

APORTACIONES DE LAS CIENCIAS COMPUTACIONALES EN SISTEMAS INTELIGENTES Y CIBERSEGURIDAD



**APORTACIONES DE LAS CIENCIAS
COMPUTACIONALES EN SISTEMAS
INTELIGENTES Y CIBERSEGURIDAD**

**APORTACIONES DE LAS
CIENCIAS COMPUTACIONALES
EN SISTEMAS INTELIGENTES Y
CIBERSEGURIDAD**

María del Carmen Santiago Díaz
Gustavo Trinidad Rubín Linares
Ana Claudia Zenteno Vázquez
Yeiny Romero Hernández
María de Lourdes Sandoval Solís
(Editores)

Gustavo Trinidad Rubín Linares
(Coordinador)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Yeiny Romero Hernández, María de Lourdes Sandoval Solís
(editores BUAP)

Gustavo Trinidad Rubín Linares
(coordinador BUAP)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Yeiny Romero Hernández, Judith Pérez Marcial, José Luis González Compeán, Alejandro Galaviz, Heidy Marisol Marín Castro, Abelardo Gómez Andrade, Paola Eunice Rivera Salas, , Ana Bertha Ríos Alvarado, María Concepción Landa Arnaiz, Osslán Osiris Vergara Villegas, Carina Toxqui Quil, Juan Manuel González Calleros, Miguel Morales Sandoval, Abel Alejandro Rubín Alvarado, María Eugenia Narcisa Sully Sánchez Galvez, Jaime Julián Cid Monjaraz, Oleg Starostenko Basarab, Carlos Soubervielle Montalvo, Víctor Manuel Morales Rocha, Germán Ardul Muñoz Hernández, Lorna Verónica Rosas Téllez, Fernando Reyes Cortes, Luis Carlos Altamirano Robles, Abraham Sánchez López, María Josefa Somodevilla García, Maya Carrillo Ruíz, Luis Enrique Colmenares Guillén, Josefina Guerrero García , Meliza Contreras González, Gabriel Juárez Díaz, José de Jesús Lavalle Martínez, Ivo Humberto Pineda Torres, Roberto Contreras Juárez, Héctor David Ramírez Hernández, Nelva Betzabé Espinoza Hernández, Rogelio González Velázquez, Pedro García Juárez, Beatriz Beltrán Martínez, Ana Luisa Ballinas Hernández, Nicolás Quiroz Hernández, Luz del Carmen Reyes Garcés, Alba Maribel Sánchez Gálvez, Bárbara Emma Sánchez Rinza, José Andrés Vázquez Flores, Hermes Moreno Álvarez, Raúl Antonio Aguilar Vera, Julio Cesar Diaz Mendoza, Juan Pablo Ucán Pech, Francisco Marroquín Gutiérrez, Jéssica Nayeli López Espejel, Eden Belouadah
(revisores)

Primera edición: 2022
ISBN: 978-607-8857-30-2

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

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Rectora:
Dra. Ma. Lilia Cedillo Ramírez

Secretario General:
Mtro. José Manuel Alonso Orozco

Vicerrector de Investigación y Estudios de Posgrado:
Dr. Ygnacio Martínez Laguna

Directora de la Facultad de Ciencias de la Computación:
M.I. María del Consuelo Molina García

Contenido

Prefacio	6
Hacking Ético en Tiempos de Covid <i>Karen Aislinn Gonzalez Lopez</i> <i>Yeiny Romero Hernández</i> <i>María del Carmen Santiago Díaz</i> <i>Gustavo Trinidad Rubín Linares</i>	7
Use of Artificial Intelligence in the Diagnosis of Idiopathic Pulmonary Fibrosis <i>Juan Arturo Pérez Cebreros</i> <i>Eric Efraín Solano Uscanga</i> <i>Gerardo Soto Vega</i> <i>Oscar Sebastian Plascencia Reyes</i>	13
Geolocalización en Dispositivos Móviles: una Aproximación al Diseño de Rutas <i>Erika Rodriguez Peña</i> <i>Ana Claudia Zenteno Vázquez</i> <i>María del Carmen Santiago Díaz</i> <i>Leslie Abril Gómez Mora</i>	21
Chatbot de Asesoría Universitaria para el Desarrollo de Aplicaciones Web <i>Cristopher Quecholac Guerrero</i> <i>Judith Pérez Marcial</i> <i>María del Carmen Santiago Díaz</i> <i>Gustavo Trinidad Rubín Linares</i>	28
Análisis de los Modos de Ahorro de Energía en Microcontroladores Usados en Sistemas Embebidos Basados en Tecnología IoT <i>Juan Mora Ramírez</i> <i>Nicolás Quiroz Hernández</i> <i>Ana María Rodríguez Domínguez</i>	34
Arquitectura de Microservicios: Caso de Estudio Servicio de RUTA en la Ciudad de Puebla <i>Pilar Hernández Zambrano</i> <i>Judith Pérez Marcial</i> <i>Gustavo Trinidad Rubín Linares</i> <i>Ana Claudia Zenteno Vázquez</i>	42
e-COMPA: Acompañante Emocional <i>Karina Torres Rodríguez</i> <i>María del Carmen Santiago Díaz</i> <i>Yeiny Romero Hernández</i> <i>Gustavo Trinidad Rubín Linares</i>	48
Uso de Inteligencia Artificial para el Reconocimiento de Imágenes de Vicuñas <i>Manuel Jesus Ibarra Cabrera</i> <i>Robinson Buitrón Barreto</i> <i>Yalmar Ponce Atencio</i> <i>Omar Sigvas Robles</i>	54

Estado del Arte del Diseño de una Nube para Archivos de Grandes Dimensiones

Ricardo Pájaro Coatl

Adriana Mendieta Leon

Said Constantino Cárcamo García

María del Carmen Santiago Díaz.62

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 cátedra 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**

Hacking Ético en Tiempos de Covid

Yeiny Romero Hernández, María del Carmen Santiago Díaz,
Gustavo Trinidad Rubín Linares, Karen Aislinn Gonzalez Lopez

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
Av. San Claudio y 14 sur, Col. San Manuel
C.P. 72000 Puebla, México
{yeiny.romero,marycarmen.santiago, gustavo.rubin }@correo.buap.mx,
karen.gonzalezlo@alumno.buap.mx

Resumen. La información es uno de los elementos más importantes de cualquier organización, y es necesario buscar estrategias para garantizar la protección de los datos, desde un simple archivo hasta grandes volúmenes de información. En este trabajo tenemos como objetivo hacer uso adecuado de herramientas de protección de la información, es decir, el Hacking Ético, que es un método de la aplicación de los conocimientos y técnicas de un hacker con un propósito de defensa de forma legal. Tendremos a bien probar algunas de las muchas herramientas existentes para la protección de los datos en este caso seleccionaremos 3 de ellas como son: Nmap, Whireshark que ambas son herramientas de monitoreo de redes y Nessus que es una herramienta de detección de vulnerabilidades. Dichas herramientas serán probadas en Sistema Operativo en Alma Linux para verificar si se complementan en la protección de la información.

1 Introducción

En la actualidad la información es lo más importante dentro de las empresas u organizaciones, por ello, es necesario buscar estrategias que nos ayuden a proteger la información, con el fin de garantizar información segura en la red y libre de ataques de cualquier tipo.

El Hacking Ético es utilizado para probar posibles ataques y determinar de una mejor manera la protección de los datos¹. 2. Para ello, necesitamos saber que un hacker es una persona con el conocimiento suficiente en tecnologías que hace uso del mismo para vulnerar sistemas o programas. Los hackers éticos son las personas que con esos conocimientos detectan posibles vulnerabilidades, previniendo de manera efectiva ataques. Se especializan en pruebas de penetración para analizar, evaluar, encontrar vulnerabilidades, para así perfeccionar la seguridad de un sistema o red³.

Las pruebas de penetración son necesarias para protección de los sistemas y para ello es necesario tener conocimientos de programación, conocer de metodologías, manejar herramientas de seguridad, criptografía, etc. Para realizar las pruebas de penetración es necesario conocer y probar diversas herramientas entre las que destacan Nmap, METASPLOIT FRAMEWORK, Wireshark, Nessus, Kali Linux (que cuenta con un conjunto de herramientas de penetración), que son las pruebas básicas del Hacking Ético. En este trabajo probaremos tres de las herramientas mencionadas con el objetivo de hacer pruebas de penetración para comprobar su efectividad en software libre y determinar que se complementan para una buena protección. [4][5][6][7]

2 Metodología

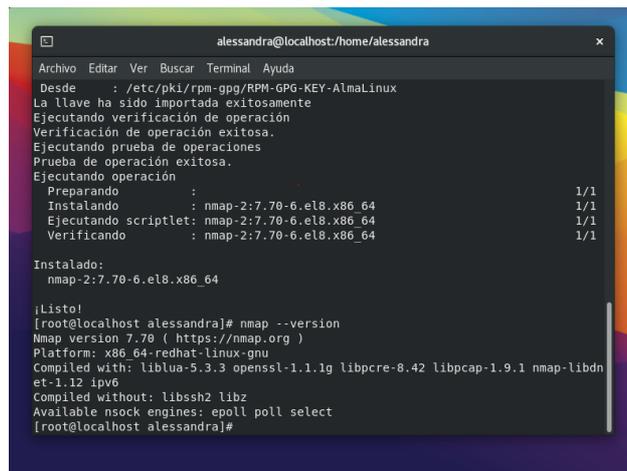
Para este trabajo primeramente necesitamos saber en qué consisten las herramientas que hemos seleccionado y que forman parte del proceso de pruebas de penetración.[8][9][10]11.

2.1 Herramientas de Hacking Ético

Las pruebas seleccionadas deben de evaluar las vulnerabilidades generales, así como monitorear el sistema, permitiendo proteger a las organizaciones según sea su arquitectura 12.

Nmap: Una herramienta de escaneo de red llamada mapeador de redes, que permite determinar qué servicios se están ejecutando en el dispositivo monitoreado, así como determinar el sistema operativo, la presencia de filtros o firewalls, entre otras cosas. Es compatible con sistemas operativos Linux, Windows, Mac.[11].

La instalación se realizará por medio de terminal con el comando dnf para instalar el paquete correspondiente. Como parte final solo nos queda verificar que la instalación se realizó correctamente consultando la versión de Nmap (ver Fig.1).



```
alessandra@localhost/home/alessandra
Archivo Editar Ver Buscar Terminal Ayuda
Desde : /etc/pki/rpm-gpg/RPM-GPG-KEY-AlmaLinux
La llave ha sido importada exitosamente
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando : 1/1
Instalando : nmap-2:7.70-6.el8.x86_64 1/1
Ejecutando scriptlet: nmap-2:7.70-6.el8.x86_64 1/1
Verificando : nmap-2:7.70-6.el8.x86_64 1/1

Instalado:
nmap-2:7.70-6.el8.x86_64

¡Listo!
[root@localhost alessandra]# nmap --version
Nmap version 7.70 ( https://nmap.org )
Platform: x86_64-redhat-linux-gnu
Compiled with: liblua-5.3.3 openssl-1.1.1g libpcrc-8.42 libpcap-1.9.1 nmap-libdn
et-1.12 ipv6
Compiled without: libssh2 libz
Available nsock engines: epoll poll select
[root@localhost alessandra]#
```

Fig. 1. Pantalla final de la instalación de Nmap

Nessus: Esta herramienta tiene una gran base de datos de vulnerabilidades que se ejecutan para determinar si existe una vulnerabilidad o no en el dispositivo a monitorear. Al ejecutar Nessus sin ciertas especificaciones esta verifica miles de vulnerabilidades y devuelve una lista de aquellas encontradas 11.

Nessus es una herramienta de paga, pero tiene su versión de prueba por lo cual es necesario registrarnos en su página oficial, lo que permitirá recibir un correo electrónico con un código de activación y un enlace que contiene el paquete de instalación. Nessus trabaja por seguridad con el puerto 8834, una vez dentro se lleva a cabo una segunda instalación donde se registra el código de activación que recibimos previamente, un nombre de usuario y una contraseña, al terminar solo queda esperar la inicialización de Nessus como se muestra a continuación [12] (ver Fig.2).

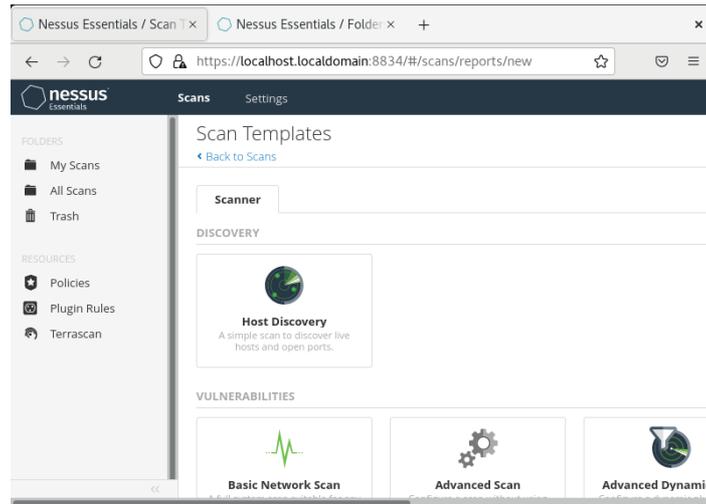


Fig. 2. Pantalla final de la instalación de Nessus.

Wireshark: Es una herramienta conocida como analizador de protocolos, Un software gratuito que permite analizar en tiempo real el tráfico de red, detecta problemas como la latencia o ataques maliciosos.¹³

Esta herramienta es de fácil instalación a través de la terminal, posteriormente podemos consultar la versión del software, lo cual nos permitirá comprobar que se descargó de manera correcta. Finalmente, para abrir la interfaz lo podemos hacer desde la terminal escribiendo el comando wireshark (véase Fig.3).

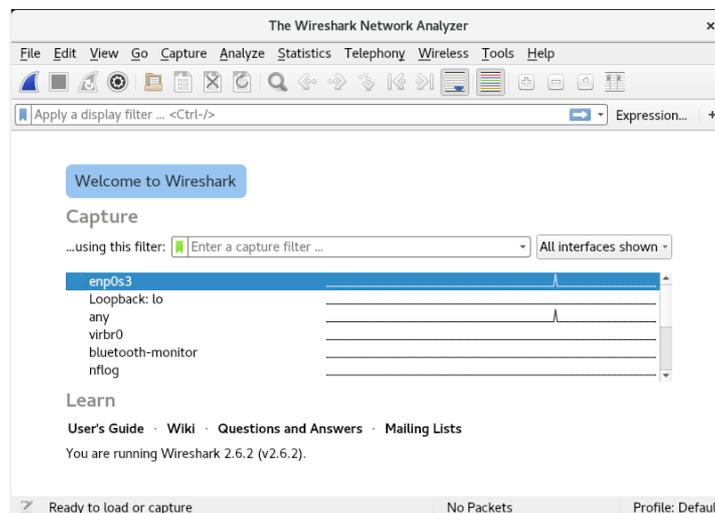


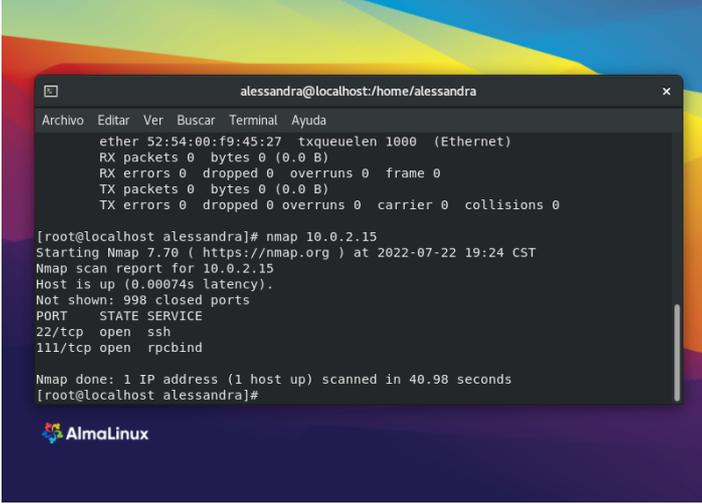
Fig. 3. Pantalla final de la instalación de Whireshark.

3 Resultados

A continuación, se muestra el uso de cada una de las herramientas descritas previamente, con el fin de probar la eficacia a la hora de realizar un monitoreo de red o una detección de vulnerabilidades.

Nmap

En los resultados obtenidos con la herramienta Nmap se puede observar una prueba de escaneo de la dirección IP 10.0.2.15 para la búsqueda de puertos abiertos, donde el host se encuentra activo con una latencia de 0.00074s, no se muestran 998 puertos esto debido a que se encuentran cerrados y únicamente se visualizan los puertos: 22/tcp, en estado abierto, con un tipo de servicio ssh y el puerto 11/tcp en estado abierto, con un tipo de servicio rpcbind (ver Fig.4).



```
alessandra@localhost/home/alessandra
Archivo Editar Ver Buscar Terminal Ayuda
ether 52:54:00:f9:45:27 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost alessandra]# nmap 10.0.2.15
Starting Nmap 7.70 ( https://nmap.org ) at 2022-07-22 19:24 CST
Nmap scan report for 10.0.2.15
Host is up (0.00074s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 40.98 seconds
[root@localhost alessandra]#
```

Fig. 4. Escaneo de la dirección ip 10.0.2.15 en Nmap.

Nessus

Con Nessus creamos nuestro primer escaneo llamado Scan1 donde introducimos la dirección ip 192.168.122.1 colocando a su vez la frecuencia, la hora y fecha de inicio, la zona horaria y cada cuando queremos que se repita el escaneo. Cabe destacar que Nessus pone una paleta de colores de acuerdo con la vulnerabilidad encontrada que van de la más básica hasta la que requiere atención urgente. Cuando no muestra resultados puede deberse a el bloqueo de un firewall, sin embargo, si fuera el caso contrario nosotros podríamos visualizar las vulnerabilidades desplegadas en una barra y grafica de pastel con una gama de colores donde el rojo hace referencia a una vulnerabilidad crítica, el naranja a una alta, el amarillo a una media, el verde a una baja y el azul a únicamente información. Finalmente, si damos clic en una vulnerabilidad especifica nos dará una descripción completa y la solución para erradicar la vulnerabilidad. (Ver Fig. 5)

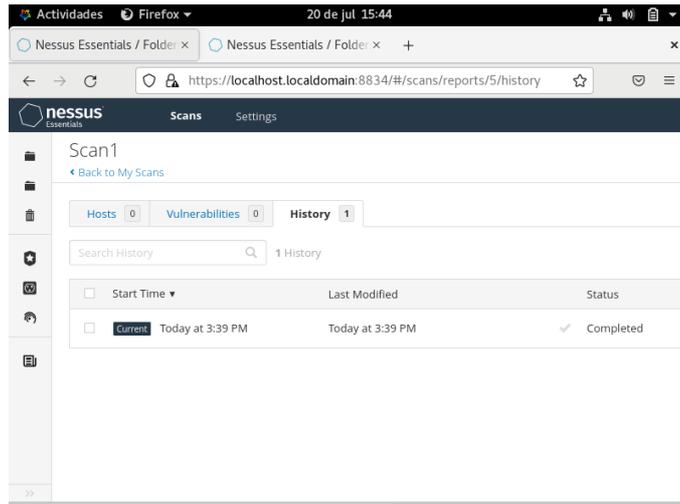


Fig. 5. Escaneo de la dirección ip 192.168.122.1 en Nessus.

Wireshark

La herramienta Wireshark nos permite mostrar un escaneo en tiempo real de paquetes de datos, como resultado del proceso de captura de tráfico a través de nuestra tarjeta de red, pudimos observar en la primera zona de datos un listado de al menos 11 paquetes capturados con información relevante como número, tiempo de captura en segundos, origen, destino, el protocolo involucrado, tamaño y un campo extra de información. Dichos paquetes a su vez cuentan con un esquema de color que nos permite identificar que es lo que acontece en cada paquete, como ejemplo seleccionamos el paquete 9 que está marcado en color negro con letras verdes, tratándose de un informe de error ICMP, donde puede deberse a un host de red inactivo o que nuestro firewall este bloqueando alguna solicitud. En la segunda zona se puede observar a detalle los datos del paquete seleccionado y finalmente en la tercera zona de datos se detalla la cabecera Ethernet II (ver Fig.6).

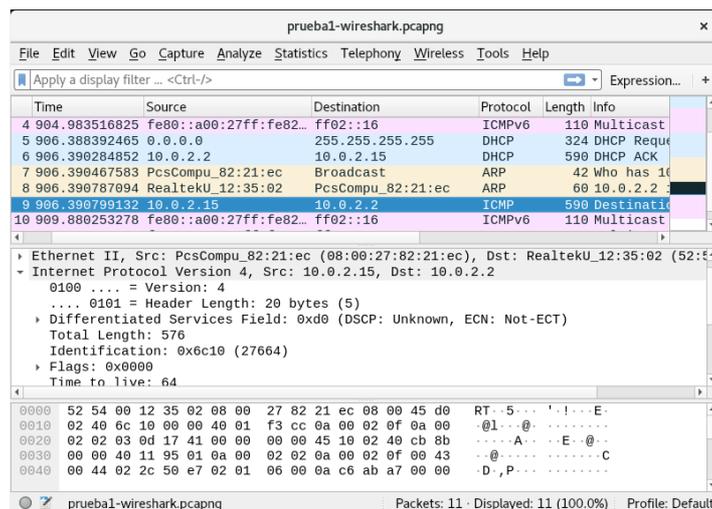


Fig. 6. Escaneo de la interfaz de red enp0s3 en Wireshark.

4 Conclusiones

Para concluir, podemos decir que en la época actual donde por motivos de pandemia se hizo un uso mayoritario del ambiente virtual, también fue importante proteger el exceso de información que viajaba a través de la red, así que hemos detallado información relevante sobre la importancia de proteger la información así como la función del Hacking Ético en la búsqueda de vulnerabilidades con los equipos de cómputo en cualquier sector, Conocimos las pruebas de penetración que son la aplicación del Hacking Ético. Vimos el funcionamiento de Nmap, Nessus y Wireshark, cada una con diferentes funciones específicas y modos de uso. Una vez instaladas y probadas las herramientas se comprobó que cumplen con el objetivo de detección y protección contra vulnerabilidades haciendo posible el uso correcto del hacking ético, cabe destacar que dichas herramientas no se contraponen por el contrario pueden trabajar juntas y complementándose en cualquier sistema operativo Linux como fue este caso. De acuerdo con los resultados mencionados se considera que esta investigación hizo el uso correcto de las herramientas para proteger cualquier equipo conectado a la red, el uso de cada una dependerá de las necesidades o información específica que deseemos proteger para obtener un entorno de máxima seguridad.

Referencias

1. Tori, C. (2008). *HACKING ÉTICO* (1.a ed., Vol. 1) [Libro electrónico]. Mastroianni Impresiones.
2. Asale, R. (s. f.). *jáquer | Diccionario de la lengua española*. «Diccionario de la lengua española» - Edición del Tricentenario. Recuperado 3 de julio de 2022, de <https://dle.rae.es/j%C3%A1quer#TLIznqw>
3. Peña, C. (2020). *Hacking Etico - Vol.1* (1.a ed., Vol. 1) [Libro electrónico]. RedUsers.
4. Berrio Lopez, J. D. (s. f.). *Hacking Ético VS Defensa en Profundidad*. DS Team Seguridad Informática. Recuperado 25 de junio de 2022, de http://www.dsteamseguridad.com/museo/HACKIN%20ETICO_VS_DEFENSA_PROFUNDIDAD_JUANBERRIO.pdf
5. Murguia, F. C. (2022, 26 febrero). *ETHICAL HACKING: TIPOS DE HACKERS*. LinkedIn. Recuperado 4 de julio de 2022, de <https://es.linkedin.com/pulse/tipos-de-hackers-fernando-conislla-murguia>
6. Gutierrez Del Moral, L. (2014). *Curso de Ciberseguridad y Hacking Ético 2013* (1.a ed.) [Libro electrónico]. Punto Rojo Libros S.L.
7. ACISSI. (2018). *Seguridad informática* (4.a ed.) [Libro electrónico]. Ediciones ENI.
8. L. (2020, 31 agosto). *Pentest: ¿qué es y cuáles son los principales tipos?* OSTEC | Seguridad digital de resultados. Recuperado 3 de julio de 2022, de <https://ostec.blog/es/seguridad-perimetral/pentest-concepto-tipos/>
9. Bayón Cascajo, E. (2011, 24 septiembre). *Resumen Conceptos básicos sobre seguridad y hacking ético*. Scribd. Recuperado 27 de junio de 2022, de <https://es.scribd.com/document/68095909/Resumen-Conceptos-basicos-sobre-seguridad-y-hacking-etico>
10. Avila, S. M. A. (2019, 13 marzo). *Hacking ético: impacto en la sociedad*. Universidad Piloto de Colombia. Recuperado 3 de julio de 2022, de <http://repository.unipiloto.edu.co/handle/20.500.12277/4919>
11. Bortnik, S. (s. f.). *Pruebas de penetración para principiantes: 5 herramientas para empezar | Revista .Seguridad*. Seguridad - Cultura de Prevención para TI. Recuperado 3 de julio de 2022, de https://revista.seguridad.unam.mx/numero-18/pruebas-de-penetracion-para-principiantes-5-herramientas-para-empezar#_ftn1
12. Penagos, C. (2019, 24 julio). *ANÁLISIS DE METODOLOGÍAS DE ETHICAL HACKING PARA LA DETECCIÓN DE VULNERABILIDADES EN LAS PYME*. ccpenagosm.pdf. Recuperado 4 de julio de 2022, de <https://repository.unad.edu.co/bitstream/handle/10596/30302/ccpenagosm.pdf?sequence=1&isAllowed=y>
13. *¿Qué es Wireshark? Así funciona la nueva tendencia esencial en seguridad*. (2018, 19 septiembre). IDG Communications S.A.U. Recuperado 6 de julio de 2022, de <https://cso.computerworld.es/tendencias/que-es-wireshark-asi-funciona-la-nueva-tendencia-esencial-en-seguridad>

Use of Artificial Intelligence in the diagnosis of Idiopathic Pulmonary Fibrosis

Juan Arturo Pérez-Cebreros^{1,3}, Eric Efraín Solano-Uscanga²,
Gerardo Soto-Vega¹, Oscar Plascencia-Reyes¹

¹ Instituto Politécnico Nacional ESIME Culhuacán,
Departamento de Ingeniería en Computación, México.

