

REVISTA ODIGOS



CIENCIAS DE LA INGENIERÍA
Y CIENCIAS EXACTAS

Vol. 3 Num. 1

2022

FEBRERO MAYO



Universidad
Israel

CONTENIDO

5 Página legal

7 EDITORIAL
Mg. Renato M. Toasa
Editor de la Revista ODIGOS

9 Android service to interface mosquitto messaging broker (MQTT)
Cristian Mauricio Gallardo Paredes
Patricia del Rocío Rodríguez Fiallos
Francisco Javier Galora Silva

25 Automatización con aplicación móvil para agendamiento de cita en peluquerías
Miguel Angel Pino Andrade
Arthur Wilson Chávez Mora
Juan Daniel Carrion Robles

49 Diseño de aplicación móvil para promoción de productos
Andrea Guissela Pérez Lozada
Edison Moreno Jurado
Luis Alberto Pérez Viana



69

“Second Mind” una aplicación móvil para el almacenamiento de memorias del usuario

Robert Alexander Almache Vallejos

Jorge Ramiro Castro Chavez

Diego Alejandro González Calderon

Johanna Daniela Ruano Cardenas

89

Analizando Internet de las Cosas y la nube informática

Darwin Stalin Ramírez Supe¹

Estefanía de las Mercedes Zurita Meza²

Francisco Javier Galora Silva³

104

NORMAS DE PUBLICACIÓN
REVISTA ODIGOS

PÁGINA LEGAL

DIRECTOR CIENTÍFICO	PhD. Fidel David Parra Balza Universidad Tecnológica Israel, Ecuador
EDITOR GENERAL	Mg. Paúl Francisco Baldeón Egas Universidad Tecnológica Israel, Ecuador
EDITOR REVISTA ODIGOS	Mg. Renato Mauricio Toasa Guachi Universidad Tecnológica Israel, Ecuador
COMITÉ EDITORIAL	PhD. Victor Hugo Andaluz Ortiz Universidad de las Fuerzas Armadas ESPE, Ecuador PhD. David Raimundo Rivas Lalaleo Universidad de las Fuerzas Armadas ESPE, Ecuador PhD(c). César Leonardo Guevara Gordillo Universidad Técnica Federico Santa María, Chile PhD(c). John Reyes Vasquez Universidad Técnica de Ambato, Ecuador PhD(c). José Luis Varela Aldás Universidad Tecnológica Indoamerica, Ecuador PhD(c) . Fernando A. Chicaiza Universidad Nacional de San Juan, Argentina PhD(c). Christian Carvajal Universidad Nacional de San Juan, Argentina PhD(c). Javier Santiago Vargas Paredes Universidad de Chile, Chile M.Sc. Cristian Mauricio Gallardo Paredes Universidad Politécnica de Tomsk, Rusia M.Sc. Flores García Yolanda Graciela Universidad Politécnica de Tomsk, Rusia Mg. David Omar Guevara Aulestia Universidad Técnica de Ambato, Ecuador MSc. Nataly Pozo Viera Universidad San Francisco de Quito, Ecuador MSc. Juan Pablo Guevara Gordillo Universidad Central del Ecuador, Ecuador MSc. Carlos Alfredo Silva Villafuerte Universidad Técnica de Manabí , Ecuador MSc. Christhel Alejandra Andrade Díaz Universidad Técnica de Manabí, Ecuador MSc. Darwin Stalin Ramirez Supe Universidad Internacional de la Rioja, España Mg. Estefanía de las Mercedes Zurita Meza Instituto Tecnológico Superior Pelileo, Ecuador MSc. Edgar Fabián Rivera Guzmán Instituto Tecnológico Superior Oriente, Ecuador Mg. Edison Andrés Gómez Reyes Instituto Ecuatoriano de Seguridad Social, Ecuador Mg. David Martínez Villacrés



