodo j.

Consideremos que el nodo Steiner, es un servidor de capacidad ilimitada, es decir que tan pronto un cliente entra a la circulación del sistema existe un canal que le da el servicio, que está en función de la distancia recorrida en su traslado, es decir la línea de espera se comporta como si cada cliente tuviera su propio servidor, el nodo Steiner se comporta como un sistema de línea de espera [5].

En un sistema de líneas de espera [8] $M/G/\infty$ el tiempo promedio de estadía en el sistema es igual al tiempo promedio de servicio en la línea de espera. En este caso el tiempo de estadía en el sistema es el tiempo que el cliente tarda en el nodo Steiner que es el tiempo necesario para trasladarse del nodo *i* al nodo *j* en sistema. Ya que el tiempo de estadía para los clientes del nodo Steiner al nodo *i* es $1/\mu_i$, aplicando la ley de Little, el tiempo promedio de los clientes en el nodo Steiner al nodo *i* es $\lambda_i(1/\mu_i)$. El número total de clientes en el sistema es: $\lambda_i p_{ij} t_{ij} \equiv q_{ij} \frac{d_{ij}}{v}$

Donde $q_{ij}=\lambda_i p_{ij}$, ($i=1,2,\ldots,n$; $j=1,2,\ldots,n$). se trata de encontrar la asignación de *n* servicios a *n* locales, es decir una permutación $p \in \Pi_N$ que minimice:

$$\frac{1}{v}\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d_{p(i)p(j)} \qquad (6)$$

Consideremos el caso específico mencionado antes, se requieren las matrices de distancias *D*, la matriz de transición *P* (Probabilidad) y la matriz de flujo *F*, el vector columna λ_i y tomando v=1 [7].

5 Resultados

Para la obtención de los resultados se tomaron las instancias obtenidas de QAPLIB [16], diseñadas por C.E. Nugent, T.E. Vollmann and J. Ruml en [14], de dimensión 12,

14, 15, 20, 21, 22, 24, 25 y 30, cada instancia está integrada por tres parámetros de entrada que son: la dimensión, la matriz de distancias D y matriz de flujo F. La matriz de flujo fue modificada para obtener una matriz de transición. La implementación propuesta de GRASP en este artículo se ejecuta con 500 iteraciones fijas, los parámetros para la LRC son $\alpha = 0.2$ y $\beta = 0.3$ pero con la primera fase de GRASP con dos opciones se genera una solución inicial, la segunda opción es generar múltiples soluciones de arranque y se sigue con una fase de post procesamiento con dos opciones una búsqueda por vecindades del tipo 2-intercambio y otra por vecino adyacente

El programa se diseñó en lenguaje de programación JAVA y las ejecuciones se llevaron a cabo en una computadora laptop TOSHIBA con procesador i7. Este programa fue probado también en instancias de QAPLIB [16] de Skorin Kapov [6] con las dimensiones que van de dimensión 42 hasta la 100e como se muestra Granillo (2018) [15], en estas pruebas se obtuvieron excelentes resultados aproximación a los óptimos como en el tiempo de ejecución. La tabla 1 muestra el resumen de los mejores resultados obtenidos en las cuatro combinaciones de la ejecución del programa propuesto, en la primera columna las siglas DM significa dimensión de la matriz, C_PFP significa combinación primera fase y fase de post-procesamiento de GRASP,

MSG2-I es primera fase múltiples soluciones greedy y segunda fase 2-intercambio, G_2-I es primera fase algoritmo greedy con una sola solución y segunda fase 2-intercambio, MVE es mejor valor encontrado por la combinación, PVE es peor valor encontrado por una combinación, TCPU/milisegundos tiempo de ejecución de la corrida de la mejor combinación. En la tabla 2 se muestran las permutaciones de asignación de cada instancia de prueba, que indican la asignación de tiendas a locales en un centro comercial y los valores objetivos es la cantidad de circulación en un pasillo central.

Tabla 1. Resultados de ejecuciones del programa secuencial de GRASP en JAVA

DM	C_PFP	MVE	PVE	TCPU/milisegundos
12	MSG2-I	44.99	49.6	90
14	G_2-I	80.79	90.60	291
15	MSG2-I	91	101.20	191
20	G_2-I	194.99	215.99	1061
21	G_2-I	197.40	209	1348
22	G_2-I	300.40	310.20	1719
24	G_2-I	279.20	293.6	2252
25	G_2-I	308.40	340.59	2670
30	G_2-I	486	523.8	5556

Tabla 2. Se muestran las soluciones de cada instancia

DM	Permutación
12	4,6,11,8,5,10,2,1,12,7,9,3
14	9,3,8,13,1,12,5,7,6,10,11,14,2,4
15	4,2,11,14,6,13,7,12,3,10,1,8,9,5,15
20	17,5,7,4,13,8,6,2,19,20,1,10,12,11,9,3,18,15,14,16

21	5,2,9,14,6,11,4,20,21,3,18,16,8,15,12,13,7,19,10,1,17
22	17,15,4,16,8,11,22,10,7,21,20,5,6,14,18,1,3,19,12,13,9,2
24	2,19,1,18,6,5,21,13,12,3,9,14,7,10,22,11,16,23,20,24,17,8,15,4
25	5,12,4,20,22,24,8,25,10,15,17,23,14,7,13,18,3,9,19,16,2,6,11,21,1
30	24,25,19,7,21,28,17,1,10,2,13,30,26,12,22,8,20,29,5,23,11,3,9,4,1 5,6,18,27,16,14

El programa también fue ejecutado para un problema propuesto por Wu-Ji [7] que presenta instancias de tamaño $n = 12$, cuya solución óptima está dada por la permutación 7,3,2,11,8,12,9,10,1,4,6,5 con 172 clientes circulando en el pasillo.

6 Conclusiones y Trabajos Futuros

Buscar soluciones para el SQAP no es tarea fácil ya que pertenece a la clase NP-hard y los objetivos planteados se lograron primero es la implementación de una metaheuristica en lenguaje de programación JAVA que permita tener una herramienta flexible y eficiente [15] para la búsqueda de soluciones, segundo diseñar instancias de prueba y ejecutar el algoritmo robusto que ofrezca soluciones en un tiempo razonable de computo ver TCPU en la tabla 1 , así como soluciones de aproximación a los valores óptimos para el SQAP. Adicionalmente visualizamos una aplicación en este caso modelar el tráfico de las personas circulando en un centro comercial con un pasillo central.

Como trabajo futuro es ampliar la experiencia computacional a matrices de mayor dimensión como las matrices de Skorin Kapov [6]que tienen dimensiones de 42 a 100 y diseñar otra metaheuristaica para realizar experimentos de comparación, con la finalidad de acrecentar las herramientas que puedan coadyuvar en la solución de problemas de aplicación prácticos de la clase NP-hard.

Referencias

1. Loiola, E. M., Maia de Abreu, N.M., Boaventura, P.O., Hahn, P., Tania Querido, T.: A survey for the quadratic assignment problem. *European Journal of Operational Research* 176(2), 657-690 (2007).
2. Mishmast, H., Gelareh, S.: A Survey of Meta-Heuristic Solution Methods for the Quadratic Assignment Problem. *Applied Mathematical Sciences* 46(1), 2293 – 2312 (2007).
3. Li, Y., Pardalos, P.M., Resende, M.G.: A Greedy Randomized Adaptative Search Procedure for the Quadratic Assignment Problem. *DIMACS Series on Discrete Mathematics and Theoretical Computer Sience, American Mathematical Society*, Vol 16, 37-261 (1994).
4. Resende, M.G., Pardalos, P.M., Li, Y.: Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problem using GRASP. ACM Transactions on mathematical software, 22(1), 104-118 (1996).
5. Pardalos, P.M., Pitssoulis, L.S., Resende, M.G.: A Parallel GRASP implemetation for the Quadratic Assignment problem. *In A. Ferreira and J. Rolim, editors, Parallel Algorithms for Irregularly Structured Problems, Klower Academic Publishers*, 94, 111-130 (1995).

6. Skorin-Kapov, J.: Tabu Apllied to the Quadratic Assignment Problem. *ORSA Journal on Computing* 2(1), 33- 45 (1990).
7. Li, W.J., Macgregor, J.: Stochastic Quadratic Assignment Problems. *DIMACS Series in Discrete Mathematics and Theoretical Science*, Vol. 16, 221-236 (1993).
8. Li, W.J., Macgregor J.: Quadratic Assignment Problems and M/G/C/C State Dependent Network Flows. *Journal Combinatorial Optimization*, Vol 1, 421-443 (2001).
9. S. Sahni, S., Gonzalez, T.: P-complete aproximations problems. *J. Asssoc. Comp. Machine*. vol. 23, 555-565 (1976).
10. Díaz, A.D., F. Glober, F., Ghaziri, H.M., Gonzalez, J.L., Moscato, P., Tseng, F.T.: *Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería*. Editorial Paraninfo (1996).
11. Roucairol, C.: A parallel branch and baund algorithm for the quadratic assignment problem. *Discrete Applied Mathematics*, vol. 18, pp 211-225, (1987).
12. Koopmans, T.C., Beckmann, M.J.: Assignment problems and the location of economic activities. *Econometrica*, Vol 25: pp 53-76,(1957).
13. Festa, P., Resende, M.G.: GRASP: An annotated Bibliography. *Ensays and Surveys on Metaheuristics. P. Hansen and C.C. Riveiro, eds., Kluwer Academic Publihers*, (2000).
14. Cela, E.: The Quadratic Assignment Problem: Special Cases and Relatives. *Tesis doctoral. Institut für Mathematik B Technische Iniversität Graz. Graz Austria*. (1995).
15. Granillo, E., González, R., Beatriz, M., Martínez J.: A Neighborhood Combining Approach in GRASP's Local Search for Quadratic Assignment Problem Solutions. *Computación y Sistemas,* 22(1), 179–187 (2018)
16. QAPLIB Homepage, *http://anjos.mgi.polymtl.ca/qaplib/,* última visita 2020/08/15.

Algoritmo genético para buscar soluciones aproximadas al Problema de Secuencia de Tareas

J.E. Powell-González[1], R. González-Velázquez[2], E Granillo-Martinez[3], A. López y López[4]
[1,2]Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Prol. 14 sur Esq. Av. Sn. Claudio, C.P 72590, Puebla, Pue., México [1]jairoe.powell@viep.com.mx,[2]rogelio.gzzvzz@gmail.com
[3]Facultad de Administración, Benemérita Universidad Autónoma de Puebla,Av. Sn. Claudio, C.P 72590, Puebla, Pue., México
[3]erika.granillo76@gmail.com
[4] Depto. de Ciencias Básicas, Universidad Autónoma de Tlaxcala, Apartado postal No. 140, Apizaco, Tlax., México
[4]araceli.lopez@uatx.mx

Resumen. El problema de secuencia de tareas es un problema clásico de optimización combinatoria que pertenece a la clase de problemas NP-hard. Ante la necesidad de ofrecer soluciones con tiempos de cómputo que no crezcan exponencialmente al tamaño del problema se diseñó en este artículo un algoritmo genético en lenguaje Phyton. El algoritmo se implementó usando una instancia de prueba de OR-Library, calibrando los parámetros de ejecución para aproximar la solución óptima. Las soluciones obtenidas con el algoritmo se compararon con las soluciones óptimas conocidas y se realizó un análisis del ajuste de parámetros y su efecto en los resultados.

Palabras Clave: Secuencia de Tareas, NP-hard, Algoritmos Genéticos, Metaheurísitcas.

1. Introducción

1.1 Secuencia de tareas

En la industria de manufactura existen problemas en los que se requiere modelar y optimizar el proceso de producción a través de máquinas, con tal de cumplir objetivos como minimizar tiempos totales de manufactura o tiempos de espera entre procesos[1]. De acuerdo a la literatura, existen diversos modelos de secuenciación de tareas debido a la complejidad que existe para modelar matemáticamente los procesos de producción. Un proceso de manufactura se puede modelar como un conjunto de máquinas individuales que realizan tareas dentro de ciertas restricciones, por ejemplo: orden de tareas, tiempos de procesamiento, tiempos de espera, entre otros [19]. La secuencia de tareas es una especificación de los tiempos de proceso de tareas y el problema principal consiste en encontrar secuencias óptimas que cumplan uno o varios objetivos dentro de las restricciones existentes [9]. En esta sección se detallan dos de los principales

modelos en problemas de secuenciación de tareas: *Flow Shop Scheduling Problem (FSSP)* y *Job Shop Scheduling Problem (JSSP)*. La implementación se enfocará en el *FSSP*. El *FSSP*, o Problema de Programación de Flujo de Tareas, modela el proceso de manufactura como un conjunto de N tareas ($i = 1,2, \dots, n$) que son procesadas en un conjunto de M máquinas ($j = 1,2, \dots, m$). Este problema tiene las siguientes restricciones: todas las tareas son procesadas secuencialmente en el mismo orden, cada tarea debe ser procesada en cada máquina, y las máquinas no pueden procesar más de una tarea a la vez [1]. Uno de los objetivos principales en los problemas de secuencia de tareas es minimizar el tiempo de *makespan* o criterio de marcación, que consiste en el tiempo total que toma el proceso completo.

Los problemas de secuenciación tienen complejidad correspondiente a problemas NP-Hard cuando se busca una solución óptima exacta [19], por lo cual se han diseñado métodos heurísticos y metaheuristicos para hallar soluciones aproximadas a la óptima. Las restricciones específicas al *FSSP* son las siguientes:

- Todas las tareas se encuentran disponibles para empezar en la máquina 1 en el tiempo 0.
- Una tarea no puede sobrepasar a otras. Debe permanecer en la misma secuencia.
- Cada tarea se procesa en una sola máquina por intervalo de tiempo.
- Solo existe una máquina de cada tipo y cada una procesa una tarea a la vez.
- Los tiempos de procesamiento de cada tarea *i* en cada máquina *j* están predeterminados.

El *JSSP* tiene mayor complejidad computacional que el *FSSP* debido a un conjunto de factores diferentes. Por ejemplo: requisitos de fecha de vencimiento, restricciones de costos, niveles de producción, capacidad de la máquina en procesos de producción alternativos, características del pedido, características del recurso y disponibilidad, entre otras. Prinicpalmente, es la naturaleza combinatoria del JSSP la que determina su complejidad computacional [13]. Los únicos casos en los que la solución del JSSP se puede obtener de forma exacta son aquellos donde el número de trabajos y de tareas es igual a dos [11][12][16]. Para una instancia de *JSSP* o *FSSP* de N tareas y M máquinas, el problema tiene $(n!)^m$ soluciones posibles.

En la sección 1.2 se describen los algoritmos genéticos, en la sección 1.3 se destacan los trabajos previos al respecto, en la sección 2 se hace la propuesta de los métodos y resultados, y en la última sección se plantean las conclusiones y los trabajos futuros.

1.2 Algoritmos genéticos

Conforme se menciona en Sastry et al [2] los algoritmos genéticos son métodos de búsqueda basados en principios de selección natural y genética. Estos algoritmos se usan en problemas de optimización y computación. Se busca optimizar una función objetivo que representa la calidad de una solución a un problema. Los algoritmos genéticos tienen una "población" de soluciones posibles, a la cual se aplicarán operaciones para evolucionarla hasta encontrar una solución óptima factible. Estas operaciones son: Selección, Recombinación, Mutación y Reemplazo. Los miembros de la población son conocidos como "cromosomas" y consisten en cadenas finitas de tamaño determinado sobre las cuales se aplican las operaciones. En un problema de

combinaciones como *JSSP* y *FSSP* cada cromosoma es una combinación de secuencia de tareas.

La selección consiste en dar preferencia a las soluciones mejores para asignarles un mayor número de copias. La recombinación es un proceso en el que dos cromosomas de soluciones se combinan para formar otras mejores. La mutación modifica localmente una solución de forma aleatoria y el reemplazo tiene como objetivo que las nuevas soluciones obtenidas por los otros procesos reemplacen a las soluciones anteriores. Existen diferentes métodos para llevar a cabo las operaciones sobre la población, que serán específicas al diseño del algoritmo.

1.3 Trabajos previos

Se han utilizado algoritmos metaheurísticos para resolver el problema de Secuencia de Tareas en su forma combinatoria. Se propone en el artículo de Pranzo [3] un algoritmo heurístico *Iterated Greedy*, donde se aplica un proceso de destrucción y construcción de las soluciones y se obtienen resultados que mejoran a los algoritmos de estado del arte en problemas de secuencia de tareas con restricciones de bloqueo. Zhang y Liu, en su trabajo [4] combinan tres algoritmos metaheurísticos: Salto de Rana, Gotas de Agua Inteligentes y Re-enlazamiento de Caminos; obteniendo una tasa de éxito de 100% en 4 *benchmarks*, así como solución a un problema de línea de producción real. Hosseinabadi en su artículo [5] propone un algoritmo genético extendido para resolver el problema de *Open Shop Scheduling*, que tiene un mayor espacio de soluciones que la secuencia de tareas y se considera NP. Demuestra que el algoritmo puede obtener mejores resultados que otros algoritmos propuestos, y que la selección de operaciones influye en la calidad de las soluciones.

2. Métodos y resultados

En el problema de *FSSP* existen conjuntos de datos para probar la efectividad de algoritmos heurísticos y metaheurísticos en aproximar soluciones óptimas. Para la experimentación con el algoritmo de este artículo se usó una matriz de 20 trabajos y 10 máquinas, definida por Taillard [7] cuyo valor óptimo de flujo es 1582 y que se puede observar en la Tabla 1, donde se representa el número de tarea (T) y el número de máquina (M), con los tiempos de procesamiento correspondientes a la máquina y tarea determinada.