² Universidad de Xalapa, Escuela de Ingeniería, México.

³ Universidad Jesuita de Guadalajara. Departamento de Electrónica, México.

juan.perezc@iteso.mx, efrain@ux.edu.mx, gsotov@alumno.ipn.mx,
oplascenciar@alumno.ipn.mx

Abstract. Respiratory diseases have long been part of a growing problem, and this has been complicated by the emergence of new viruses. In particular, Idiopathic Pulmonary Fibrosis (IPF) is defined as a chronic fibrotic interstitial pneumonia, located in the lung. We performed dense convolutional neural network processing to classify different computed tomography (CT) images. The results obtained show accuracy values of up to 99% in the case of IPF diseases.

Keywords: Artificial Intelligence, Neural Networks, Pneumonia.

1 Introduction

Artificial intelligence (AI) is the term used to describe the use of computers and technology to simulate intelligent behavior and critical thinking comparable to that of a human being; including machine learning (ML), natural language processing (NLP), computer vision, biometrics, virtual and augmented reality [1]. ML can be divided into traditional ML and deep learning (DL). In traditional ML, methods include logistic regression, decision trees, random forests, nearest neighbor, hidden Markov models, vector support machine, among others. DL is a subset of ML which represents a learning method for building deep neural network models [2].

The 1980s and 1990s saw an increase in interest in AI, but it was not until 2016 that a greater number of investments were made in this field of medicine compared to other areas, and this is expected to continue increasing [3]. AI in medicine can be classified into two subtypes: virtual and physical [4]. The virtual aspect ranges from applications such as medical record systems to neural network-based guidance in decision making for treatment. Then, the physical aspect concerns robots that help to perform surgeries and intelligent prostheses for disabled people, among others. The objective of this study is to compare the information from studies of "artificial intelligence in medicine", and specifically, computer-assisted diagnosis.

Medical radiographs are one of the first options for diagnosis due to their capacity for revealing some unsuspected pathological alterations [5]. There are many benefits to developing computer-aided tools for X-ray analysis. However, computer-aided diagnosis is a phrase that provokes strong reactions among radiologists, and many of these reactions are negative.

The term computer-aided detection (CAD) emerged during the 1980s and 1990s, during the second era of artificial intelligence (AI). As with the first era of AI, in the 1950s and 60s, as well as during the current third era in which we live; the second era emerged from many exciting new technologies that promised to solve all of our problems. However, the second era fell far short of these lofty expectations and led to a period of disappointment, often described as the "AI winter" [6]. Given the history of AI, it's no wonder that CAD is viewed by many as a disappointment. Radiologists in

particular have more reason than most to be disappointed, because CAD in medical imaging represented only an unfulfilled promise [7].

Firstly, these automated tools help the radiologist make a well-informed and quantitative decision. As the volume of radiographs increases, it will become increasingly difficult for radiologists to maintain the same level of efficiency. Therefore, automation will be very necessary to maintain the quality of diagnosis. Several investigations concluded that traditional CAD has not been beneficial but has reduced the radiologist's precision, leading to higher biopsy rates. Similarly, we can say that the medical results were not only disappointing, but also represented an estimated 20% increase in the time needed to interpret each study, because of the need for the radiologist to rule out the false alarms produced by these systems. Given these bad experiences and broken promises, why are radiologists now expressing an interest in CAD? The answer is, new technology has been developed that far exceeds the previous methods, a technology known as deep learning that has become popular for solving a variety of problems and seems capable of fulfilling the promises made by CAD pertaining to the second era of AI [8].

This brings us to the first difference between deep learning and traditional computer vision systems, as at least outside of the field of medicine, deep learning works. Proof of this can be found in the well-known image analysis competition; the best traditional methods from the second era produced five times more errors than a human, when asked to identify everyday objects in photographs, such as bicycles, dogs and cars [9]. Likewise, it is apparent how deep learning algorithms contribute to many tasks in other eras only attributable to the human being; for example for the identification of people in the middle of a crowd [10], intervening in complex games such as chess, Go, Shogi [11], handwriting recognition systems [12], natural language processing [13] or in the case of autonomous vehicles, which were practically unable to leave the parking lot but are now being tested on hundreds of real roads [14, 15].

The problems that the third era of AI addresses relate to the management of large quantities of data to derive useful information for the solution of complex problems associated with reasoning, perception, planning, learning [16, 17]. With reference to the above examples, we want to say that AI has become a multidisciplinary field with broad potential to generate benefits for society. Although these applications are not related to medical images, this represents a change in the perception capacity of our computer systems. Thus, during the second era of AI, computer vision systems could not perform visual tasks that a small child would find very easy, but the AI of the third era, is succeeding in tasks that were originally the exclusive domain of humans. However, we do not yet have the level of evidence required to show that deep learning can do the job of doctors, although preliminary results show systems that appear to work at the human level in common medical tasks, such as retinal assessment [18] and skin lesions [19]. We have also seen similar results in radiology; for example, [20] shows a system that can detect a wide variety of critical tomographic findings. This paper is structured as follows. Section 2 explores relevant literature; Section 3 describes the methodology and the results of the experimentation. Finally, Section 4 presents the conclusions for this work.

2 Related work

In medicine, research is constantly attempting to incorporate more areas of activity; aiming to address unmet needs or improve existing solutions. In the area of endoscopy, AI has been widely used as a tool to improve lesion classification. Theoretically, AI has a promising future as a decision-making aid, which would improve on the performance of less experienced endoscopists and theoretically reduce the number of undetected lesions. In colonoscopy, research has focused on the detection and characterization of early lesions, an example of this being the meta-analysis presented in [21], where 4754 images were used with the aim of defining levels of accuracy, in terms of detecting colorectal polyps. Accuracy for detecting neoplastic lesions achieved 96%, with a sensitivity of 92% and a specificity of 89%. This same author carried out a comparative

study to assess the percentage of unperceived lesions resulting either from AI colonoscopy or those undertaken by a colonoscopist; in this study the colon was checked, while an investigator using a different monitor observed the result produced by the machine, and if the machine found a polyp unperceived by the endoscopist, he was notified. Ultimately, it was apparent that AI had missed 79% of adenomas (almost all these in the proximal colon).

Regarding the detection of osteoarthritis, in [22], a deep learning approach was presented to distinguish between knees with osteoarthritis that are painful and those that are not. These prediction results reached 80% of the AUC metric (Area Under Curve). In [23], a classification method was proposed on the Kellgren-Lawrence scale, using a CNN DenseNet121 as a classification method, for which X-ray images, and correct predictions reached 73%.

Concerning the detection of breast lump locations and their classification, as either benign or malignant in a mammogram, we have access to the YOLO algorithm [24], and can make comparisons between convolutional neural network architectures, as opposed to other architectures, such as Inception v3, Vgg16, ResNet 40; with CNN architecture reaching an accuracy of 98.94% [25]. As for algorithms that classify skin cancer using images, here they have compared the diagnostic accuracy of DNN (Deep Neural Network) algorithms, to that of dermatologists. In a study using a dataset of 130,000 images, 21 certified dermatological experts were at least as accurate as the algorithm [26]. Subsequently, in [27], a comparative study was carried out using a variety of automatic classification algorithms to detect skin cancer.

Regarding ophthalmology, we can also find some studies, where the performance of algorithms in relation to ophthalmologists is compared. After training with about 130 thousand retinal photographs that were labeled by 54 ophthalmologists, an accuracy of 91% was obtained for the classification of retinal abnormalities [28].

When it comes to cardiology, the vast majority of images cardiologists currently use consist of electrocardiograms (ECG) and echocardiograms. In this line of research, they also have a negative history, as for almost forty years, very imprecise rule-based algorithms were used [29]. In [30], a health monitoring system was proposed that aimed to predict cardiac deaths in relation to data obtained from patients. For this, a convolutional neural network was used, obtaining an accuracy of 99%, which is comparable to that of cardiologists.

Likewise, a context that has attracted a lot of attention for the application of AI is radiology. Chest X-rays are the most common, with more than two billion scans performed worldwide each year. In [31] informs us that intelligent algorithms for chest radiography interpretation will have great impact and will become more common in the medical sector. More recently, in another study [32], 99% accuracy is indicated for the detection of COVID using a set of X-ray-type images. Similarly, in [33], diagnostic comparisons of radiography images are presented, using different machine learning algorithms.

In the literature, we can discern many other useful clinical circumstances, such as facilitating the diagnosis of autism [34], helping anesthesiologists [35], selecting viable embryos for in vitro fertilization [36], and also for interactive chatbots [37], mental health issues [38] and automatic segmentation of brain tumors [39], among others.

3 Methodology and Results

In [40], we compare various articles and observe that most authors who use a Deep neural network (DNN) for the diagnosis or prediction of disease obtain a better outcome. This is partly because a DNN can function without the constraints of nonlinearity. In our proposal, we use medical imaging, particularly computed tomography (CT) scans, to assess the accuracy of detecting and diagnosing idiopathic pulmonary fibrosis (IPF). It appears that IPF is difficult to diagnose, as this requires, not only an expert in pulmonology, but also a pathologist and a radiologist [41]. In Mexico, currently 63% of medical centers do not have specialists who can make an adequate diagnosis [42]. The

process to detect IPF begins with the acquisition of a data set, which consists of two files (training and test images). Each file consists of two types of images; those from patients with healthy lungs and those with IPF.

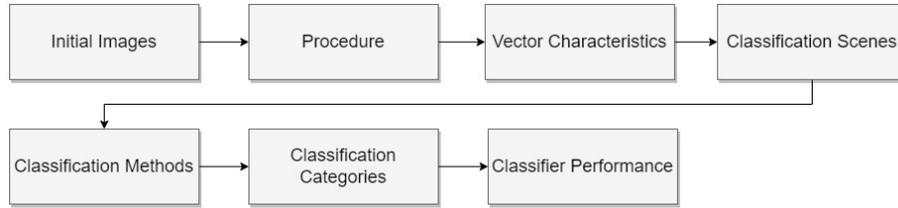


Fig. 1. Schematic diagram of the stages necessary for detection of IPF

In the following, we proceed to describe the tools and configurations used to accomplish the research project:

- **Data set.** To train our model, we use a Kaggle dataset [43]. This data set contains 21,465 CT images, which help doctors detect cancer, along with details such as the shape and size of a tumor. The procedure for generating a tomography takes an average of 20 minutes.
- **Data Augmentation.** This technique is used to balance the amount of data, which allows us to generalize the model and reduce overfitting [44]. In particular, different procedures were carried out on each image, for example rotations, displacements, zooms, cropping, and horizontal reflections.

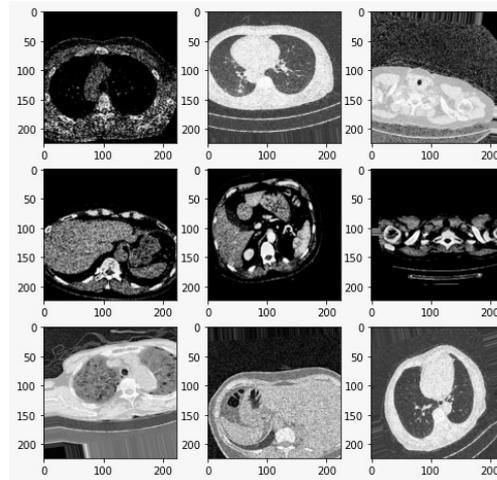


Fig. 2. CT Data Augmentation

- **Software and hardware configuration,** Experiments were carried out using a conventional computer with 16GB in RAM, 500GB in SSD, NVIDIA GTX 1650 graphics card and a 3.4 GHz Ryzen 5 processor. The operating system was Windows 10 PRO with Python 3.9, Keras 2.4 and Tensorflow 2.5
- **DenseNet.** Dense convolutional network architecture represents a joint venture between Cornwell University, Tsinghua University and Facebook AI Research. [45]. This architecture consists of five convolutional layers and three pooled layers; followed by another three fully interconnected layers.
- **ResNet.** Residual network architectures have the ability to learn more complex functions and consequently achieve better performance. However, sometimes adding more layers has a negative effect on the final performance. This phenomenon

is known as the degradation problem, alluding to the fact that while better parameter initialization techniques and batch normalization allow deeper networks to converge, they sometimes converge with higher error rates than less complex ones.

- Normalization. A pre-training of our model was carried out with the final intention; that rather than beginning with random weights, to instead begin with pre-trained weights, in order to reduce training time.
- Model compilation. Finally, the model was trained over 20 epochs, with a batch size of 40. Once the model training was completed, performance was evaluated using the data from the data set tests, reaching an accuracy of 99.3%.

Table 1. Comparison of models.

Algorithm	Accuracy	Precision	Recall	F1-score
ResNet50	99.37	100.00	98.81	99.40
Densenet201	99.30	99.88	98.35	99.11

In Mexico, there are not enough specialist doctors to ensure a quick and safe diagnosis, and as this is a disease with a high mortality rate, it is very important to detect it in time if recuperation is to occur. We decided to work with CT because this is used for the diagnosis of IPF and the automation of processes is one of the most important steps required in the health sector. The aim is to produce a mobile application that can be used by a specialist doctor.

4 Conclusions

Finally, we can conclude that more evidence is required, given the history of experiences with traditional CAD, but the success of deep learning in so many human domains is unprecedented and justifies our degree of optimism. Likewise, this deep learning ability that allows you to be successful in a variety of perception tasks is also distinguishable from traditional DAC technology. Thus, the AI of the second era, focused on a specific task, where each system had to be designed specifically for each task. Proof of this is that specific subgroups of CAD have been defined based on their intended use. We have the most commonly used terms; CADx (computer-assisted diagnosis) and CADe (computer-assisted detection), but there are also other variants such as CADq (computer-assisted quantification) and CAST (simple computer-assisted classification), among others. These subgroups make sense for second-age technology, as their technology could only perform a single task.

In contrast, deep learning is independent of the task; but in order to be able to train the machine to learn to classify new cases successfully, it is necessary to have a large volume of accumulated knowledge referring to thousands of patient cases, along with the characteristics of each disease. From this data set, you can create an algorithm that generates equations to correctly classify new cases, that becomes more robust over time, as you learn from new cases. This characteristic of self-learning, as well as the success in recognizing objects in natural images represents the factor that has stimulated renewed interest for also applying deep learning to medical images. Undoubtedly, this multidisciplinary approach will change the way medicine is practiced, so health professionals will have to prepare, learn and adapt.

We think that the future of Artificial Intelligence (AI) will involve an integrated multidisciplinary approach; we make this assertion because AI is a data-driven technology, where scientists can integrate small datasets to create large volumes of data to improve the accuracy of models. However, regulations exist that do not allow different data sources to be shared and integrated. Therefore, an option is to use distributed learning; a method that distributes and shares the trained values of the model, rather than distributing and sharing the data.

Our next task will be to undertake an in depth study of the accuracy of distributed learning, where models can be trained in different hospitals in an isolated way, so that only the trained models are sent to the main servers, in such a way that we can assure that no data is directly read. We consider that an integrated multidisciplinary approach and distributed learning represents a future line of research because we have ample research by different specialists from different medical fields (such as pulmonology, cardiology, dermatology, ophthalmology, otorhinolaryngology among others) that need to be integrated, with the intention of building multidisciplinary predictive models that take into account information from different diagnoses and medical specialties.

References

1. Wang, P. (2019). On Defining Artificial Intelligence. *J. Artif. Gen. Intell.*, 10(2), 1-37.
2. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5), 1-36.
3. Pakdemirli, A., & Leblebici, A. (2020). AI in Medicine versus AI in Prehospital. *Medicine*, 9(2), 293-7.
4. Hamet, P., & Tremblay, J. (2017). Artificial intelligence in medicine. *Metabolism*, 69, S36-S40.
5. Campadelli, P., & Casiraghi, E. (2005, August). Lung field segmentation in digital postero-anterior chest radiographs. In *International Conference on Pattern Recognition and Image Analysis* (pp. 736-745). Springer, Berlin, Heidelberg.
6. Hendler, J. (2008). Avoiding another AI winter. *IEEE Intelligent Systems*, 23(02), 2-4.
7. Bahl, M. (2019). Detecting breast cancers with mammography: will AI succeed where traditional CAD failed?. *Radiology*, 290(2), 315-316.
8. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8(1), 1-74.
9. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
10. Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429, 215-244.
11. Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., ... & Silver, D. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839), 604-609.
12. Mishra, A. (2021). Handwriting Recognition using Neural Networks, Deep Learning and Support Vector Machines. *Deep Learning and Support Vector Machines (March 4, 2021)*.
13. Landolt, S., Wambsganss, T., & Söllner, M. (2021). A taxonomy for deep learning in natural language processing. Hawaii International Conference on System Sciences.
14. Martínez-Prado, G., Pescador-Hernandez, M., García-Ponce, J., & Pérez-Cebreros, A. (2021). Model for the identification of license plates in Mexico City using machine learning algorithms. *Llankasun*, 3(1), 07-13.
15. Chan, T. K., & Chin, C. S. (2021). Review of autonomous intelligent vehicles for urban driving and parking. *Electronics*, 10(9), 1021.
16. Sossa, J. (2020). El papel de la inteligencia artificial en la industria 4.0. In Rodríguez, P., editor, *Inteligencia artificial y datos masivos en archivos digitales sonoros y audiovisuales*, chapter 1. *UNAM*, first edition, pp. 21–58.
17. Vergara Villegas, O. O., Nandayapa, M., Sossa Azuela, J. H., Cossio Franco, E. G., & Rubín Linares, G. T. (2021). Artificial Intelligence for Industry 4.0 in Iberoamerica. *Computación y Sistemas*, 25(4).
18. Srivastava, K. R., & Girdhar, N. (2022). Retinal Image Segmentation based on Machine Learning Techniques. In *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 252-257). IEEE.
19. Lucieri, A., Bajwa, M. N., Braun, S. A., Malik, M. I., Dengel, A., & Ahmed, S. (2022). ExAID: A Multimodal Explanation Framework for Computer-Aided Diagnosis of Skin Lesions. *Computer Methods and Programs in Biomedicine*, 106620.
20. Chen, Y. T., Chen, Y. L., Chen, Y. Y., Huang, Y. T., Wong, H. F., Yan, J. L., & Wang, J. J. (2022). Deep Learning-Based Brain Computed Tomography Image Classification with Hyperparameter Optimization through Transfer Learning for Stroke. *Diagnostics*, 12(4), 807.

21. Lui, T. K. L., Yee, K., Wong, K., & Leung, W. K. (2019). Use of Artificial Intelligence Image Classifier for Real-Time Detection of Colonic Polyps. *Gastrointestinal Endoscopy*, 89(6), AB135.
22. Chang, G. H., Felson, D. T., Qiu, S., Guermazi, A., Capellini, T. D., & Kolachalama, V. B. (2020). Assessment of knee pain from MR imaging using a convolutional Siamese network. *European radiology*, 30(6), 3538-3548.
23. Kondal, S., Kulkarni, V., Gaikwad, A., Kharat, A., & Pant, A. (2022). Automatic Grading of Knee Osteoarthritis on the Kellgren-Lawrence Scale from Radiographs Using Convolutional Neural Networks. In *Advances in Deep Learning, Artificial Intelligence and Robotics* (pp. 163-173). Springer, Cham.
24. Al-Masni, M. A., Al-Antari, M. A., Park, J. M., Gi, G., Kim, T. Y., Rivera, P., ... & Kim, T. S. (2018). Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Computer methods and programs in biomedicine*, 157, 85-94.
25. Chougrad, H., Zouaki, H., & Alheyane, O. (2018). Deep convolutional neural networks for breast cancer screening. *Computer methods and programs in biomedicine*, 157, 19-30.
26. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639), 115-118.
27. Gaytán-Campos, Morales-Castro, Priego-Sanchez, Fitz-Rodríguez, Guzmán-Cabrera (2022). Automatic Classification of Images with Skin Cancer Using Artificial Intelligence. *Computación y Sistemas*, 26(1), 2007-9737.
28. Grewal, P. S., Oloumi, F., Rubin, U., & Tennant, M. T. (2018). Deep learning in ophthalmology: a review. *Canadian Journal of Ophthalmology*, 53(4), 309-313.
29. Ajitkumar Singh, S., Ashinikumar Singh, S., Dinita Devi, N., & Majumder, S. (2021). Heart Abnormality Classification Using PCG and ECG Recordings. *Computación y Sistemas*, 25(2), 381-391.
30. Malarvizhi Kumar, P., Hong, C. S., Chandra Babu, G., Selvaraj, J., & Gandhi, U. D. (2021). Cloud-and IoT-based deep learning technique-incorporated secured health monitoring system for dead diseases. *Soft Computing*, 25(18), 12159-12174.
31. Wang, G., Kalra, M., & Orton, C. G. (2017). Machine learning will transform radiology significantly within the next 5 years. *Medical physics*, 44(6), 2041-2044.
32. Singh, D., Kumar, V., & Kaur, M. (2021). Densely connected convolutional networks-based COVID-19 screening model. *Applied Intelligence*, 51(5), 3044-3051.
33. Lopez-Betancur, D., Bosco Duran, R., Guerrero-Mendez, C., Zambrano Rodríguez, R., & Saucedo Anaya, T. (2021). Comparación de arquitecturas de redes neuronales convolucionales para el diagnóstico de COVID-19. *Computación y Sistemas*, 25(3).
34. Kalikar, S., Sinha, A., Srivastava, S., & Aggarwal, G. (2022). Early Detection of Autism Spectrum Disorder (ASD) Using Machine Learning Techniques: A Review. In *Proceedings of Third International Conference on Communication, Computing and Electronics Systems* (pp. 1015-1027). Springer, Singapore
35. Singh, M., & Nath, G. (2022). Artificial intelligence and anesthesia: A narrative review. *Saudi Journal of Anaesthesia*, 16(1), 86.
36. Berntsen, J., Rimestad, J., Lassen, J. T., Tran, D., & Kragh, M. F. (2022). Robust and generalizable embryo selection based on artificial intelligence and time-lapse image sequences. *Plos one*, 17(2), e0262661.
37. Kurup, G., & Shetty, S. D. (2022). AI Conversational Chatbot for Primary Healthcare Diagnosis Using Natural Language Processing and Deep Learning. In *Computational Intelligence in Pattern Recognition* (pp. 259-272). Springer, Singapore.
38. Pérez-Ceberos, J. A., Vázquez-Fernández, E., Partida-Herrera, A., & Peña-Ramírez, G. (2022). SVM based Learning System for the Detection of Depression in Social Networks. *Computación y Sistemas*, 26(1).
39. Kharrat, A., & Neji, M. (2020). A System for Brain Image Segmentation and Classification Based on Three-Dimensional Convolutional Neural Network. *Computación y Sistemas*, 24(4), 1617-1626.
40. Rodríguez, J. C. D. J. M., Biswal, R. R., & de la Cruz, E. S. (2019). Algoritmos de aprendizaje automático de vanguardia para el diagnóstico de enfermedades. *Res. Comput. Sci*, 148(7), 455-468.
41. Mejía, M., Buendía-Roldán, I., Mateos-Toledo, H., Estrada, A., Espinoza-Hernández, M., Juárez-Hernández, F., ... & Selman, M. (2016). Primer consenso mexicano sobre fibrosis pulmonar idiopática. *Neumología y cirugía de tórax*, 75(1), 32-51.
42. Barreto-Rodríguez, J. O., Mejía, M. E., & Buendía-Roldán, I. (2015). Panorama actual de la fibrosis pulmonar idiopática en México. *Neumología y cirugía de tórax*, 74(4), 256-261.
43. Kaggle. OSIC Pulmonary Fibrosis Progression. 2, 7
44. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1), 1-48.

45. Jiménez Gaona, Y., Rodríguez-Alvarez, M. J., Espino-Morato, H., Castillo Malla, D., & Lakshminarayanan, V. (2021, July). DenseNet for Breast Tumor Classification in Mammographic Images. In *International Conference on Bioengineering and Biomedical Signal and Image Processing* (pp. 166-176). Springer, Cham.