**GESTIÓN DE LA REVISTA
ELECTRÓNICA
RESPONSABLE DE ESTILO**

**RESPONSABLE DE DISEÑO,
MAQUETACIÓN Y
DIAGRAMACIÓN**

**RESPONSABLE
PROGRAMADOR**

PERIODICIDAD DE PUBLICACIÓN - CUATRIMESTRAL

ENTIDAD EDITORA

Universidad de Guayaquil, Ecuador
MSc. Francisco Javier Galora Silva
Universidad Internacional de la Rioja, España
MSc. Mauricio Xavier López Flores
Universidad Internacional de la Rioja, España
MSc. Elvis Román López Flores
Universidad Internacional de la Rioja, España
MSc. Juan Carlos Barrera León
Instituto Politécnico de Leiria, Portugal

Mg. Paúl Francisco Baldeón Egas
Universidad Tecnológica Israel, Ecuador

Lcda. Carla Cristina Florez
Universidad Tecnológica Israel, Ecuador

Mg. José Alejandro Vergelín Almeida
Universidad Tecnológica Israel, Ecuador

Ing. Carlos Alberto Rivadeneira Proaño
Universidad Tecnológica Israel, Ecuador

Universidad Tecnológica Israel
Dirección: Marieta de Veintimilla E4-142 y Pizarro, Quito
Código postal EC-170522
editorial@uisrael.edu.ec - Teléfono: (02) 255-5741 ext. 113



EDITORIAL

Nos complace presentar el número 1, volumen 3 de la Revista ODIGOS, el primero del año 2022. En esta oportunidad ponemos a disposición de toda la comunidad académica 05 artículos como resultado de las investigaciones elaboradas con alta rigurosidad científica y metodológica, y que aportan significativamente a diversas áreas del conocimiento.

Como en todas nuestras publicaciones, los trabajos presentados han pasado por un proceso de selección, arbitraje, corrección y edición, que van en correspondencia con las líneas aprobadas por la Universidad Tecnológica Israel, entidad editora de nuestra revista.

En este sentido, los trabajos que se presentan son:

“Android service to interface mosquito messaging broker (MQTT)” es el título del primer artículo que se presenta, en este, los autores determinan como objetivo desarrollar un servicio sobre Android, que sirva de interfaz con el broker de mensajería Mosquitto (MQTT) que proporcione un mecanismo adecuado para recibir peticiones de una aplicación Android y al mismo tiempo también notifique que han llegado unos datos, mismo que se almacenan en el gestor SQLITE, que utiliza pocos recursos computacionales y es adecuado para pequeñas aplicaciones, que están probando protocolos y servidores de mensajería.

El segundo trabajo publicado lleva por nombre *“Automatización con aplicación móvil para agendamiento de cita en peluquerías”*, en el cual se idea la aplicación desde dos perspectivas, la de usuario; es decir, clientes de la peluquería y la de administrador. El desarrollo de la aplicación se realizará bajo Visual Studio XAMARIN. Como parte de las conclusiones, se establece que, si bien la aplicación se encuentra en su fase inicial, los resultados experimentales demuestran que los negocios (peluquerías) consolidados, y que manejan cantidad masiva de clientes, se podrían beneficiar con la automatización en su proceso de agendamiento de citas.

Por otra parte, los autores de *“Diseño de aplicación móvil para promoción de productos”*, tras un análisis sobre la necesidad que contemplan los emprendedores para dar a conocer sus productos o servicios, se plantearon desarrollar un prototipo para dispositivos móviles Android, que permita crear un catálogo de productos, con base en métodos de inferencia, fuentes bibliográficas y encuestas resumidas sobre las preferencias comerciales de los consumidores y las marcas favoritas, tres



aspectos que son importantes para los usuarios cuando compran son: direcciones, precios y ubicación, que son los puntos clave para diseñar la aplicación planteada.

Mientras que en *“Second Mind” una aplicación móvil para el almacenamiento de memorias del usuario*, fue una parte primordial identificar la necesidad del usuario, así como también los requerimientos para que el prototipo sea efectivo. Con tan solo el uso de esta app móvil, que tiene una interfaz amigable, se podrá disponer de textos, tutoriales y grabar incluso una URL o imágenes, usando su propia cámara del dispositivo, así traerá y recordará cualquier información que necesite de forma inmediata.