Se implementó un algoritmo genético utilizando el lenguaje de programación Python, en el cual cada cromosoma es una solución combinatoria que es candidata para ser la óptima en el problema planteado. Los cromosomas son vectores de 20 números enteros, que indican el orden a seguir en la secuencia de 10 máquinas. La función de *fitness* está dada por el tiempo de flujo, o *makespan,* calculado con la secuencia usada.

Tabla 1. Matriz 1 de 20 máquinas, 10 trabajos de Taillard.

T\M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	74	21	58	4	21	28	58	83	31	61	94	44	97	94	66	6	37	22	99	83

2	28	3	27	61	34	76	64	87	54	98	76	41	70	43	42	79	88	15	49	72
3	89	52	56	13	7	32	32	98	46	60	23	87	7	36	26	85	7	34	36	48
4	60	88	26	58	76	98	29	47	79	26	19	48	95	78	77	90	24	10	85	55
5	54	66	12	57	70	82	99	84	16	41	23	11	68	58	30	5	5	39	58	31
6	92	11	54	97	57	53	65	77	51	36	53	19	54	86	40	56	79	74	24	3
7	9	8	88	72	27	22	50	2	49	82	93	96	43	13	60	11	37	91	84	67
8	4	18	25	28	95	51	84	18	6	90	69	61	57	5	75	4	38	28	4	80
9	25	15	91	49	56	10	62	70	76	99	58	83	84	64	74	14	18	48	96	86
10	15	84	8	30	95	79	9	91	76	26	42	66	70	91	67	3	98	4	71	62

El *makespan* se calcula como la suma de tiempos de proceso dados por la secuencia. Específicamente: una tarea i en la máquina j se procesa en un tiempo $C(i\ ,j)$, correspondiente a su valor en la matriz de tiempos. La primera tarea en la secuencia tomará un tiempo de proceso total igual a: $\sum_{j=1}^{m} C(1,j)$.

Mientras que para las siguientes tareas, su tiempo dependerá de los tiempos de proceso de la tarea anterior en la secuencia. El tiempo de proceso de la tarea i para *makespan* será el máximo de las tareas anteriores; por ejemplo, en esta matriz de Taillard la tarea 1 tarda más en la máquina 3 que la tarea 2 en la máquina 2, si estuvieran seguidas en la secuencia, la máquina 2 terminaría antes y debe esperar a que la tarea 1 termine para avanzar. El *makespan* es equivalente a las sumas de tiempos de proceso tomando en cuenta estos tiempos de espera.

Se utilizaron las operaciones de Cruza, Mutación y Reemplazo sobre los cromosomas. En cada iteración se ordenan los cromosomas por mejor valor de fitness, para asegurar que las mejores soluciones se combinarán entre sí, formando nuevos miembros de población. El algoritmo se observa en el Algoritmo 1.

Como experimento principal, se varió el número de iteraciones y el tamaño de población para observar su efecto sobre la calidad de las soluciones, manteniendo los mismos parámetros de probabilidad de cruza y mutación, y las mismas operaciones de cambio sobre los cromosomas. Se mide el tiempo de ejecución del programa en segundos, y como referencia se usó Python 3.7, sin manejo de librerías específicas, y un equipo de cómputo con un procesador Intel i3 de dos núcleos de 2.27GHz cada uno.

Se experimentó con ejecuciones que tuvieran un tiempo de cálculo menor a 400 segundos, con tamaños de población de 20, 50, 100, 400 y 800 cromosomas; y números de iteraciones de 500, 1000, 2000, 4000 y 8000. Esto con tal de tener combinaciones de parámetros dentro de un rango determinado que permitan realizar comparaciones de los efectos de variación de los parámetros elegidos.

Algoritmo 1. Algoritmo genético implementado

```
AlgoritmoGenetico (Matriz de tiempos)
  Definir tamaño de población.
  Definir número máximo de iteraciones.
  Probabilidad de cruza <- 0.2
  Probabilidad de mutación <-0.4
```

```
  Crear una población aleatoria de cromosomas.
  Mientras no se tenga un número máximo de iteraciones:
    Ordenar la población de cromosomas por mejor valor de
    función fitness.
  Para cada par de cromosomas en la población ordenada:
    Calcular un valor aleatorio de cruza entre 0 y 1.
    Si el valor aleatorio es mayor a probabilidad de cruza:
        Calcular un número aleatorio (índice) dentro del
        tamaño de cromosoma.
        Dividir cada cromosoma en dos partes, a partir del
        índice.
        Intercambiar la segunda parte de los dos cromosomas
        entre ellos, para formar dos "hijos".
  Para cada cromosoma:
    Calcular un valor aleatorio de mutación entre 0 y 1.
      Si el valor aleatorio es mayor a la probabilidad de
      mutación:
      Calcular dos números aleatorios índice.
        Intercambiar los números de tarea dados por estos
        índices, dentro de cada cromosoma.
    Para cada cromosoma:
    Calcular su función de fitness.
      Si su valor de fitness es menor que la mejor solución
      registrada asignar como nueva mejor solución.
Salida: Cromosoma de la mejor solución y valor fitness.
```

Se realizaron 5 ejecuciones con cada combinación de parámetros y se registraron el mejor y peor valor obtenidos por la combinación. En la tabla 2 se observan los resultados de este experimento.

En las figuras 1, 2 y 3, se puede observar los resultados de 3 tamaños de población con sus variaciones de iteración respectivas, representados en forma gráfica.

Se puede observar, tanto en las figuras como en la tabla, que los tamaños menores de población y menores números de iteraciones tienen soluciones más lejanas a la óptima que aquellas con mayor población y más iteraciones. Como ejemplo, manteniendo 500 iteraciones como constante y variando el tamaño de población se obtiene una relación que puede ser vista como lineal entre el tiempo de ejecución, la calidad de la solución y el tamaño de la población.

Por otro lado, manteniendo constante la cantidad de cromosomas en la población y variando el número de iteraciones, se observa igualmente una relación entre el tamaño de población y el tiempo de ejecución. En las figuras 4 y 5 se observan las relaciones, graficadas entre el número de iteraciones y los mejores valores obtenidos, así como el tiempo de ejecución. En estas figuras se representan valores de iteraciones: 500, 1000, 2000 y 4000; y valores de tamaño de población: 20, 50 y 100; con tal de ilustrar la posible existencia de una relación entre las variables mencionadas

Tabla 2 . Resultados de ejecuciones.

Tamaño de Población	Número de Iteraciones	Mejor Valor	Peor Valor	Tiempo de ejecución
20	500	1735	1745	7
	1000	1688	1730	13
	2000	1701	1726	30
	4000	1675	1728	60
	8000	1694	1706	120
50	500	1692	1723	17
	1000	1700	1726	38
	2000	1678	1705	70
	4000	1664	1684	150
	8000	1642	1705	300
100	500	1697	1709	44
	1000	1677	1708	90
	2000	1674	1694	176
	4000	1676	1696	300
200	500	1677	1705	80
	1000	1673	1703	180
	2000	1677	1694	366
400	500	1682	1704	175
	1000	1673	1689	376
800	500	1668	1684	400

Se puede afirmar que para este algoritmo genético aplicado al *FSSP* existe correlación entre el número de iteraciones o el tamaño de población con el tiempo de ejecución y que es lineal. La relación con el mejor valor obtenido es, por otro lado, menos exacta pero igualmente se puede observar linealidad. Esto se debe a la aleatoriedad dentro del método metaheurístico de algoritmo genético. Con estos datos no se puede determinar que, en el tamaño de la población o el número de iteraciones, una variable sea significativamente más dominante que la otra para mejorar las soluciones o el tiempo de ejecución. Para ambas variables (población e iteraciones), las soluciones y el tiempo mejoran con su aumento, de forma lineal.

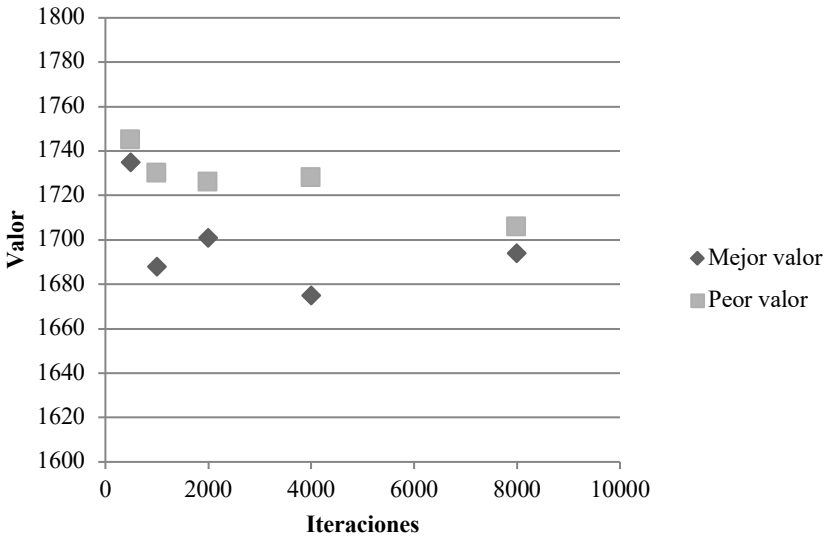


Fig.1. Variación de iteraciones con población de 20 cromosomas.

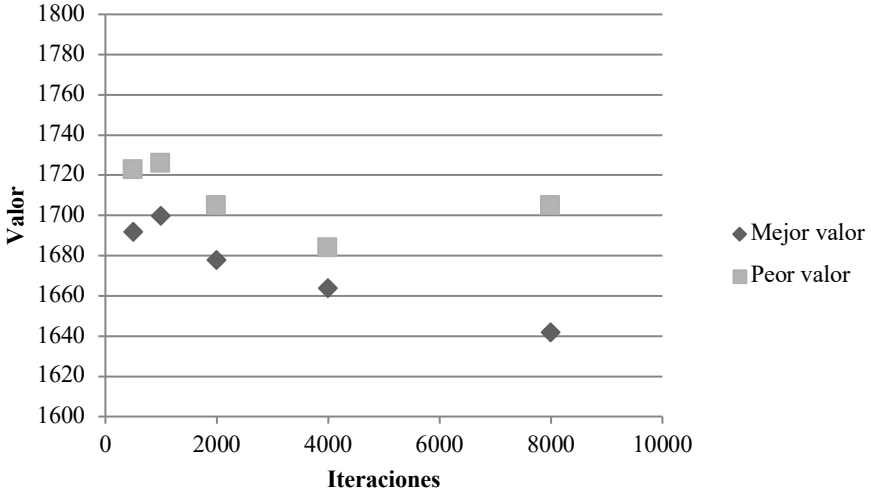


Fig.2. Variación de iteraciones con población de 50 cromosomas.

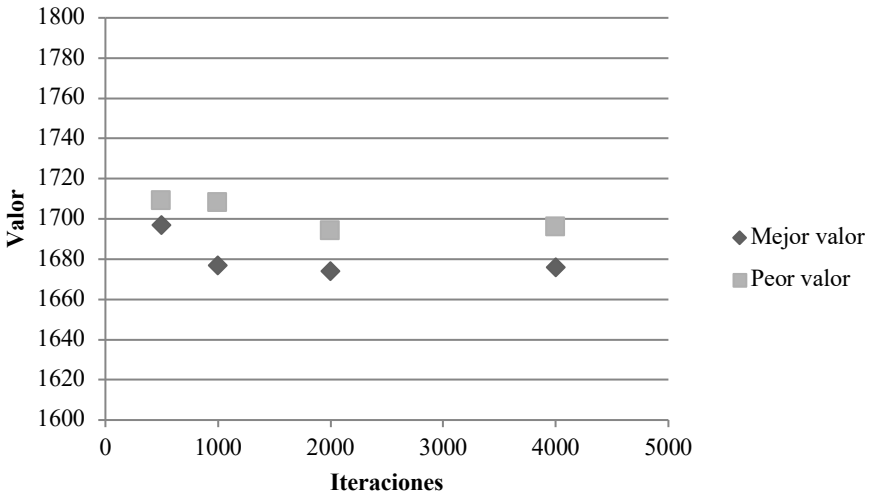


Fig. 3. Variación de iteraciones con población de 100 cromosomas.

.

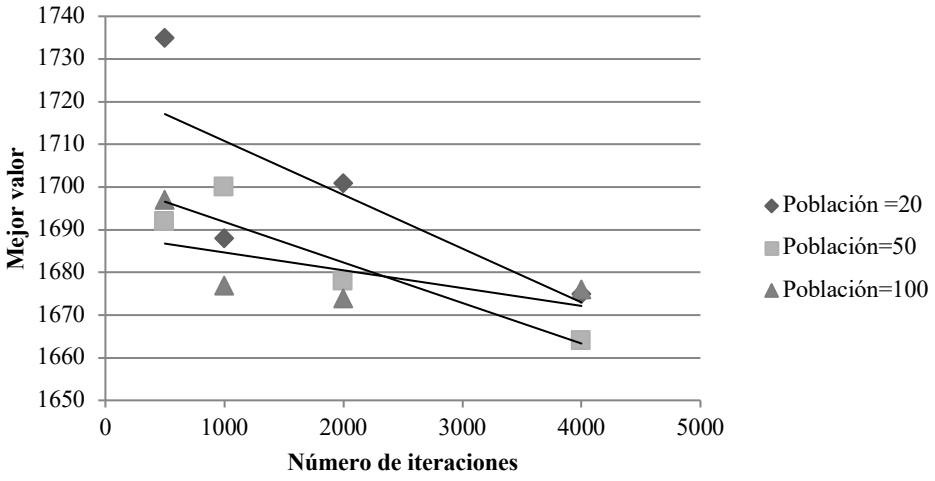


Fig. 4. Relación del mejor valor obtenido con el número de iteraciones.

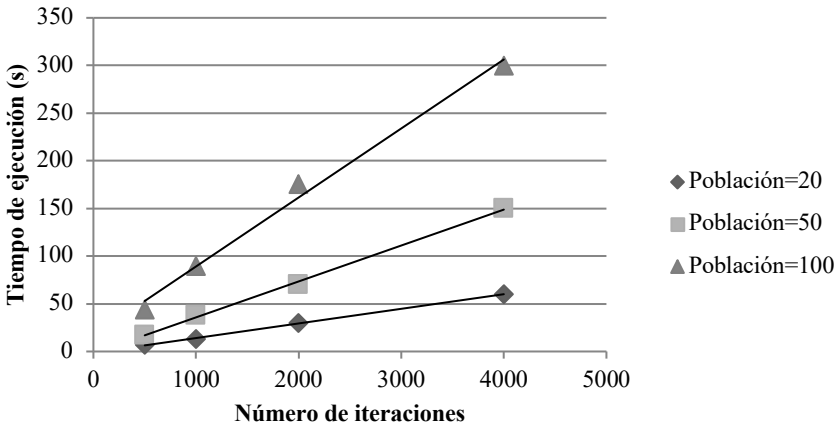


Fig. 5. Relación del tiempo de ejecución con el número de iteraciones.

3. Conclusiones y trabajo futuro

Se implementó un algoritmo genético con parámetros que fueron mencionados, de los cuales se eligieron dos para estudiar su relación con la calidad de las soluciones obtenidas. Se observó el efecto individual de estos parámetros sobre el tiempo de ejecución de programa y la solución obtenida. Se determinó la posible existencia de correlación lineal, dentro de las restricciones de tiempo y tamaño de los parámetros, con los resultados de la metaheurística en un problema *FSSP* específico. El experimento en este artículo estuvo limitado a trabajar con una matriz, dos variables diferentes, y con las restricciones de tiempo y equipo computacional establecidas. En trabajos futuros se podrían variar otros parámetros, como la selección, los métodos de Mutación, Reemplazo y Cruza, y comparar la calidad de las soluciones y el tiempo de ejecución.

También se podrían usar más matrices con una mayor variabilidad y rango en el tamaño y cantidad de variables, para experimentar con las relaciones entre estas variables y los resultados obtenidos. Igualmente existe la posibilidad futura de demostrar matemáticamente las relaciones entre parámetros del algoritmo genético con sus resultados, así como de otros métodos metaheurísticos. Existe la posibilidad de combinar múltiples métodos y comparar su desempeño con los algoritmos individuales.

Además del *FSSP* se puede experimentar con el *JSSP*, el cual tiene un espacio de soluciones mayor debido a menores restricciones de secuencia.

References

1. Arisha, A; et al. Flow shop scheduling problem: a computational study. Sixth International Conference on Production Engineering and Design for Development (PEDD6),Cairo, Egypt, pp 543 – 557.(2002)
2. Sastry, K., Goldberg, D., & Kendall, G. Genetic algorithms. In *Search methodologies* (pp. 97-125). Springer, Boston, MA. (2005).