Geolocalización en Dispositivos Móviles: Una Aproximación al Diseño de Rutas

Erika Rodriguez, Ana C. Zenteno, Carmen Santiago, Leslie Gómez
Facultad Ciencias del Computación, Benemérita Universidad Autónoma de
Puebla, C. 4 Sur 104, Centro histórico de Puebla, 7200, Puebla, Pue,
México.
erikarp28@gmail.com, {ana.zenteno, marycarmen,santiago}@correo.buap.mx,
leslie.gomez@alumno.buap.mx

Resumen. Noche de museos es un evento de la ciudad de Puebla para fomentar el acercamiento a la cultura, sin embargo, tanto el sitio web y página de Facebook de promoción al evento no muestran la información tomando en cuenta al usuario y tampoco le brinda apoyo para planificar sus visita y recorridos. Ante la situación y aprovechando los grandes beneficios que brinda el uso de la tecnología como la geolocalización y las API (interfaz de programación de aplicaciones) de empresas como Google, se desarrolló una aplicación móvil que permite obtener información clara y útil sobre los museos en la Ciudad de Puebla de una forma interactiva y ofreciendo mapas guía y con ello brindar una experiencia satisfactoria en la visita de museos.

Palabras Clave: Aplicación Móvil, Geolocalización, API, Google, Mapas.

1 Introducción

La ciudad de Puebla de los Ángeles, ubicada en el corazón de la República Mexicana, ha experimentado un importante desarrollo en infraestructura cultural en los últimos años. El crecimiento de las bases culturales y la constante apertura de nuevas exposiciones en la capital histórica y artística no hacen más que confirmar la importancia del patrimonio cultural que existe desde la fundación de la ciudad y ha crecido en la última década hasta incluir el imparable paisaje urbano [1].

“La Noche de los Museos” es un proyecto que surge en 2009 por la Secretaría de Cultura de México para atraer nuevos visitantes a través de una combinación de historia, arte y cultura [2]. La ciudad de Puebla es una de las muchas ciudades que participan en este proyecto. Durante este evento, la información se difunde con métodos poco innovadores los cuales no logran reunir un gran número de participantes provocando poco fomento en la cultura. Según lo que se logra observar en la página de Facebook de "Noche de Museos Puebla" las interacciones, los me gusta y los comentarios, no tienen un gran impacto. En la figura 1 se muestra una publicación donde se dan a conocer los museos y los programas que existen, sin embargo, con las interacciones que muestra el post, se concluye que el alcance que tuvo la publicación fue muy bajo para una página que tiene 43,000 seguidores [3].



Fig. 1. Publicación en Facebook Noche de Museos Puebla.

Es importante mencionar que no hay un sitio oficial del programa, en el sitio web oficial Museos Puebla, el cual se puede observar en la figura 2 el contenido digital de los museos tiene un diseño poco llamativo y estático, el sitio web podría ser más atractivo con elementos visuales incluyendo iteraciones con el usuario [4].

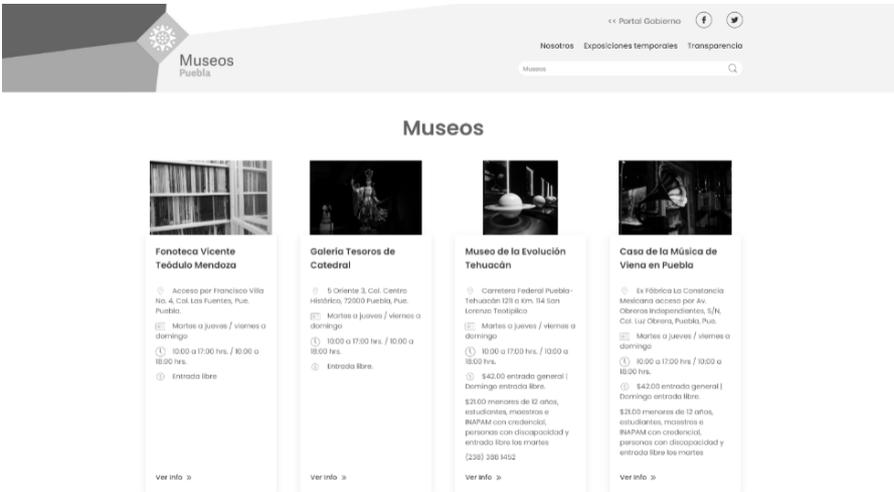


Fig. 2. Sitio Web Museos de Puebla

La propuesta para mejorar el proceso y la experiencia del asistente a al evento es promedio del desarrollo de una aplicación móvil que permita al usuario interactuar y diseñar una ruta a los museos tomando en cuenta su ubicación actual y la de los museos para facilitar la visita y fomentar la cultura promoviendo el turismo.

2 Métodos

La búsqueda de museos en la Ciudad de Puebla debería ser en un único sitio y también precisa para que este a su vez se ofrezca una solución eficiente. Con estas recomendaciones se podría promover el turismo y fomentar la identidad cultural de manera que se pueda mostrar a los turistas de forma directa el patrimonio y la historia que rodea la ciudad.

La aplicación permite encontrar y obtener información de forma unificada, haciendo un esfuerzo para jerarquizar y priorizar los contenidos adecuadamente, evitando que el usuario se pierda buscando la información sobre un museo.

2.1 Requerimientos Funcionales

La aplicación consta de 4 módulos los cuales se detallan a continuación:

- *Visualizar en un mapa los museos*: En este módulo se visualiza el tiempo estimado, la distancia y la ruta dibujada desde el punto origen hasta el museo seleccionado.
- *Visualizar en una lista los museos*: En este módulo se muestra una lista de todos los museos que se encuentran en la ciudad con datos generales como el nombre, la dirección y el rating.
- *Visualizar un museo seleccionado*: En este módulo se puede ver a detalle la información del museo de interés, los datos mostrados son nombre, dirección, rating, si se encuentra abierto o cerrado, la página web, número telefónico, precio y finalmente el tiempo estimado de visita.
- *Itinerario*: En este módulo se muestra el itinerario realizado por el usuario, en el cual podrá visualizar el tiempo estimado que le tomara completarlo y los museos que fueron seleccionados.

2.2 Requerimientos No Funcionales

Los requerimientos no funcionales para esta aplicación son aquellos se refieren a las propiedades emergentes del sistema, tales como:

- *Rendimiento de la Aplicación*: La capacidad de ofrecer respuesta (datos) al usuario en tiempo real.
- *Disponibilidad*: Posibilidad de estar disponible 24 horas al día, los siete días a la semana. Dado que una aplicación nativa se instala en el dispositivo móvil, y además depende de una conexión a internet para el envío y la recepción de información.

2.3 Desarrollo

El primer paso en la fase de diseño fue definir la arquitectura del sistema a alto nivel. Se genera la especificación de la estructura a nivel global del sistema. La arquitectura es de cliente/servidor, ya que existe un cliente (aplicación Android) que solicita y envía datos a la API de Google Maps [5].

La figura 3 muestra los módulos que conforman el alcance del sistema. También los que no lo están, pero que cumplen una función en el ciclo de vida de las partes de trabajo. En este diagrama se distinguen tres módulos distintos.

- *Usuario*: El cliente realiza peticiones y envía información a la API de Google Maps la cual brinda acceso a las bases de datos de *Directions API* y *Places API*.
- *Google Cloud Platform*: Para usar las API de Google Cloud. Google Cloud Platform se encarga de la comunicación entre el usuario (aplicación) y los datos que se encuentran en la base de datos de las API's.
- *Google Maps Platform*: Esta es la base de datos central de la Google Maps Platform contiene la información de lugares, direcciones, rutas, etc.

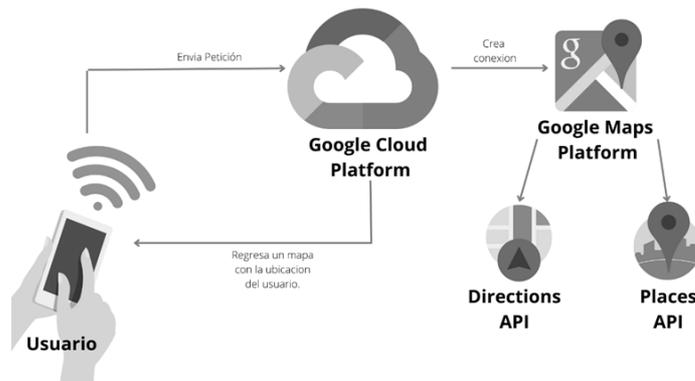


Fig. 3. Arquitectura del Sistema

Para poder darle estructura a nuestra aplicación se tomó en cuenta cuántas pantallas se necesitan, en el diagrama de navegación de la figura 4, se muestra la relación entre las diferentes pantallas que contiene la aplicación *Enpueblate*.

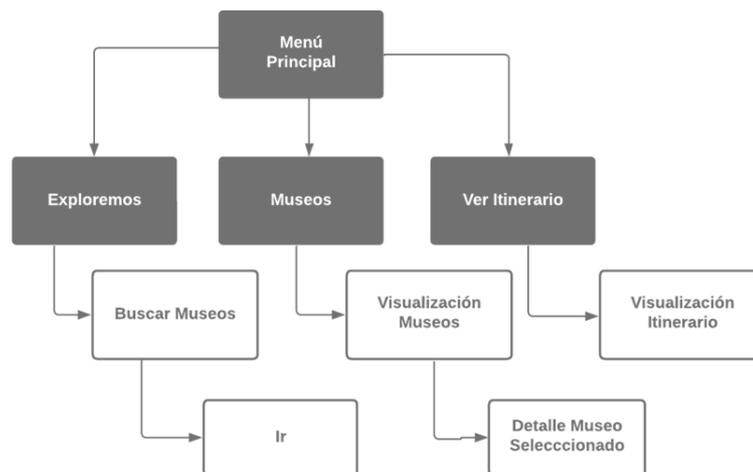


Fig. 4. Diagrama de Navegación

La aplicación cuenta con diferentes módulos, entre ellos, el módulo *Exploremos*, donde se visualiza una pantalla que muestra en un mapa la ubicación actual del usuario y los museos que se encuentran a su alrededor cuando seleccione el botón *Buscar Museos*. El módulo *ruta trazada* para el usuario se puede observar en la figura 5, por medio de una línea que se dibuja desde la ubicación actual del usuario hasta el punto final, que será el museo seleccionado al seleccionar el botón *Ir*.



Fig. 5. Segunda pantalla Exploremos.

En el módulo *Museos*, se muestra en la pantalla los museos que se encuentran en la Ciudad de Puebla, por lo que es posible visualizar una especie de catálogo donde se muestra el nombre, la dirección y el rating de los museos, la idea general se muestra en la figura 6.

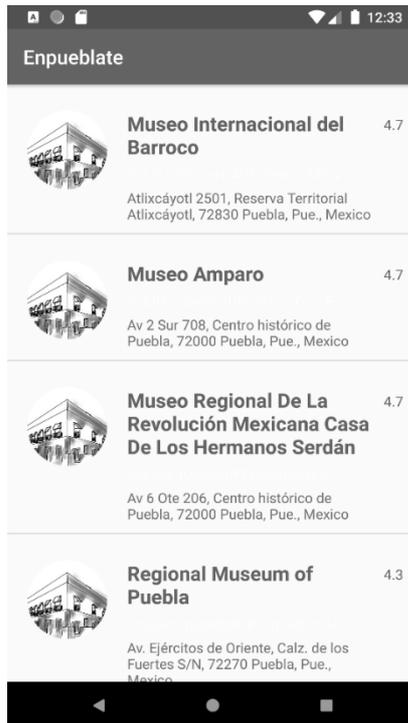


Fig. 6. Pantalla Museos

Una vez seleccionado un museo, se muestra una pantalla donde se visualizan datos específicos sobre el museo en el que se encuentra interesado un usuario. Los datos mostrados son: nombre, dirección, si se encuentra abierto o no, número telefónico, página web, el rating, los costos de entrada y el tiempo estimado que le toma a un usuario visitar el museo. Además, contendrá un botón que le permitirá al usuario añadir ese museo a su itinerario.

En el módulo *Ver Itinerario*, se visualizan los museos que el usuario a agregado a su itinerario y un tiempo total de lo que le tomara cumplir con este.

3 Resultados

En este trabajo se desarrolló una aplicación móvil que puede ser de gran utilidad para uso del turista que visita museos en la ciudad de Puebla. Con la finalidad de incrementar las visitas y fomentar la cultura y el turismo. La metodología de desarrollo resultante se describe en la figura 7.

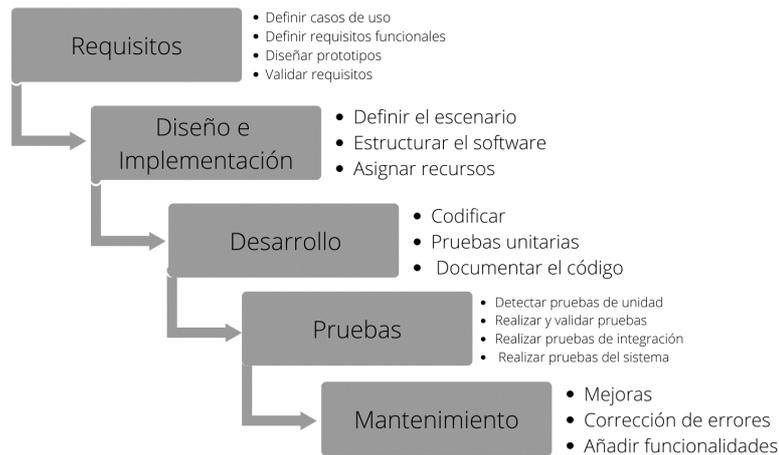


Fig. 7. Metodología de desarrollo de aplicación móvil.

Con los resultados obtenidos del presente proyecto se tiene una propuesta de desarrollo desde la fase definición de Requisitos pasando por el Diseño e Implementación, en este caso con la API de Google para el uso de mapas y rutas, para que el Desarrollo se realicen pruebas unitarias y posteriormente Pruebas mayores y el proceso de Mantenimiento.

La geolocalización por medio del uso de teléfonos inteligentes y aplicaciones móviles ayuda a que el proceso para buscar museos dentro de la ciudad de Puebla sea eficiente. Se ofrece una alternativa para la experiencia de visita de museos y se automatiza el proceso de obtener la información de cada uno. El usuario puede revisar su ubicación con respecto a los museos a su alrededor y llegar a ellos por medio de un trazado en el mapa que la aplicación calcula y ofrece.

4 Conclusiones

El desarrollo de la aplicación móvil fue para Android, dado que es un sistema operativo gratuito (de código abierto), y da a los usuarios la libertad de obtenerlo y usarlo. La implementación en este sistema operativo permitió desarrollar la aplicación integrando herramientas gratuitas, incluyendo también la API de Google Maps.

Las expectativas del presente proyecto se cumplieron en las diferentes etapas, desde el levantamiento de información, la identificación de las necesidades e implementación y posteriormente el desarrollo y pruebas en la aplicación móvil. La importancia del manejo de la API de Google, brinda toda la funcionalidad al responder a mostrar la ubicación actual del usuario y en la generación de rutas a los museos que desea visitar.

Al implementar la aplicación móvil se redujo la pérdida de tiempo que se generaba al buscar información específica sobre cada museo. Ahora la aplicación mejora los

procesos de obtención y visualización de la información contribuyendo a el fomento de la cultura y el turismo en la ciudad.

Se tiene como trabajo futuro realizar implementación de la aplicación en otros sistemas para dispositivos móviles y de esta forma tener un mayor alcance en la población. Además de recabar información sobre el uso de la aplicación para hacer análisis de datos y proponer otros módulos que se puedan integrar y enriquecer la aplicación.

Referencias

1. Fraile, I. (2018) Puebla de los Ángeles: 30 años de patrimonio a través de su política museística. NORBA, Revista de Arte, vol. XXXVIII (2018)
2. Álvarez, A. & Reyes, F. & Rodríguez, J. (2020) Noche de museos en tiempos de COVID-19. Museos Revista Digital, Año 2, No. 2, enero-junio 2020
3. Noche de Museos Puebla (2022) Facebook. <https://www.facebook.com/NochedeMuseosPuebla/>
4. Secretaría de Cultura (2014) Noche de Museos conquista Puebla , Recuperado de <https://www.gob.mx/cultura/prensa/noche-de-museos-conquista-puebla> el día 16 de junio de 2022.
5. Ordorica, I. (2020) ¿Qué es Google Cloud y para qué sirve? Incentro. <https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform>
6. Alfonso, F. & Martínez, L. & Segovia, F. (2005) Introducción a la Ingeniería del software. Delta Publicaciones, 2005.
7. Garita-Araya, R. (2013) Tecnología Móvil: desarrollo de sistemas y aplicaciones para las Unidades de Información. Revista electrónica semestral, Volumen 3, número 2, artículo 2, Julio - diciembre, 2013.
8. ¿Qué es geolocalización? Macrosig. Recuperado de <https://www.macrosig.com/que-es-geolocalizacion> el 9 de junio de 2022
9. Urriolabeytia, J. (2020) Android al máximo. (C, Peña Milahual, Ed) (2020) SIX EDITIONS.
10. Historial de versiones de Android (2014) Wikipedia. http://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android.
11. Prieto, J. & Ramírez, R. & Morillo, J. & Domingo, M. (2011) Tecnología y desarrollo en dispositivos móviles. Material Docente de la UOC. Primera edición: septiembre 2011
12. Google Maps. (s.f.). Universidad Politécnica de Valencia. <http://www.androidcurso.com/index.php/223>

Chatbot de Asesoría Universitaria para el Desarrollo de Aplicaciones Web

Cristopher Quecholac, Judith Pérez, María. del C. Santiago, Gustavo T. Rubín

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 14 Sur s/n, Cd Universitaria, C.P. 72592 Puebla, Pue.
cristopher.quecholac@alumno.buap.mx, {judith.perez, marycarmen.santiago, gustavo.rubin}@correo.buap.mx

Resumen. En el presente artículo se describe el diseño, implementación y pruebas de un Chatbot de asesoría universitaria. El chatbot se implementa bajo un modelo de procesamiento de lenguaje natural (PLN) y es capaz de asesorar a estudiantes en el área de desarrollo Web. Los Chatbot responde a preguntas oportunas de las materias, es una versión de asesoría personalizada. Se implementa a través la metodología Scrum para la gestión de los requerimientos y Power Virtual Agents para la implementación de una interfaz usable, como resultado se tiene un Chatbot que brinda asesoría con el fin de apoyo al trabajo del docente para dar asesorías y seguimiento. El uso de la aplicación alcanza el 35% de los alumnos inscritos en comparativa a la asesoría presencial.

Palabras Clave: Chatbot, PLN, Power Virtual Agents, Asesoría Universitaria.

1 Introducción

La conversación entre el usuario y el bot, basándose en la comunicación directa a través del diálogo. Después de lo anterior se busca establecer una relación de confianza para poder responder a sus preferencias e inquietudes. Estos medios conversacionales dominan la retórica digital a través de un lenguaje natural, que debe ser lo más parecido al de un ser humano: “el potencial de los bots, está en su distribución personalizada y en su poder conversador con las audiencias, mediante aplicaciones de mensajería instantánea”. Los chatbots incorporan cuatro características fundamentales que detallan su forma de comportamiento: la capacidad de actuar de forma autónoma, la automatización total o parcial, requieren de una orientación inicial y se utilizan a través de los medios de comunicación en línea. Además, su sistema de funcionamiento es muy sencillo y cualquier persona está capacitada para manejarlo ya que el programa está preparado para responder a sus preguntas. En general, parte de su atractivo reside en que estos bots están disponibles instantáneamente las 24 horas del día, los 7 días de la semana. El chatbot puede hacer que la información esté habilitada a una escala mayor y también mejora el servicio al cliente, porque proporciona una rápida respuesta al interesado.[1]

Se puede definir a un chatbot de forma rápida como un programa computacional que toma una entrada en lenguaje natural y este devuelve contenido que satisface y/o enriquece dicha entrada. Aunque existe también la opción de llamar a los Chatbots módulos de software, ya que esto implica que la implementación no este dada por solo un lenguaje o paradigmas, más bien que puede acoplarse con otros y complementarse con más funcionalidades. Por último, podemos definir la disponibilidad del chatbot, estos suelen estar disponibles las 24 horas del día, los 7 días a la semana y los 365 días del año.[2]

2 Estructuración de un chatbot

2.1 Introducción a los chatbots

Para empezar el desarrollo del chatbot se requiere elegir qué tipo de chatbot será de acuerdo con las necesidades del proyecto. Dentro de las decisiones a tomar se encuentra la funcionalidad del chatbot, el objetivo, las herramientas, etc. El primer tema por tratar es el tipo de chatbot, consideramos las siguientes dos ramas principales:

- Chatbots orientados a tareas.
- Chatbots basados en datos predictivos.

Los chatbots orientados a tareas tratándolos como módulos son los que se dedican a cumplir una función específica, esto quiere decir que siguen un flujo predeterminado, la principal característica de un chatbot orientado a la tarea es que no se incluye una comprensión del lenguaje natural en cambio se utilizan palabras desencadenadoras que activan eventos para que el chatbot responda a las peticiones.

Los chatbots basados en datos predictivos utilizan métodos relacionados con la inteligencia artificial para predecir datos como respuestas, estos son construidos con modelos para la comprensión del lenguaje natural (NLU) en conjunto del procesamiento de lenguaje natural (NLP) y el aprendizaje automático (ML).

El presente artículo presenta la estructuración y creación de un chatbot orientado a la tarea. El funcionamiento de estos tipos de chatbots están regidos por un dominio en concreto, ya que su escalabilidad suele ser difícil.[3]

2.2 Pasos previos al desarrollo de un chatbot.

Antes de empezar el desarrollo de un chatbot se tiene que considerar tres aspectos importantes:

- Plantear los objetivos del proyecto.
- Definir el desarrollo.
- Seguimiento y escalabilidad del chatbot.

Estos tres aspectos se encadenan para tener un mayor rendimiento al crear el chatbot. Con objetivos claros se encuentra el camino al resultado deseado. Además, de esta manera se conocen las restricciones que se tienen al desarrollar el proyecto.

Para definir el desarrollo se requiere de una planeación teniendo en cuenta su implementación a corto o largo plazo, un chatbot definido para resolver distintas tareas tiene un mayor margen de desarrollo que uno para resolver un problema en específico.

El seguimiento del chatbot se basa en analizar los datos resueltos por el mismo, creando y obteniendo estadísticas esta tarea se vuelve imprescindible para que se tenga una escalabilidad, ya que se atienden las necesidades según grado de importancia. La escalabilidad también está ligada con el desarrollo del chatbot, un chatbot con implementación a largo plazo es susceptible a ser escalable. [4]

2.3 Herramientas para el desarrollo

Existen herramientas que permiten el desarrollo de chatbots como plataformas (web, móviles, etc). El uso de estas herramientas debe ser analizadas ya que la que elijamos debe darnos la facilidad de acceder al público objetivo y completar la meta propuesta algunos se describen en la Tabla 1.

Tabla 1. Tabla comparativa de herramientas para el desarrollo de un chatbot.

Herramienta	Plataforma	Precio/mes (peños mx)
Intercom	Web	\$ 1480
Cliengo	Web/móvil	\$ 0
amoCRM	Web	\$ 300
Power Virtual Agents	Web/ Escritorio	\$ 4220

*Todos los precios fueron elegidos por el plan más básico.

Dentro de nuestro catálogo de herramientas se tiene que discriminar el mayor número de estas, con un catálogo más reducido, la tarea de elección nos hace fijarnos más en las ventajas y desventajas de nuestras herramientas seleccionadas.

3 Desarrollo del chatbot.

3.1 Plataforma para el desarrollo del chatbot

La asesoría universitaria es un espacio de orientación educativa que se cimienta en la experiencia del asesor y en sus condiciones de apoyo académico a estudiantes con diferentes modalidades, se desarrolla un chatbot de asesoría universitaria a través de la plataforma elegida para el desarrollo del chatbot es Power Virtual Agents, la cual es una plataforma desarrollada por Microsoft para el ambiente Teams. La razón de esta elección es por la integración del chatbot a un ambiente ya conocido por los estudiantes, además, se espera que con esto los estudiantes tengan un fácil acceso sin tener que recurrir a otras plataformas. Con esto creamos una herramienta sin tener que acudir a otro entorno, con una fácil instalación dentro de la plataforma Teams y en una plataforma concurrida por los estudiantes.

Power Virtual Agents es un framework para crear chatbots escalables utilizando una interfaz visual. Esta interfaz visual cuenta con temas, graficas de árbol, eventos, variables, entre otros tantos elementos que se verán en la siguiente sección.

Para instalar el entorno se requiere de una cuenta en Teams, es tan simple como ingresar en el buscador de aplicaciones de Teams, Power Virtual Agents.

3.2 Creación del chatbot.

Para crear el chatbot es necesario conocer la estructura que este tiene, una vez iniciado el proceso, Power Virtual Agents muestra una interfaz con un menú y elementos útiles, los cuales son:

- Inicio: Página de inicio donde podrá ver los chatbots que ha creado.
- Temas: Parte del chatbot que se encargara de resolver los temas.
- Entidades: variables utilizadas para guardar informacion.
- Análisis: Datos acerca del rendimiento del chatbot.
- Publicar: Publicar el contenido más reciente del chatbot.
- Administrar: Opciones acerca del chatbot (eliminar, renombrar, etc).