Finalmente, el último trabajo *“Analizando Internet de las Cosas y la nube informática”*, presenta un estudio de 2 nuevas tecnologías que han ido creciendo a pasos agigantados en los últimos años, una de ellas es el Internet de las Cosas (IoT), el cual permite a los usuarios conectar miles de millones de máquinas inteligentes e intercambiar información, monitorear y controlar servicios. Mientras que la otra es el Cloud Computing, que se refiere al acceso, configuración y operación de recursos en ubicaciones remotas. Como parte de las conclusiones se obtiene que la combinación entre el IoT y Cloud Computing podrían construir una superpotencia de TI creará tecnologías que actualmente son inimaginables y nuevos campos de investigación que beneficiará a la humanidad.

El impacto de estas investigaciones, dentro de la comunidad científica, permitirá replantear modelos y herramientas para generar propuestas de intervención que contribuyan con la solución de ciertos problemas existentes en la sociedad, relacionados con los temas aquí tratados.

Mg. Renato M. Toasa
Editor de la Revista ODIGOS

Android service to interface mosquitto messaging broker (MQTT)

Fecha de recepción: 2021-12-26 • Fecha de aceptación: 2022-01-05 • Fecha de publicación: 2022-02-10

Cristian Mauricio Gallardo Paredes¹

Universidad Politécnica de Tomsk, Rusia

kristianmauricio1@tpu.ru

<https://orcid.org/0000-0002-5361-9144>

Patricia del Rocío Rodríguez Fiallos²

Ministerio de Educación Distrito 18D02, Ecuador

patriciad.rodriguez@educacion.gob.ec

<https://orcid.org/0000-0002-5213-1008>

Francisco Javier Galora Silva³

Universidad Internacional de la Rioja, Ecuador

francisco.galora049@comunidadunir.net

<https://orcid.org/0000-0002-5464-5336>

ABSTRACT

Real-time messages are used at a high level by different computer applications, there are different servers, protocols, etc. that can be used and are available on the Internet, however, these are created in such a way that the consumption of computational resources is very high and would only be optimal if we use a server with great features. This research work aims to develop a service on Android service to interface mosquitto messaging broker (MQTT) and provide a suitable mechanism to receive requests from an Android application at the same time also notify that some data has arrived, data are stored in the SQLite manager, which uses few computational resources and is suitable for small applications, which are testing protocols and messaging servers. The experimental results show that the service meets the initial objectives of the work, allowing to create of a messaging system based on publishers and subscribers for the Android platform.

KEYWORDS: service, android, Mosquito, MQTT

RESUMEN

Los mensajes en tiempo real son utilizados en un alto nivel por diferentes aplicaciones informáticas, existen diferentes servidores, protocolos, etc., que se pueden utilizar y están disponibles en Internet; sin embargo, estos están creados de tal manera que el consumo de recursos computacionales es muy elevado y solo sería óptimo si se utiliza un servidor con grandes prestaciones. Este trabajo de investigación tiene como objetivo desarrollar un servicio sobre Android, que sirva de interfaz con el broker de mensajería Mosquitto (MQTT) que proporcione un mecanismo adecuado para recibir peticiones de una aplicación Android y al mismo tiempo también notifique que han llegado unos datos, mismo que se almacenan en el gestor SQLITE, que utiliza pocos recursos computacionales y es adecuado para pequeñas aplicaciones, que están probando protocolos y servidores de mensajería. Los resultados experimentales muestran que el servicio cumple con los objetivos iniciales del trabajo, permitiendo crear un sistema de mensajería basado en editores y suscriptores para la plataforma Android.

PALABRAS CLAVE: servicio, Android, Mosquito, MQTT

Introduction

With the notoriety of cell phones, individuals are progressively subject to their cell phones to receive continuous messages, so texting is especially significant (Zhang et al., 2007). As of now, we can sidestep the transporters, through a standard TCP/IP organization to send messages straightforwardly to the cell phone (Peiji & Yanai, 2001).

It's obviously true that over the most recent couple of years, cell phones have altered the entire world, and a piece of this significant world pattern has involved changing the manner in which individuals speak with each other; one of the components of this current has been texting administration. The push message administrations were executed different convention structures like XMPP, CoAP, and MQTT. These different conventions are utilized for each unique circumstance, specifically the MQTT convention was intended to chip away at low-power gadgets pleasantly as a lightweight convention and has been utilized in numerous IoT gadgets and texting frameworks (Hwang et al., 2016).