3. Pranzo, M., & Pacciarelli, D. (2016). An iterated greedy metaheuristic for the blocking job shop scheduling problem. *Journal of Heuristics*, *22*(4), 587-611.
4. Zhang, H., Liu, S., Moraca, S., & Ojstersek, R. (2017). An effective use of hybrid metaheuristics algorithm for job shop scheduling problem. *International Journal of Simulation Modelling*, *16*(4), 644-657.
5. Hosseinabadi, A. A. R., Vahidi, J., Saemi, B., Sangaiah, A. K., & Elhoseny, M. (2019). Extended genetic algorithm for solving open-shop scheduling problem. *Soft computing*, *23*(13), 5099-5116.
6. Kumar, S; Pooja, J. A Novel Hybrid Algorithm for Permutation Flow Shop Scheduling. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4), 5057-5061. (2014)
7. Taillard, E. (1993). Benchmarks for basic scheduling problems. *european journal of operational research*, *64*(2), 278-285.
8. Parveen, S; Hafiz, U. Review on job-shop and flow-shop scheduling using multi criteria decision making. Journal of Mechanical Engineering, Vol. ME 41, No. 2. (2010).
9. Emmons, H. Vairaktarakis, G. Flow Shop Scheduling. Theoretical results, algorithms and applications. Springer. New York. (2013)
10. Blazewicz, J., Domschke, W., Pesch, E. (1996) The job-shop scheduling problem: Conventional and new solution techniques. Eur J Oper Res 93:1–33
11. Brucker, P. (1988) A polynomial algorithm for the two-machine job-shop scheduling problem with a fixed number of jobs. OR Spektrum 16:5–7
12. Brucker, P. (1994) An efficient algorithm for the job-shop problem with two jobs. Computing 40:353–359
13. Charalambous, O. (1991) Knowledge based job-shop scheduling. PhD thesis, University of Manchester Institute of Science and Technology, ManchesterConway, R.W., Maxwell, W.L., Miller, L.W. (1967) Theory of scheduling, Addison-Wesley
14. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. (1979) Optimization and approximation in deterministic sequencing and scheduling: asurvey. Ann Discrete Math 5:287–326
15. Hefetz, N., Adiri, I. (1982) An efficient optimal algorithm for the two-machine,unit-time, jobshop, schedule-length, problem. Math Oper Res 7:354–360
16. Jackson, J.R. (1956) An extension of Johnson's result on job lot scheduling.Int J Prod Res 36:1249–1272
17 Lenstra, J.K., Rinnooy Kan, A.H.G. (1979) Computational complexity of discrete optimization problems. Ann Discrete Math 4:121–140
18. Fatos, X., Ajith, A. (2008) Metaheuristics for Scheduling in Industrial and Manufacturing Application. Studies in computational Intelligence. Vol 128. Springer. Berlin

Uso de un Sistema Embebido Programable en Chip PSoC para Aplicaciones de Domótica

Gerardo Sánchez-Alba, Felipe J. Torres, Carlos Ignacio Gómez, Gustavo Capilla, Edson González y Brayan Martínez
Universidad de Guanajuato, División de Ingenierías Campus Irapuato-Salamanca, Carretera Federal Salamanca-Valle de Santiago km. 3.5+1.8, 36885, Salamanca, Guanajuato, México.
fdj.torres@ugto.mx

Resumen. Este trabajo presenta el desarrollo de aplicaciones de domótica que permiten la automatización y control de dispositivos eléctricos de una casa-habitación, por medio de la codificación de un sistema programable en chip, conocido como PSoC, la cual es una tarjeta utilizada como un sistema embebido que está obteniendo gran relevancia en el desarrollo e investigación de sistemas mecatrónicos. En este proyecto se probó la comunicación serial asíncrona transmisor-receptor (UART) que la tarjeta puede soportar para manipular los puertos de entrada/salida (E/S) para activar el módulo de relevadores, encargados de controlar el encendido y apagado de los elementos conectados a la red eléctrica doméstica. Los resultados de las implementaciones realizadas, muestran la capacidad y alcance de la PSoC para llevar a cabo aplicaciones de domótica de forma asequible y segura en un entorno digital.

Palabras clave: Domótica, PSoC, sistema embebido, comunicación UART.

1. Introducción.

La automatización en las edificaciones como casas habitación, ha sido un tema de desarrollo tecnológico e investigación que ha tenido gran interés desde hace varias décadas, actualmente es conocido bajo el término de domótica, por sus raíces francesas de la palabra “domotique” aparecida a finales de los 90’s.

Paralelamente al avance de la tecnología, las aplicaciones de domótica han sido implementadas por medio de distintos componentes, desde aquellos sistemas de relativa simpleza como un temporizador o sensor analógico de luz para llevar a cabo el encendido y apagado de luces de manera autónoma, hasta aquellos sistemas que conjuntan elementos electrónicos, de control, actuadores mecánicos y eléctricos, y un equipo programable para ejecutar las acciones que derivan en un sistema mecatrónico aplicado en la automatización de una casa.

El uso de microcontroladores ha traído consigo una amplia variedad de implementaciones, sin embargo, las limitaciones propias del dispositivo generan una incesante búsqueda por replicar las acciones de control y automatización a través de otros dispositivos que están siendo objeto de interés mundial por su alta capacidad de integración de componentes y una programación libre, aunado a la tendencia en la incorporación de un sistema embebido programable en chip, PSoC. Así, por ejemplo,

en [1] se diseña una fuente de corriente con una mayor protección al ruido mediante el uso del convertidor de corriente analógico-digital (IDAC), incluido en el chip PSoC 3. Además, de un transistor de paso y una resistencia de ajuste de corriente. En [2] se desarrolla un sistema de control de nivel de agua por medio de la apertura de una compuerta a través del amplificador de ganancia programable (PGA), un comparador de voltaje y convertidor analógico digital (ADC), todos ellos componentes de la tarjeta PSoC. Otra aplicación usando los recursos PGA y ADC de la PSoC se detalla en [3], para realizar un arreglo de resistencias de detección de fuerza para la prevención de colisiones en robots. En [4] se expone el trabajo realizado de un controlador de retroalimentación de estado aplicado a un cuadricóptero, en donde la comunicación de baja energía por bluetooth (BLE) del joystick de la PSoC 6, es usada como control remoto. En [5] se presenta un dispositivo portátil de medición electroquímica para analizar sustancias líquidas que puedan ser clasificadas por sus propiedades gustativas, denominada lengua electrónica e implementada con tecnología PSoC. En [6] se utiliza la tarjeta PSoC 5LP para desarrollar un generador de pulsos. En [7] se construye un potenciostato para realizar pruebas de voltametría cíclica y amperometría basado en una PSoC; sin requerir componentes externos. En [8] se presenta una variedad de proyectos de laboratorio tomando como base la PSoC 5LP, con resultados favorables.

El objetivo de este trabajo consiste en el desarrollo de la implementación de aplicaciones de domótica a través del uso de un sistema embebido programable en chip, PSoC, para mostrar la capacidad y alcance que ésta tarjeta puede tener en la automatización de un hogar. El uso de la tarjeta PSoC presenta ventajas de ser asequible y de programación abierta basada en bloques, permitiendo una integración de diversos componentes incluidos en la propia tarjeta, de acuerdo a las funciones y características de las tareas a ejecutar.

El resto del documento está organizado de la siguiente manera: en la sección 2 se presenta la descripción de los componentes de la automatización, las características de la tarjeta PSoC y el módulo de relevadores son mencionados; en la sección 3 se detalla la metodología usada; posteriormente, en la sección 4 y 5 se exponen las aplicaciones de domóticas logradas a través del análisis de los resultados obtenidos y, por último, en la sección 6 se dan las conclusiones del trabajo realizado.

2. Dispositivos para la automatización.

2.1 Descripción y características de la PSoC 6.

PSoC, por sus siglas en inglés, es un sistema programable en chip, esto quiere decir que se trata de un dispositivo que contiene diversos componentes como ADC, PGA, IDAC, etc., por tal motivo también es conocido como un sistema embebido. Estos componentes tienen la característica de ser configurables, de acuerdo con las necesidades del programador. Por ejemplo, si se requiere mayor ganancia al amplificador, basta con realizar la configuración en el bloque de programación del PGA para lograr una mayor amplificación; haciéndolo de mayor utilidad que aquellos dispositivos de único propósito a los cuales no se les pueden configurar sus funciones esenciales.

En el portal de internet de Cypress, corporación que ha lanzado al mercado la tecnología PSoC, la describe como la tarjeta que cierra la brecha entre los procesadores de aplicaciones caros y con alto consumo de energía y los microcontroladores de bajo rendimiento (MCU). Actualmente se está ofertando al público la tarjeta de evaluación CY8CPROTO-062-4343W, mostrada en la Fig. 1, con arquitectura MCU PSoC 6 de potencia ultra baja y ofrece el rendimiento de procesamiento que necesitan los dispositivos del Internet de las Cosas (IoT por sus siglas en inglés), eliminando las compensaciones entre potencia y rendimiento. La MCU PSoC 6 contiene una arquitectura de doble CPU, con ambos CPU en un solo chip. Tiene un Arm® Cortex®-M4 para tareas de alto rendimiento y un Arm® Cortex®-M0+ para tareas de baja potencia [9].

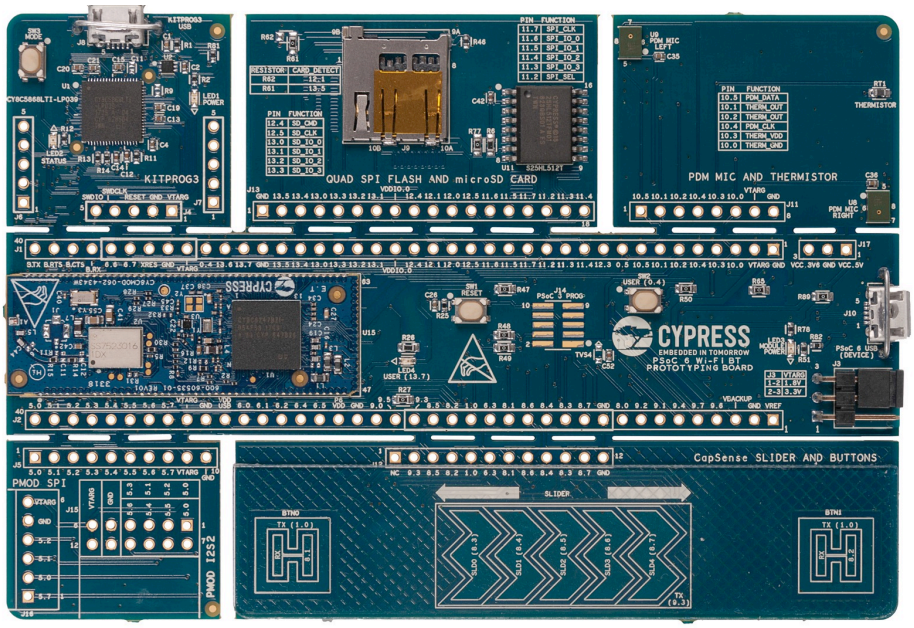


Fig. 5. Tarjeta de evaluación CY8CPROTO-062-4343W que contiene a la PSoC 6.

La tarjeta CY8CPROTO-062-4343W contiene las características siguientes:

- Procesador primario de 150 MHz y procesador secundario de 100 MHz.
- 2 MB de memoria flash y 1 GB de SRAM.
- Módulo para conexión Wi-Fi y Bluetooth.
- Bajo consumo de potencia, operando de 1.7 V a 3.6 V CD.
- Opciones flexibles de reloj con osciladores de cristal variables.
- 9 bloques de comunicación serial configurables: SPI, I2C, UARTs.
- Interface para USB de alta velocidad.
- 32 bloques (24 de 16 bits y 8 de 32 bits) de contadores/temporizadores/ moduladores de ancho de pulso PWM.

- Convertidor analógico digital (ADC) y digital analógica (DAC) de 12 bits con 16 canales.
- Más de 100 pines de entrada/salida de propósito general (GPIO).
- Un módulo de sensado capacitivo.
- 12 bloques lógicos programables, 2 amplificadores operacionales, 2 comparadores de baja potencia.

2.2 Módulo de relevadores

La mayoría de los aparatos eléctricos de una casa habitación, así como las luces de la misma, son alimentados por 110 V de CA. La PSoC, de acuerdo con sus niveles de voltaje de máximo 3.6 V CD, no puede manipularlos directamente. Por tal razón, se hace uso de un módulo de relevadores de 4 canales como lo muestra la Fig. 2, en donde la PSoC envía un bit al canal del relevador correspondiente para activarlo, haciendo que sus contactos normalmente abierto y normalmente cerrado conmuten para controlar el dispositivo eléctrico conectado en esas terminales.

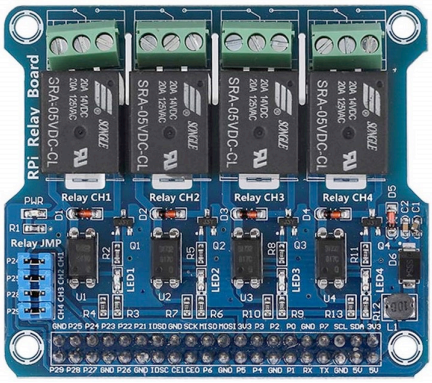


Fig. 6. Módulo de relevadores de 4 canales.

3 Metodología

En la literatura disponible no existe basta información respecto a la programación de la PSoC, particularmente la PSoC 6 utilizada en este trabajo, la cual es un modelo reciente por lo que se inició con la investigación y puesta en marcha de las acciones que llevaron a las aplicaciones de domótica como objeto de estudio. Así, se atendieron las necesidades de configuración de los componentes, a través de los bloques, para ejecutar las tareas de automatización.

La idea general del código de programa es controlar múltiples relevadores. Esto se realiza presionando distintas teclas del teclado de una computadora conectada por un cable USB a la tarjeta PSoC 6. La tarjeta PSoC se encuentra en modo “SLEEP”, de bajo consumo energético, y al detectar la activación de alguna de las teclas, activa o

desactiva una de las salidas GPIO dispuestas para controlar los relevadores (en el programa implementado, se controlan dos de los cuatro relevadores del módulo, pero es sencillo modificar el mismo para controlar más relevadores). El modo de conexión es mostrado en la Fig. 3.

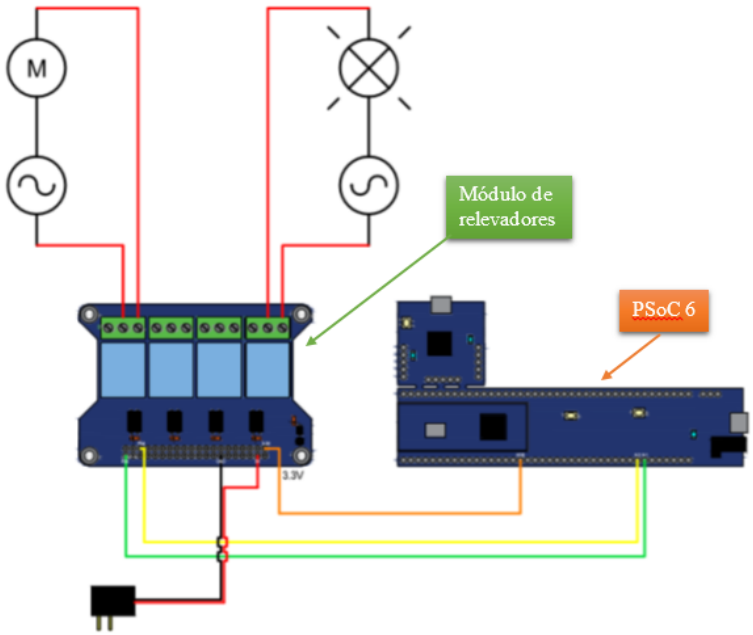


Fig. 7. Conexión de la PSoC con el módulo de relevadores.

En la Fig. 4 se presenta el diagrama de flujo para exponer la metodología general utilizada, es importante notar que se emplearon diversas estrategias para activar el módulo de relevadores a partir de interrupciones que la programación de la PSoC 6 debía atender de manera inmediata. Sin embargo, se observó que el tiempo y costo computacional no satisfacían los criterios de un consumo bajo de potencia por mantenerse en un ciclo hasta que existiera la interrupción en el código. De esta manera, se abordó la problemática desde la opción de la comunicación serial por medio de los bloques UART (Transmisor-Receptor Asíncrono Universal).