Para administrar las entidades es necesario crear un diagrama de acciones donde se incluyen los temas los cuales son activados mediante frases desencadenadoras, es decir, dada cierta frase o grupos de frases, el tema que contenga esas frases desencadenadoras se activara desencadenando una serie de eventos programados, todo esto siguiendo una estructura ramificada como se muestra en la Figura 1.

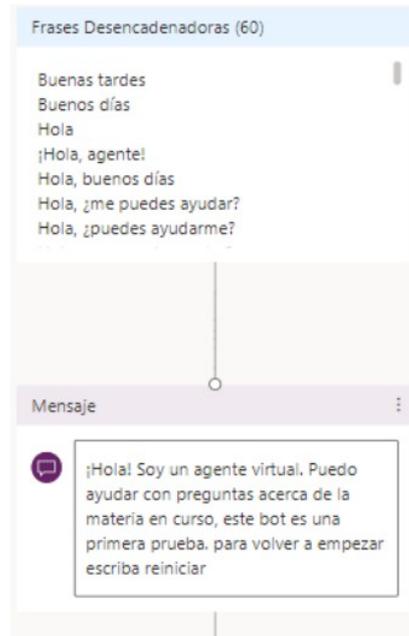


Fig. 1. Frases desencadenadoras de un tema.

Después de creado el diagrama de acciones es necesario definir la ruta de flujo de la información, para eso podemos crear preguntas agregar condiciones ó llamar a otras acciones. Con el flujo de la información ya creado, el siguiente paso es crear más temas, conectarlos con la opción ir a otro tema y estructurar el chatbot.

Una vez listo el chatbot el último paso es publicar el chatbot, dentro del apartado de publicar podemos encontrar varias opciones dentro de las cuales podemos elegir añadirlo a algún equipo de trabajo en Teams.

3.3 Exportación e importación del chatbot.

Para la exportación e importación del chatbot se requiere del uso externo de un manejador que administra las aplicaciones en Teams, esta aplicación se llama Power Virtual Apps, se considera esta herramienta para exportar e importar los chatbots.

4 Resultados.

Se logro crear un chatbot que contiene 6 unidades con material didáctico en cada una de estas unidades. Cada unidad contiene un flujo de conversación diferente, la cual contiene información asociada a cada unidad de contenido, además de apoyo complementario de cada uno de los temas y subtemas. De esta forma el alumno puede navegar entre unidades y buscar la información que requiera como se muestra en la figura 2.

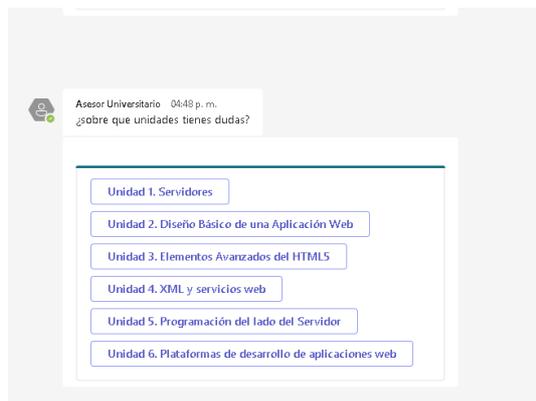


Fig. 2. Chatbot de asesoría universitaria de Desarrollo de Aplicaciones Web

Se emplea también una rama para poder contactar al docente, de esta forma existe un feedback para aclarar mayores dudas sobre la materia en curso como se muestra en la figura 3.

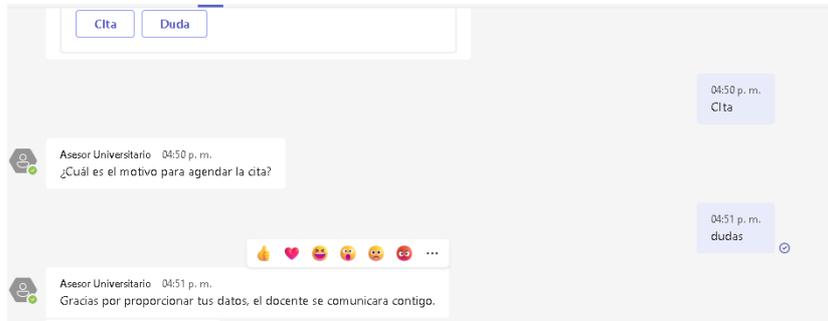


Fig. 3. Agenda solicitud asesoría personal.

El chatbot es apoyo a la asesoría académica del desarrollo de aplicaciones web ya que permite un fácil acceso a la información de los contenidos del curso por lo que la asesoría incrementa en un 35% como se muestra en la figura 4. de la población a través de esta propuesta. Por lo que se puede concluir que es un sistema de apoyo para los estudiantes universitarios ya que pueden consultarlo 24/7/365.

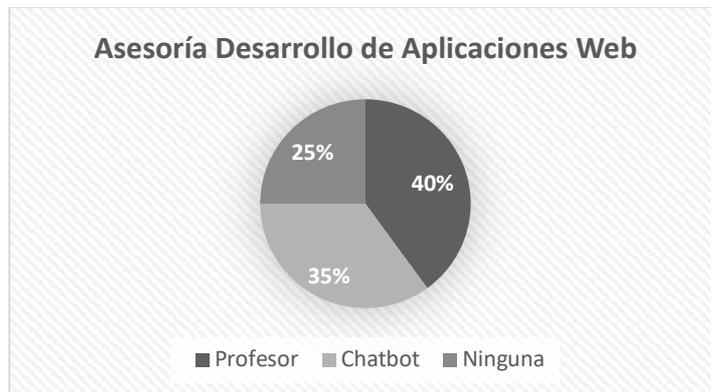


Fig. 4. Estadística de asesoría del Desarrollo de Aplicaciones Web

5 Conclusiones

El desarrollo de chatbots está ligado a la automatización de tareas, estas pueden ser resueltas de una u otra forma según el paradigma que proponamos. Un chatbot para la asesoría universitaria es ideal ya que automatiza la resolución de las peticiones de los alumnos (dudas), sin embargo, se debe tomar en cuenta acerca de lo que es el entendimiento para un chatbot, ya que este puede ser nulo o con algoritmos que procesen algún tipo de lenguaje. En este caso el chatbot logra demostrar una buena fluidez en la conversación, además, la herramienta utilizada proporciona otras herramientas externas para complementar la funcionalidad de este chatbot, tales como el envío de correos electrónicos para agendar citas con el maestro en curso.

En conclusión, el chatbot para asesoría universitaria cumple sus funciones como chatbot orientado a la tarea, dándole gran ventaja al docente para resolver dudas y agendar citas para asesorías.

Referencias

1. Brandtzaeg P., Folstad A. (2017). Why People Use Chatbots. Internet Science. doi.org/10.1007/978-3-319-70284-1_30.
2. Sánchez-Gonzales, Hada M.; Sánchez-González, María (2017). Los bots como servicio de noticias y de conectividad emocional con las audiencias. El caso de Politibot. *Doxa comunicación*, 25, 63-84.
3. Chatbot web. (s. f.). Cliengo. Recuperado 1 de julio de 2022, de <https://www.cliengo.com/chatbot>.
4. Gogovski, D. (2020, 16 noviembre). Cómo crear un chatbot para Whatsapp CRM. amoCRM. Recuperado 1 de julio de 2022, de <https://www.amocrm.com/es/blog/chatbot-para-whatsapp-crm/>.
5. Daura, F. T. (2010). La asesoría académica universitaria: un espacio propicio para la promoción del aprendizaje autorregulado. *Revista de orientacion educacional*, 47, 49-63. <https://dialnet.unirioja.es/descarga/articulo/3860048.pdf>

Análisis de los Modos de Ahorro de Energía en Microcontroladores Usados en Sistemas Embebidos Basados en Tecnología IoT

Juan Mora Ramírez, Nicolás Quiroz Hernández, Ana María Rodríguez Domínguez

Benemérita Universidad Autónoma de Puebla, 18 Sur y Avenida
San Claudio S/N, San Manuel, Puebla, México.
juan.morar@alumno.buap.mx, nicolas.quirozh@correo.buap.mx,
anamaria.rodriguez@correo.buap.mx

Resumen: El trabajo analiza los resultados de las pruebas realizadas a diferentes microcontroladores de 8 y 32 bits para medir el consumo de corriente en los diferentes modos de ahorro de energía y en el modo de datos de los fabricantes, se analizan microcontroladores ARM Cortex-M de 32 bits. Utilizando herramientas de laboratorio como multímetro de banco para poder medir las corrientes que consumen los microcontroladores que son del orden de los micro o nano amperios. Se evalúa el funcionamiento mediante la implementación de algoritmos simples de prueba y se muestra el análisis de los resultados obtenidos. Estos resultados son necesarios para el desarrollo de sistemas empotrados, sobre todo en aplicaciones de IoT.

Palabras claves: Ahorro de energía · ARM · *Deep Sleep* · Internet de las Cosas · Microcontroladores.

1 Introducción

El concepto de Internet de las Cosas (IoT) surgió en la primera década de este siglo, actualmente en el mundo existe aproximadamente 35.82 millones de dispositivos IoT, y se estima que para el año 2025 existan 75.44 millones de dispositivos [1,2]. Los dispositivos IoT se pueden encontrar en los relojes inteligentes, asistentes de voz, hasta en procesos industriales.

El Internet de las Cosas, es una red de objetos que se encuentran conectados a internet, estos objetos recopilan e intercambian datos utilizando sensores integradas en ellas [2], como los dispositivos IoT son cada vez más inteligentes requieren más memoria y mejores procesadores para realizar sus tareas [3], esto conlleva a que el microcontrolador tenga un mayor consumo de energía. Es importante que los microcontroladores tengan diferentes modos de ahorro de energía para disminuir su consumo y prolongar la carga útil de la batería de los dispositivos.

Los dispositivos IoT se pueden tener ubicados a distancias de decenas o centenas de metros entre ellos y se espera que funcionen en un entorno con restricciones de consumo energía. En ese contexto, se necesita un consumo de energía ultra bajo para que el funcionamiento del sistema aumente la vida útil de las baterías [4].

Se estima que se requieren 27 millones de reemplazos de baterías todos los días, incluso si los dispositivos IoT funcionan con baterías que puedan durar 4 años [5,6]. El constante reemplazo de baterías ocasiona un impacto ambiental negativo. Se generan problemas ambientales y de salud, debido a los materiales tóxicos que contienen las baterías y al inadecuado manejo de las mismas cuando se desechan [7]. La importancia de seleccionar un microcontrolador y componentes de bajo consumo es crucial para cualquier sistema embebido, esto permitirá que el dispositivo no sea desechado de forma inmediata y las baterías no sean cambiadas constantemente.

1.1 Modos de ahorro de energía

Los microcontroladores pueden administrar el manejo de sus módulos internos, así como de los dispositivos externos para hacer más eficiente el consumo de energía. Un

dispositivo en modo de ejecución normal polarizado con una batería podría estar funcionando durante un par de meses. [8,9] El mismo dispositivo con técnicas apropiadas y una óptima configuración de sus recursos en modo operación normal y en reposo, reduce el consumo de energía y así se extiende la carga útil de la batería [10].

Diversos fabricantes de microcontroladores cuentan con características de ahorro de energía integradas. Dependiendo del modelo y de la marca de los microcontroladores puede manejar diferentes tipos de operación, en el caso de STMicroelectronics, sus 3 principales modos de operación son:

Sleep. (Suspensión.) En el modo *Sleep*, solo se detiene el núcleo mientras todos los periféricos continúan funcionando no es necesario disminuir la frecuencia del sistema y todos los periféricos del dispositivo están disponibles para su uso. Cualquier interrupción o evento disponible en el modo de ejecución puede reactivar el dispositivo.

Stop. (Detenido.) En el modo *Stop*, el núcleo se detiene y los únicos osciladores que pueden funcionar son el LSE (Reloj externo de baja velocidad), LSI (Reloj interno de baja velocidad) y el HSI (Reloj interno de alta velocidad) en una capacidad limitada. Los relojes de baja velocidad permiten que el RTC (Reloj de tiempo real) y el IWDG (Perro guardián) continúen funcionando y activen el dispositivo. El HSI puede proporcionar una funcionalidad limitada a los periféricos capaces de funcionar en modo de parada.

Standby. (Espera.) A diferencia del modo de *Stop*, entrar en el modo de *Standby* es bastante sencillo porque el usuario tiene menos opciones. Los únicos osciladores disponibles son LSI y LSE, los únicos periféricos que pueden funcionar son RTC e IWDG, el regulador de voltaje está completamente deshabilitado y todos los pines de E/S están configurados como de alta impedancia.

El microcontrolador ESP32-WROOM tiene 4 modos de ahorro de energía:

Sleep. En el modo de *Sleep*, todo está activo excepto WiFi, Bluetooth y radio. También mantiene la CPU en funcionamiento y el reloj es configurable. En este modo el chip consume alrededor de 3 mA a baja velocidad y 20 mA a alta velocidad.

Light Sleep. (Suspensión Ligera.) El *Light Sleep* es similar al *Sleep*. La única diferencia es que, durante el modo de *Light Sleep*, los periféricos digitales, la mayor parte de la RAM y la CPU, están controlados por reloj. Esto da como resultado un menor consumo de energía que el modo de suspensión del módem, que es de alrededor de 0,8 mA.

Deep Sleep. (Suspensión Profunda.) En el modo de *Deep Sleep*, las CPU, la mayor parte de la RAM y todos los periféricos digitales están apagados. Las únicas partes del chip que permanecen operativas son: Coprocesador ULP (Potencia Ultra-baja), Controlador RTC, Periféricos RTC, Memoria RTC rápida y lenta, El chip consume alrededor de 10 μ A.

Hibernation. (Hibernación.) A diferencia del modo de *Deep sleep*, en el modo de hibernación, el chip desactiva el oscilador interno de 8 MHz y el coprocesador ULP. La memoria de recuperación RTC también está apagada, lo que significa que no podemos conservar ningún dato mientras estamos en modo de hibernación. En este modo el chip consume 5 μ A.

En el modo *Sleep* de un microcontrolador PIC se coloca en su estado de consumo de corriente más bajo. El dispositivo entra en modo de suspensión al ejecutar una instrucción SLEEP. El oscilador del dispositivo está apagado, por lo que no se producen relojes de sistema en el dispositivo. Sin embargo, los puertos de E/S mantienen el estado que tenían antes de que se ejecutara la instrucción SLEEP. Por lo tanto, con el fin de minimizar el consumo de energía en el modo de *Sleep*, los puertos de salida no deben generar ni absorber la corriente antes de pasar al modo de suspensión.

1.2 Bajo consumo de energía

El concepto de bajo consumo de energía varía de una a aplicación a otra, para este experimento lo que se quiere lograr es reducir al máximo la energía consumida por el

microcontrolador. En algunos dispositivos IoT, el consumo de energía no es relevante en su aplicación, sin embargo, los dispositivos que la energía es suministrada con baterías es necesario optimizar el consumo para prolongar la carga útil del dispositivo.

En los dispositivos electrónicos que utilizan tecnología CMOS, como los microcontroladores y microprocesadores, la energía total que se consume puede dividirse en dos rubros: la energía dinámica y estática. La energía dinámica es cuando el dispositivo está ejecutando alguna rutina del proceso. La energía estática es la que consume el dispositivo al estar en reposo y energizado [11].

La energía dinámica incluye la pérdida de energía cuando los circuitos CMOS conmutan y las corrientes que polarizan los circuitos analógicos (osciladores, ADC, DAC, entre otros). Dicha energía se calcula para cada uno de los circuitos de la siguiente manera:

$$P = V \cdot \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \cdot I_{OH}\} + \sum (V_{OL} \cdot I_{OL}) \quad (1)$$

donde P es la potencia de consumo del microcontrolador, V es el voltaje de operación (VDD), e IDD es la corriente de consumo del microcontrolador, [11, p. 2].

En la Fórmula 1, Se aprecia la potencia que está relacionada directamente con la suma de las corrientes de entrada y salida del microcontrolador. Si la corriente IDD aumenta, la potencia del microcontrolador también. Por consecuencia el consumo de energía se eleva.

1.3 Tarjetas de desarrollo

Una tarjeta de desarrollo es un circuito electrónico que contiene un microcontrolador o dispositivo lógico y elementos como puertos, conectores y reguladores que les permita a los usuarios acceder fácil y rápidamente a los periféricos para realizar aplicaciones y pruebas.[12].

Las marcas como STMicroelectronics, Espressif, y Microchip, tienen un gran número de familias de microcontroladores tanto de 8, 16 y 32 bits, en este trabajo se utilizaron las siguientes tarjetas de desarrollo:

- Nucleo-F302R8
- Nucleo-F429Z1
- STM32F746G-Discovery
- Nucleo-L432KC
- Nucleo-L011k4
- PIC18LF4550
- PIC16f887
- ESP-01 ESP32
- NodeMCU

Estas tarjetas se encuentran disponibles en el laboratorio SLED de la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla.

2 Desarrollo experimental de la medición de corriente de un microcontrolador

En la elaboración de estas pruebas, se utilizó un multímetro digital de precisión de 6.5 dígitos, marca Keysight Truevolt modelo 34460A, ofrece un nivel alto de exactitud, y tiene la capacidad de medir pA en la resolución de 100 μ A [13].

Tabla 1. Características básicas del multímetro 34460A

Características	34460A
Resolución	6.5 dígitos
Exactitud Básica VCD	75 ppm
Tasa de lectura	300 lecturas/s
VCD	100 mV - 1000 V
VCA	100 mV - 750 V
Corriente CA, CD	100 μ A - 3 A

También se utiliza una fuente de alimentación modelo 2231A-30-3, tiene una potencia de salida máxima de 195 W proporcionando los niveles de potencia necesarios para este experimento. Tiene un nivel de ruido de menos de 5 mVpp y los ajustes de voltaje y corriente tiene una precisión de 0.06 % y 0.2 % [14].

2.1 Medición de corriente

La configuración para la medición de corriente de los microcontroladores se muestra en la Figura 1. La metodología consta de la medición de corriente usando un multímetro de banco, esta medición consta de dos etapas. En la primera etapa, el microcontrolador ejecuta un código con funciones aritméticas enteras. La segunda etapa entra en modo de reposo. El microcontrolador utiliza el RTC interno para reiniciar su proceso cada 5 segundo. Se realizan mediciones con los osciladores internos en baja, mediana y alta frecuencia de operación.

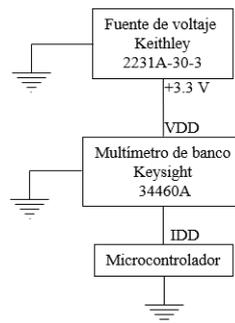


Fig. 1. Configuración de la medición de energía.

2.2 Diseño del Hardware para la medición de corriente

Las tarjetas de ST contienen un puente para hacer la medición del consumo de corriente del microcontrolador, este puente está indicado en la PCB como “IDD” y en el esquemático se localiza como “JP6” mostrado en la Figura 2.

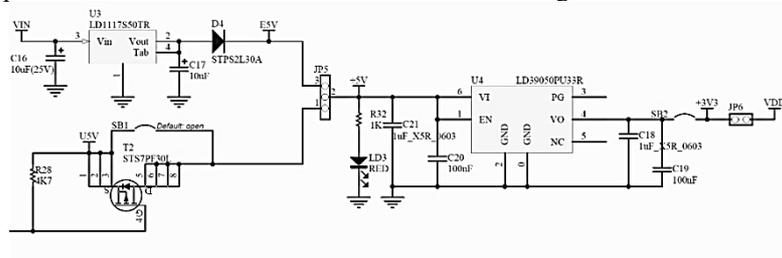


Fig. 2. Circuito de alimentación de la tarjeta Núcleo STM32F302R8.

Al realizar la medición de corriente de los microcontroladores de Microchip fue necesario utilizar un regulador de 3.3 V, y un voltaje de entrada con la fuente de

alimentación de 4.2 V simulando el voltaje de una batería de Li-PO que comúnmente se utiliza en dispositivos IoT.

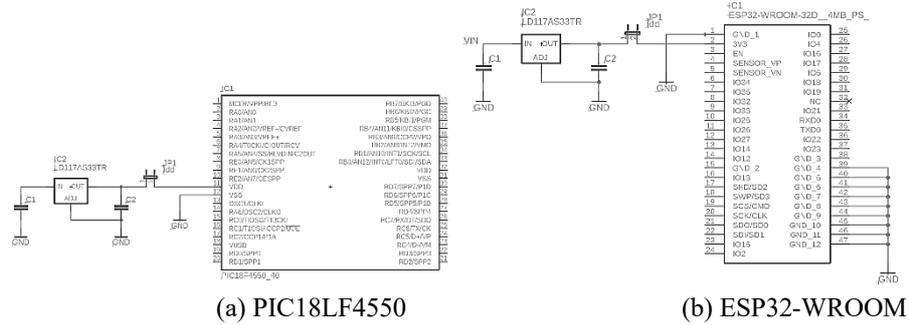


Fig. 3. Esquemático para medir la corriente.

En el esquemático de la Figura 3a, se puede observar que el puente “JP1” es el punto donde se midió la corriente. Los dispositivos ESP8266EX y ESP32-D0WD-V3 tienen un voltaje de operación nominal de 3.3 V, se utilizó un regulador de voltaje para alimentar el dispositivo y poder realizar las pruebas de consumo de corriente. En la Figura 3b, se muestra que en el *jumper* “JP1” se realizó la medición de corriente de los dispositivos.

2.3 Implementación de un algoritmo para determinar la corriente de consumo de un microcontrolador

La implementación del algoritmo de modo ahorro de energía para microcontroladores ST, se utiliza el software STM32CubeIDE versión 1.9.0, este software usa lenguaje de programación en C, C++ y está basado en el estándar CMSIS y bibliotecas HAL.

Se utilizó CCS C Compiler versión 5.091 para la implementación del modo ahorro en microcontroladores PIC de Microchip, este software utiliza lenguaje de programación basado en C.

En el caso del microcontrolador ESP8266EX se utilizó el IDE de Arduino versión 1.8.16, este IDE usa lenguaje de programación basada en C y C++.

En la Figura 4a, se describe en el diagrama de flujo el programa que se utilizó para realizar la medición de corriente en modo *Sleep*, el programa principal ejecuta un parpadeo de led, cuando se presiona el botón de *Sleep* el microcontrolador se va al modo de ahorro de energía que esté configurado, y este se quedará en *Sleep* hasta que se reinicie el microcontrolador. Para hacer la medición de corriente para el modo normal de ejecución, el microcontrolador ejecutará una suma hasta que el resultado sea 0xFFFFFFFF, después se irá al modo de ahorro de energía y se esperará 5 segundos para volver a reiniciar la suma. Con este programa mostrado en la Figura 4b, podremos medir tanto el consumo de corriente en modo normal de ejecución y el modo de ahorro de energía.

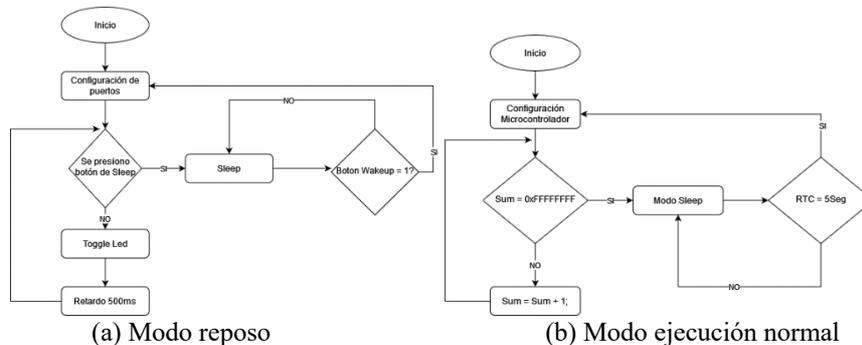


Fig. 4. Algoritmos para determinar la corriente de consumo.

3 Resultados

Las mediciones en este trabajo se realizaron en el laboratorio SLED de la Benemérita Universidad Autónoma de Puebla, se validó el funcionamiento en modo normal de ejecución con el algoritmo mostrado en la Figura 4b, se utilizó la fuente de alimentación configurada a 3.3 V, se midieron las corrientes con el multímetro en la escala de mA y μ A, los resultados obtenidos se muestran en las Tablas 2, 3, 4. Es posible medir las corrientes bajo una misma unidad de medición para poder compararlas.

En la Figura 5a, se observa la medición de corriente del microcontrolador ESP32-D0WD-V3 en modo de ejecución normal, tiene un consumo aproximadamente de 56.5 mA. Para los siguientes microcontroladores se repite el procedimiento cambiando la frecuencia máxima de operación, los resultados se registraron en la Tabla 2.

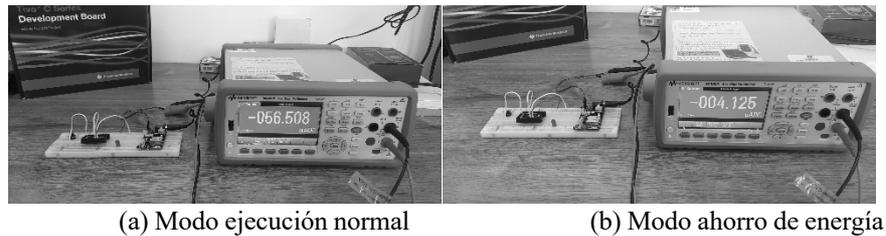


Fig. 5. Medición de corriente en sus diferentes modos

En la Tabla 3, se observan los resultados en modo de ejecución normal con una frecuencia de operación baja. La frecuencia baja de operación se selecciona a partir de las características que el fabricante especifica.