MQTT protocol become our best option due for its undeniable potential benefits. It is an open, basic, lightweight, and simple to-carry out informing convention. At first, it was intended to associate huge quantities of far-off sensors and control gadgets (Silva et al., 2017). The convention has been applied in an assortment of inserted frameworks. Clinics utilize this convention to speak with pacemakers and other clinical gear providers. Oil and gas organizations use it to screen oil pipelines large number of miles away (Shaoyue et al., 2012).

A portable application that utilizes MQTT sends and gets messages by calling a MQTT library. Messages are traded through a MQTT informing server. The MQTT customer and server handle the intricacies of conveying messages dependably to the versatile application and monitor the cost of organization the board.

MQTT applications run on cell phones, for example, cell phones and tablets. MQTT is likewise utilized in telemetry to get information from sensors and to control them from a distance. For cell phones and sensors, MQTT offers a profoundly adaptable distribution/membership convention with secure conveyance.

A few review papers are tending to various attributes of distribute/buy in based frameworks, like plan, execution, quality necessities, directing calculations, and so forth In "Nature of Administration in Wide-Scale Distribute Buy in Frameworks", the creators research the cutting edge modern and scholarly distribute/buy in arrangements zeroing in on adaptability and quality prerequisites, while in "The Many Essences of Distribute/Buy in", the creators present order dependent on correspondence, different distribute/buy in plans, plan, and executions, yet as far as we could possibly know, there is no overview which recognizes key-necessities for IoT distribute/buy in arrangements.

The following research is also carried out in this important area, such as:

A. Design and implementation of a reliable message transmission system based on



MQTT protocol in IoT

This paper planned and executed a dependable message transmission framework utilizing MQTT convention to keep up with requesting between messages for the workplace. This (framework) comprises of MQTT convention, solid message transmission server, and customer module (Hwang et al., 2016).

B. Design and implementation of push notification system based on the MQTT protocol

This paper depicted a technique for pushing warning framework dependent on the MQTT convention. It tends to be utilized to tackle the issue of moment pushing different messages from the server to the versatile customer (Tang et al., 2013).

C. System of acquisition, transmission, storage and visualization of pulse oximeter and ECG data using android and MQTT

The creators present an arrangement of wellbeing information assortment, transmission, and capacity intended for electrocardiography (ECG) and Heartbeat Oximetry results, the objective is to address this test by making a framework for gaining and communicating information through Bluetooth to an Android versatile stage, which sends it to a distant server, where the information is put away in a data set and opens up for perception. This empowers any far off client to get to imperative information on a patient without being actually present (Barata et al., 2013).

D. Correlation analysis of MQTT loss and delay according to QoS level

This examination breaks down the MQTT message transmission process which comprises of truly wired/remote distribute customer, specialist server, and buy in customer. By communicating messages through 3 degrees of QoS with different sizes of payloads, we have caught parcels to examine start to finish deferrals and message misfortune (Lee et al., 2013).

All these papers are of great help as a base for the realization of this project, because they allow understanding the benefits that can be obtained when using mosquito MQTT and its linkage with other technologies that would greatly benefit people, this can be evidenced by the investigations carried out in this important area.

The purpose of this research is to perform an android service to interface Mosquitto Message Broker (MQTT), offering a suitable mechanism to receive requests from Android applications and also to notify that some data has arrived.

Methodology

In this chapter, we propose a method for sending and receiving messages, this is done through an android service, which allows you to create the topic, make subscriptions, perform publication,

display all messages. For the experiment, we used Linux-based Debian 8 for the server and an open-source project Mosquitto MQTT for Broker server software.

It is very important to keep the guarantee of sending messages reliably, the service developed in Android through the mobile application communicates with other IoT devices messages must have atomicity, consistency, and performance (Schiper & Raynal, 1996). In addition, the mobile application for reliable message transmission should be able to view the previous messages for reliable message transmission or communication history.