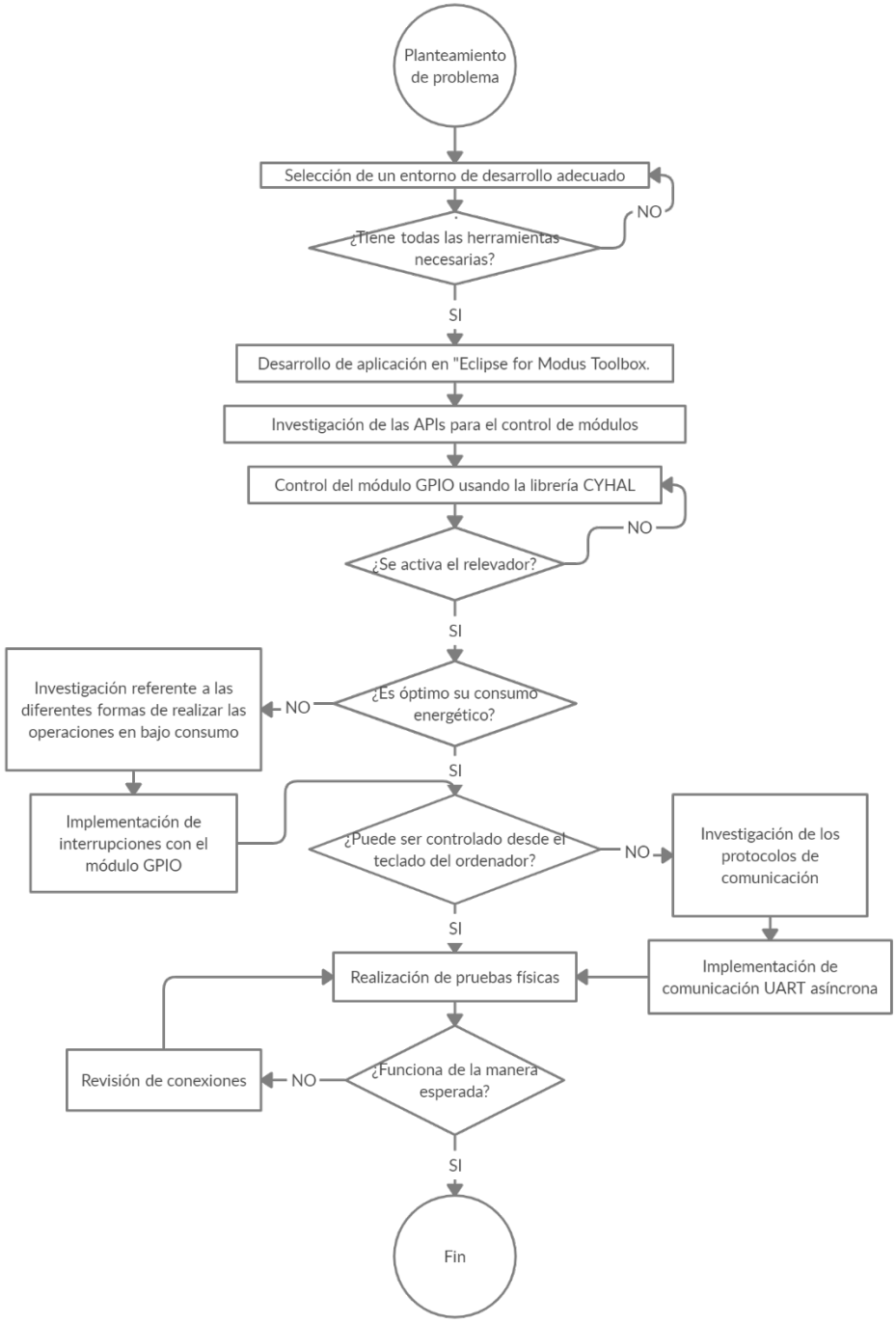


Fig. 8. Diagrama de flujo del trabajo realizado con la configuración de la PSoC6.

4 Resultados

La PSoC 6 está conectada por USB a la computadora como se aprecia en la Fig. 5. Esto permite ser alimentada con el voltaje necesario para su operación, además de transferir datos por medio de la conexión UART con un emulador de terminal (en nuestro caso, el Tera Term). Así, desde la computadora se ingresan los caracteres que el código espera recibir para el control de las salidas GPIO.

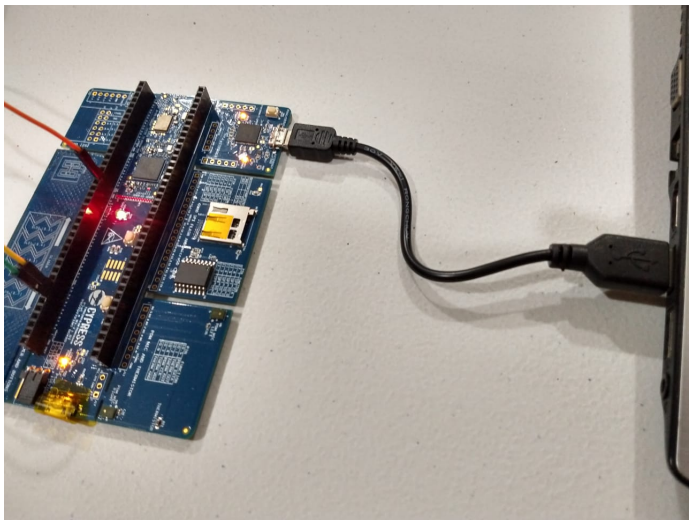


Fig. 9. Conexión microUSB de la PSoC 6 a USB de la computadora.

De la PSoC 6, se utilizaron los pines P9_1 y P9_2 para enviar la señal que recibirán los pines P29 y P24 del módulo relevador, como se muestra en la Fig. 6. Así, es posible controlar la activación o desactivación de las bobinas de los relevadores, permitiendo o cortando el flujo de corriente en el circuito de potencia. Es importante resaltar, el requerimiento de conectar el puerto Vdd de la PSoC 6 al pin 3V3 del módulo relevador, y debido a que la PSoC 6 no puede otorgar los 5 V CD que necesita el módulo relevador para activar las bobinas, se usa una fuente externa que suministre ese voltaje.

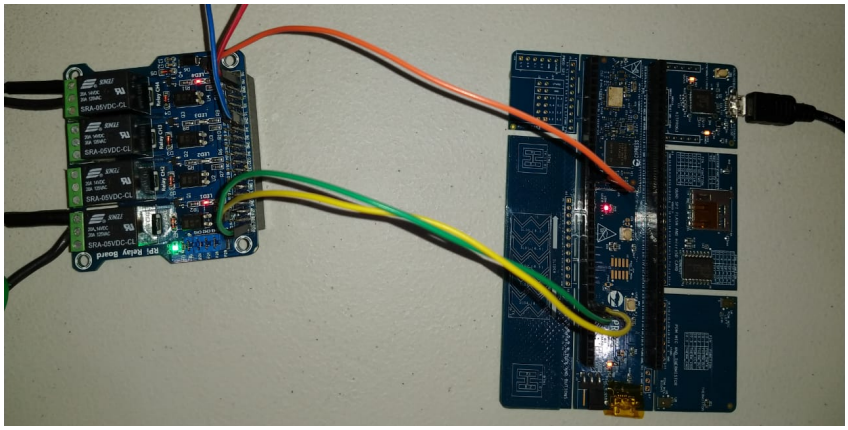


Fig. 10. Conexión física de la PSoC 6 con el módulo de relevadores.

Por último, el circuito de potencia se dispone interrumpiendo la línea de fase (110 V CA), conectándose a la terminal común del relevador y el otro extremo a la terminal del contacto normalmente abierto de un canal del módulo relevador, siguiendo el diagrama de la Fig. 3.

5 Pruebas

Se realizaron pruebas de funcionamiento utilizando 2 canales del módulo de relevadores para encender un foco de 60 W y un ventilador de pedestal, como se observa en la Fig. 7 y Fig. 8, respectivamente. El encendido y apagado del foco se hace por medio de oprimir las teclas “A” y “S”, respectivamente. El ventilador de pedestal obedece a las teclas “D” y “F”.

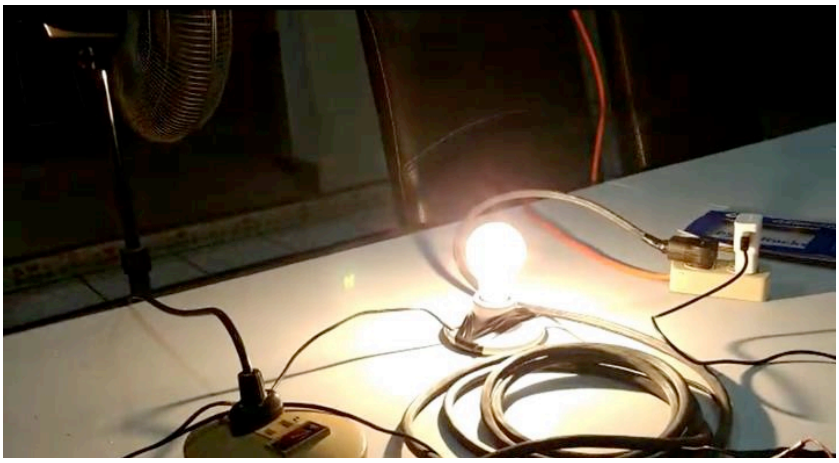


Fig. 11. Encendido de foco de 60 W a través de la activación realizada por medio de la PSoC 6.

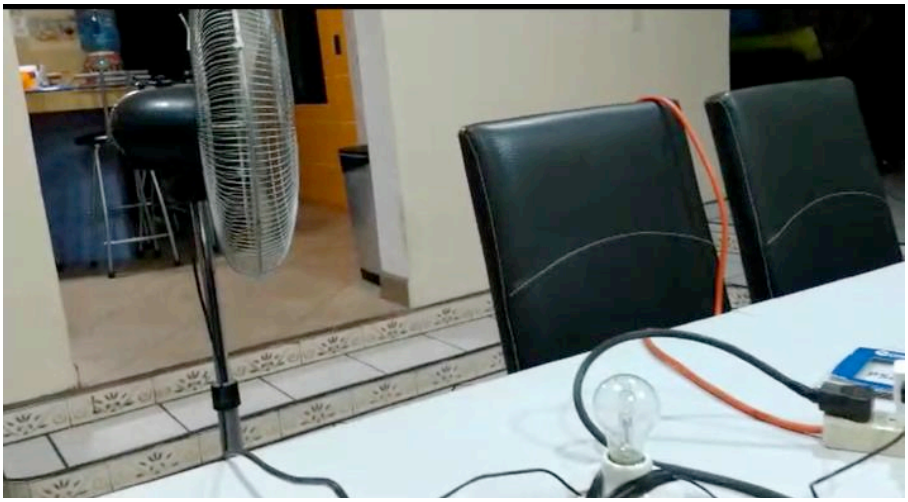


Fig. 12. Encendido de ventilador de pedestal a través de la activación realizada por medio de la PSoC 6.

Los videos de las pruebas de funcionamiento pueden ser consultados en la siguiente liga: https://1drv.ms/u/s!AlM3cRth52xajUay3F__4ek7VPw5?e=8BOiEJ

6 Conclusiones

Los sistemas de domótica dirigidos a la automatización del hogar, los cuales requieren de la manipulación de distintos niveles de voltaje, pueden ser implementados a través de un sistema embebido programable en chip, PSoC. Notar que las implementaciones realizadas ya han sido llevadas a cabo a través de microcontroladores de 8 bits con resultados probados desde hace varios años, sin embargo, la tarjeta PSoC 6 posee características que permiten optimizar los recursos de bajo consumo de energía (es posible operar ésta tarjeta con un voltaje de 1.7 V de CD). Por tanto, ha sido mostrado que la PSoC 6 puede llevar a cabo tareas de automatización a través de la activación de relevadores que controlen aparatos eléctricos de una casa habitación.
En la configuración de la PSoC 6 fue necesario el uso del bloque de comunicación serial UART para comunicarse desde una computadora con la PSoC 6 y así lograr el buen funcionamiento del sistema domótico. El alcance de la integración de otros componentes de la PSoC 6 en este proyecto, será tratado en los trabajos futuros para incorporar la comunicación WiFi.

Referencias

1 M. Mohan, N. Seenu and M. Sarfudeen, “Low Cost, Wider Range On-Chip Programmable Current Source For Robotic Applications,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 7, pp. 73-76, 2018.

2 Y. Liu, W. Wang, B. Liu, T. Huang, Z. Wang, J. Bao y Y. Bi, “Water Level Control System Based on PSoC,” *DEStech Transactions on Computer Science and Engineering*, 2016.

3 J. Castellanos, A. Trujillo, R. Navas, F. Barbero, J. Sánchez, O. Oballe y F. Vidal, “Adding Proximity Sensing Capability to Tactile Array Based on Off-the-Shelf FSR and PSoC,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4238-4250, 2020.

4 B. Alakananda y N. Venugopal, "Development of a Programmable System on Chip (PSoC) based Quadcopter", *Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020),* pp. 93-98, 2020.

5 A. Arrieta y O. Fuentes, " Electronic Tongue: New Tool for Food Analysis Based a PSoC (Programmable System-on-Chip) Technology," *Advance Journal of Food Science and Technology,* vol. 12, no. 11, pp. 603-608, 2016.

6 P. Acevedo, C. Díaz, M. Vázquez y J. Durán, " Design of a Pulse Generator Based on a Programmable System-on-Chip (PSoC) for Ultrasonic Applications," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering,* vol. 10, no. 3, pp. 453-456, 2016.

7 P. Lopin y K. Lopin, “PSoC-Stat: A single chip open source potentiostat based on a Programmable System on a Chip”, *PLoSONE,* vol. 13, no. 7, 2018.

8 S. Strom y D. Loker, “Programmable System-On-Chip (PSoC) Usage in an Engineering Technology Program” In *ASEE Annual Conference and Exposition, Conference Proceedings*, 2016.

9 Cypress, “32-bit Arm® Cortex®-M4 Cortex-M0+ PSoC® 6”, https://www.cypress.com/products/32-bit-arm-cortex-m4-cortex-m0-psoc-6, [en línea], consultado el 01 de agosto de 2020.

Metodología para el Proceso de Recuperación de la Plataforma Moodle Actualizando a Versiones Superiores

Nahum Pérez Rodríguez, Ana Claudia Zenteno Vázquez, María del Carmen Santiago Díaz, Gustavo Rubín Linares, José Luis Pérez Rendón, Antonio Álvarez Núñez
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 Sur, Ciudad. Universitaria, 72592 Puebla, Pue.
nahum.per@hotmail.com, {ana.zenteno, marycarmen.santiago, gustavo.rubin, jose.perezren }@correo.buap.mx, antonio.alvarez@alumno.buap.mx

Resumen. La educación semipresencial o a distancia requiere el uso de plataformas educativas que pongan a disposición del alumno materiales y que sirvan como medio de comunicación con el docente por medio de herramientas como foros. Moodle es una plataforma sencilla, potente, ecológica y económica que ha extendido su uso en los centros de enseñanza a nivel mundial También es ideal para gestionar la organización de las comunidades educativas y permitir la comunicación y el trabajo en red entre sus distintos integrantes y con otros centros, y que ha incrementado su uso a raíz de la pandemia por covid-19. Para su correcto uso requiere configuraciones y actualizaciones entre versiones, principalmente en gestores de bases de datos en conjunto al sistema operativo que lo hospeda. Se busca garantizar la disponibilidad del servicio y de los contenidos en la plataforma y para ello se propone una metodología de recuperación, actualización y mantenimiento de Moodle.

Palabras Clave: Configuración, Actualización, Bases de datos, Moodle.

1 Introducción

Hoy en día las plataformas web educativas son herramientas muy útiles que apoyan los procesos educativos. Proporcionan un espacio virtual para el aprendizaje y en los últimos años, se han convertido en la herramienta que facilita y dinamiza la formación a distancia.

La educación a distancia es una forma de enseñanza que no requiere la presencia de los alumnos en los centros educativos. El alumno recibe material de estudio por correo electrónico y otras posibilidades que ofrece internet permitiendo que sea autodidacta y se fomente la autogestión en el proceso de aprendizaje. Se tiene entonces un proceso educativo que es flexible apoyado en las tecnologías y plataformas educativas.

La pandemia por covid-19 ha traído consigo la necesidad de implementar herramientas que contribuyan a la educación más allá de las aulas y los apuntes. Si bien el internet ha contribuido a manejar grandes y diversas fuentes de información, hace falta una herramienta que proporcione la capacidad de organización y gestión de aulas, pero ahora virtuales. También que permita la distribución de materiales de estudio y carga de actividades del estudiante de tal forma que el aprovechamiento académico no se vea afectado.

La educación a distancia desaparece la limitación geográfica (espacio), de movimiento y contribuye a generar interés y más participación del alumnado potencial al abarcar nuevos ámbitos geográficos. Empresas y profesionales adoptan esta modalidad debido a la escasez de tiempo y la falta de flexibilidad horaria de las clases presenciales. Las actualizaciones, diplomados, maestrías, entre otros tienen una importante demanda y solventan las necesidades de facilitar la comunicación y transmitir información, propias de la educación continua.

Para lograr educación a distancia se crean espacios virtuales de aprendizaje (EVA) que permiten dar un seguimiento a los estudiantes, habilitan canales de comunicación para asesorías y/o tutorías, habilitan la colaboración en trabajos en línea y de esta forma se recupera en gran medida la educación tradicional. También, como características tenemos el acceso a la información y contenidos de autoaprendizaje, uso de simuladores e incluso la adaptación de evaluaciones en línea [1]. Los entornos de aprendizaje virtuales (EVA) se han convertido en una Tecnología Educativa que ofrece oportunidades los centros educativos en todo el mundo. Es un programa interactivo de carácter pedagógico que posee una capacidad de comunicación y colaboración integrada.

Actualmente, las opciones de plataformas educativas disponibles para crear e implementar cursos son muchas, y es importante analizar algunos factores antes de escoger una. Las plataformas comerciales solicitan un pago que va en función de las características que se habilitan para los usuarios. Las plataformas de Software libre son conocidas también como plataformas de código abierto, y han sido diseñadas para ser distribuidas y usadas sin costo. Forman parte del dominio público. Y existen otras cuyo uso es a través de la nube, pero también se exige un pago en función de los requisitos de los cursos.

Un LMS ((Learning Management Systems)) se puede definir como software que permiten la creación y gestión de entornos de aprendizaje en línea de manera sencilla y automatizada, pudiendo ser combinados o no con el aprendizaje presencial. Algunos de los ejemplos más significativos a nivel mundial son Moodle, Blackboard, Chamilo, EvolCampus, Canvas, entre otros. Permiten que los usuarios interactúen, se comuniquen y realicen trabajos colaborativos a través de herramientas síncronas o asíncronas. Las primeras posibilitan ofrecen comunicación instantánea, simulando un aula presencial como por ejemplo los chats o las videoconferencias. Por otra parte, las asíncronas establecen una interacción diferida en el tiempo, como por ejemplo los foros de discusión, blogs o el correo electrónico [4].

Moodle ha sido ampliamente adoptado debido a su flexibilidad, derivada de su estructura modular que garantiza dar soporte a cualquier estilo docente. Además, permite la creación de espacios destinados a la enseñanza que entornos virtuales de aprendizaje (EVA) o entornos virtuales de enseñanza aprendizaje (EVEA) [5].