Tabla 2. Consumo de corriente en frecuencia máxima de operación

VDD = 3.3 V; Ta = 26 °C; Multímetro = 34460A;

	Frecuencia máxima del procesador (MHz)	Consumo de corriente (mA)
STM32F302R8	72	25.73
STM32F429ZI	180	76.48
STM32F746NG	216	129.71
STM32L432KC	80	10.42
STM32L011k4	32	6.13
PIC18F4550	48	31.05
PIC16F887	8	4.25
ESP8266EX	80	63.65
ESP32-D0WD-V3	160	56.55

Para los resultados del modo de ahorro de energía, los microcontroladores se programaron con el algoritmo de la Figura 4a, en la salida de los microcontroladores se eliminó la salida del led, se configuró el dispositivo para deshabilitar sus periféricos y así evitar mediciones erróneas. En la Figura 5b, se observa la medición de corriente del microcontrolador ESP32-D0WD-V3 en modo de hibernación, la corriente que consume es de 4.12 μ A. En la Tabla 4, se observan los resultados de los modos de ahorro de energía de los microcontroladores seleccionados para este trabajo.

Tabla 3. Consumo de corriente en frecuencia mínima de operación

VDD = 3.3 V; Ta = 26 °C; Multímetro = 34460A;

	Frecuencia baja del procesador (kHz)	Consumo de corriente (μ A)
STM32F302R8	125	840.65
STM32F429ZI	2 MHz	2.85

STM32F746NG	25 MHz	11.98
STM32L432KC	16	133.1
STM32L011k4	32	40.3
PIC18F4550	32	35.89
PIC16F887	31	
ESP8266EX	-	-
ESP32-D0WD-V3	-	-

Tabla 4. Consumo de corriente en los diferentes modos de ahorro de energía.

VDD = 3.3 V; Ta = 26 °C; Multímetro = 34460A;

	Sleep (mA)	Light Sleep (mA)	Deep Sleep (μ A)	Hibernation (μ A)	Stop (μ A)	Stop RTC on (μ A)	Standby (μ A)
STM32F302R8	8.5	-	-	-	9.53	10.58	5.8
STM32F429ZI	17.8	-	-	-	294	356	2.6
STM32F746NG	34.7	-	-	-	263	324	22
STM32L432KC	2.18	-	-	-	108	171	0.9
STM32L011k4	1.41	-	-	-	0.34	0.62	0.2
PIC18F4550	5.71 μ A	-	-	-	-	-	-
PIC16F887	3.6 μ A	-	-	-	-	-	-
ESP8266EX	16.7	0.42	18	-	-	-	-
ESP32-D0WD- V3	18.4	0.76	13.6	4.1	-	-	-

4 Conclusiones

El consumo de energía de un microcontrolador dependerá del modo de operación en el que se encuentre, así como el de la frecuencia en la que se esté operando. A medida que la frecuencia aumenta, también lo hace la corriente. El consumo de energía no es lineal, esto quiere decir que, si aumentamos la frecuencia al doble, la corriente no lo hará en la misma proporción. Esto se debe a que la corriente dependerá del tiempo en el que el microcontrolador realice una tarea. Si el dispositivo se encuentra en modo de ahorro de energía deshabilitando sus dispositivos internos y permanece en este estado en la mayor parte del tiempo, el consumo de energía puede prolongar la carga útil de la batería hasta por años. La metodología utilizada en este trabajo para la medición de corriente es aplicable a cualquier microcontrolador de 8, 16, o 32 bits, ya que se ocupan instrumentos que comúnmente se encuentran en los laboratorios donde se utilizan microcontroladores. Para trabajos futuros, se pretende evaluar el consumo de corriente, al realizar cambios de la frecuencia, mientras se ejecuta el código, así como evaluar el consumo antes y después de entrar en un ciclo infinito (*for* y *while*).

Referencias

1. Jack, M.: Más de 21 estadísticas, Hechos y Tendencias de Internet de las cosas para 2022, <https://findstack.com/es/internet-of-things-statistics/>.
2. Li, X., Xu, L.D.: A review of internet of things—resource allocation. *IEEE Internet of Things Journal*. 8, 8657–8666 (2021).
3. Watts, S.: *The internet of things (IOT): Applications, technology, and privacy issues*. Nova Publisher's, New York (2016).
4. Shafiee, N., Tewari, S., Calhoun, B., Shrivastava, A.: Infrastructure circuits for lifetime improvement of Ultra-Low Power IOT devices. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 64, 2598–2610 (2017).
5. Suciú, G., Petrache, A.L., Badea, C., Buteau, T., Schlachet, D., Durand, L., Landez, M., Hussain, I.: Low-power IOT devices for measuring environmental values. 2018 IEEE 24th

- International Symposium for Design and Technology in Electronic Packaging (SIITME). (2018).
6. Teresa Penella, M., Albesa, J., Gasulla, M.: Powering wireless sensor nodes: Primary batteries versus energy harvesting. 2009 IEEE Instrumentation and Measurement Technology Conference. (2009).
 7. Cortina, S., Dominguez, J., Cotler, H.: La contaminación por pilas y baterías en México. , <https://www.redalyc.org/pdf/539/53905501.pdf>. Último Acceso 6 Jul 2022
 8. Belvedere, B., Bianchi, M., Borghetti, A., Nucci, C.A., Paolone, M., Peretto, A.: A microcontroller-based power management system for standalone microgrids with hybrid power supply. *IEEE Transactions on Sustainable Energy*. 3, 422–431 (2012).
 9. Luo, G., Guo, B., Shen, Y., Liao, H.Y., Ren, L.: Analysis and optimization of embedded software energy consumption on the source code and algorithm level. 2009 Fourth International Conference on Embedded and Multimedia Computing. (2009).
 10. Sridhara, S.R.: Ultra-low power microcontrollers for Portable, wearable, and implantable medical electronics. 16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011). (2011).
 11. Ive, B.: AN1416, low-power design guide - microchip technology, <https://ww1.microchip.com/downloads/en/Appnotes/90001416a.pdf>.
 12. Castro, C.D.P., Osorio, S.R.: Tarjetas de Desarrollo: Herramientas para el diseño. *Letras ConCiencia Tecnológica*. 11, 43–52 (2018).
 13. Digital multimeters 34460A, 34461A, 34465a, <https://www.mouser.com/datasheet/2/878/5991-1983EN-1143116.pdf>. Último Acceso 6 Jul 2022
 14. 2231A-30-3 195w triple Channel DC power supply, <https://www.farnell.com/datasheets/1884835.pdf>. Último Acceso 6 Jul 2022.

Arquitectura de Microservicios: Caso de Estudio

Servicio de RUTA en la Ciudad de Puebla

Pilar Hernández Zambrano, Judith Pérez Marcial, Gustavo T. Rubín Linares, Ana
Claudia Zenteno Vázquez

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 14 Sur s/n, Cd Universitaria, C.P. 72592 Puebla, Pue.
pilar.hernandez@alumno.buap.mx {judith.perez, gustavo.rubin,
ana.zenteno}@correo.buap.mx

Resumen. El sistema de Red Urbana de Transporte (RUTA) a través de 3 líneas da servicio a la Población de la Ciudad de Puebla, así como foránea. Se realiza el análisis de un sistema microservicios donde se identifican servicios los cuales se comunican entre ellos y se agrupan implementado en HTML5, Javascript, CSS en un contenedor Docker, para realizar su implementación a través de la metodología de SCRUM y ejecución del plan en Sprint. Como resultado se obtiene un sistema con un rendimiento en tiempo de despliegue y de actualización con la arquitectura de microservicios. Con los servicios independientes, se definen mecanismos de comunicación rápida para la interacción entre ellos, el resultado es que cada componente se toma como una unidad independiente, lo que sus operaciones no afectan a las demás, garantizando alta disponibilidad.

Palabras Clave: Microservicios, Arquitectura, Método, funcionalidad, sistema, Servicio, implementación

1 Introducción

Los microservicios cambian la lógica de negocio en un repositorio único, lo que hace que el componente crezca en tamaño y complejidad de implementación. Lo que es importante identificar que tan pequeño debe ser, además de que número de líneas de Código es problemática, pues cada lenguaje de programación tiene diferentes sintaxis. Jon Evans en RealEstate.com.au en Australia caracteriza los microservicios como algo que pueda ser reescrito en dos semanas. [1]

El agregar una nueva funcionalidad tiene por consecuencia el incremento del código, entonces al ir agregando cada vez más funcionalidades, con el tiempo se vuelve complicado deducir donde es que se debe realizar un cambio.

Es aquí donde tienen relevancia los microservicios ya que al tener un enfoque de servicios más ligeros es más fácil lograr que el desarrollador se familiarice con el producto y por ende sea factible el realizar un cambio.

1.1 Microservicios

Las ventajas de la gestión de los procesos como microservicios es que se pueden publicar aplicaciones grandes como un conjunto de módulos de aplicaciones (microservicios) que se pueden desarrollar, implementar, escalar, manejar y visualizar de forma independiente[2]. Los microservicios les dan a las empresas la posibilidad de manejar aplicaciones con una base de código que sea grande ya que al utilizar esta metodología que es más práctica permite que las mejoras sean realizadas por pequeños equipos con despliegues independientes.

La rapidez, los costes bajos y la escalabilidad granular, son retos de los sistemas distribuidos y las prácticas de gestión de los equipos de desarrollo que deben ser

considerados. Según Richards, hay diferentes formas de llevar a cabo un patrón de arquitectura de microservicios, pero se destacan tres topologías principales que son las más comunes y populares: basada en API REST, basada en aplicaciones REST y la centralizada de mensajería [2].

Aplicación monolítica. Una aplicación monolítica se destaca porque todas o la mayoría de sus funcionalidades se ejecutan en un único servidor de aplicaciones. Esta es extensa y se construye por diversos equipos, el inconveniente de esta arquitectura es que hay que tener mucho cuidado a la hora de realizar un cambio ya que requiere de una moderada organización de la ejecución. Además, existen servicios de API que en realidad son una aplicación web grande, estas son tomadas en cuenta como aplicaciones monolíticas. En ambos la arquitectura de microservicios puede ser una opción. Arquitectura de microservicios [2]

Tabla 1. Comparativa entre Arquitectura Monolítica y

Categoría	Arquitectura Monolítica	Arquitectura de Microservicios
Código Fuente	Código centralizado para toda la aplicación	Múltiples módulos de código, cada microservicio cuenta con su propia base.
Comprensión	Difícil de Administrar.	Fácil lectura y de fácil mantenimiento
Lanzamiento Lenguaje	Implementación compleja. Desarrollado en un solo lenguaje de programación	Implementación sencilla Cada microservicio puede ser desarrollado en lenguaje de programación diferente
Escalamiento	Se requiere escalar la aplicación completa	Se pueda escalar como un nuevo servicio por modulo.

2 Preparación de la contribución

Las organizaciones como Amazon, eBay y Netflix, implementan sus servicios a través del patrón de arquitectura de microservicios. En lugar de crear una sola aplicación monolítica monstruosa, la idea es dividir su aplicación en un conjunto de servicios interconectados más pequeños”. [4]

Regularmente en un servicio se pueden implementar un grupo de varias funcionalidades, como por ejemplo la gestión de empleados, la gestión de inventario, etcétera.

Cada microservicio es una miniaplicación que tiene su propia arquitectura que consta de lógica comercial junto con varios complementos. Algunos microservicios expondrían una API que consumen otros microservicios o los clientes de la aplicación. Otros microservicios pueden implementar una interfaz de usuario web. En tiempo de ejecución, cada instancia suele ser una máquina virtual en la nube o un contenedor.

2.1 Arquitectura de microservicios para estandarizar la RUTA en la ciudad de Puebla

El Plan de Desarrollo de arquitectura microservicio, describe el plan usado para el desarrollo del “Servicio Web estandarizado para la RUTA en la ciudad de Puebla”. El detalle de las iteraciones individuales se describe en la pila de cada sprint, se pueden visualizar las actividades a realizar en cada una de ellas.

Para comenzar con el Plan de Desarrollo del Software, tomamos como base los datos proporcionados por la página de RUTA y por medio de documentos brindados por otras dependencias de la CDMX, las cuales nos dan información y estadísticas más estudiadas que las de la Ciudad de Puebla, con las que se realizara un análisis.

El servicio de la Red Urbana de Transporte Articulado no muestra información suficiente sobre la logística para su correcto uso, con testimonios anónimos de

ciudadanos poblanos asegurando que no saben cómo pagar el servicio, como adquirir una tarjeta para el mismo, así como los horarios, además las rutas de las “alimentadoras” no ofrecen información a los usuarios de sus paraderos ni de sus diferencias con respecto al Metrobús principal. El la arquitectura microservicios planteada proporciona a los usuarios un catálogo completo de servicios para el sistema RUTA de pago a través de la aplicación, seguimiento de las unidades móviles y articuladas en tiempo real y notificaciones actualizadas sobre percances o retrasos en las unidades de las 3 líneas, así como de las alimentadoras RUTA.

2.2 Metodología Scrum

La metodología Scrum es una metodología ágil, se aplica de manera regular a un conjunto de buenas prácticas para trabajar de manera colaborativa a través de iteraciones e incrementos, que se enfoca a la gestión de procesos de desarrollo, equipos de mantenimiento de software. Tiene como características la colaboración entre los miembros del equipo de desarrollo, los clientes y los equipos, la autoorganización para la gestión de cargas de trabajo, controlar el tiempo, etc. Por todo esto, el equipo es el que marca su ritmo de trabajo y el desarrollo es incremental a medida que se avanza en el tiempo, el equipo madura y gana valor [5].

Scrum se definen un conjunto de buenas prácticas para el desarrollo de un proyecto de software. Los roles principales de la metodología de Scrum son Administrador Scrum, Dueño del Producto y el Equipo de desarrollo.

Toma como referencia una Pila de Producto la cual es una lista de requerimientos priorizados de alto nivel lo que define el qué se va a realizar, los requerimientos se gestionan a través de Sprint, es decir se van a desarrollar por iteraciones incrementales, y se planifican durante la reunión de planeación de Sptint. Durante dicha reunión de planificación, el dueño del producto identifica que requerimientos de la pila del producto se deben ver completados y se los transmite al equipo que determina la cantidad de trabajo que puede comprometerse a completar durante el sprint de se está definiendo, además en la reunión se define la duración, el objetivo y el componente de entrega [5].

En los Scrum diario se describen las actividades diarias y la Retrospectiva que consiste en que el equipo mejore su proceso, además que se puedan realizar cambios en los Sprints y por último la revisión de Sprint donde se revisa con el cliente el Componente funcional. Cada Sprint dura entre 1 y 4 semanas el equipo crea un microservicio como se muestra en la Fig 1.

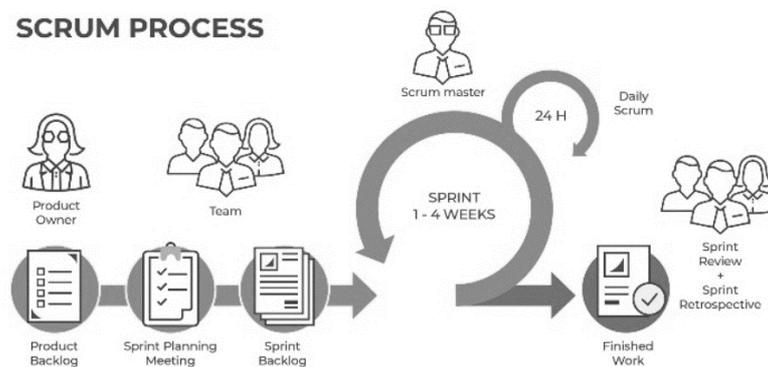


Fig. 1. Metodología Scrum

Como se propuso anteriormente, la arquitectura que se implementó en la plataforma es la de microservicios y en este es que se basa el patrón de diseño, descomposición de un sistema se muestra en la figura 2.

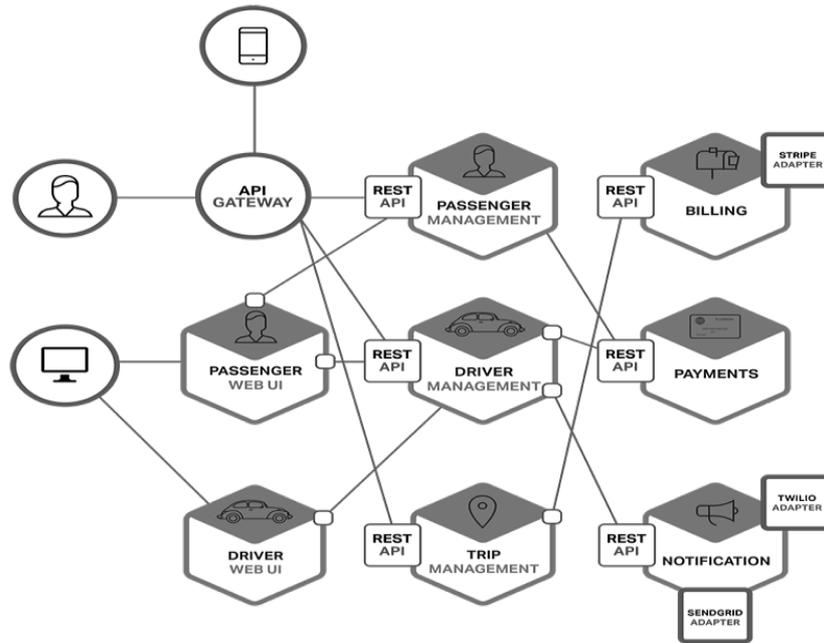


Fig. 2. Un diagrama que representación Microservicios

Lado del cliente: Es representado por el frontend Bootstrap para el diseño de interfaces responsivas, además de HTML5 y CCC3 y JavaScript, del lado del BackEnd la arquitectura del sistema propuesto está dividida en 4 capas:

- Seguridad. Administra los roles y permisos de acceso a la aplicación.
- Gateway. En esta sección de la Arquitectura, se implementa un proceso de puerta de enlace ente los requerimientos de consumo del cliente y por otra parte el servidor.
- Microservicios. En la Tabla 2 se detalla cada uno de los microservicios implementados.

Tabla 2. Detalle de los microservicios implementados en el sistema de Servicio de RUTA en la Ciudad de Puebla

Microservicio	Objetivo
Descripción y horarios de Rutas	Permite a los usuarios conocer los horarios de las 3 líneas de la Ruta en la ciudad de Puebla
Gestión de Tarjeta de Prepago	Este microservicio permite identificar las estaciones para la adquisición de tarjetas y de saldo, además de las condiciones de uso de la tarjeta de prepago para el uso de la RUTA
Pago de viaje	Permite realizar el pago del servicio de transporte a través de la tarjeta de prepago.
Transferencia de saldo	Permite a los usuarios transferir saldo a sus familiares y amigos.
Datos de viaje y conductor de la unidad	Permite conocer datos del viaje y el chofer del viaje que toma, es un módulo implementado por la seguridad del viajero.
Eventos del medio de transporte	Permite conocer los cierres y los eventos que afectan los horarios y la trayectoria de la RUTA
Reportes y quejas	Permite tener implementado un módulo para que los usuarios expresen sus comentarios y quejas.

Cada micro servicio es gestionado como un sprint para la implementación a través de la metodología de SCRUM como se muestra en la Tabla 2.

Tabla 2. Planeación en Sprint, metodología Scrum

Nombre de la tarea	Responsable	Fecha de inicio	Fecha final	Días	Estado
Sprint 1	Pilar B.	9/3	9/13	10	Finalizado
Base de datos	Juan C.	9/3	9/7	3	Finalizado
Horarios de RUTA	Alejandro S.	9/7	9/12	5	Finalizado
Datos de viaje y conductor	Alejandro S.	9/9	9/13	4	Finalizado
Sprint 2	Alejandro S.	9/16	9/24	8	Finalizado
Gestión de Tarjeta de Prepago	Pilar B.	9/16	9/17	1	Finalizado
Pago de transporte	Juan C.	9/17	9/21	4	Finalizado
Transferencia de Saldo	Sarahi R.	9/22	9/24	2	Finalizado
Sprint 3	Sarahi R.	9/25	10/5	10	Finalizado
Eventos del medio de transporte	Pilar B.	9/25	9/29	4	Finalizado
Reporte y quejas	David S.	9/24	10/2	8	Finalizado
Reporte de viajes	Alejandro S.	10/2	10/5	3	Finalizado

El modelado de los microservicios se describe a través de UML del cual se muestra en la Fig. 3 el cual nos permite tener la arquitectura de la base de datos para gestionar los servicios implementados.

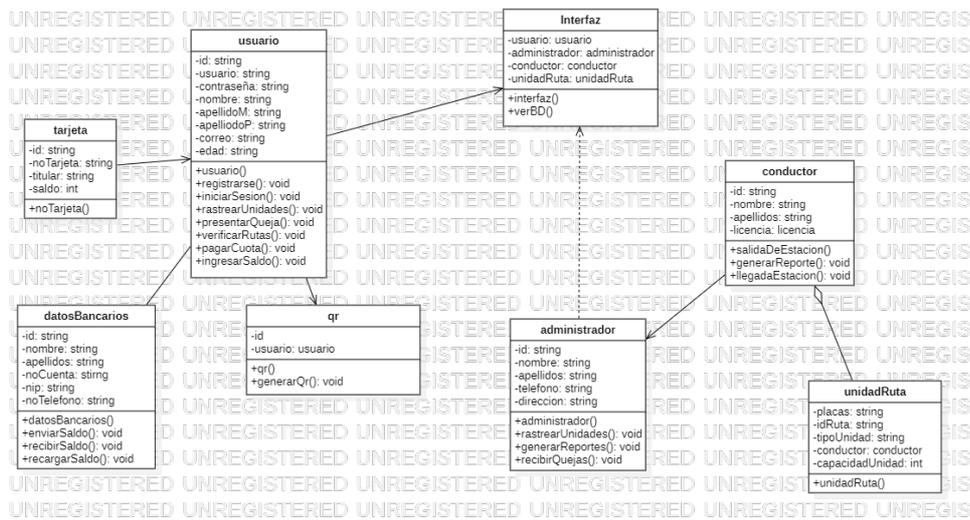


Fig. 3. Diagrama de clases Sistema de Servicio de RUTA en la Ciudad de Puebla

3 Conclusiones y trabajos futuros

Se implementó un sistema con arquitectura de microservicio con resultados favorables para el servicio de transporte de la RUTA en la ciudad de Puebla, brindando mejor experiencia a los usuarios, la implementación se realizó con el patrón arquitectónico de microservicios el cual permite realizar cambios a cada servicio y en caso de fallo o actualizaciones por requerimientos no afecta la disponibilidad del sistema además de que permite el escalamiento rápido y eficiente, además de la implementación de mejoras, añadiendo nuevas funcionalidades a la aplicación de forma ágil y sin interrupciones.

Como resultado se obtiene un sistema con un rendimiento en tiempo de despliegue y de actualización con la arquitectura de microservicios. Con los servicios independientes, se definen mecanismos de comunicación rápida para la interacción entre ellos, el resultado es que cada componente se toma como una unidad independiente, lo que sus operaciones no afectan a las demás, garantizando alta disponibilidad. El tener la independencia de los servicios les permitirá realizar un mantenimiento y desarrollo con menor complejidad, ya que se enfoca en una sola tarea a la vez.

Referencias

1. Barrios Contreras, D. A. (2018). Arquitectura de Microservicios. *Tecnología Investigación y Academia*, 6(1), 36–46. Recuperado a partir de <https://revistas.udistrital.edu.co/index.php/tia/article/view/9687>.
2. Acero, D. A. R. (2018). Modelo de composición de microservicios para la implementación de una aplicación web de comercio electrónico utilizando kubernetes. *Revista de Investigaciones: Escuela de Posgrado de la Universidad Nacional del Altiplano de Puno*, 7(3), 7. <https://doi.org/10.26788/epg.v7i3.371>.
3. Quevedo-Avila, P., Zhindón-Mora, M. G., & Quevedo-Sacoto, A. S. (2020). Arquitectura de microservicios para compras en línea: caso de uso «ala orden». *Polo del Conocimiento: Revista científico - profesional*, 5(1), 151-162. <https://doi.org/10.23857/pc.v5i1.1884>.
4. Torres-Berru, Y., Camacho-Macas, J., Solano-Cabrera, J., & León-Pinzón, L. F. (2020). Migración de un monolito a una arquitectura basada en microservicios, caso de estudio sistema “kbus”. *Dominio de las Ciencias*, 6(2), 763-781. <https://doi.org/10.23857/dc.v6i2.1193>.
5. McCarthy, R. (2021). *Agile y Scrum: Descubra el poder de la gestión de proyectos Agile, Lean Thinking, el proceso Kanban y Scrum*. Primasta.

e-COMPA: Acompañante Emocional

Karina Torres Rodríguez, María del Carmen Santiago Díaz, Yeiny Romero Hernández, Gustavo Trinidad Rubín Linares

Facultad de Ciencias de la Computación, Benemérita Autónoma
Universidad de Puebla, Avenida San Claudio y 14 sur, Colonia San
Manuel, Ciudad Universitaria, C.P. 72570, Puebla, Puebla. México
karina.torresro@alumno.buap.mx, {marycarmen.santiago,
yeiny.romero, gustavo.rubin}@correo.buap.mx

Resumen. En nuestra vida cotidiana experimentamos varias emociones como tristeza, alegría, enojo o amor y también generamos una cantidad enorme de pensamientos que la mayoría de estos son negativos. [1] Estos pueden causar muchas enfermedades mentales como la depresión o ansiedad, que hoy en día han ido en aumento. Para esto hay que tener en cuenta que una de las raíces de estos pensamientos es la autoestima o amor propio. La autoestima o el amor propio es muy importante para el ser humano ya que un nivel bajo de esta puede causar muchos problemas en la vida del mismo. Es por ello que se propone en este trabajo el desarrollo de un acompañante emocional virtual que pueda permitir e impulsar a las personas con baja autoestima a mejorar en su persona diariamente a través de diversas técnicas como la plática, registro o incluso llevar un diario.