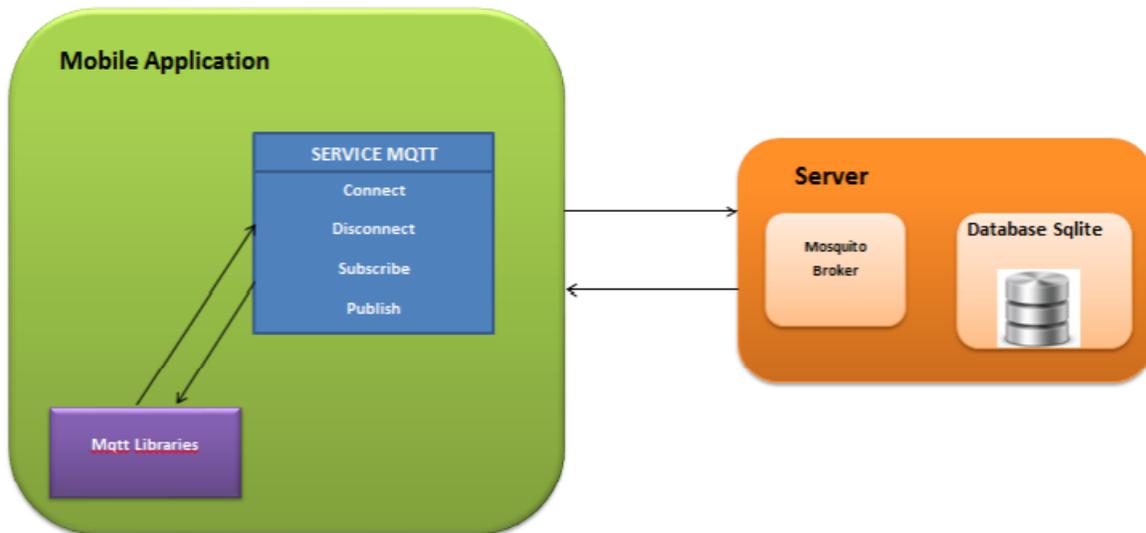
The definition of all components of both software and hardware the existing relationships between them are the most relevant aspects at the time of execution of a project. Next, all the architecture adjacent to this project will be described and specified.

2.1 MQTT service architecture

The service is within the mobile application, the service uses MQTT libraries to perform the functions of: Connect, Disconnect Publish, subscribe, QoS. By executing a .sh file the messages are stored in an SQLite database (see *Figure 1*).

Figure 1.

Service Architecture



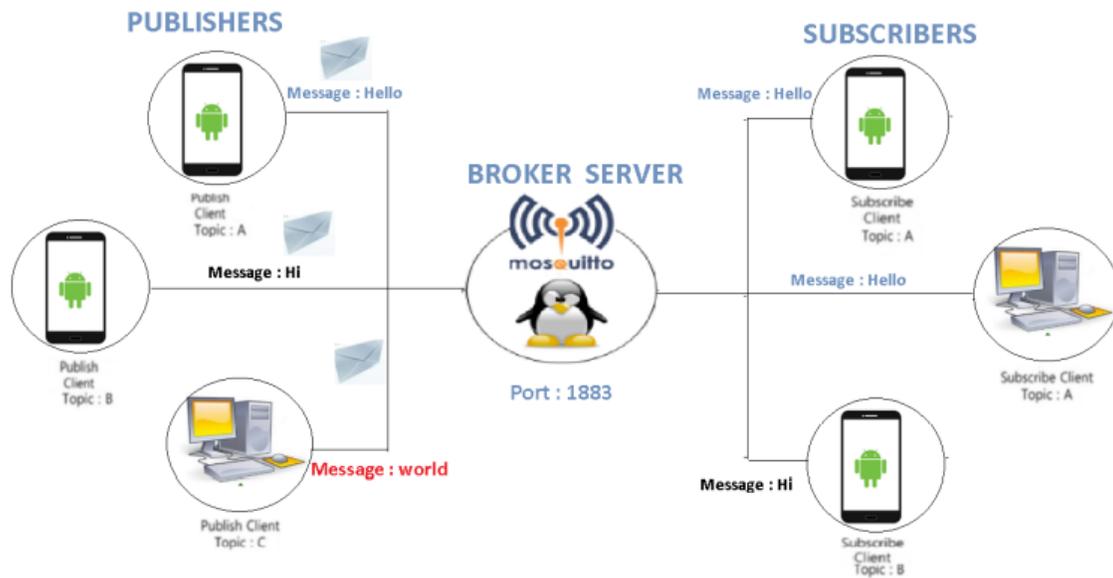
The central node that goes about as a server or representative, equipped for working up to 500000 messages. It is the merchant definitively the component responsible for dealing with the organization and sending the messages. Correspondence dependent on themes, the customer distributes the message, makes and the hubs that wish to get it should prefer it. The intermediary server associates with a SQLite information base utilizing a clump cycle to store the messages coming from the Distributers, MQTT customers preferred a point.