La plataforma Moodle fue desarrollada por Martin Dougiamas, basada en el Constructivismo social de Vigotsky, bajo el acrónimo de "Modular Object-Oriented Dynamic Learning Environment" (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular). Tres son los grandes recursos de moodle: gestión de Contenidos, comunicación y evaluación. Al utilizar la plataforma Moodle como herramienta para el aprendizaje, los usuario cuentan con un nuevo espacio, en donde tienen a su disposición actividades innovadoras de carácter colaborativo y con aspectos creativos que les permiten reforzar lo que aprenden, además de construir su conocimiento en conjunto

con el docente en su rol de guía o facilitador El usuario tiene la oportunidad de explorar el ambiente, con la seguridad de contar con los medios suficientes y necesarios para salir de dudas cuando existan, pero sobre todo tenido la oportunidad de leer, reflexionar, investigar y trabajar a sus tiempos [2].

2 Metodología

Sin importar la plataforma elegida, se requiere que la instalación de la plataforma sea estable ya que es crucial para mantener la disponibilidad y acceso a la plataforma.

Por lo que contar con una serie de pasos que garanticen que en todo momento se brinda el servicio, nos permite definir una metodología.

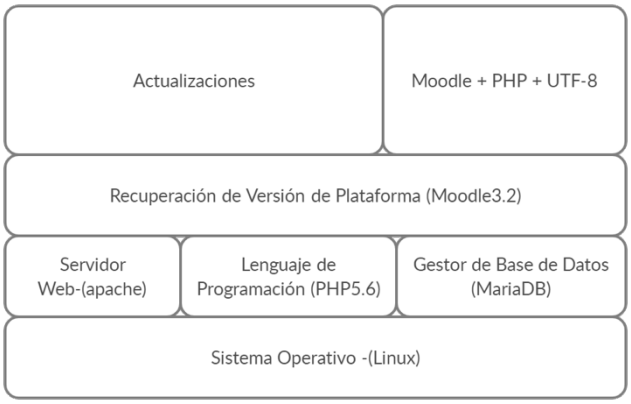


Fig 1. Metodología de Recuperación y Actualización de Plataforma Moodle

En la fig. 1 se observa la relación de sistemas y herramientas necesarias para actualizar las versiones de la plataforma Moodle.

Esta prueba tiene como propósito recuperar una versión de respaldo de Moodle para instalarlo en un servidor nuevo y realizar las actualizaciones necesarias para dejarlo operativo, en línea y con cursos y contenidos disponibles para los usuarios.

2.1 Servicio web

El servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. En la arquitectura de servicios web existen tres partes: proveedor de servicios web, el que pide el servicio web y el publicador. Dentro de las ventajas que provee es que aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas

sobre las que se instalen. También fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. Además permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
Apache es uno de los servicios que funciona para distintas plataformas, implementa el protocolo HTTP [6]. Una vez que está instalado el servicio es posible iniciar, detener, recargar, activar, desactivar o ver el estado de ejecución en el servidor, ya que Moodle es accesible por medio de un navegador.

2.2 Lenguaje de Programación

Cada plataforma requiere de un lenguaje de programación que le permite ejecutar plugins con propósitos específicos como realizar encuestas, adecuaciones en editores, entre otras y con ellos brindar nuevas experiencias de aprendizaje.
PHP (Hypertext Preprocessor) es un lenguaje de programación web simple [2] el cual requiere Moodle para su correcto funcionamiento. Se recomienda la actualización del lenguaje en el servidor para una correcta visualización de los elementos de Moodle.

2.3 Gestor de Base de datos

Hoy en día, los sistemas de gestión de base de datos son necesarios y muy importantes en la creación y gestión de los datos de una organización. Almacenan la información de manera organizada y permiten acceder a la información de forma ágil. Los gestores de base de datos o gestores de datos hacen posible administrar todo acceso a la base de datos ya que tienen el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones. Los gestores de bases de datos que se utilizan típicamente para Moodle son MySQL/MariaDB, siendo este último derivado de MySQL [3], el cual manipula los datos de los usuarios y de los contenidos bajo las reglas de tablas de datos y códigos.

2.4 Moodle

Es importante poner especial atención a parámetros de configuración importantes, entre ellos el idioma, directorio web, Gestor de base de datos, así como: *usuario y contraseña, nombre de la base de datos, nombre del sitio, una palabra clave del sitio*, los que no hayan sido mencionados deben dejarse por defecto. Los permisos sobre las carpetas añaden consideraciones de seguridad a la información, por lo que también se deben atender en base a las recomendaciones de privilegios para el usuario *root*.

2.4.1 Consideraciones de codificación

La codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural (como el de un alfabeto o silabario) en un símbolo de otro sistema de representación aplicando normas o reglas de codificación.

El formato de las tablas de datos de Moodle debe ser compatible con las versiones posteriores para asegurar la conversión de caracteres y representación de la información [7]. Convertir tablas InnoDB a Barracuda permite optimizar y mejorar el rendimiento y optimización web además de contribuir a la compresión de datos.

Debemos recordar que InnoDB es un mecanismo de almacenamiento de datos de código abierto para la base de datos MySQL, incluido como formato de tabla estándar en todas las distribuciones de MySQL AB a partir de las versiones 4.0. Su característica principal es que soporta transacciones de tipo ACID y bloqueo de registros e integridad referencial. InnoDB ofrece una fiabilidad y consistencia muy superior a MyISAM, la anterior tecnología de tablas de MySQL.

2.4.2 Actualizaciones Moodle

Para cualquier proceso de actualización se recomienda recuperar las configuraciones de la plataforma en la versión anterior. La relevancia de actualizar el lenguaje de programación recae en la mejora de rendimiento tanto en los cambios de estándares de idioma y en su mayoría en la interpretación del código [8].

Es de vital importancia realizar este proceso para cada una de las versiones además de resaltar que importante notar que las actualizaciones deben hacerse una a una (sin saltarse alguna versión) ya que cada una de ellas presenta nuevas características y funciones que puede generar errores al cargar si no se lleva acabo de esta forma el proceso.

3 Resultados

El mantenimiento, actualización de plataformas, migración de bases de datos requieren de una metodología para completar con éxito la tarea. En el caso plataformas educativas con Moodle, es indispensable ejecutar una a una las actualizaciones de versiones, debido a que cada una contiene características que pueden modificar incluso la estructura de los directorios y la forma de tratar la información. Otra consideración importante son las bases de datos, debido a los formatos de representación y el tratamiento de caracteres especiales.

Dentro del proceso de mantenimiento y/o actualización es importante poner el sitio en mantenimiento para detener a usuarios no administradores, evitando el ingreso y posibles cambios a los contenidos de las plataformas.

Además, es importante poner especial atención que los gestores de bases de datos pueden añadir características de seguridad que también se deben atender en el momento de realizar la migración de la información, de tal forma que en la nueva actualización

de Moodle sea transparente para el usuario la configuración de backend del servidor que aloja, la versión de Moodle y del gestor de base de datos.

Un caso de éxito se llevó a cabo en la Facultad de Ciencias de la Computación, con la recuperación un respaldo de Moodle, y actualización para proveer todas las funcionalidades de Moodle en su versión 3.2 y actualizar a su versión 3.7. También se llevó a cabo la instalación del servidor web, gestor de base de datos, lenguaje de programación y actualizaciones correspondientes.

4 Conclusiones y trabajos futuros

Un proceso de instalación limpio permite que las funciones de cualquier sistema estén más estables por lo que debemos tener cuidado a la hora de manipular varios servicios que nos prestaran uno más complejo, por ello se debe mencionar que la instalación de MariaDB así como como la manipulación correcta de las bases de datos son fundamentales y no se han mencionado en este artículo para hacer pauta a posibles trabajos futuros.

Como observamos un proceso delicado, pero con las bases correctas llega a ser una tarea fácil y completa, y con ello lograr ser un instalador eficaz.

Hay tres áreas que debe respaldar antes de actualizar, a decir, Moodle y todo lo que concierne al servidor. También los Archivos subidos a Moodle, el contenido y, por último, la Base de Datos de Moodle.

Referencias

1. Rojas M., "Moodle como herramienta de comunicación y enseñanza aprendizaje, desde un enfoque constructivista ", Revista Digital Universitaria, 1 de noviembre de 2016, Vol. 17, Núm. 11. Disponible en Internet: http://www.revista.unam.mx/vol.17/num11/art79/index.html ISSN: 1607-6079.
2. Pineda P. (2013) Los LMS como herramienta colaborativa en educación ISBN-13: 978-84-15698-29-6
3.Sánchez J., (2012) Usos pedagógicos de Moodle en la docencia universitaria desde la perspectiva de los estudiantes REVISTA IBEROAMERICANA DE EDUCAIÓN. N.º 60, pp. 15-38 (ISSN 1022-6508)
4. Ros, I. (2008). Moodle, la plataforma para la enseñanza y organización escolar. Ikastorratza, e- Revista de Didáctica 2. Retrieved from http://www.ehu.es/ikastorratza/2_alea/moodle.pdf (issn: 1988-5911)
5. Sánchez, I.J. (2010): *Plataforma educativa Moodle administración y gestión,* México: Alfaomega
6.El servidor HTTP número uno en Internet; https://httpd.apache.org/; Accedido el 1 de mayo de 2019.
7. Soporte Unicode MySQL https://docs.moodle.org/32/en/MySQL_full_unicode_support; Accedido el 1 de mayo de 2019.
6. https://docs.moodle.org/dev/Moodle_and_PHP7; Accedido el 1 de mayo de 2019.
8. MySQL soporte completo de Unicode; https://docs.moodle.org/32/en/MySQL_full_unicode_support; Accedido el 1 de Mayo de 2019.

9. Actualización a Moodle; https://docs.moodle.org/all/es/Actualizaci%C3%B3n_de_moodle; Accedido el 1 de Mayo de 2019.
10. Mallett, A. (2014). CentOS System Administration Essentials. Packt Publishing.
11. Point, T. (2016). MariaDB. Tutorials Point.
12. West, A. W. (2018). Practical PHP7, MySQL8, and MariaDB Website Databases. Apress.
13. Wild, I. (2017). Moodle 3.x Developer's Guide. Packt Publishing - ebooks Account.
14. William E. Shotts, J. (2012). The Linux Command line. Copyright. San Francisco.
15. Wood, W. (2018). Migrating to MariaDB. Apress.

Análisis de los Principales Algoritmos de Ciberataques Utilizados en 2019-2020

Yeiny Romero, María del Carmen Santiago, Gustavo T. Rubín,
Miguel Ángel Pérez, José Alberto Rendón
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 14 sur, Col. San Manuel, Puebla México
{yeiny.romero, marycarmen.santiago, gustavo.rubin}@correo.buap.mx,
miguel.perezxi@alumno.buap.mx, albertors280996@gmail.com

Resumen. Las vulnerabilidades en los sistemas de información son el principal problema hoy en día en la red, en los últimos años han aumentado considerablemente, gracias a ello se hace uso de la criptografía que surge como una solución en los sistemas de seguridad de la información contra ataques maliciosos. En este documento, se analizan los ataques más importantes y las técnicas de cifrado que se han utilizado con la finalidad de dar un mejor soporte en los sistemas de seguridad.
Palabras clave: Criptografía, seguridad, información, cifrado.

1 Introducción

Las vulnerabilidades de los sistemas de información son el principal problema para solucionar dado que conlleva fuertes consecuencias la fuga de información, son los puntos débiles de un sistema donde su seguridad informática no tiene defensas en caso de un ataque, [1] en los últimos años el ciberataque ha sido considerado como una guerra no solo por el avance tecnológico que existe, sino por la cantidad de información que es robada. El Foro Económico Mundial en 2019 publico su estudio "Riesgos Globales" en donde cita que los ciberataques se posicionan en el quinto lugar de Riesgos Globales.[2]. El terrorismo electrónico, se puede definir como el uso de las tecnologías de información para intimidar, coaccionar o para causar daños a grupos sociales, con objeto de lograr una serie de fines políticos, religiosos y finalmente monetarios.[3][4][5][6][7]

Los ataques y las filtraciones de datos han causado estragos en grandes compañías. La ciberseguridad es cada vez más importante,[5][6][7] en especial para quienes manejan servicios directos a usuarios La fuga de información puede ocurrir como resultado de una acción de un ataque, la aparición de vulnerabilidades en los sistemas y los métodos de encubrimiento de los atacantes, lo convierten en una práctica en aumento. Algunos de los principales ataques en la red son:[6][7][8]

Malware: se refiere a software malicioso que tiene por objetivo infiltrarse en un sistema para dañarlo. [5][9] [11][12][13]

Virus: código que infecta los archivos del sistema mediante un programa maligno, se necesita que el usuario lo ejecute directamente. [5][9]

Gusanos: es un programa que, una vez infectado el equipo, realiza copias de sí mismo y las difunde por la red. [5][9]

Troyanos: busca abrir una puerta trasera para favorecer la entrada de otros programas maliciosos. [5][9]

Spyware: programa espía, cuyo objetivo es recopilar información de un equipo y transmitirlo a una entidad externa sin el consentimiento del propietario. [5][9]

Ransomware: su objetivo es secuestrar datos (encriptándolos) y pedir un rescate por ellos, generalmente el pago es en bitcoin. [5][9] [11][12][13]

Escaneo de puertos: Su empleo permite al atacante realizar un análisis preliminar del sistema y sus vulnerabilidades. [5][9]

Phishing: se trata de suplantación de identidad para obtener datos privados de las víctimas, como contraseñas o datos bancarios. [5][9] [11][12][13]

Botnets: son computadoras o dispositivos infectados y controlados remotamente, que se comportan como robots o "zombis". [5][9]

Denegación de servicios: inhabilita el uso de un sistema o computadora, con el fin de bloquear el servicio para el que está destinado. [5][9][11][12][13]

En el presente trabajo, se hablará sobre el análisis de los ataques a los sistemas de seguridad, el tipo de ataque y el algoritmo de cifrado que usan para proponer una posible solución a este tipo de ataques e ir un paso delante de los atacantes.

2 Estado del Arte

Para poder realizar alguno de estos ataques, deben contar con una finalidad, y oportunidad que facilite el desarrollo del ataque, como podría ser un fallo en la seguridad del sistema informático elegido o alguna vulnerabilidad en el mismo. Algunos ataques importantes en los años 2019 y 2020 fueron:

- Según Kaspersky Lab, el año pasado (2019) el ataque más sonado fue ransomware, que son códigos informáticos maliciosos que secuestran los equipos de cómputo para pedir un rescate económico.[14]
- Ryuk una variante de ramsomware ha sido una de las más virulentas y responsable de muchos de los casos como PEMEX que fue una de las víctimas de este ataque. [15]
- Sodinokibi o Bitpaymer son del tipo de malware ramsomware que han impactado en muchas empresas también. Uno de los casos más llamativos que han tenido lugar en 2019 es el ciberataque a Vodafone [15]
- La empresa VASS comenta que han sido 3 los principales ciberataques ocurridos en el 2019 Ransomware, Phising y Ataques DDos [16]
- Shlayer es un malware que causa daño en los Dispositivos Apple lo que hace es interceptar el tráfico de internet, poniendo anuncios en las páginas que visitan que pueden llevar a la descarga de más malware [17]
- Los hackers de todo el mundo aprovecharon la contingencia han lanzado 600 nuevas campañas de phishing a nivel mundial por día [18]
- Durante la pandemia, el comercio electrónico se convirtió en el único medio de acceso a múltiples productos. Muchas personas que tenían desconfianza de hacer compras en línea dejaron de desconfiar y gracias e ello se han visto bombardeados

por nuevos y sofisticados ataques de spam, phishing, ransomware e ingeniería social.[19][20]

- El hospital Moisès Broggi de Sant Joan Depí, en Barcelona, víctima de un ataque ransomware donde no se vio comprometida información confidencial.[21]

El phishing y el ransomware se están utilizando para infiltrarse en los sistemas y redes, especialmente en correo, en páginas educativas y de salud, en obras de caridad que necesitan donaciones, en mensajes de la industria o temas relacionados con el covid.[22]

El éxito del ransomware ha preocupado en la industria y el FBI ha emitido advertencias sobre la amenaza de alto impacto [23]

Hemos visto que las grandes empresas son las más propensas a los ataques dado que los datos almacenados en servidores o repositorios son muy rentables, por lo que se recurre al uso de herramientas criptográficas para extraer la información. La criptografía se encarga de cifrar la información para evitar que su contenido pueda ser visto por personas no autorizadas; la función principal se centra en algoritmos de cifrado que buscan proteger los datos y que los atacantes han usado a su favor.

Estos algoritmos de cifrado han existido a lo largo de la historia por ejemplo el cifrado por sustitución (mono alfabética, poli alfabética) donde se remplaza cada letra del alfabeto por otra y en la clave se especifica el tipo de sustitución, el cifrado por transposición donde se intercambian las posiciones de las letras de una palabra siguiendo un esquema definido y los más actuales como cifrado simétrico (hace uso de un método monoclave, es decir,usa la misma clave para cifrar y descifrar como en el caso de AES Advanced Encryption Standard, DES Data Encryption Standard Y 3DES hace uso de un cifrado múltiple, es decir, se vuelve a cifrar el criptograma una o más veces con otras claves; esto hace que la clave efectiva aumente de tamaño, asi que se necesitan más operaciones para poder romperla) y cifrado asimétrico (se utilizan claves distintas para el cifrado y el descifrado, la clave pública y la clave privada sus ejemplos son RSA Rivest, Shamir y Adleman con tamaño de clave mayor a 1024 bits, DSA Digital Signature Algorithm con tamaño de clave de 512 bits a 1024 bits usado comúnmente para firmas digitales [8]

Durante esta investigación se buscará analizar cuáles son los algoritmos criptográficos más usados por los atacantes en Ransomware, DDos y Phishing para poder evitar o mitigar estos ataques, explicando que estos ciberataques no sólo representan una amenaza momentánea para las empresas y gobiernos del mundo, sino que será una lucha continua para salvaguardar la información.