Palabras Clave: Acompañante Emocional, Inteligencia Artificial, Salud Emocional.

1 Introducción

La pandemia del COVID-19 es un acontecimiento que ha traído muchos cambios en la vida cotidiana de las personas, el principal fue el aislamiento que trajo muchas consecuencias negativas en la salud de la gente sobre todo en la mental.

Se ha realizado varios estudios en América por parte de la Organización Panamericana de la Salud donde los datos analizados muestran que más de cuatro de cada diez brasileños han tenido problemas de ansiedad y seis de cada diez de depresión; los síntomas de depresión se quintuplicaron en Perú, y la proporción de canadienses con altos niveles de ansiedad se cuadruplicó como resultado de la pandemia. Un estudio realizado en México documentó síntomas de estrés postraumático clínicamente significativo en casi un tercio de la población. En Estados Unidos, las tasas de ansiedad y depresión alcanzaron hasta el 37% y el 30%, respectivamente, a finales de 2020, frente al 8,1% y el 6,5% respectivamente en 2019.[2]

Adicionalmente, la Unicef realizó un sondeo rápido a 8,444 adolescentes y jóvenes de 13 a 29 años en nueve países y territorios de la región. El reporte da cuenta de los sentimientos que enfrentaron en los primeros meses de respuesta a la pandemia y la situación en el mes de septiembre del 2020. Entre las y los participantes, 27% reportó sentir ansiedad y 15% depresión en los últimos siete días. Para el 30%, la principal razón que influye en sus emociones actuales es la situación económica. La situación general en los países y sus localidades ha afectado el día a día de las personas jóvenes pues 46% reporta tener menos motivación para realizar actividades que normalmente disfrutaba. 36% se siente menos motivada para realizar actividades habituales. De todas estas solo el 73% ha sentido la necesidad de pedir ayuda en relación con su bienestar físico y mental. Pese a lo anterior, el 40% no pidió ayuda. [3]

Estas cifras son preocupantes ya que la salud mental es un aspecto fundamental en nuestra vida ya que sustenta nuestras capacidades individuales y colectivas para tomar decisiones, establecer relaciones y dar forma al mundo en el que vivimos. Además, es un derecho humano y un elemento esencial para el desarrollo personal, comunitario y socioeconómico.

También hay que tener en cuenta que en México la terapia es costosa ya que puede variar el precio desde doscientos hasta dos mil pesos mientras que el salario mínimo es de 172.87 pesos diarios, esto puede ser perjudicial para la gente que realmente necesite una ayuda, pero no pueda obtenerla.

Por todo lo mencionado anteriormente se ha decidido generar un acompañante emocional con inteligencia artificial que implemente técnicas para aportar una mejora en la salud mental. Se decidió una inteligencia artificial ya que la tecnología fue uno de los aliados durante la pandemia pues facilitó varias tareas como el trabajo y el estudio, además que de esta manera podría ser accesible a la mayoría de las personas.

En materia de cuidados paliativos, el acompañamiento emocional es generalmente definido como “La capacidad que se puede para llevar o inducir a un paciente en dificultades, a que él mismo movilice sus propios recursos internos y de entorno a efectos de mejor poder sobrellevar una situación”.

Actualmente existen muchas herramientas que realizan un acompañamiento emocional, los más frecuentes son chatbots que son agentes conversacionales automatizados, entre ellos están YANA (You Are Not Alone) el cual ayuda a fortalecer el bienestar emocional a través de un diario en el cual el usuario puede desahogar las penas y emociones con el objetivo de reflexionar sobre sus pensamientos. Otro es JENNY la cual usa inteligencia artificial y emplea un modelo teórico basado en evidencia (la terapia cognitivo conductual) para brindar atención psicológica de primer nivel y recomendar herramientas útiles para el bienestar emocional. También existen otras herramientas como Bird Alone que es un juego móvil donde mediante preguntas te hace reflexionar sobre la vida; Meyo es otra aplicación que consiste en un asistente personal que te enseña a potenciar el autoestima, controlar la ansiedad y el estrés por medio de contenido audiovisual, cortometrajes y ejercicios prácticos; Replika es una inteligencia artificial que su objetivo es ser tú amigo y relacionarse contigo, por chat o por voz; Por último tenemos Aiko que está basado en la Inteligencia Artificial (IA) emocional y tiene como finalidad ayudar a los usuarios a combatir la soledad, a exteriorizar sus sentimientos o emociones y a mejorar sus habilidades comunicativas.

1.1 Detección de Trastornos Depresivos

La depresión es un trastorno mental muy frecuente, en las que destaca la falta de energía, pérdida de interés, alteraciones de sueño y de humor, no poder disfrutar u obtener placer, que a la larga puede provocar incapacidad igual que una enfermedad crónica [3][4]. Tener un software a la mano que diagnostique el nivel de este problema y de un resultado preliminar, será de mucha ayuda ya que al identificarlo se podrá actuar de manera oportuna y así esta enfermedad no tenga consecuencias graves.

Un análisis de investigación no sistemático sobre los instrumentos de tamizaje menciona que entre los mejores resultados a nivel internacional están la escala de depresión del centro de estudios epidemiológicos (CES-D), con una validez del 0,001 y una confiabilidad del 0,90, el Inventario de depresión de Beck (BDI), con una validez del 0,9 y una confiabilidad del 0,69 y, el cuestionario sobre la salud del paciente (PHQ-9) con una validez del 0,75 y una confiabilidad del 0,83 [5].

Entre las distintas pruebas destaca el inventario de depresión de Beck, específicamente en su versión BDI-IA en español donde se ha concluido que tiene una buena capacidad discriminativa, sensible y específica, lo cual nos sirve para apoyarnos de la probabilidad de cuando encuentre la presencia del trastorno depresivo o cuando no se encuentre un rastro de ella.

Si bien la mayoría de estas tecnologías no están enfocadas a la ayuda específicamente del trastorno depresivo, sino a una ayuda emocional más general, si abordan este problema, tal es el caso de What's Up? quienes utilizan algunos de los mejores métodos de terapia cognitivo-conductual (CBT) y terapia de aceptación y compromiso (ACT) para ayudar a sobrellevar la depresión, la ansiedad, la ira, el estrés y más [7], tenemos a Woebot un entrenador personal para los altibajos de la vida [8] y Replika: My AI Friend es un chatbot con inteligencia artificial que te levanta el ánimo, esta aplicación puede ser tu mentor, tu mejor amigo o incluso tu pareja [9].

Algo interesante en la historia de los psicólogos chatbot sucedió con ELIZA quien es considerada el primer chatbot de la historia, aunque no fue programada con un fin de ayuda emocional algunos usuarios en la conversación revelaban sus problemas psicológicos, esto da pie a crear un sistema tipo chatbot orientado a la ayuda emocional para la detección del trastorno depresivo.

2 Desarrollo del Acompañante Virtual

Para empezar, se necesita tener una estructura muy bien definida para el sistema de chatbot, en la cual tendremos las intenciones (las demandas o requerimientos que pide el usuario), las entidades (distinciones relativas de una intención) y el diálogo (el conjunto de preguntas o frases para interactuar con el usuario).

Un reconocimiento de voz es un sistema que toma como entrada las señales acústicas del ambiente, procesa la información obtenida y decodifica los datos a un lenguaje entendible para el sistema. A continuación, se presentan las etapas de análisis espectral, conversión (sonido a texto), modelo mixto de Márkov-Redes Neuronales Artificiales (HMM-ANN), Procesamiento del Lenguaje Natural (PNL) y árboles de decisión, etapas por las cuales pasa un sistema de reconocimiento de voz de tipo chatbot.

Analizamos las entradas que nos proporciona el usuario, por medio de las características de su voz estas nos proporcionan características del contexto en el que está respondiendo el usuario como el tono, intensidad, la duración o la entonación pueden ser datos de gran ayuda para los expertos en la salud emocional esto al identificar si el usuario tiene una voz baja o le está gritando al sistema.

Existen varias técnicas para descomponer el audio en sonidos individuales y convertir estos sonidos en formatos digitales, cuando las personas hablan, crean vibraciones en el aire y por medio de un dispositivo, transforma esta onda de sonido en datos binarios que el sistema puede entender, estas entradas se comparan con sonidos pregrabados de una base de datos y así se obtiene un posible resultado.

El lenguaje humano no es tan simple, muchas personas hablan con diferentes acentos que nos ha heredado la zona en la que nos desarrollamos, jergas y malas pronunciaciones, estas variaciones no se encuentran en el diccionario de la computadora, por esto se utiliza el modelo oculto de Márkov (HMM) el cual es un modelo perfecto para seguir un discurso y junto con las redes neuronales artificiales (ANN) quienes con sus entrenamientos hacen una distinción muy eficiente de las diferentes sílabas que lo componen generando una solución a estos inconvenientes.

3 Desarrollo de una Inteligencia Artificial

Es importante considerar el nivel de aprendizaje que este acompañante virtual tendrá y las herramientas que utilizará para su autonomía, por lo cual se debe desarrollar una Inteligencia Artificial y entrenarla a fin de que el Acompañante tenga autonomía basada en un conjunto de hechos históricos previos, para éste fin describiremos algunos elementos que se utilizarán.

3.1 Redes Neuronales Artificiales

La neurona artificial, este sistema tendrá datos de entrada donde recibirá estímulos externos, con los valores recibidos realizará un cálculo interno y generará un valor de salida. Internamente la neurona hará una suma ponderada de estos valores, la ponderación de cada entrada viene dada con el peso que se le asigna a cada uno de los valores de entrada. Los pesos de entrada sirven para definir la intensidad con la que afecta la entrada a la neurona, veámoslo como un regulador de datos. A todo esto, se le sumará un sesgo o vías para tener mayor control con los datos de salida y para

finalizar se le agrega una función de activación que evaluará el resultado de los datos procesados. En la figura 1 se muestra una ilustración de este sistema.

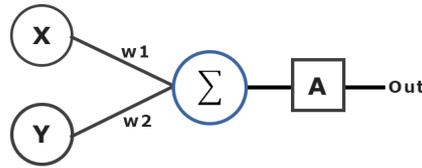


Fig. 1. Arquitectura de la neurona artificial.

El problema que presentó este sistema fue que las soluciones son muy limitadas, solo dan solución a pocos problemas, por este motivo se agregaron más neuronas a un solo sistema para abarcar más problemas a resolver, esto trajo como consecuencia las redes.

Al juntar las neuronas de forma vertical, a este conjunto le denominaremos capas, entonces las neuronas que se encuentran en la misma capa recibirán la información de la capa anterior y los cálculos que realicen los pasarán a la siguiente capa. De esta manera se separa en la primera capa (capa de entrada), la capa de en medio (capa oculta) y la última capa (capa de salida) creando un sistema de redes neuronales artificiales (ANN). En la figura 2 se muestra el esquema ilustrativo de este modelo.

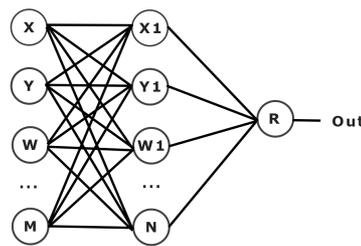


Fig. 2. Arquitectura de una red neuronal artificial (ANN).

3.2 Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) es una rama de la Inteligencia Artificial, específicamente de la lingüística computacional, asimila en un lenguaje de programación definido para poder comunicarse con el ser humano en su lenguaje natural.

El PLN tiene varios objetivos, entre ellos está facilitar la comunicación de la computadora con usuarios que no están especializados con estas, modelar procesos para la comprensión del lenguaje y así diseñar sistemas que realicen tareas lingüísticas como la traducción del lenguaje, resúmenes de textos, recuperación de información o en nuestro caso aplicación de test, entre otros.

3.3 Árbol de Decisión

Anteriormente se vieron algunos test que pueden detectar el trastorno depresivo, para la aplicación de estos utilizaremos los árboles de decisión. Estos son modelos bayesianos de inteligencia artificial, en donde se representan esquemas gráficos de izquierda a derecha o de arriba a abajo hacia las posibles alternativas que puede tomar

el sistema. Entre estos existen modelos descriptivos que sirven para distinguir entre objetos de diferentes clases y los modelos predictivos que pueden predecir la clase a la que pertenece el objeto conociendo sus características anteriores, este es el uso que se le da a la clasificación.

4 Resultados y Conclusiones

El producto de esta plataforma fue una interfaz con un ambiente cálido para las personas de prueba, en estas pruebas se tuvo al programador como observador, guía para registrar errores y problemas de uso.

La interacción fue positiva en función del diseño y usabilidad, sin embargo, se identificaron dos posibles puntos de problemas:

- El primero fue en la interacción, algunos usuarios responden aun cuando el bot no terminaba de hablar, lo cual registraba una combinación de palabras entre las preguntas y la respuesta del usuario, una solución a este problema sería desactivar el micrófono cuando el sistema detecte la entonación.
- El segundo problema se debe al entrenamiento, al ser un sistema que se maneja con el lenguaje natural humano, los vocabularios y frases de cada persona son muy distintos, esta característica funge un lado positivo al dejar que el usuario se pueda expresar más, la solución a esto es entrenar con más datos y pruebas de charla al sistema para lograr una mayor eficiencia.

De esto podemos concluir:

- Aún hay camino por recorrer, pero estamos en la dirección correcta, la sustitución del personal médico de enfermedades mentales aún no es posible, pero con un mayor entrenamiento y añadiendo un receptor visual al sistema para saber las expresiones faciales que tiene el usuario (factor que da información al aplicar el análisis del trastorno), podemos obtener resultados muy prometedores.
- Existe la posibilidad de que el usuario se expresa más abiertamente al estar interactuando con un robot y no una persona.
- Las pruebas de detección del trastorno depresivo (en este caso aplicado por un chatbot) que niegan el bienestar o confirman el buen estado mental de la persona, pueden ayudar a prevenir futuros incidentes en la población joven y adulta.
- El reconocimiento de voz es una gran alternativa para una primera atención a la salud emocional y así dar apoyo a los profesionales de salud emocional.

Referencias

1. brunier, a., & drysdale, c. (2020). los servicios de salud mental se están viendo perturbados por la covid-19 en la mayoría de los países, según un estudio de la oms. recuperado el 28 de junio de 2022, de <https://www.who.int/es/news/item/05-10-2020-covid-19-disrupting-mental-health-services-in-most-countries-who-survey>
2. migala. (2020). podcast migala 22: la depresión. migala. recuperado el 28 de junio de 2022, de <https://www.youtube.com/watch?v=upd6mixy8us&t=4600s>
3. alarcón, r., gea, a., martínez, j., pedreño, j., & pujalte, m. (2003). guía de práctica clínica de los trastornos depresivos. recuperado el 28 de junio de 2022, de <https://consaludmental.org/publicaciones/gpctrastornosdepresivos.pdf>
4. navas, w., & vargas, m. (2012). abordaje de la depresión: intervención en crisis. recuperado el 28 de junio de 2022, de <https://www.binasss.sa.cr/bibliotecas/bhp/cupula/>, 19–35.
5. bustos, v., galvis, e., & rojas, s. (2015). instrumentos de tamizaje de depresión para niños, adolescentes y adultos, revisión narrativa de la literatura. pontificia universidad javeriana.
6. beltrán, m., freyre, m., & hernández, l. (2012). el inventario de depresión de beck: su validez en población adolescente. https://www.scielo.cl/scielo.php?script=sci_serial&pid=0718-4808&lng=en&nrm=iso, 30(1), 5–13. recuperado el 28 de junio de 2022, de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=s0718-48082012000100001&lng=en&nrm=iso&tlng=en

7. jackson temprá. (2021). what's up? - mental health app. google play. recuperado el 28 de junio de 2022, de <https://play.google.com/store/apps/details?id=com.jacksontempra.apps.whatsup>
8. daniels, j. (2017). woebot. facebook. recuperado el 28 de junio de 2022, de https://www.facebook.com/hiwoebot/?ref=page_internal.
9. replika: my ai friend. (s/f). google play. recuperado el 28 de junio de 2022, de <https://play.google.com/store/apps/details?id=ai.replika.app>
10. fernández, y. (2017). así era eliza, el primer bot conversacional de la historia. xataka. recuperado el 28 de junio de 2022, de <https://www.xataka.com/historia-tecnologica/asi-era-eliza-el-primer-bot-conversacional-de-la-historia>

Uso de Inteligencia Artificial para el Reconocimiento de Imágenes de Vicuñas

Manuel J. Ibarra-Cabrera¹[0000-0001-6711-4916], Robinson Buitron¹[0000-0003-4904-7507], Yalmar Ponce²[0000-0003-4379-4341] and Omar Siguas³[0000-0001-6023-3496]

¹ Universidad Nacional Micaela Bastidas de Apurímac - Perú

² Universidad Nacional José María Arguedas - Perú

³ Universidad Nacional de Huancavelica- Perú

mibarra@unamba.edu.pe, robinson1617@gmail.com,
yalmar@unajma.edu.pe, omar.siguas@unh.edu.pe

Resumen. Este trabajo de investigación propone un modelo para realizar el reconocimiento de vicuñas utilizando redes neuronales convolucionales. El problema es que es difícil hacer el conteo de las vicuñas, debido a que es un animal silvestre que generalmente viven en cautiverio o semicautiverio y que no permiten el acercamiento de las personas. Las imágenes fueron recolectadas del centro poblado de Iscahuaca del distrito de Cotaruse en Apurímac-Perú. Posteriormente, las imágenes fueron etiquetadas con el software LabelImg. Finalmente, se entrenó el modelo y los resultados muestran que el rendimiento tiene una precisión de 0,7886, una recuperación de 0,94 y una puntuación F1 de 0,97. Se utilizaron 36 imágenes para entrenamiento, 6 imágenes para validación y 4 imágenes para pruebas.

Palabras clave: Inteligencia artificial, vicuña, reconocimiento imágenes, aprendizaje máquina.

1 Introducción

Los camélidos sudamericanos se clasifican en cuatro especies: la vicuña (*Vicugna vicugna*), la alpaca (*Lama pacos*), llama (*Lama glama*) y el guanaco (*Lama guanicoe*); de estas cuatro especies la vicuña y el guanaco son especies silvestres; y la alpaca y la llama son especies domésticas [1].

En la Fig. 1, se muestra las características de la silueta de cada uno de los camélidos sudamericanos, en la cual se puede observar que la vicuña es el más pequeño en tamaño [2].

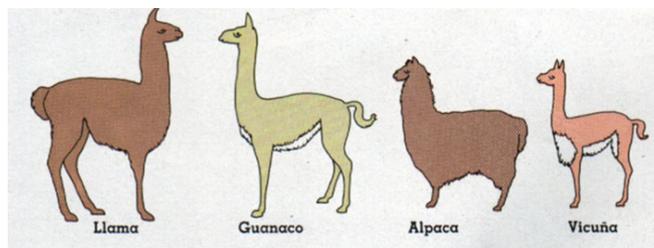


Fig. 6. Siluetas de los camélidos sudamericanos

Los camélidos sudamericanos se encuentran distribuidos a lo largo de la Cordillera de los Andes en América del sur: Ecuador, Bolivia, Chile y Argentina; la mayor parte se concentran el norte de Chile y Argentina y también en la zona del altiplano de Perú y Bolivia. Generalmente viven entre 3000 y 5000 metros sobre el nivel del mar Ecuador [3]. La Fig. 2 muestra una distribución geográfica de las vicuñas en Sudamérica.

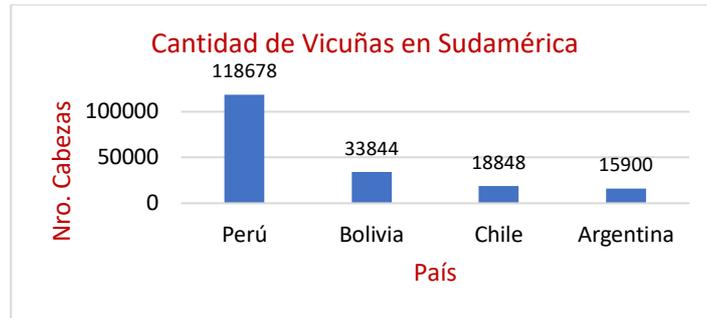


Fig. 7. Distribución geográfica de las vicuñas en Sudamérica

El Perú es uno de los principales productores de camélidos sudamericanos del mundo, con alrededor de 3.5 millones de cabezas entre las cuatro especies [4]. La Fig. 3 muestra una distribución geográfica de las vicuñas en el Perú.

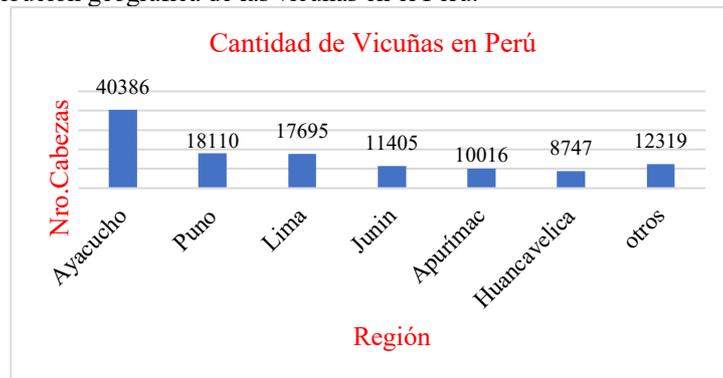


Fig. 8. Distribución geográfica de las vicuñas en el Perú

La vicuña es uno de los animales más preciados, porque su lana es una de las mejores debido a su finura muy particular, y es una de las más valorizadas en el mercado nacional e internacional [5, 6], pudiendo costar hasta 500 dólares el kilo. Las prendas que se elaboran con esta fibra son bastante caras, así por ejemplo una bufanda puede costar 1500 dólares en el mercado internacional [7]; por tanto, es un recurso importante para la economía y el desarrollo de las zonas andinas. La vicuña es una especie que estuvo al borde de la extinción en los años 60s debido a la caza furtiva [8].

Los métodos tradicionales de conteo poblacional de animales silvestres, como por ejemplo de las vicuñas, están basados en el conteo por aproximación [9]; o también en el uso de transeptos para estimar su abundancia o densidad poblacional [10]. Sin embargo, ninguno de estos métodos es enteramente satisfactorio en términos de costos y fiabilidad; por tanto, es difícil conocer con exactitud la población total de vicuñas que existe en el Perú.

El uso de un Vehículo Aéreo No Tripulado (VANT) para el reconocimiento de imágenes de especies animales, emerge como una alternativa para automatizar el proceso de conteo de poblaciones a escala espacio-temporal [11]. Existen investigaciones que han dado buenos resultados para el reconocimiento de imágenes de animales, así por ejemplo bovinos [12, 13]; canguros [14]; ciervos y cerdos salvajes [15].

El problema es que la vicuña es uno de los animales difícil hacer el conteo, debido a su naturaleza de ser silvestre, éstos huyen rápidamente al detectar la presencia de personas. Existen técnicas de conteo basado en aproximaciones o estimaciones, que se realizan cada uno o dos años en los eventos denominados "chaco".

Este trabajo de investigación propone un método para detectar y realizar el conteo de vicuñas en cautiverio o semicautiverio en áreas rurales, en base a imágenes obtenidas por una cámara fotográfica y un dron.

2 Trabajos relacionados

En Australia, Nguyen H. et al realizaron una investigación [16] para el monitoreo eficiente y confiable de los animales salvajes en sus hábitats naturales porque consideran que es esencial la conservación y manejo de dichos animales. Se utilizaron cámaras ocultas o “cámaras trampa” para recolectar las imágenes de manera discreta, continua y en gran cantidad. Aprovechando los avances recientes en técnicas de aprendizaje profundo en visión por computadora, utilizaron técnicas de redes neuronales convolucionales profundas de última generación, para entrenar un sistema computacional capaz de filtrar imágenes de animales e identificar especies automáticamente. Los resultados experimentales muestran que lograron una precisión del 96,6% para la tarea de detectar imágenes que contienen animales, y del 90,4% para identificar las tres especies más comunes entre el conjunto de imágenes de animales salvajes tomadas en el centro-sur de Victoria, Australia.