2.2 Architecture of MQTT message transmission between multiple clients

Customers under remote organization climate were tried under Android climate. The remote climate can be viewed as reasonable, since the correspondence from portable climate goes through 3G organization and arrives at specialist waiter (see *Figure 2*).

Figure 2.

The general architecture of MQTT message transmission between multiple clients



It investigates message misfortune and start to finish delay by gathering bundles among customer and server during the 5 minutes' estimation. With regards to start to finish delay, we use timestamps shaped as bundles move from the membership server to distribute the customer through the representative server. The retransmission demand bundles were counted when of five minutes to gauge message misfortune.

2.3 Implementation

In the first place, performed standard tests to comprehend the MQTT standards with online test agents (test.mosquitto.org, broker.mqttdashboard.com) and the order line: preferring points and distributing messages in those subjects. From that point onward, research was done to assess what was at that point made with respect to Java.

MQTT customers to characterize a beginning stage. We ran over various programming like Java customer Application Programming Points of interaction (APIs) and dealers, which are recorded in *Table 1*.

Table 1.*Implementation features*

Brokers	License	Java Client APIS
Active MQ	Open-source	Eclipse Paho - a Java client developed by the Eclipse Foundation (Iyer et al., 2018)
Hive MQ	Commercial	MQTT-client - a Fuse source Java MQTT client with a variety of API styles (Manh Pham et al., 2019)
IBM WebSphere Message broker	Commercial	IBM WebSphere MQ Telemetry provides a Java client API (Katsikeas et al., 2017)
Mosquito	Open Source	Free

2.3.1 Local Server MQTT

First import the repository package signing key:

- wget <http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key>
- sudo apt-key add mosquitto-repo.gpg.key

Then make the repository available to apt:

- cd /etc/apt/sources.list.d/

Then one of the following, depending on which version of Debian

- sudo wget <http://repo.mosquitto.org/debian/mosquitto-wheezy.listsudo>
- wget <http://repo.mosquitto.org/debian/mosquitto-jessie.list>