Usando las técnicas anteriores de cifrado o la combinación de varias de ellas los atacantes buscan perjudicar, inhabilitar o provocar intrusiones en los sistemas informáticos. Por ejemplo, los ataques más sonados desde 2019 son:

Ransomware: Lo que hace comúnmente es cifrar algunos o todos los archivos con una determinada clave, que sólo el atacante conoce, con lo cual exige al usuario una recompensa para obtener dicha clave. Existen 2 tipos de ransomware el bloquea pantalla conocido como Lockscreen y que tal vez con extraer el disco y copiar la información a otro equipo lo podamos eliminar para no perder la información y el que cifra archivos, conocido como Cryptolockers siendo este el más peligroso ya que

cambia la estructura de los archivos dejándolos ilegibles e inutilizados hasta pagar el rescate dado que la clave de encriptación para regresar todo a su estado original es la que ofrece el atacante a cambio de dinero. Las consecuencias de un ataque por ransomware son la pérdida de información de forma temporal o permanente, la interrupción de los servicios regulares, y las pérdidas financieras [24] normalmente lo hace a través de la criptografía simétrica (cifrar, cifrar, cifrar pero sin parar) un ejemplo será AES algoritmos de cifrado bien estudiados y de seguridad demostrada, caracteriza por: Ser muy rápidos y requerir claves cortas, por ejemplo, de 128 ó 256 bits.

Phishing: Usa técnicas basadas en frecuencia de patrones (machine learning). Existen varios tipos de phishing, como por teléfono llamada vishing, por mensajes de texto llamado SMiShing a través de la red se encuentra el Spray and Pray usa técnica de emails con el asunto "urgente" donde pide actualice su contraseña. El Spear Phishing ataca a organizaciones específicas para obtener su información personal y hacerse pasar por ellas para obtener datos de sus clientes. El fraude del CEO ataca a empleados, ganándose su confianza y posteriormente pidiendo información personal del resto de los empleados. Phishing de Alojamiento de Archivos que buscan acceder a los datos que sus víctimas tienen almacenados en la red para ver si tienen algo valioso que usar. Phishing de Criptomoneda crean páginas de registro falsas a páginas de criptomoneda para extraer los fondos de sus víctimas. [25][26][30]

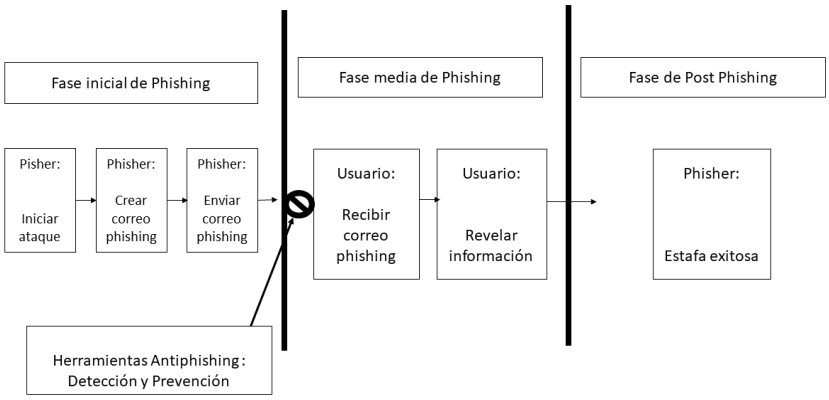


Fig. 1 Ciclo de vida de un ataque Phishing que muestra el proceso de infección y extracción de información

Existen diferentes formas por las cuales un usuario puede ser afectado por este tipo de intrusiones o ataques, entre ellas se encuentra la Ingeniería social (el arte del engaño) que cuenta con todos los recursos que utilizan los ciberdelincuentes para hacer que las víctimas realicen movimientos para vulnerar la seguridad y acceder a la información, [4] gracias a que simulan ser una persona para que la víctima se confíe y haga una acción que revele la clave o dé clic en la liga maliciosa. lo que lo hace más peligro cuando a través de este descuido se logra acceso a diferentes sistemas dentro una empresa.[27]

DDOS: Distributed Denial of Service se basa en utilizar un conjunto de máquinas distribuidas para que produzcan el ataque o sobrecarga de servicio. Con este ataque se ven afectados los servicios con una carga de trabajo determinada que no soportan la saturación del sitio, entre ellos se encuentran los sitios web y los servidores DNS. Los mecanismos mayormente utilizados para este tipo de ataque son la inundación de paquetes TCP, inundación de datagramas UDP e inundación de mensajes ICMP. Existen algunos factores que posibilitan estos ataques como la seguridad de Internet, los recursos limitados de los equipos, ancho de banda y capacidad de tráfico mayor a las redes finales, el uso de políticas locales definidas por sus propietarios. Para que estos ataques tengan éxito se necesitan personas que se dedican a crear grandes redes de equipos "zombie" (botnets) y posteriormente rentarlas.[28] Los ataques DDoS dirigidos a los servicios de cifrados utilizan cifrado SSL / TLS para ocultar su actividad, los sistemas de prevención de intrusiones son insuficientes para enfrentar ataques cifrados dirigidos a servicios encriptados.[29]

3 Metodología

Se propone hacer uso de Hacking ético para contrarrestar este tipo de ataques recordando que el hacking es la detección de vulnerabilidades de seguridad y explotación de estas. El Hacking ético hace referencia al buen uso de las herramientas de ataque para beneficio de nuestros sistemas apoyando a la detección de vulnerabilidades. Los ataques actuales son complicados de contrarrestar, existen aparte de pérdidas de datos, perdida de recursos monetarios y pérdida de tiempo valioso, así que, dicho esto, es más recomendable la prevención de ellos, entre los mecanismos de prevención encontramos la detección de vulnerabilidades y los ataques de fuerza bruta. Un ataque de fuerza bruta ocurre cuando el atacante emplea determinadas técnicas para probar combinaciones de contraseñas y así lograr acceso a una cuenta o sistema. En el caso de este proyecto se busca que mediante esta técnica se obtenga una protección de los datos.
Existen diversas aplicaciones libres las cuales tratan de dirigir o guiar los requerimientos de las evaluaciones en seguridad. La idea principal de utilizar una metodología es ejecutar diferentes tipos de pruebas paso a paso, para poder juzgar con alta precisión la seguridad de los sistemas. Se probarán 2 herramientas en particular:

TCH Hydra = Utiliza el ataque de fuerza bruta para crackear prácticamente cualquier servicio de autenticación remota. Hace uso de algoritmos criptográficos capaces de descifrar DES. Admite ataques rápidos de diccionario para más de 50 protocolos, incluidos el ftp, https, telnet, etc.
Puede usarse para entrar en escáneres web, redes inalámbricas, Crafter de paquetes, Gmail, etc. La cual se ejecutó mediante línea de comandos.

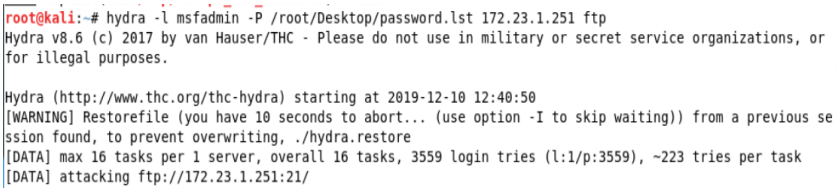


Fig.2 Ataque de fuerza bruta en TCH Hydra mostrando el listado de usuarios del host atacado

Una vez identificados usuarios en el sistema, se puede poner en funcionamiento un ataque de fuerza bruta el cual prueba que la contraseña elegida por el usuario es sencilla, los diccionarios se encargan de contener muchas de las más comunes contraseñas débiles que se suelen usar por descuido. THC Hydra hace el ataque pasando como argumento el usuario que ya previamente se encontró agregado y se le paso un diccionario que no es más que un archivo de texto con todas las palabras posibles que pueden coincidir como contraseña, si la contraseña es fácilmente explotable la herramienta la proporciona en una salida.

John the Ripper es otra herramienta utilizada en la comunidad de pruebas de penetración; es capaz de visualizar las contraseñas de los usuarios aun estando cifradas. Logra romper la seguridad en cifrados o hash como DES, SHA-1, Hace uso de los archivos shadow y passwd los cuales son los archivos críticos que contienen los usuarios del sistema, así como su contraseña. Particularmente hace la unión de los dos archivos en unos mismo para efectuar la desencriptación. Con el archivo fusionado y la ayuda de un diccionario predeterminado que contiene cientos de contraseñas posibles se ejecuta el ataque pasando como argumento la clave --wordlist=password.lst que es el diccionario que ocupara para llevar a cabo la fuerza bruta y el archivo fusionado con el contenido de los usuarios y contraseñas ataque-fusion.txt

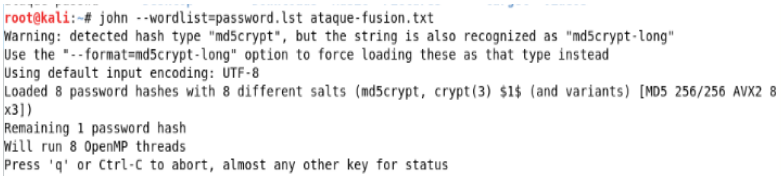


Fig. 3. Ejecución de la herramienta John The Ripper

4 Resultados

Los resultados obtenidos hacen ver la facilidad y exactitud de las herramientas para acceder a la información. Haciendo uso de TCH Hydra se descubrió la vulnerabilidad en NFS se hizo el montado del directorio en la maquina atacante para obtener todos los archivos y la información de la máquina vulnerable obteniendo un resultado exitoso ya que se pudo montar el directorio NFS y al acceder a él se pudo ver toda la información de la maquina atacada del usuario root (usuario principal del sistema operativo linux).

Se creó un nuevo usuario en la maquina atacada, se le asigno una contraseña fácilmente descifrable. Con el ataque de John The Ripper la contraseña puede ser fácilmente descubierta debido a su popularidad a pesar de su encriptación, se observó que en el archivo fusionado con el nombre y la contraseña del usuario estaba cifrada pero al realizar el ataque con John The Ripper esta encriptación fácilmente se descifro, así como todas las contraseñas que los parámetros indican como fáciles que la herramienta puede detectar con ayuda del diccionario.

5 Conclusiones

Es importante estar al día con el tipo de ataques en la red, saber que los atacantes hacen uso de técnicas conocidas para obtener beneficios conlleva a revisar con gran detenimiento toda la información que implique encriptación y herramientas de detección de vulnerabilidades. Los algoritmos de encriptación son conocidos a lo largo de la historia lo que los hace de fácil acceso, teniendo el debido conocimiento y manejo de estas herramientas se pueden realizar las pruebas de seguridad y penetración siempre y cuando se haga con ética en beneficio de sus sistemas o servidores. Las herramientas se pueden combinar entre sí, para generar un plan de detección de vulnerabilidades en los sistemas y con ello reforzar la seguridad y cuidar la información. Hydra y John the Ripper son sorprendentes dado que al usarlas hay que estar conscientes de que son para cuidar los accesos y sobre todo las contraseñas de nuestros usuarios. Se debe recalcar a los usuarios la importancia de generar una contraseña difícilmente descifrable (combinación de números, letras, caracteres, mayúsculas, minúsculas, etc.) además, se debe apoyar a cuidar la información, a mantener actualizados los sistemas e instalar aplicaciones de protección como antimalware, antispam, antivirus y sobre todo proponer las adecuadas políticas de seguridad necesarias que nos brinden un mejor soporte.

Referencias

1.Vulnerabilidad. Obtenido de Significados: [en línea] https://www.significados.com/vulnerabilidad/
2. Forbes Staff. (2018). América Latina registra 9 ciberataques por segundo. Forbes. Recuperado de https://www.forbes.com.mx/america-latina-registra-9-ciberataques-por-segundo/
3. Mark M.Pollitt. (1998). Cyberterrorism — fact or fancy? Science Direct. Recuperado de https://www.sciencedirect.com/science/article/pii/S1361372300870098
4. Blogthinkbig.com (22 febrero 2019). Mayores filtraciones del 2018. Obtenido de: [en línea] https://blogthinkbig.com/filtraciones-datos-ciberseguridad-2018
5.b2b Consultores (enero 10, 2019). Amenazas y vulnerabilidades de los sistemas informáticos. Obtenido de: [en línea] http://btob.com.mx/ciberseguridad/amenazas-y-vulnerabilidades-de-los-sistemas-informaticos/
6.OpenWebinars (19 agosto 2019) Que es el hacking. Obtenido de: [en línea] https://openwebinars.net/blog/que-es-el-hacking/
7. Gómez Vieites, Álvaro. (2011). "Enciclopedia de la seguridad informática", 2da edición, RA-MA, 830 p.

8. Jordi Herrera Joancomartí (coord.) Joaquín García Alfaro, Xavier Perramón Torni. (2004). Aspectos avanzados de seguridad en redes. Catalunya: UOC Formación de Posgrado.
9. Daniel Bechimol.(2011).Hacking desde cero. Argentina: Fox Andina S.A. Recuperado de: https://www.freelibros.me/ Consultado el 20 de julio de 2020
10.David Jiménez Domínguez. (2018). Privacidad de la Información. Ataques Dirigidos. Agosto 2020, De Revista .Seguridad | 1 251 478, 1 251 477 | Revista Bimestral Sitio Web: Https://Revista.Seguridad.Unam.Mx/Numero-05/Privacidad-De-La-Informaci%C3%B3n-Ataques-Dirigidos
11.López Gómez, Julio. (2010). Guía de campo de hackers aprender a atacar y defenderte, México, Alfaomega, 180 p.
12.Buch, Jordi; Jordan, Francisco. (2006). "Seguridad en las transacciones en internet"
13.Picouto Ramos, Fernando; Lorente Pérez, Iñaki; Ramos Varón, Antonio. (2008). "Hacking y seguridad en internet", 1ra Edición; México; Editorial Alfaomega.
14. Miguel Ángel Mendoza . (2020). Ciberataques: una de las principales amenazas para el 2020. 10/09/2020, de Welivesecurity Sitio web: https://www.welivesecurity.com/la-es/2020/02/13/ciberataques-principales-amenazas-2020/
15. Monica Valle. (2020). Estos fueron los mayores ciberataques del 2019. 09/2020, de Bit Life Media Sitio web: https://bitlifemedia.com/2020/01/estos-fueron-los-mayores-ciberataques-de-2019/
16. TICPymes. (2020). 3 ciberataques que nos seguirán al 2020. 30 oct 2020, de BPS Business Publications Spain S.L. Sitio web: https://www.ticpymes.es/formacion/noticias/1115978049404/3-ciberataques-seguiran-al-2020.1.html
17. MARTA GASCÓN. (2020). ¿Usas Mac? Cuidado, este malware ha conseguido saltarse las medidas de seguridad de Apple. 29-oct-2020, de 20 Minutos Editora, S.L. Sitio web: https://www.20minutos.es/noticia/4370065/0/usas-mac-cuidado-este-malware-ha-conseguido-saltarse-las-medidas-de-seguridad-de-apple/
18. Maigo Gómez. (2020). 2.1 billones de ataques cibernéticos en lo que va de 2020. 0ctubre 2020, de GeeksTerra Sitio web: https://geeksterra.com/seguridad/billones-de-ataques-ciberneticos-en-2020/
19. Christopher Holloway. (2020). El estado de la seguridad IT en 2020: Las superficies de ataque se amplían y las personas nunca fueron tan importantes para la defensa. octubre 2020, de IT Masters Mag. Sitio web: https://itmastersmag.com/informes-whitepapers/el-estado-de-la-seguridad-it-en-2020-las-superficies-de-ataque-se-amplian-y-las-personas-nunca-fueron-tan-importantes-para-la-defensa/
20. Redacción. (2020). Pronostican 40% más de ataques cibernéticos. octubre 2020, de eSemanal – Noticias del Canal Sitio web: https://esemanal.mx/2020/05/pronostican-40-mas-de-ataques-ciberneticos/
21. Javier Pastor. (2020). Un ataque ransomware en el hospital Moisès Broggi, en Barcelona, provoca el colapso de algunos servicios secundarios. octubre 2020, de Xataka Sitio web: https://www.xataka.com/seguridad/ataque-ransomware-hospital-moises-broggi-barcelona-colapsa-algunos-servicios-secundarios
22. Fernando Román. (2020). COVID-19: Nuevos ciberataques, nuevos retos La seguridad de la información no solo depende de la tecnología. octubre 2020, de Cybersecurity & Privacy Services, PwC México Sitio web: https://www.pwc.com/mx/es/gestion-de-crisis/covid-19/covid-19-ciberseguridad.html
23. Antonio Hernández. (2020). Advierten aumento de ciberataques en 2020. octubre 2020, de El Universal Sitio web: https://www.eluniversal.com.mx/cartera/advierten-aumento-de-ciberataques-en-2020