Por otro lado, Godsk T. and Kjaergaard M. [17] realizaron una investigación para monitorear ganado vacuno, utilizaron un collarín que incluye un GPS (Global Positioning System) para rastrear la posición del ganado, específicamente para detectar el ganado que necesita atención adicional debido a problemas de salud. A menudo, los síntomas de los cambios en el estado de salud y bienestar afectan, así por ejemplo, cambios en el tiempo dedicado a actividades como estar de pie, acostarse, comer o caminar. Para proporcionar datos para este estudio, se montaron receptores GPS de bajo costo en 14 vacas lecheras, durante un día mientras se observaba a distancia y sus actividades se registraban manualmente para que sirvieran como verdad en el terreno. Para el conjunto de datos, se logró obtener una tasa de éxito de clasificación promedio del 86.2% de las cuatro actividades: comer / buscar (90.0%), caminar (100%), mentir (76.5%) y estar de pie (75.8%) optimizando tanto el pre-procesamiento de los datos GPS sin procesar y la extracción de características subsiguientes.

Por otro lado, Pérez R. et al [18] realizaron un trabajo de investigación para analizar estado de preñez en ovejas, a través de imágenes termográficas capturadas de la zona ventral en diferentes etapas de gestación. Los investigadores consideran que la preñez es importante en la producción ovina, para realizar atenciones de salud, monitoreo y nutrición específica a cada oveja, de esta manera garantizar la obtención de una cría sana. Se usó el algoritmo supervisado de segmentación K-means y para la detección del espectro fetal, se llevó a cabo con la implementación del algoritmo Haar-Cascade propuesto por Viola y Jones, fue entrenado con 500 imágenes positivas que contienen la zona de interés, y un grupo de 1000 imágenes negativas donde se muestra la ausencia del feto. Con el método propuesto se obtuvo un porcentaje de asertividad del 80% en la detección automática de espectros térmicos fetales, incluso logrando una múltiple detección de la zona de interés, que permitirá incrementar el índice de asertividad en la etapa de identificación del feto.

Jiang B. et al [19] realizaron un análisis para la detección de manera precisa de las partes más importantes de una vaca lechera. En este estudio, se probó una red de aprendizaje profundo llamada Filter Layer Yolov3 para lograr la detección de las partes importantes de una vaca lechera en escenarios complejos. Dado que las imágenes son inestables durante el proceso de entrenamiento se llevó a cabo el algoritmo de filtrado medio y se aplicó una función de rectificador con fugas (Leaky ReLU) para integrar la capa Filter Layer personalizada para reducir la interferencia de entrenamiento. Se seleccionaron al azar 500 imágenes de 1000 imágenes de vacas como imágenes de entrenamiento originales, y las imágenes restantes se utilizaron como imágenes de prueba. El método propuesto se comparó con el algoritmo Faster R-CNN y YOLOv3 mediante el uso de indicadores como precisión, frecuencia de recuperación, frecuencia de imagen promedio y precisión promedio. Los resultados de las pruebas mostraron que la precisión del algoritmo Yolo fue del 99,18%, la tasa de recuperación fue del 97,51%, y la precisión promedio fue del 93,73%.

3 Materiales

3.1 Cámara

Para tomar las imágenes frontales de la imagen de una vicuña se usó una cámara oculta (escondida entre piedras), cuyas especificaciones técnicas se muestran en la Fig. 4.

<i>Device</i>	<i>Descripción</i>	<i>Foto</i>
Rango de ISO	100- 6400	
Lentes	18-55mm	
Sensor de imagen	CMOS	
Tamaño máximo de imagen	6016 x 4000 pixeles	
Pixeles efectivos	24Megapixeles	
Procesador de video	Full HD 1,920x1,080 / 30 fps	

Fig. 9. Especificaciones técnicas de la cámara fotográfica Nikon D3200.

3.2 Dron

Para tomar las imágenes aéreas de una vicuña y desde una altura de unos 80 metros, se usó el dron DJI Air 2, cuyas especificaciones técnicas se muestran en la Fig. 5.

<i>Device</i>	<i>Descripción</i>	<i>foto</i>
Altura máxima de vuelo	5000 m	
Sensor de imagen	CMOS 1pulg	
Tamaño máximo de imagen	20 MP	
WiFi	2.4 y 5.8 GHz	
Tiempo de vuelo	Hasta 30min	
Velocidad	Hasta 19 m/s	
Distancia máxima del control remoto	8 Km	
Batería	3500 mAh	

Fig. 10. Especificaciones técnicas del DJI Air 2.

3.3 Servidor para el entrenamiento del modelo

Para el entrenamiento, se ha utilizado un servidor con las siguientes características: GPU video GeForce RTX 2080 SUPER; memoria RAM de 32 GB DIMM DDR4 4 bancos que funcionan a una velocidad de 2666 Hz; Disco duro SSD-M2; CPU procesador Intel(R) Core (TM) i9-9900KS CPU; 4.00GHz; 8 cores; 16 threads Mainboard Intel Z390 GAMING X-CF. En este servidor se instaló el software necesario para el entrenamiento del modelo (LabelImg, Python, Darknet, Cuda y otros).

3.4 Software utilizado

LabelImg. Es una herramienta de uso libre y sirve para realizar la anotación de imágenes gráficas y cuadros delimitadores de objetos de etiquetas en imágenes. Está escrito en lenguaje de programación Python y usa QT para el diseño de la interfaz gráfica. Las anotaciones se pueden guardar en archivos con formato XML, Json y texto plano.

Visual Studio Code. Es un editor de texto que ayuda a realizar la codificación del programa.

Python. Lenguaje de programación de código abierto que permite realizar programas los algoritmos de machine learning.

Pytorch. Es una librería de código abierto cuyo propósito es que un algoritmo de Machine Learning se desenvuelva de mejor manera en usabilidad y en velocidad. Es fácil la depuración, es compatible con otras librerías y es compatible con las configuraciones de una GPU.

4 Método

4.1 Recolección de datos

No se ha encontrado imágenes vicuñas en base de datos como por ejemplo Kaggle o Imagenet. Por tanto, se ha tenido que tomar fotos con drones y con cámaras “ocultas”. Una cámara “oculta” (o trampa) es una cámara fotográfica que se coloca de manera oculta entre las piedras y la paja en el entorno geográfico donde habitan las vicuñas, de tal forma que el animal no lo percibe a primera vista, y es posible tomar fotos de cerca.

Para la recolección de datos se utilizaron dos técnicas 1) mediante un dron que tiene una video cámara, este dron es sobrevolado por donde habitan las vicuñas para tomar fotos y para grabar videos; 2) mediante una “cámara oculta”

Para el entrenamiento se utilizaron 36 imágenes y 6 imágenes para validación que fueron seleccionadas al azar. Las imágenes tienen una resolución máxima de 1080p (1920 pixeles de largo por 1080 pixeles de ancho). Las imágenes fueron seleccionadas de tal forma que tengan la mejor calidad y la mejor resolución. La Fig. 6 muestra un diseño de la recolección de datos en el campo.

4.2 Filtrado de datos

Después de recolectar las imágenes (datos), se tiene que hacer un “filtrado” de tal forma que se separe las imágenes que no serán parte del experimento, debido a las siguientes razones: imágenes que no tienen claridad, imágenes borrosas, imágenes muy oscuras, imágenes que no enfocan al animal, entre otras razones. Por eso, antes de ser etiquetados deben ser consideradas solo aquellas imágenes que servirán para el entrenamiento del modelo.

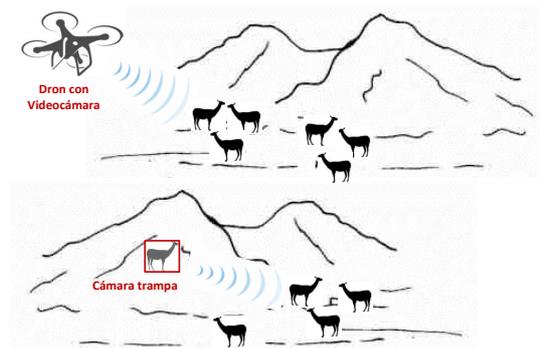


Fig. 11. Recolección de datos con dron y cámara trampa

4.3 Etiquetado de datos

Para etiquetar los datos se utilizó el software LabelImg. La Figura 7 muestra un ejemplo de etiquetado de una vicuña.



Fig. 12. Etiquetado con LabelImg

5 Metodología para el proceso de reconocimiento utilizando inteligencia artificial

5.1 Recolección de datos

En esta parte se realiza la recolección de datos, para este caso específico los datos son “imágenes de vicuña”. Se divide en 3 fases: la fase de obtención (A), consiste en tomar las fotos desde varios ángulos utilizando una cámara digital o la cámara del dron; luego la fase de filtrado (B), consiste en filtrar las imágenes que tienen buena resolución y que muestran la imagen con vicuñas que no están superpuestas, también se descartan imágenes borrosas o imágenes que tienen baja resolución o que están desenfocadas; y luego la fase de etiquetado (C), que consiste en etiquetar cada una de las imágenes mediante el software LabelImg y que a su vez genera un archivo tipo texto con todas las especificaciones de la imagen etiquetada.

5.2 Reconocimiento de imagen de vicuña

El proceso de reconocimiento de imágenes tiene 3 fases: Entrenamiento (D) consiste en proporcionar las imágenes etiquetadas (ingreso de datos) a la técnica de detección de objetos Yolo4, este proceso puede durar horas e incluso días, dependiendo de la cantidad de imágenes a utilizar, entrenar y la computadora en la que entenas; Generación del modelo (E), una vez entrenado, Yolo genera un modelo que se utiliza para posteriormente realizar las pruebas; y la prueba (F) consiste en probar el modelo con cualquier otra imagen que no haya sido etiquetada o entrenada, ver Figura 8.

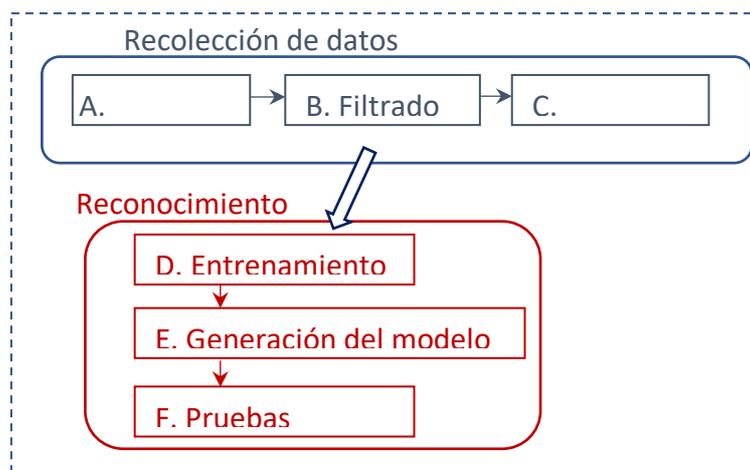


Fig. 13. Metodología para el reconocimiento de la imagen de una vicuña

6 Resultados

Los resultados muestran que después de haber corrido el entrenamiento, se ha obtenido un modelo que es bastante aceptable, y se ha logrado tener una precisión de 0,7886, una recuperación de 0,94 y una puntuación F1 de 0,97. La Figura 9 muestra un ejemplo de la detección de imágenes de vicuña una vez que el algoritmo fue entrenado.



Fig. 14. Detección de imágenes con machine learning

Estos resultados están alineados con los resultados obtenidos por la investigación realizada por Nguyen H et. Al [16], en la que también utilizaron cámaras escondidas o trampa para la recolección de imágenes de manera discreta y continua. Utilizaron una red neuronal convolucional con la función de activación ReLU. La diferencia es que en la investigación propuesta por Nguyen dividió las imágenes en dos segmentos etiquetados como “es animal” (pájaro, bandicut, rata y otros) o “no es animal”, es decir aprendizaje supervisado; por otro lado, en la investigación que se propone el reconocimiento de vicuñas de manera específica se realiza mediante un aprendizaje no supervisado en la que no se etiqueta si es o no animal; sino solo se etiqueta la región en la cual se encuentra la vicuña. En ambos casos la computadora aprende y luego puede detectar la imagen si es o no un animal.

7 Conclusiones y trabajo futuro

En este trabajo de investigación, se propone un mecanismo de reconocimiento de imágenes de vicuñas, las imágenes para el proceso de formación se obtuvieron de 3 localidades del Perú (Iscahuaca). Las imágenes fueron obtenidas mediante cámara fotográfica y vuelo aéreo no tripulado (dron) y posteriormente etiquetadas con el software Labellmg. Para una correcta formación, las imágenes tienen una resolución mínima de 1080p y máxima de 4K.

El rendimiento de detección se mejoró al optimizar la resolución de entrada y obtuvimos una precisión de 0,7886, una recuperación de 0,94 y una puntuación F1 de 0,97. Se utilizaron 36 imágenes para entrenamiento, 6 imágenes para validación y 4 imágenes para pruebas.

Como trabajo futuro, se pretende entrenar con más imágenes y con otras técnicas como por ejemplo Fast-RCNN, RetinaNet, y otros.

Referencias

1. Pinto Jiménez, C.E., Martín Espada, C., Cid Vázquez, M.D.: South american camelids: classification, origen and characteristics. Revista Complutense de Ciencias Veterinarias. 4, 23–36 (2010).
2. Católica, U.: Producción de camelidos, http://www7.uc.cl/sw_educ/prodanim/notrad/siv4.htm.
3. Laker, J., Baldo, J., Arzamendia, Y., Yacobaccio, H.D.: La vicuña en los Andes. Investigación, conservación y manejo de vicuñas. 37–50 (2006).
4. MINAGRI: Origen y situación actual de los camélidos sudamericanos, <https://www.minagri.gob.pe/portal/40-sector-agrario/situacion-de-las-actividades-de-crianza-y-produccion/298-camelidos-sudamericanos>.

5. Quispe, E.C., Rodríguez, T.C., Iñiguez, L.R., Mueller, J.P.: Producción de fibra de alpaca, llama, vicuña y guanaco en Sudamérica. *Animal Genetic Resources/Resources génétiques animales/Recursos genéticos animales*. 45, 1–14 (2009).
6. Quispe Coaquira, J.E., Herrera Mamani, T., Apaza Zúñiga, E., Clavetea Quisca, L., Maquera Marón, Z.: Características tecnológicas de la fibra de vicuñas en semicautiverio de la Multicomunal Picotani-Región Puno. *Revista de Investigaciones Veterinarias del Perú*. 29, 522–532 (2018).
7. Vilá, B.: Los camélidos como parte del patrimonio biocultural de los Andes, http://www7.uc.cl/sw_educ/prodanim/notrad/siv4.htm.
8. Rodríguez García, G.: Animales...? En peligro de extinción o en peligro de que los extingamos? *Ius et Praxis*. 14, 301–315 (2008). <https://doi.org/http://dx.doi.org/10.4067/S0718-00122008000100011>.
9. Hoffmann, R., Otte, K., Ponce, C.: El manejo de la vicuña silvestre Tomo I. Sociedad Alemana de Cooperación Técnica- GTZ, Alemania (1983).
10. Gallardo, G., Nuñez, A., Pacheco, L.F.: Transectos lineales como opción para estimar abundancia de vicuñas (*Vicugna vicugna*): Estudio de caso en el Parque Nacional Sajama, Bolivia. *Ecología en Bolivia*. 45, 64–72 (2010).
11. Christie, K.S., Gilbert, S.L., Brown, C.L., Hatfield, M., Hanson, L.: Unmanned aircraft systems in wildlife research: Current and future applications of a transformative technology. *Frontiers in Ecology and the Environment*. 14, 241–251 (2016). <https://doi.org/10.1002/fee.1281>.
12. Barbedo, J.G.A., Koenigkan, L.V., Santos, P.M., Ribeiro, A.R.B.: Counting cattle in UAV images-dealing with clustered animals and animal/background contrast changes. *Sensors (Switzerland)*. 20, (2020). <https://doi.org/10.3390/s20072126>.
13. Kumar, S., Pandey, A., Satwik, K.S.R., Kumar, S., Singh, S.K., Singh, A.K., Mohan, A.: Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement*. 116, 1–17 (2018). <https://doi.org/https://doi.org/10.1016/j.measurement.2017.10.064>.
14. Brunton, E., Bolin, J., Leon, J., Burnett, S.: Fright or flight? Behavioural responses of kangaroos to drone-based monitoring. *Drones*. 3, 41 (2019). <https://doi.org/https://doi.org/10.3390/drones3020041>.
15. Witczuk, J., Pagacz, S., Zmarz, A., Cypel, M.: Exploring the feasibility of unmanned aerial vehicles and thermal imaging for ungulate surveys in forests - preliminary results. *International Journal of Remote Sensing*. 39, 5504–5521 (2018). <https://doi.org/10.1080/01431161.2017.1390621>.
16. Nguyen, H., Maclagan, S.J., Nguyen, T.D., Nguyen, T., Flemons, P., Andrews, K., Ritchie, E.G., Phung, D.: Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 40–49 (2017). <https://doi.org/https://doi.org/10.1109/DSAA.2017.31>.
17. Godsk, T., Kjærgaard, M.B.: High classification rates for continuous cow activity recognition using low-cost GPS positioning sensors and standard machine learning techniques. In: *Industrial Conference on Data Mining*. pp. 174–188 (2011). https://doi.org/https://doi.org/10.1007/978-3-642-23184-1_14.
18. Pérez, R.S., Olivera, M.V.M., García, O.S., de la Cruz, J.D.D., Herrera, V.G.H., García, L.G.M.: Detección de ovinos preñados usando algoritmos de inteligencia y visión artificial (sheep pregnancy detection using artificial intelligent and artificial vision algorithms). *Pistas Educativas*. 40, (2018).
19. Jiang, B., Wu, Q., Yin, X., Wu, D., Song, H., He, D.: FLYOLOv3 deep learning for key parts of dairy cow body detection. *Computers and Electronics in Agriculture*. 166, 104982 (2019). <https://doi.org/https://doi.org/10.1016/j.compag.2019.104982>.

Estado del Arte del Diseño de una Nube para Archivos de Grandes Dimensiones

Ricardo Pájaro Coatl, Adriana Mendieta Leon, Said Constantino Cárcamo
García, María del Carmen Santiago Díaz

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
Avenida San Claudio, Blvd 14 Sur, Cdad. Universitaria, 72592 Puebla, Pue.
{said.carcamo, adriana.mendietal, ricardo.pajaro}@alumno.buap.mx,
marycarmen.santiago@correo.buap.mx

Resumen. Los sistemas de gestión de contenido son herramientas que permiten la creación, modificación, almacenamiento y organización de cantidades de información de una forma eficaz para su posterior manejo o consulta. En la actualidad, el manejo de grandes cantidades de información es un tema de interés debido a que el almacenamiento digital se ve delimitado por la capacidad de los servidores. El Big data es un término que se asocia a la administración de grandes volúmenes de datos que pueden provenir de una sola fuente o de varias, engloba diversas tecnologías. Cloud Services son servicios que una empresa ofrece a terceros para la administración y gestión de información a organizaciones o clientes que necesiten almacenarla y no cuenten con servidores locales para tratar su información. Este trabajo es el estado del arte del diseño de un gestor de contenido de archivos de grandes dimensiones del tipo no estructurado, se implementa una metodología de desarrollo orientada a una aplicación web (inicial) con Cloud Services. Con el trabajo se busca exponer los factores involucrados para el desarrollo de un sistema de gestión de contenido de grandes dimensiones, se presenta una documentación sobre el proceso desde una investigación previa hasta la presentación de un prototipo base.

Palabras Clave: Gestor de Contenido, Servicios en la Nube, Servidores, Almacenamiento, Formato Marc, Formato Dublin Core, Base de Datos, Desarrollo Web Responsivo, Big Data.

1 Introducción

El crecimiento en el volumen de datos generados por diferentes sistemas y actividades cotidianas en la sociedad ha forjado la necesidad de modificar, optimizar y gestionar toda esta información almacenada. A raíz de esta necesidad de almacenamiento surge el término Big Data, el cual incluye diferentes tecnologías asociadas a la administración de grandes volúmenes de datos provenientes de diferentes fuentes y que se generan con rapidez [1].

Cuando acumulamos grandes volúmenes de datos, se nos plantea la necesidad de ver qué podemos hacer con ellos. Esto nos remite a la necesidad de gestionarlos con una finalidad organizativa, a disponer de tecnología y metodologías específicas. Es correcto decir que cuando hablamos de datos masivos, estamos hablando también de gestión de la información y de generación de conocimiento para la acción [2].

McKinsey Global Institute define el término Big Data de la siguiente manera: “Big Data se refiere a los conjuntos de datos cuyo tamaño está más allá de las capacidades de las herramientas típicas de software de bases de datos para capturar, almacenar, gestionar y analizar. A medida que la tecnología avanza en el tiempo, el tamaño de los conjuntos de datos también crecerán”[3].

Los tipos de datos se clasifican en tres grandes categorías: estructurados (datos transaccionales de las base de datos relacionales), no estructurados (audio, video, fotografía, textos, etc.) y semiestructurados (HTML,XML). La mayoría de las fuentes de datos tradicionales son datos estructurados, datos con formato o esquema fijo que poseen

campos fijos, los datos vienen en un formato bien definido que se especifica en detalle, y que conforma las bases de datos relacionales. Los datos estructurados se componen de piezas de información que se conocen de antemano, vienen en formato específico y se producen en orden específico. Los datos no estructurados son datos sin tipos predefinidos, se almacenan como documentos u objetos sin estructura uniforme, y se tiene poco o ningún control sobre ellos. Datos de texto, video, audio, fotografía son datos no estructurados. Al menos el 80% de la información de las organizaciones no reside en las bases de datos relacionales o archivos de datos, si no que se encuentran esparcidos a lo largo y ancho de la organización; todos estos datos se conocen como datos no estructurados[3].

La producción de datos ha sido la condición social emergente más productiva de la última generación, y por ende empresas y organizaciones de sectores públicos y privados se han visto en la necesidad de incursionar en el ámbito tecnológico donde se promueva el uso de los datos producidos por los sistemas de información y con ello permitir a las organizaciones la emancipación de las ataduras a una localidad física para compartir toda su información. La integración de datos facilita su organización y con ello se puede emplear técnicas tecnológicas para aprovechar todos esos datos generados dentro de la organización, se debe llevar a cabo una buena gestión de almacenamiento de datos.

Una gestión eficaz es la clave para garantizar que las organizaciones utilicen los recursos de almacenamiento de forma eficaz y que almacenen los datos de forma segura, cumpliendo con las políticas de la empresa y la normativa gubernamental. La gestión del almacenamiento garantiza que los datos estén disponibles para los usuarios cuando los necesiten. La política de retención de datos define los datos que una organización retiene por necesidades operativas o de cumplimiento, describe por qué la organización debe conservar los datos, el periodo de retención y el proceso de eliminación, ayuda a una organización a determinar cómo puede buscar y acceder a los datos y es especialmente importante ahora que los volúmenes de datos aumentan continuamente, y puede ayudar a reducir el espacio y los costos de almacenamiento[4].

Debido a la pandemia de COVID-19 y la necesidad de adaptarnos a la nueva normalidad que ha limitado la convivencia a una forma más digital y menos física se ha promovido el incremento de datos digitales provocando que diversas instituciones se vean en la necesidad de modernizar sus sistemas de almacenamiento o incluso comenzar con estrategias de gestión de contenido digital.

El sector salud busca aprovecharse del Big Data para producir información concerniente a: medición de los índices de epidemiología presentes en una determinada localidad, indicadores de enfermedades infecciosas y condiciones presentes en el ambiente que promueven su brote y proliferación. Hospitales privados y públicos tienen al alcance una cantidad finita pero inmensa de información clínica sobre diversos casos médicos que su preservación es importante para llevar a cabo investigaciones médicas o incluso con ese historial médico dar una mejor atención a los pacientes. En México, el sistema de salud más grande es el IMSS que tiene a su disposición un almacenamiento de datos clínicos de tipo estructurado (ej. número de derechohabientes adscritos a clínicas del IMSS) y no estructurados (ej. radiografías, estudios clínicos), gestiona una gran cantidad de información importante para su estudio y preservación que en los últimos años proporciona al público conjuntos de datos abiertos sin restricciones para ser reutilizados en favor a la ciudadanía, estos conjuntos de datos son del tipo semiestructurado (.csv, .rdf, .json y xml)[1].

El complejo Cultural Universitario fue inaugurado el día 10 de Noviembre de 2008, como un conjunto de espacios arquitectónicos y plazuelas interactivas, cuyo objetivo es la promoción de todas las expresiones culturales, artísticas, científicas y tecnológicas que la Benemérita Universidad Autónoma de Puebla genera. Cuenta con una gran cantidad de material cultural digital que ha ido acumulando con el paso de los años y que por la nueva normalidad en la que nos encontramos se ha visto en la necesidad de contar con un sistema de gestión administrativa del contenido, para poder administrar, gestionar y almacenar archivos de grandes dimensiones.

Se encarga del resguardo de producciones digitales de diversas compañías donde el registro de actividades puede ser rescatado en videos de alta calidad con una duración aproximada de 1 hr según sea el evento realizado por la compañía en el que destacan

actividades culturales como bailes folklóricos, eventos musicales de la orquesta, producciones universitarias a cargo de estudiantes de las distintas carreras del área de humanidades localizadas en el complejo cultural, aunado con actividades producidas en casa por cuestiones de la pandemia. Además de material audiovisual, el complejo cultural lleva un control y registro de cápsulas informativas, podcast y registro de imágenes de resolución alta de eventos artísticos, científicos y culturales de la cultura poblana. Antes de la pandemia, el complejo cultural ponía a disposición del público todo este acervo pero en su mayoría ayudaba a la organización de eventos públicos en diversos espacios universitarios, con la nueva normalidad, se busca agilizar el proceso de búsqueda y distribución del material digital para diversos eventos y organizadores que los soliciten, así como facilitar el trabajo y la producción a larga distancia dentro del equipo de desarrollo y producción.

El almacenamiento en la nube es un servicio que permite almacenar datos transfiriéndose a través de Internet o de otra red a un sistema de almacenamiento externo que mantiene un tercero. Los sistemas de almacenamiento suelen ser escalables para adaptarse a las necesidades de almacenamiento de datos de una persona o una organización, accesibles desde cualquier lugar o independientes de aplicaciones para ofrecer accesibilidad desde cualquier dispositivo.

En este reporte, se presenta el estado del arte del diseño de una Nube para archivos de grandes dimensiones, específicamente el manejo de archivos digitales multimedia como videos de alta calidad e imágenes de índole cultural que forman parte del patrimonio cultural de la universidad y del estado. El resultado que se presenta consiste en el modelo integrado por un prototipo de un sistema web para la gestión, administración y almacenamiento de información, así como el análisis de datos para la implementación de una nube de almacenamiento basada en servicios tecnológicos de terceros.

2 Antecedentes

2.1 Gestores de Contenido

Los CMS (sistema de contenido) pueden definirse como: herramientas informáticas estructuradas sobre la base de módulos que permiten la creación, almacenamiento, actualización, recuperación y visualización de los contenidos, a distintos niveles de gestión y acceso, en función del producto implementado. El CMS tiene muchos beneficios por ejemplo: facilidad de creación y mantenimiento, facilidad de uso, gran cantidad de recursos adicionales y posibilidad de optimización para búsqueda orgánicas [5].

El sistema ALEPH (Automated Library Expandable Program) es un software integral que nació de la necesidad de automatizar las bibliotecas universitarias, con el apoyo de un grupo de especialistas en bibliotecología y cómputo. Entre las características de ALEPH se encuentran su arquitectura cliente-servidor, la realización de transacciones por medio de correo electrónico, accesos a través del www y compatibilidad con la norma Z39.50, funciona bajo plataforma UNIX y soporta millones de registros y más de 1,000 usuarios concurrentes [8].

Módulos de ALEPH

Los módulos que integran el sistema ALEPH son: OPAC (catálogo público en línea): catalogación, publicaciones periódicas, catálogos de autoridad, circulación, adquisiciones, utilerías, préstamo, etc., permite tener acceso a las colecciones de la bibliotecas realizando búsquedas por autor, título, encabezamientos de materia. CC(catalogación), se importan los registros en formato MARC, los cuales se validan contra los catálogos de autoridad definidos por la institución [8].

ALEPH ha contado con sus altas y bajas en la UNAM, debido a la complejidad en los procesos de automatización. Almacena y administra catálogos y bases de datos de la biblioteca Conjunta de Ciencias de la Tierra en CU, Unidad Académica Mazatlán ICML, Unidad Académica Puertos Morelos-ICML, Estación Académica de “El Carmen”-ICML,

Estación Regional Noroeste(ERNO)-IGL. Centraliza su documentación lo que le permite tener un mayor control y define los metadatos de las colecciones [9].

ISID es una compañía especializada en el desarrollo de Software de Gestión y Análisis de Contenido Multimedia cuyo objetivo es mejorar la gestión de archivos de vídeo, audio e imagen. ISID tiene como producto Videoma content manager, un gestor inteligente de archivos multimedia, integra en una sola plataforma todas la capacidades necesarias para la ingesta, clasificación automática, archivo y distribución de contenido multimedia. Incorpora además inteligencia artificial para automatizar y facilitar la indexación de contenidos, ahorrando tiempo y permitiendo un uso más eficiente de los recursos. Incluye todas las herramientas para una edición profesional del contenido permitiendo al responsable de contenidos aunar en una única plataforma para dar respuesta a las peticiones de distintos departamentos.

Tabla 1. Características de cada gestor de contenido presentado

Características	ALEPH	Videoma content Manager	Quantum Manager
Crear contenido desde el sitio	SI	SI	SI
Editar el contenido ya existente	SI	SI	SI
Borrar contenido	SI	SI	SI
Búsqueda Ágil	SI	SI	SI
Agregar Contenido	SI	SI	SI
Catología	SI	NO	NO
Manejo de roles y permisos	SI	SI	SI
Descargar contenido	NO	SI	SI
IA implementada	NO	SI	NO
Tipo de dato que manejan	ESTRUCTURADO Y NO ESTRUCTURADO	NO ESTRUCTURADO	NO ESTRUCTURADO

2.2 La nube y sus servicios

La nube hace referencia a los servidores a los que se accede a través de Internet, y al software y bases de datos que se ejecutan en esos servidores. Los servidores de la nube están ubicados en centros de datos por todo el mundo. Con la informática en la nube, no es necesario que los usuarios y las empresas gestionen los servidores físicos ni que ejecuten aplicaciones de software en sus propios ordenadores. La nube permite a los usuarios acceder a los mismos archivos y aplicaciones casi desde cualquier dispositivo, ya que los procesos informáticos y de almacenamiento tienen lugar en servidores en un centro de datos, y no de forma local en el dispositivo del usuario.

3 Metodología

Como parte del trabajo colaborativo para el diseño de un gestor de contenido de grandes dimensiones, se empleó una propuesta metodológica. La metodología consta de 3 etapas principales, propuestas de acuerdo a la naturaleza del proyecto y a los tiempos de desarrollo. Como primera etapa: el análisis, un proyecto de gran alcance debe ser analizado antes de proponer una solución, después la etapa que denotamos como “pre-

producción”, la cual es la etapa donde se lleva a cabo la ingeniería de software, definir tecnologías, arquitectura e infraestructura del gestor, es el intermediario entre el análisis teórico y la implementación práctica del proyecto. Finalmente definimos una tercera etapa, llamada producción, en la cuál se trabaja el desarrollo del código para crear el gestor de contenido. En la Figura 1 se muestra el esquema de la propuesta metodológica.

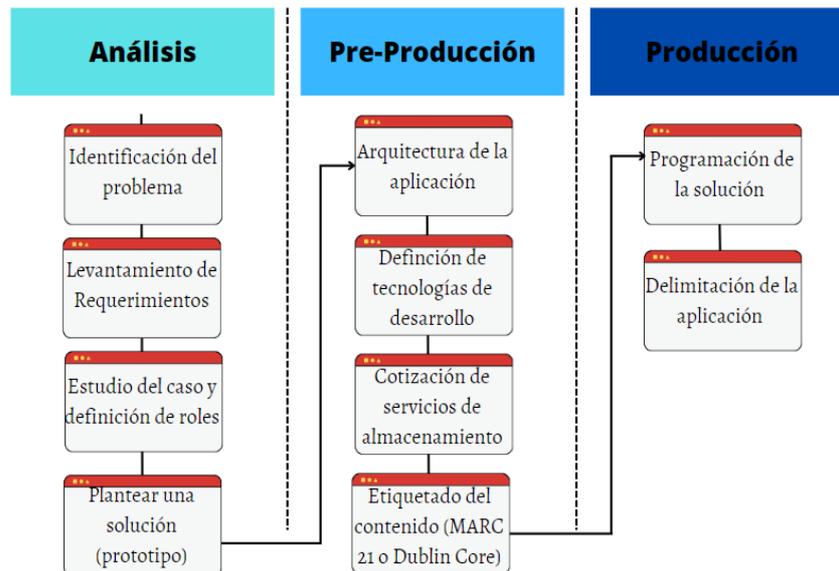


Fig. 1. Diagrama de bloques de la metodología utilizada para el desarrollo de este proyecto, se muestran las etapas y las actividades realizadas en cada una de ellas.

3.1 Etapa de análisis:

Etapa de análisis , establecida con la finalidad de conocer el problema antes de plantear una posible solución con el personal administrativo. Dentro de las actividades realizadas por el equipo de desarrollo para esta primera etapa encontramos la identificación del problema, levantamiento de requerimientos y una primera solución. Personal administrativo planteó la necesidad de un servicio o plataforma que permitiera administrar, gestionar y canalizar el material digital producido con los años en el Complejo Cultural Universitario (CCU) del Estado de Puebla, perteneciendo a la máxima casa de estudios la Benemérita Universidad Autónoma de Puebla, como principal problema el material no se encuentra localizado en un servidor centralizado, la magnitud de almacenamiento supera los discos de almacenamiento locales con los que se cuenta, llevando al mismo personal a distribuir el material y con ello tener un problema de optimización al necesitar consultar, trabajar o compartir el material.

A partir del levantamiento de requerimientos y la identificación del problema, se define una propuesta de trabajo. Las necesidades del cliente fueron planteadas y debido a tiempos de trabajo, la cantidad de personal para administrar y desarrollar el proyecto y al factor de falta de espacio de almacenamiento, planteamos un diseño escalable, donde se propone una mejora constante de funcionalidades dentro del mismo gestor.

El proyecto es abordado bajo el perfil de ingenieros en ciencias de la computación. Se definen las clases que conforman el modelo del sistema, utilizando como referencia los requerimientos del cliente, como se muestra en la Figura 2.

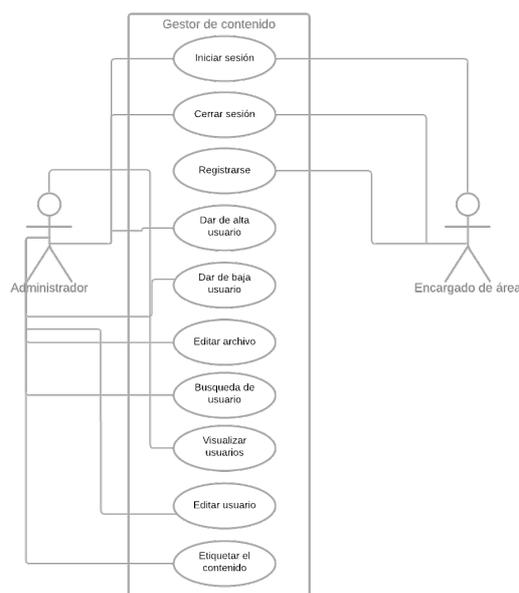


Fig. 2. Diagrama de clases del primer modelo del gestor de contenido según la primera etapa de la metodología.

3.2 Etapa de pre-producción:

Una arquitectura de aplicaciones describe los patrones y técnicas que se utilizan para diseñar y desarrollar aplicaciones. La arquitectura le proporciona un plan y las prácticas recomendadas que debe seguir para diseñar una aplicación bien estructurada. Una aplicación web son sistemas a los que los usuarios acceden a través de Internet. En la aplicación habrá servicios de frontend y de backend, donde el frontend se refiere a la experiencia del usuario con la aplicación, mientras que el de backend implica proporcionar acceso a los datos, los servicios y otros sistemas que permiten el funcionamiento de la aplicación [19].

En la etapa de pre-producción encontramos el proceso de determinar la arquitectura del sistema, tanto backend como frontend y la definición de las tecnologías de desarrollo para ambas partes de la arquitectura. Para la parte del backend, el almacenamiento masivo en un solo sitio es un problema, el Complejo Cultural no cuenta con un servidor con la capacidad masiva de almacenamiento que se necesita.

Para almacenamiento masivos, la universidad cuenta con la Dirección General y Tecnologías de la Información y Comunicaciones (DCyTIC), la cual maneja un Centro de Datos como parte de los servicios de TI que ofrece la dirección, el centro de datos ofrece servidores para el almacenamiento de los datos, sin embargo para este proyecto no fue posible el préstamo de estos servidores, por ello se propone incursionar en almacenamiento en la nube como una alternativa para el gran almacenamiento.

Se desea almacenar datos no estructurados (videos, música, entrevistas, fotografía) en un servicio de almacenamiento de terceros. Debido a la dimensión del conjunto total de los datos, un plan free no es suficiente, por ello, como parte del trabajo se llevó a cabo la cotización aproximada con algunas de las empresas que podrían resolver el problema de capacidad con el que nos enfrentamos al desarrollar el gestor.

El gestor no sólo debe contener el catálogo de material producción con los años en el complejo si no que además debe tener el espacio suficiente para poder guardar y administrar nueva información. De todos los datos que serán almacenados en el gestor, el gran problema son los videos de alta calidad que tienen una duración aproximada de 1 hr. En la Figura 3 podemos ver cuánto espacio ocupa solo guardar en el gestor un video de 1 hr según la resolución de este.

Nombre común	Pixel Tamaño	También conocido como	Tamaño de archivo para 1 hora de video
720p	1280 x 720	HD o "HD Ready"	800-900 MB
1080p	1920 x 1080	FHD o "Full HD"	1.2 - 1.4GB
2K	2048 x 1080	Cine Digital (DCI)	2.8 - 3GB
4K	3840 x 2160	UHDTV1	20 - 22GB
8K	7680 x 4320	UHDTV2	36 - 38GB

Fig. 3. Tamaños de archivos de video según las diversas calidades que se manejan.

3.3 Etapa de producción:

La etapa de producción, a partir de lo definido en las etapas anteriores se comienza el maquetado final del gestor. La interfaz para el usuario se escogió bajo el diseño que emplean algunos de los gestores de contenido existentes, se trabajó bajo el uso de Firebase como servidor de almacenamiento y autenticación con el plan free que ofrece Google.

Las funcionalidades con las que cuenta este gestor implementado en esta fase son las definidas en el sistema gestor de contenido sencillo, véase más en la descripción de la etapa de análisis donde se llevó a cabo la definición de limitaciones en cada prototipo.

El producto final desarrollado es sencillo y cuenta con limitaciones como: funcionalidades sencillas, sin manejo de roles, permitiendo hacer bajas y altas, así como el etiquetado del documento para documentos pequeños sin exceder los 1000 GB por contar con el plan free hasta el momento.

4 Resultados

Se logró tener una interfaz amigable con el usuario para realizar pruebas de navegación, altas y bajas de documentos sin exceder la capacidad del plan free con el que se cuenta hasta al momento. Así mismo fue posible implementar el etiquetado bajo el Formato Marc 21 para cada archivo que sea subido al gestor y la autenticación de usuarios.

A continuación se presenta el diseño final producido como se ve en las Figura 4.

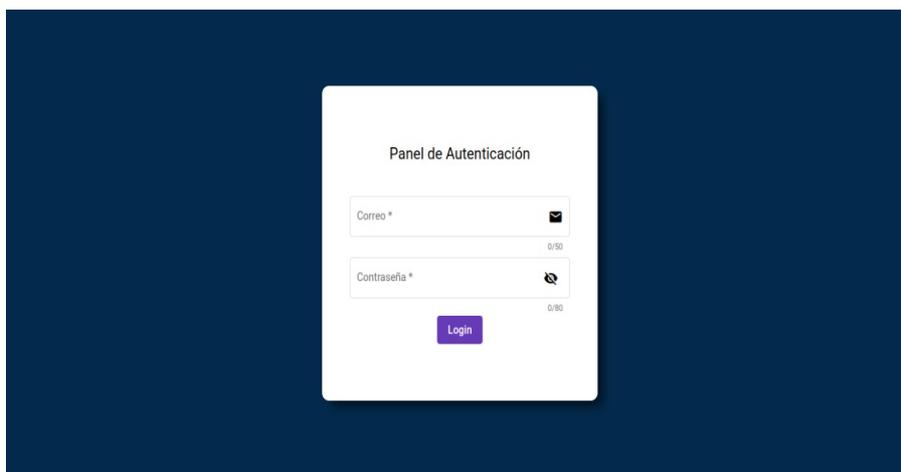


Fig. 4. Vista de inicio de sesión.

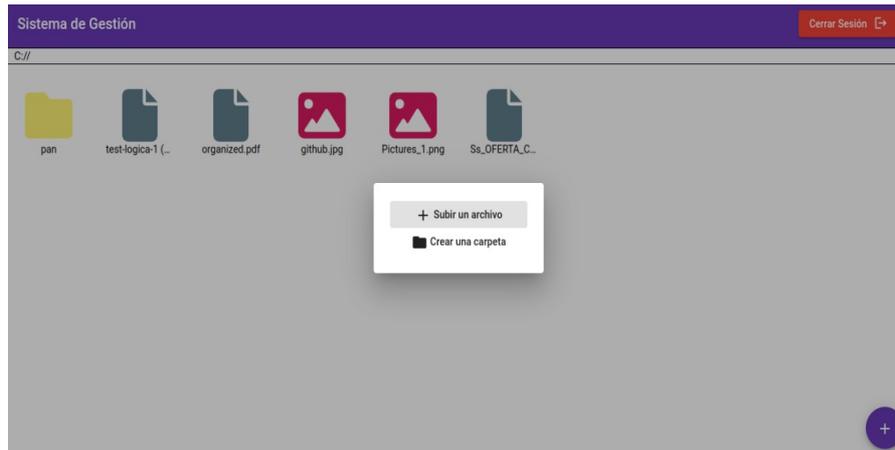


Fig. 5. Vista subir un archivo o crear una nueva carpeta.

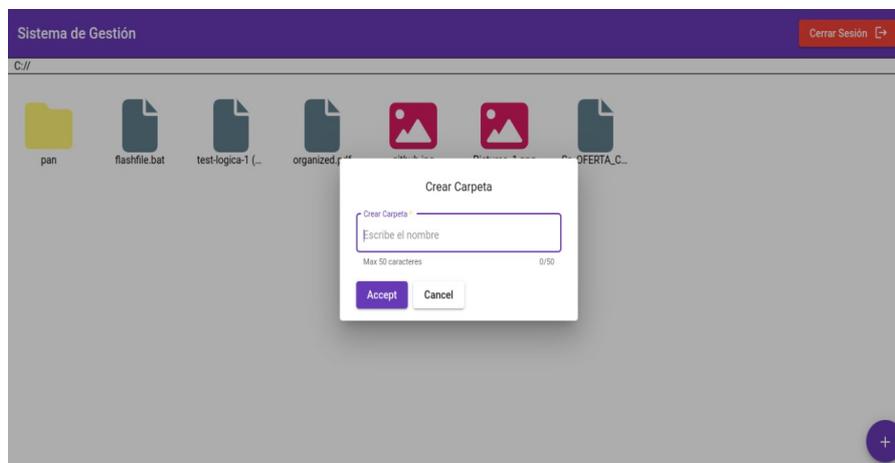


Fig. 6. Vista crear una nueva carpeta.

5 Conclusiones

El manejo de grandes dimensiones de información es un problema que se enfrenta en esta época digital. La conservación y la optimización al trabajar con información recabada por una organización o institución es de vital importancia ya que representa el trabajo de años de datos recabados en diversas actividades. Debido a ello es difícil controlar el crecimiento, ya que el incremento es de manera exponencial y su organización se vuelve bastante compleja provocando problemas como redundancia en datos, pérdida o fuga de información, etc.

El control y la organización de información ya sea estructurada o no estructurada con sistemas gestores de contenido facilitan el trabajo de los usuarios y clientes que soliciten la información digital ya sean externos o internos dentro de la institución. En la actualidad, la diversidad de sistemas de almacenamiento en servidores remotos ha permitido la creación de aplicaciones web con capacidades masivas de almacenamiento, debido a que toda la información es alojada en centros de datos diversos.

Un sistema gestor de contenido es una buena solución para administrar contenido de gran dimensión, pero se deben considerar factores para la construcción de este tipo de aplicaciones. Es un proceso que conlleva la delimitación de alcances, gestión de planes de almacenamiento, considerando en este aspectos si el almacenamiento será en servidores locales o con un proveedor del servicio, políticas de retención, etc. El encargado del desarrollo deberá considerar todos estos factores para poder desarrollar un sistema gestor de la mejor manera para su problema en particular, así mismo se debe considerar qué tipo de información se maneja ya que un sistema que trabaja con contenido

multimedia suele necesitar más espacio para almacenar un solo objeto que un sistema de contenido de datos estructurados.

En conclusión, el desarrollo de un sistema gestor de contenido multimedia es un trabajo que necesita un proceso largo y metódico, se espera que este reporte ayude a futuros trabajos de desarrollo para sistemas de este tipo.

Referencias

1. Escobar Borja, M. y Mercado Pérez, M. 2019. Big data: un análisis documental de su uso y aplicación en el contexto de la era digital. *Revista La Propiedad Inmaterial*. 28 (dic. 2019), 273–293. DOI:<https://doi.org/10.18601/16571959.n28.10>.
2. García Alsina, M. (2017). *Big Data: Gestión y explotación de grandes volúmenes de datos* (1.a ed.) [Books.google.com.mx]. Javier Guallar. <https://books.google.com.mx/books?id=SFgtEAAAQBAJ&pg=PT30&dq=big+data+almacenamiento+masivo&hl=es&sa=X&ved=2ahUKewiow4Dwxsj7AhX6MUQIHV06AZEQ6AF6BAGHEAI#v=onepage&q=big%20data%20almacenamiento%20masivo&f=true>
3. Joyanes, L. (2022). BIG DATA. Alfaomega Grupo Editor.
4. Crocetti, P. (2021, 16 noviembre). *Gestión de almacenamiento de datos: ¿Qué es y por qué es importante?* ComputerWeekly.es. <https://www.computerweekly.com/es/reporte/Gestion-de-almacenamiento-de-datos-Que-es-y-por-que-es-importante>
5. Rosell LY. Sistemas gestores de contenidos: una mirada desde las ciencias de la información. *Revista Cubana de Información en Ciencias de la Salud* (ACIMED). 2011;22(1):3-17. Recuperado de: <https://www.medigraphic.com/pdfs/acimed/aci-2011/aci111b.pdf>
6. ¿Qué es un sistema de gestión de contenido (CSM)? (s. f.). Oracle. Recuperado de: <https://www.oracle.com/mx/content-management/what-is-cms/>
7. Rodríguez-Reséndiz, Perla Olivia, Joséphine Simonnot y Dafne Citalli Abad-Martínez. 2018. “Gestor de contenidos de código abierto para archivos digitales sonoros que preservan materiales de investigación”. *Investigación Bibliotecológica: archivonomía, bibliotecología e información* 32 (77): 101-115. <http://dx.doi.org/10.22201/iibi.24488321xe.2018.77.58005>
8. Mata Acosta, V., Rosas Poblano, S. y Cortés Girón, S. (2004). El módulo de adquisiciones en Aleph: su utilización en la UNAM. *Biblioteca universitaria*, 7 (1) 23-33. Recuperado de http://biblioteca.udgvirtual.udg.mx/jspui/bitstream/123456789/3380/1/M%c3%bdulo_adquisiciones_Aleph.pdf
9. Armendáriz Sánchez, S., & Castro Escamilla, M. (2019). El uso de ALEPH en la Biblioteca Conjunta de Ciencias de la Tierra. *Biblioteca Universitaria*, 22(1). <https://doi.org/10.22201/dgb.0187750xp.2019.1.445>
10. Videoma Content Manager - Gestión multimedia con inteligencia artificial. (2020, 12 noviembre). *ISID*. <https://isid.com/es/productos-2/gestion-archivo-multimedia/>
11. (2021, 20 abril). Quantum Manager - El gestor multimedia avanzado para Joomla. *Sergio Iglesias Blog*. <https://sergioiglesias.net/blog/extensiones/quantum-manager-el-gestor-multimedia-avanzado-para-joomla>
12. ¿Qué es la nube? Definición de nube (s.f.). *Cloudflare*. <https://www.cloudflare.com/es-es/learning/cloud/what-is-the-cloud/>
13. Cruz Chávez, M. A., Peralta Abarca, J. del C., Martínez Rangel, M. G. & Cruz Rosales, M. H. (2014). *La computadora, herramienta indispensable en diversas áreas de conocimiento*. (1.a ed.). https://www.researchgate.net/profile/Marco-Cruz-Chavez/publication/270340184_La_Computadora_Herramienta_Indispensable_en_Diversas_Areas_del_Conocimiento/links/54a82e540cf256bf8bb7e072/La-Computadora-Herramienta-Indispensable-en-Diversas-Areas-del-Conocimiento.pdf#page=175
14. ecastillo@nts-solutions.com. (2022, 21 septiembre). ¿Qué es Firebase? ¿Qué ventajas ofrece en 2022 a nuestras apps? *NTS SEIDOR*. <https://www.nts-solutions.com/blog/firebase-que-es.html>
15. Cosio, L. (2017, 14 marzo). ¿Qué es AWS?. *Inbest* <https://www.inbest.cloud/comunidad/qu%C3%A9-es-aws>
16. AWS | Almacenamiento de datos seguro en la nube (S3). (s. f.). *Amazon Web Services, Inc*. https://aws.amazon.com/es/s3/?did=ft_card
17. Qué es Azure: Servicios en la nube de Microsoft. (s. f.). *Microsoft Azure*. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-azure/>
18. Presta, M. (2021, 25 junio). Backend como servicio gratuito. *Back4App Blog*. <https://blog.back4app.com/es/backend-como-servicio-gratuito/>
19. ¿Qué es una arquitectura de aplicaciones? (s. f.). <https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture>

***Aportaciones de las Ciencias Computacionales en Sistemas Inteligentes y
Ciberseguridad***

se terminó de editar en Diciembre de 2022 en la Facultad de Ciencias de la
Computación

Av. San Claudio y 14 Sur
Jardines de San Manuel
Ciudad Universitaria
C.P. 72570

*Aportaciones de las Ciencias Computacionales en Sistemas Inteligentes y
Ciberseguridad*
Coordinado por
Gustavo Trinidad Rubín Linares