24. ESET. (2017). GUÍA DE Ransomware. octubre 2020, de ESET Sitio web: https://www.welivesecurity.com/wp-content/uploads/2017/11/guia-ransomware.pdf
25. SoftwareLab.org. (2020). ¿Qué es Phishing? La definición y los 5 ejemplos principales. octubre 2020, de SoftwareLab.org Sitio web: https://softwarelab.org/es/que-es-phishing/
26. Webranded. (2018). Conoce los principales tipos de phishing y evita que los hackers te puedan sorprender. octubre 2020, de RANDED Sitio web: https://randed.com/tipos-de-phishing/
27. ESET. (2016). 5 cosas que debes saber sobre la Ingeniería Social. 30 octubre 2020, de ESET Sitio web: welivesecurity.com/la-es/2016/01/06/5-cosas-sobre-ingenieria-social/
28. Jennifer Allen. (2020). What Is a DDoS Attack? Important things to know about denial of service attacks. 31 octubre 2020, de Lifewire Sitio web: https://www.lifewire.com/what-is-a-ddos-attack-4787991
29. Fredy Yesid Ávila. (2018). Ataques a servicios encriptados. octubre 2020, de securityhacklabs.net Sitio web: https://securityhacklabs.net/articulo/ataques-a-servicios-encriptados
30. Jaime López Sánchez y Pau del Canto. (2019). Métodos y técnicas de detección temprana de casos de phishing. octubre 2020, de España de Creative Commons Sitio web: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/89225/6/jlopezsanchez012TFM0119memoria.pdf

Ciberseguridad para Aplicación contra el Calentamiento Global

María del Carmen Santiago-Díaz[1], Gustavo Trinidad Rubín Linares[1], Hermes Moreno Álvarez[2], Jéssica López Espejel[3], Antonio Álvarez Núñez[1]

[1]Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 sur, San Manuel, Puebla, Puebla, 72000, México

[2]Universidad Autónoma de Chihuahua, Circuito No. 1, Campus Universitario 2, Chihuahua, Chihuahua, C.P. 31125, México

[3]Université Paris 1399, avenue Jean-Baptiste Clément 93430 Villetaneuse

[3]Commissariat à L'énergie Atomique, Paris, Francia

[1]{marycarmen.santiago, gustavo.rubin}@correo.buap.mx, hm1713a@gmail.com, jessicalopezespejel@gmail.com, antonio.alvarez@alumno.buap.mx

Resumen. El problema de la seguridad de la información es lo más importante al utilizar aplicaciones móviles que involucren la localizacion del dispositivo. En éste trabajo se presenta el análisis de las vulnerabilidades para una aplicación móvil que se diseñará para enviar información de temperatura y localización del dispositivo a una estación de procesamiento, como parte de un proyecto mayor que se esta desarrollando a fin de medir en tiempo real la velocidad de cambio de los parámetros atmosféricos y generar un panorama de comprensión y prevención del Calentamiento Global. Para este fin se desarrolla la metodología de diseño de la aplicación y los ataques que deben realizarse a fin de probar la seguridad de la información.

Palabras clave: Temperatura, Localización, Aplicación Móvil, Calentamiento Global.

1 Introducción

El calentamiento Global es uno de los principales problemas que se deben resolver y contener a fin de mejorar las condiciones de vida no solo de parte de la población sino de todos los seres vivos que habitan el planeta tierra. Las soluciones involucran a todos los sectores de la sociedad y requieren ademas de grandes inversiones de capital un cambio en la conducta humana, dicho cambio debe ser en practicamente todos los sentidos pero algunos que han sido bastante difundidos involucran principalmente la conciencia ecologica, el reciclado de materiales, condiciones menos invasivas de utilización de los recursos naturales, etc.. Aunque muchos factores contribuyen al Calentamiento Global, tanto naturales como los generados por el hombre, el caso de los gases de efecto invernadero se puede analizar y medir las consecuencias de su ritmo de aumento y las características que poseen. En muchos países se han realizado muchos estudios y a partir de estos se han generado estrategias para contrarestar este Calenta-

miento Global, sin embargo, no hay aun una solución única, por ello en este trabajo se plantea una etapa que es importante para conocer la evolución de los parámetros ambientales en tiempo real y de forma puntual sin utilizar sistemas de medición sofisticados, incluso midiendo uno o dos parámetros ayudaría a determinar los restantes. Para este fin se propone que mediante una aplicación móvil se puede medir temperatura y ubicación, la información sea enviada a centros de procesamiento donde estarían arribando cientos de miles o millones de paquetes de información para analizarse en tiempo real, sin embargo, el éxito de esta etapa depende de que se garantice la seguridad de los datos, lo cual es objeto de este trabajo.

Por otro lado, los ataques pueden provenir de muchas fuentes y afectar desde una aplicación o servicio hasta dañar el sistema informático entero.[2][3][4] Es de suma importancia la protección de la información que está expuesta en la red, ya que cada vez hay más datos que se comparten o se guardan en algún lugar. Existen diversos mecanismos que hacen que nuestra información sea más segura, por ejemplo, los mecanismos de prevención dentro de los servidores como los firewalls, los mecanismos de corrección como los antivirus y los mecanismos de cifrado [6] para que los datos viajen seguros a través de la red. El presente trabajo detalla la elaboración del segmento de seguridad de un proyecto sobre el cambio climático, la seguridad de la aplicación consistirá en el uso de un código de cifrado,[3][4][5] capaz de codificar la información recolectada y enviarla por internet a un servidor donde se aplican los principios de seguridad como son autentificación, integridad y disponibilidad de los datos [3][4][5].

1.1 Algoritmos de Cifrado

La encriptación ha adquirido hoy en día especial importancia a la hora de proteger los datos dado que la información se recibe y envía a través de la red y hay que protegerla en su camino. Cuando se envían los datos cifrados se hace mediante fórmulas matemáticas o lógicas, de manera que sólo el destinatario tenga los códigos necesarios para poder descifrarlos y leerlos. [3][4][5]

A lo largo de la historia diversos algoritmos criptográficos han ayudado a contrarrestar estas vulnerabilidades, en la criptografía moderna se ha hecho uso de las claves simétricas (cifrado por bloques, modos de operación, cifrado de flujo, código de autenticación de mensajes y cifrado Hash) y asimétricas (firma digital, intercambio de claves, RSA-Rivest, Shamir y Adleman, etc.) han sido hasta hoy en día populares por el mecanismo de cifrado. [5][7][8][9] Ver Tabla1.

Tabla 1. Tabla de comparación de las principales características de los algoritmos AES, Triple DES y RSA. [7] donde es destacable la longitud de la clave y el tipo de clave que fueron decisivos para el uso de un algoritmo en particular.

Factores	AES	Triple DES	RSA
Tipo de cifrado	Simétrico	Simétrico	Asimétrico
Longitud de la clave	128,192 y 256 bits	K1, K2 y K3 168 bits (k1 y k2 168 bits (k1 y k2 es el mismo) 112bits	516, 1024 o 2048 bits

Tipo de cifras	Cifrado por bloques	Cifrado por bloques	Cifrado de flujo
Clave	Sola	Sola dividida en 3	Pública y Privada

Cabe destacar que el algoritmo RSA es de los más completos en el momento de la implementación por los tipos de clave que maneja, tener una clave pública y otra privada permiten minimizar los riesgos que conlleva mandar mensajes a través de la web, además con la longitud de su clave será más difícil un ataque y si a eso le sumamos que también podemos autentificar los remitentes, entonces es una buena opción.

3 Metodología

En el desarrollo se toma como referencia la metodología security by design que permiten evaluar la seguridad de las aplicaciones, como es el caso de OWASP. Dicha metodología tiene como objetivo identificar los peligros correspondientes a la seguridad de dispositivos móviles y proporcionar los controles necesarios en el desarrollo para reducir su impacto y la probabilidad de explotación de los mismos.

3.1 Análisis

Se plantea el desarrollo de una aplicación que procese información de temperatura y posición desde un dispositivo para saber los cambios de temperatura de todas las ubicaciones donde se realice un muestreo a través de un dispositivo móvil, mismas que serán enviadas en tiempo real al servidor donde se realiza el procesamiento de la información, garantizando una comunicación segura a través de la red hasta llegar al servidor. El análisis de los mecanismos de seguridad cuando la información viaja a través de la red son vulnerabilidades, por lo que la manera de mitigarlos es que los datos viajen encriptados desde el envío del móvil hasta que es recibida en el servidor. Para ello se hace uso de la criptografía en algunas de las capas del modelo TCP/IP modelo actual en el envío y recepción de datos. El modelo TCP/IP detalla cómo se transmite la información a través de la red, recordando que un modelo de red sirve para interconectar equipos computacionales que utilizan diferentes sistemas operativos, diferentes tarjetas de red. El modelo TCP /IP está dividido en 4 capas. Ver figura 1.

Modelo TCP/IP	Protocolos seguros
Capa aplicación	https, s-ftp, ssh, s-mime, cifrado de datos
Capa transporte	Ssl, Tls
Capa de internet	Ipsec, Algoritmos de cifrado
Capa de red	Ppp, autenticación

Fig. 1. Capas y sus respectivos protocolos de seguridad del modelo TCP/IP

Pueden existir distintas vulnerabilidades en cada capa del modelo que son aprovechadas por los atacantes para explotar los protocolos asociados a cada una de ellas.

3.2 Diseño

En cada una de las capas del modelo TCP/IP se debe verificar los mecanismos de seguridad descritos en la figura interior, en particular la capa de aplicación que interactúa directamente con el usuario debe proteger los datos que serán enviados con algunos protocolos que involucran cifrado por ejemplo RSA que es un sistema de cifrado asimétrico que utiliza claves hasta de 2048 bits lo que lo hace robusto en el cifrado generando dos claves una pública para encriptar y una privada para desencriptar véase en la Figura 2.

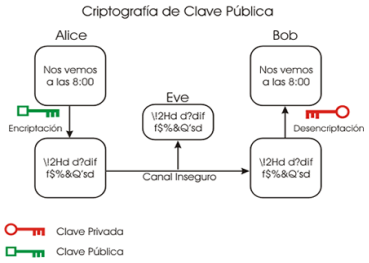


Fig. 2. Clave Pública Asimétrica

Además, se propone un Token de seguridad que se genera en el móvil y se envía para el registro y como factor de autentificación, el cual se muestra en la Figura 3.

Fig. 3 Token de autentificación.

3.3 Desarrollo

El segundo proceso de comunicación involucra la generación de un código de usuario, que se obtiene de obtener datos del dispositivo (por ejemplo MAC Address) y con ellos generar un código único. Este código se combinará con los datos atmosféricos obtenidos y pasarán por dos procesos de cifrado, asegurando que estos datos solo puedan ser interpretados una vez que el servidor los reciba.

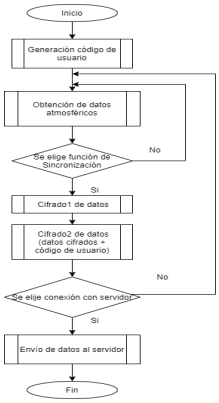


Fig. 4 Diagrama de llave pública asimétrica con los datos de latitud, longitud, temperatura

3.4 Pruebas

Para verificar el éxito del algoritmo programado se realizan pruebas de envío de datos desde el móvil al receptor de la información de manera codificada donde se pudo constatar a través de un analizador de protocolos que la información viajaba de una manera cifrada y segura y que ningún paquete fue vulnerado en el recorrido. Figura 5

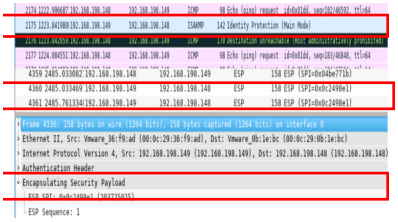


Fig. 5 Pruebas de Funcionamiento del cifrado y envío de la información

4 Resultados

A través del desarrollo seguro se identifican y analizan las amenazas, dando un valor de riesgos a la vulnerabilidad de pérdida de información en el envío desde el dispositivo móvil al receptor, por lo que se identificaron mecanismos de seguridad para mitigar las amenazas a través del cifrado de algoritmo asimétrico RSA, además el registro del usuario a través de un Token.

5 Conclusiones

El desarrollo seguro de cualquier software es indispensable en la actualidad, tomar en cuenta lenguajes de programación y mecanismos que añaden seguridad en los sistemas dado que la información que viaja a través de la red es vulnerable y puede ser interceptada, modificada o secuestrada repercutiendo en pérdidas económicas considerables y exponiendo información sensible de los usuarios. En este trabajo a partir del análisis realizado y usando el modelo propuesto para transferir datos a través de la red se incentiva el uso de protocolos y algoritmos de encriptación de los datos para legitimar las conexiones entre servidor y cliente. El uso del algoritmo RSA de tipo asimétrico fue implementado de manera satisfactoria en un lenguaje multiplataforma lo que facilitó su programación y se logró cifrar el contenido de la información enviada desde el móvil hasta el servidor que fue receptor y almacenador de la información. La propuesta de implementar un doble cifrado de la información, además de la generación de tokens para validar las conexiones del usuario garantizan que la información se encuentra protegida cuando se realizan sincronizaciones de la aplicación con el servidor para transferir los datos recolectados por el móvil.

References

1. UIT-T. (2010). Actualidades de la UIT. Septiembre 2020, de UIT-T Sitio web: https://www.itu.int/net/itunews/issues/2010/09/pdf/201009_20-es.pdf
2. Josep Jorba Esteve, Remo Suppi Boldrito. (2004). Administración avanzada de GNU/Linux. Oberta de Catalunya: UOC Formación de Posgrado.
3. Andrew S. Tanembaum y David J. Wetherall. (2012). Redes De Computadoras. México: Pearson Education. Recuperado de: https://www.freelibros.me/redes/redes-de-computadoras-5ta-edicion-andrew-s -tanenbaum Consultado el 20 de julio de 2020
4. Stallings, William. (2004). Comunicaciones Y Redes De Computadores. Madrid: Pearson Education. Recuperado de: https://www.freelibros.me/redes/comunicaciones-y-redes-de-computadores- 7ma-edicion-william-stallings Consultado el 20 de julio de 2020
5. Jordi Herrera Joancomarti, Software Libre, Aspectos Avanzados de Seguridad en Redes. Recuperado de: https://www.freelibros.me/redes/aspectos-avanzados-de-seguridad-en-redes Consultado el 22 de julio del 2020
6. Juan Ranchal. (2019). Veinte incidentes que resumen la ciberseguridad en 2019 ¡Y Feliz año nuevo!.10/sep/2019, de MuySeguridad Sitio web: https://www.muyseguridad.net/2019/12/31/ciberseguridad-en-2019/
7. "Cifrado AES y RSA", Boxcryptor.com, 2020. [En linea]. Disponible: https://www.boxcryptor.com/es/encryption/. [Ultimo Acceso: 14 agosto de 2020].
8. J. Martinez, Cifrado de Clave Privada: AES, 1st ed. Catellón de la Plana: Universitat Jaume I, 2016, pp. 45.56.
9. D. Córdoba, "RSA: ¿Cómo funciona este algoritmo de cifrado? - Junco TIC", Junco TIC, 2016. [Online]. Available: https://juncotic.com/rsa-como-funciona-este-algoritmo/. [Accessed: 14- Aug- 2020].

Diseño de un micro laboratorio como estación meteorológica para Mediciones de temperatura en el Medio Ambiente.

María del C. Santiago-Díaz, Judith Pérez-Marcial, Gustavo T. Rubín-Linares, Claudia Zenteno-Vázquez, Antonio E. Álvarez
Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 sur, San Manuel, Puebla, Puebla, 72000, México
{marycarmen.santiago, judith.perez, gustavo.rubin, ana.zenteno}@correo.buap.mx, antonio.alvarez@alumno.buap.mx

Resumen. El acceso a nuevas tecnologías y soluciones innovadores están dando resultados satisfactorios para combatir las consecuencias del calentamiento global. Este trabajo se tiene como objetivo la creación de micro laboratorios con smartphones como estaciones meteorológicas con el fin de recolectar y registrar datos en intervalos cortos y de manera segura, además de enviar los valores a un centro de datos para su posterior procesamiento, esto facilitará la identificación de variaciones anormales de temperatura en el medio ambiente que permita toma de decisiones a través de una interfaz inteligente, Para identificar riesgos acacionado por las altas temperaturas en una zona especifica. Como resultado se obtiene una app que a través del uso de tecnologías móviles como son GPS y conectividad wifi, bluetooth y 3G nos permita obtener parámetros del medio ambiente en tiempo real de una manera práctica que a diferencia de las estaciones meteorológicas automáticas, donde su infraestructura es costosa.

Palabras clave: GPS, Temperatura, Localización, Aplicación Móvil.

1 Introducción

El cambio climático es la variación global del clima de la Tierra debido a causas naturales y también a la acción del ser humano. El calentamiento global del planeta se ve acelerado por gases de efecto invernadero causados por las actividades humanas. Los principales son el dióxido de carbono (CO2), el metano (CH4) y el óxido de nitroso (N2O). El CO2, por ejemplo, aunque es el principal gas que contribuye al cambio climático, no es nocivo para la salud humana. Tiene consecuencias múltiples y de impacto global derivadas principalmente de los cambios en los patrones climáticos, el aumento del nivel del mar y los fenómenos meteorológicos más extremos. El cambio climático no supone únicamente un fenómeno ambiental, porque sus impactos negativos tienen consecuencias sociales y económicas. Las consecuencias de este calentamiento global son incontables, la desaparición de los glaciares, aumento de los incendios forestales, incremento de la temperatura de los océanos y puesta en peligro de la vida marina, entre otros[1,2].

Cada día, en diferentes puntos de la geografía mundial, el planeta nos manda mensajes sobre las enormes transformaciones que está sufriendo. El Grupo Intergubernamental de Expertos sobre el Cambio Climático de la ONU (IPPC) creado por la Organización Meteorológica Mundial (OMM) y la ONU Medio Ambiente. En 2018 el IPCC publicó un informe especial sobre los impactos del calentamiento global a 1,5°C. Este informe subraya que la limitación del calentamiento global a 1.5ºC, comparado con 2ºC, debe de ir unida al compromiso de construir una sociedad más sostenible y equitativa, e indica que gran parte del impacto del cambio climático ya se produciría con 1,5°C de aumento.

Por ejemplo, para 2100 el aumento del nivel del mar a nivel global sería 10 cm más bajo con un calentamiento global de 1,5°C. Las probabilidades de tener un Océano Ártico sin hielo durante el verano disminuirán a una vez por siglo, en lugar de una vez por década. Los arrecifes de coral disminuirían entre un 70 y 90%.

Se requieren transiciones rápidas y oportunas en la tierra, la energía, la industria, los edificios, el transporte y las ciudades. [3].

Los efectos del cambio climático son cada vez más visibles en el mundo y en México también se han hecho presentes durante 2019.

En lo que va del año, han ocurrido una serie de fenómenos ambientales que nos alertan para tomar acción sobre lo que estamos haciendo a nuestros ecosistemas.

Hielo y granizo en verano, playas infestadas de sargazo, inundaciones con saldos rojos, la contingencia ambiental por contaminación e incendio de la reserva de la biósfera de los Petenes [4].

El medio ambiente juega un papel muy importante todos los días en nuestra salud y bienestar, cuidarlo es una de las mayores preocupaciones en la actualidad, llevamos años utilizando los recursos naturales y ha tenido consecuencias tanto positivas como negativas en el aire que respiramos, el agua que bebemos, los niveles de ruido de los que estamos expuestos y el tiempo meteorológico].

Uno de los sectores que ha experimentado un mayor desarrollo en las últimas décadas ha sido el tecnológico. La tecnología, bien utilizada, también puede servir para ayudar al cuidado del medio ambiente. La conectividad permanente a Internet desde el móvil nos permite acceder a multitud de información y datos en tiempo real sobre diferentes aspectos del entorno. Ya es más que habitual consultar la previsión meteorológica, y pronto lo será también la calidad del aire que respiramos o que vamos a respirar al viajar a un lugar concreto.

Para ello se suelen utilizar los datos generados por diferentes estaciones de medición, las estaciones Meteorológica Automática las cuales a través de un conjunto de dispositivos eléctricos y mecánicos que realizan mediciones de las variables meteorológicas de forma automática. En una Estación Meteorológica Automática, está conformada por un grupo de sensores que registran y transmiten información meteorológica de forma automática de los sitios donde están estratégicamente colocadas. Su función principal es la recopilación y monitoreo de algunas Variables Meteorológicas para generar archivos del promedio de cada 10 minutos de todas las variables, esta información es enviada vía satélite en intervalos de 1 ó 3 horas por estación. Los sensores que integran la Estación son de Velocidad del viento, Dirección del viento, Presión atmosférica, Temperatura y Humedad relativa, Radiación solar y Precipitación [5].

Los teléfonos inteligentes están proporcionando una plataforma ambiental de bajo coste, la funcionalidad del GPS o la geolocalización proporciona datos para la obtención de medidas que se comparten mediante el uso de correos electrónicos, SMS o APPs. El Smartphone ofrece una interfaz común para el procesamiento de los datos, por tanto, de igual la manera y forma en la se obtiene los dato mediante el usos de sensores inalámbricos, al final, al pasar por el dispositivo móvil los datos quedan condicionados a una plataforma común, disponibles para entornos profesionales o entornos centrados en los ciudadanos[1]. El censado del medioambiente a través de tecnologías de red y comunicaciones móviles se obtiene por muchos sensores ambientales con capacidades inalámbricas que van desde Zigbee de baja potencia a redes Wi-Fi o 3G/4G[2]

Por ejemplo, aplicaciones móviles para aviso, prevención y monitorización del cuidado del medio ambiente, con aportaciones como Plume Air Report que permite saber en tiempo real los niveles de contaminación de un área concreta. Además, pronostica cómo evolucionará la calidad del aire en esa zona en las siguientes 24h a la consulta [5]. Loss of the night que obtiene la gran cantidad de luz que emana de las ciudades nos impide observar las estrellas de noche, permite reconocer los niveles de polución lumínica que se alcanzan en ciertos lugares y nos indica la visibilidad que tienen las estrellas en cada punto [6]. Go Green: Carbon Tracke, es una aplicación muy interesante, pues analiza la huella de carbono que producen nuestras actividades cotidianas: desde el consumo eléctrico en el hogar, hasta el uso de transporte público o privado. Gracias a ella, el usuario puede saber, en base a una escala que determina la propia app, si sus actividades son respetuosas con el medio ambiente. Una vez detectados los hábitos con margen de mejora, la propia app ofrecerá al usuario consejos y alternativas sobre cómo cambiar algunos de sus hábitos, adquiriendo un comportamiento más sostenible. Otra aplicación es Noise Watch donde los ciudadanos pueden aportar sus mediciones de ruido. a través de su dispositivo móvil que utiliza el micrófono del teléfono para medir el ruido del ambiente. Esa medición, acompañada con datos referentes a la ubicación geográfica obtenidos del GPS del teléfono, se envían a través de la conexión a Internet del móvil. Los datos enviados por los ciudadanos se suman a las mediciones científicas y oficiales que se obtienen[6]. Otra aplicación importante es Forest Fire Danger Meter, se trata de una calculadora para hallar el peligro de incendio (atendiendo a la clasificación McArthur Forest Fire Danger Index) en función de una serie de parámetros, Temperatura, Humedad relativa, Velocidad del viento, Factor de sequedad, Carga inicial de combustible, Pendiente. Con las funciones de Aviso de incendio, Permiso quema, Velocidad incendio, Riesgo de incendio, Cálculo coste extinción estimado.

Como referencia a los incendios forestales la Secretaría de Medio Ambiente y Recursos Naturales (Semarnat) y la Comisión Nacional Forestal (Conafor) informan que entre el 1º de enero al 23 de julio de 2020, en México se han registrado 5 mil 473 incendios forestales con afectación a una superficie de 305 mil 474 hectáreas, en su mayoría pastos y matorrales. El Estado de México (1,082), Michoacán (601) y Jalisco (586) son las entidades que más incendios han presentado durante este año, en tanto que los estados que reportan la mayor cantidad de hectáreas afectadas son: Guerrero (46,578), Quintana Roo (31,143), Baja California (30,698) y Jalisco (29,573). Cabe señalar que el acumulado de incendios en 2020 registra el 24% menos, en comparación con el mismo periodo de 2019 donde se registraron 7,245 incidentes. En cuanto a la

superficie afectada se reporta 48% menos que en el mismo periodo, ya que en 2019 el fuego sin control alcanzó a un total de 589 mil 371 hectáreas. Las causas de incendios que se tienen hasta el momento continúan siendo por actividades del ser humano entre las cuales destacan las actividades ilícitas (27%), agrícolas (27%), desconocidas (13%), pecuarias (9%), fogatas (9%) entre otras. Aunque los incendios forestales ocurren durante todo el año, la temporada fuerte para la región Centro, Norte, Noreste, Sur y Sureste del país se presenta principalmente del mes de enero a junio. En tanto que para la región Noroeste, la temporada crítica es de mayo a septiembre, ambos casos coincide con el estiaje.

2 Metodologia.

La metodología de desarrollo consiste en la implantación de micro laboratorios mediante tecnología smartphone que permitan recuperar la temperatura del sitio para ser enviadas en paquetes seguros y así ser procesados en centros de datos para identificar patrones o tendencias del cambio de temperatura y se puedan prevenir riesgos como incendios, se describe la arquitectura en la Figura 1, la cual muestra la distribución de los componentes para la obtención de la temperatura a través de smartphone. Para la comunicación entre el dispositivo móvil y el servidor de información se estableció como punto de transferencia de información digital, las redes GPS (Sistema de Procesamiento Global), portadas por el proveedor de servicio de telefonía celular, tal como se representa en la Figura 1.

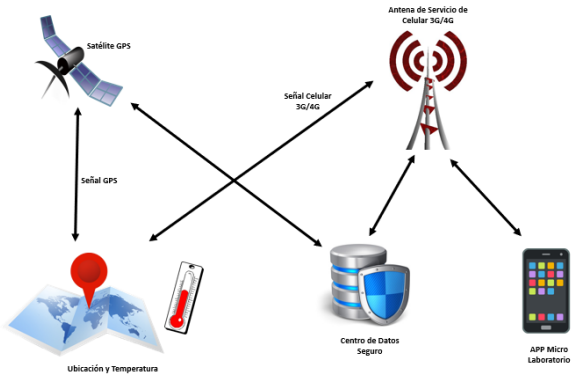


Fig. 1. Arquitectura del sistema propuesto

Para el procesamiento de datos, se establece la importancia de alojar información en el dispositivo móvil Android mediante el uso de SQL y su posterior sincronización de datos con el servidor principal como se puede ver en la Figura2.

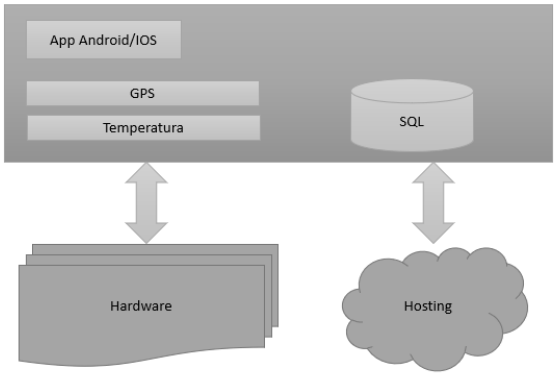


Fig. 2: Arquitectura de componentes de la aplicación móvil

Se realiza la transmisión de paquetes. Para la implementación de la aplicación del servidor se usó el marco de desarrollo de aplicaciones basado en modelo vista controlador MVC, que es un conjunto de herramientas que facilita la construcción de software con un conjunto de bibliotecas de tareas comunes, una interfaz sencilla y una estructura lógica.

3. Desarrollo

Para el desarrollo se utilizo la metodologia de desarrollo de SCRUM, La elaboración del proyecto se constituye en un proceso de construcción de conocimientos que puede caracterizarse como:

i) Evolución continua en requerimientos como en funcionalidad;
ii) Tiempos de elaboración
iii) Proceso de elaboración del proyecto es incremental;
iv) Integración de conocimientos previos y en la introducción de novedades tecnológicas.

Se considera factible aplicar las prácticas de la metodología ágil SCRUM en la gestión y control del proceso de elaboración del proyecto, Se elabora un marco de trabajo que se expone a continuación.
Planificación. La gestión de los requerimientos del proyecto, la cual consiste en una lista de tareas que conlleve a la elaboración de un producto tecnológico

SPRINT		
ID	ITEM	ESTIMACIÓN
#A1	Gestionar información de la aplicación móvil	5
#V1	Obtención de Temperatura a través de la Ubicación	4

#V2	Envío al centro de datos	3
#V3	Gestionar de Datos	4
	TOTAL:	16

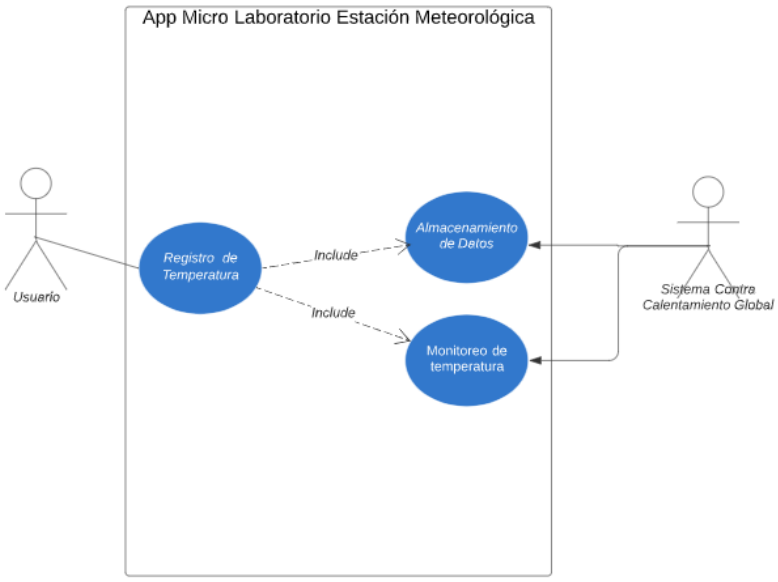
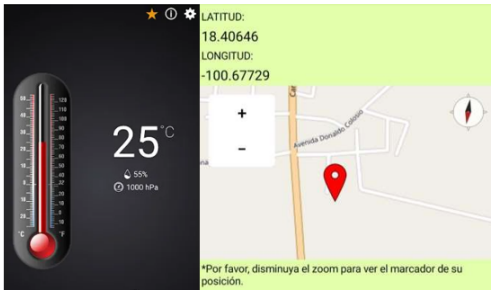


Fig. 3.

4 Resultados.

Se tiene como resultado una aplicación móvil segura que permite recolectar datos del medio ambiente como es la temperatura en tiempo real de una ubicación específica, utilizando la tecnología GPS con conectividad bluetooth, wifi, 3G, en intervalos cortos para su posterior procesamiento, que facilita el acceso a la información y a la toma de decisiones a través de una interfaz inteligente.



5. Conclusiones.

El día hoy existen un número elevado de estrategias para ayudar al medio ambiente y protegerlo de malas acciones que el ser humano provoca o que en ocasiones es la misma naturaleza la que genera algunos estragos. Así que como una suma más a ese esfuerzo por el planeta, se desarrolla la aplicación móvil con el objetivo de ayudar y proveer información en tiempo real en el caso de variaciones anormales en la temperatura ambiente ya que su uso mejora los sistemas de estimación de parámetros ambientales, los cuales nos permita medir la eficiencia de las estrategias para revertir el cambio climático, además que permita realizar diagnósticos eficientes. La aplicación integra diferentes tecnologías y conocimientos que permiten recolectar datos del entorno (temperatura y ubicación) y enviarlos a una base de conocimiento para ser procesados, se propone como trabajo a futuro recolectar información a través de otros parámetros que se pueden obtener a través de los dispositivos móviles con conectividad a internet, tales como el acelerómetro, barómetro o el de proximidad, todos estos y otros más con la intensión de proporcionar una medición más exacta a la hora de procesar la información y así determinar qué tan probable se genere algún problema ambiental, ya que se tendrá registro de todas las variables.

Referencias

1. Relación entre el cambio climático y la contaminación del aire. (s. f.). Sostenibilidad para Todos. Recuperado 15 de agosto de 2020, de https://www.sostenibilidad.com/cambio-climatico/relacion-cambio-climatico-contaminacion-del-aire/
2. Amón, R. (2020, 7 abril). El calentamiento global no frena su avance: 2020 será uno de los años más cálidos. El confidencial. https://www.elconfidencial.com/tecnologia/ciencia/2020-04-07/calentamiento-global-2020-ano-record-calor_2537964/
3. Cambio Climático. (s. f.). Naciones Unidas, Paz, dignidad e igualdad en un planeta sano. Recuperado 15 de agosto de 2020, de https://www.un.org/es/sections/issues-depth/climate-change/index.html

4. Garcia, D. O. (5 de septiembre de 2016). Análisis y diseño de un dispositivo de sensores ambientales para su integración con terminales móviles. Recuperado 14 de agosto de 2020, de oa.upm: http://oa.upm.es/47295/1/PFC_Diego_Onofre_Artes_Garcia.pdf
5. K. Mainwaring, L. Srivastava, The Internet of Thing – setting the standards, The Internet of Thing: connecting objects, pp. 208-210, 2010
6. Estaciones Meteorológicas Automáticas (EMAS). (s. f.). Gobierno de México. Recuperado 15 de agosto de 2020, de https://smn.conagua.gob.mx/es/observando-el-tiempo/estaciones-meteorologicas-automaticas-ema (Referencia OMM 20182).
7. Carcelén, S. G. (2015). Monitorización y gestión de una red de sensores inteligentes: aplicación móvil. Recuperado el 14 de agosto de 2020, de Ruinet: https://riunet.upv.es/handle/10251/54181
8. Marciszack, E. C. (27 de mayo de 2015). Sistema de monitoreo de la calidad del aire integrado a IoT. Recuperado el 14 de agosto de 2020, https://rdu.iua.edu.ar/bitstream/123456789/1126/1/Sistema de Monitoreo de la Calidad del Aire Integrado a IoT.pdf
9. 5 efectos del cambio climático en México durante 2019. (2020, 28 mayo). Enlight. https://residencial.enlight.mx/5-efectos-del-cambio-climatico-en-mexico-durante-2019
10. Bhonsle, V. (17 de marzo de 2020). Geocoding a Location Using Python and Flask. Recuperado el 14 de agosto de 2020, de Developer Here: https://developer.here.com/blog/understanding-geocoding-with-python

Aplicaciones Científicas y Tecnológicas de las Ciencias Computacionales
se terminó de editar en Diciembre de 2020 en la Facultad de Ciencias de la Computación Av. San Claudio y 14 Sur
Jardines de San Manuel
Ciudad Universitaria
C.P. 72570

Aplicaciones Científicas y Tecnológicas de las Ciencias Computacionales

Coordinado por María del Carmen Santiago Díaz