Update and install mosquitto

- apt-get update 2. apt-get install mosquitto

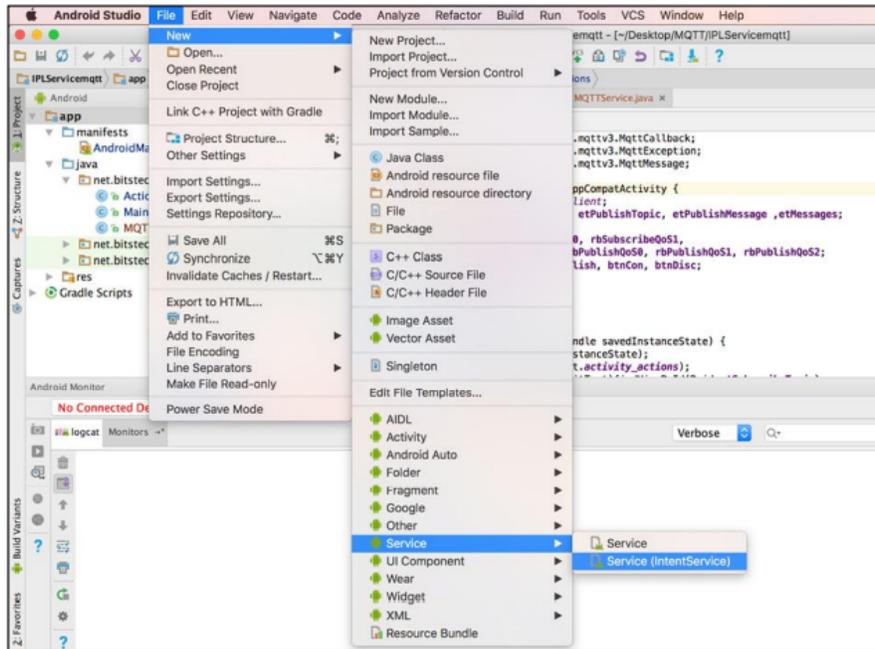
Run mosquito and see the status

- Mosquitto -c /etc/mosquitto/mosquitto.conf 2. Service mosquitto status

Create service on android studio, see *Figure 3*.

Figure 3.

Service on Android Studio



2.3.2 Message storage SQLite

Before configuring the MQTT server you need a requirement to have installed an Ubuntu or Debian server, with a non-root, sudo-enabled user and basic firewall.

Installing SQLite (Nemetz et al., 2018)

- `sudo apt-get update;`
- `sudo apt-get install sqlite3 libsqlite3-dev`

Create a database, if you are still in the `sqlite3` program, exit with `quit`. Quit in the SQLite flag, then run the command:

- `sqlite3 $HOME/MESSAGE.DB`

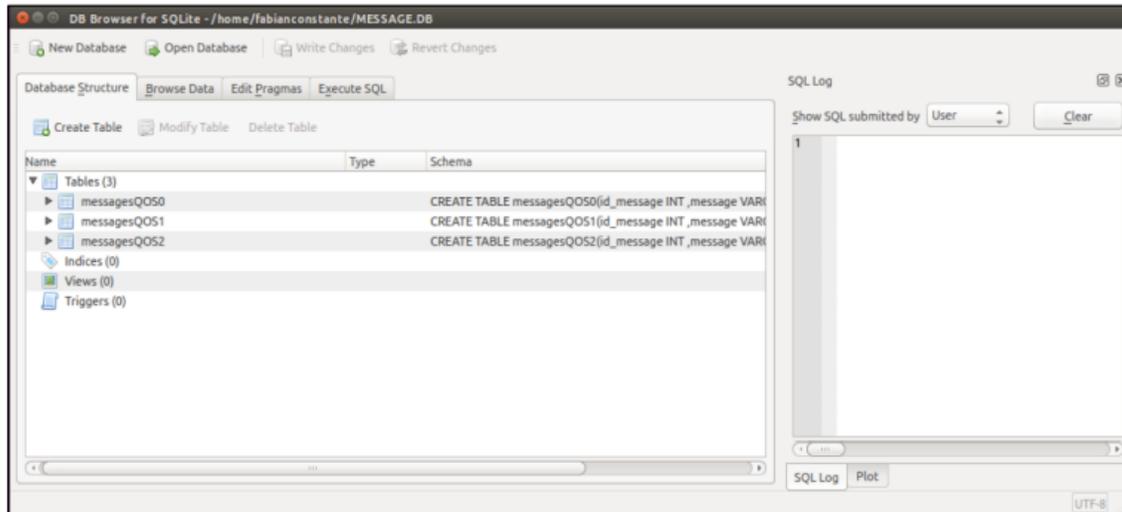
Creating tables corresponding to each QoS (see *Figure 4*).

- `sqlite3 $HOME/MESSAGE.DB "CREATE TABLE messagesQOS0(id_message INT ,message VARCHAR(200), topic VARCHAR(50),date text);"`
- `sqlite3 $HOME/MESSAGE.DB "CREATE TABLE messagesQOS1(id_message INT ,message VARCHAR(200), topic VARCHAR(50),date text);"`

- sqlite3 \$HOME/MESSAGE.DB "CREATE TABLE messagesQOS2(id_message INT ,message VARCHAR(200), topic VARCHAR(50),date text);";

Figure 4.

Message database



Results

3.1 Functional testing

Functional testing is a quality confirmation (QA) process and a kind of discovery testing that puts together its experiments with respect to the details of the product part under test. Capacities are tried by taking care of them input and inspecting the result, and inward program structure is seldom thought of (dissimilar to white-box testing). *Table 2*, portrays exhaustively the particular of the tests that are performed for usefulness

Table 2.

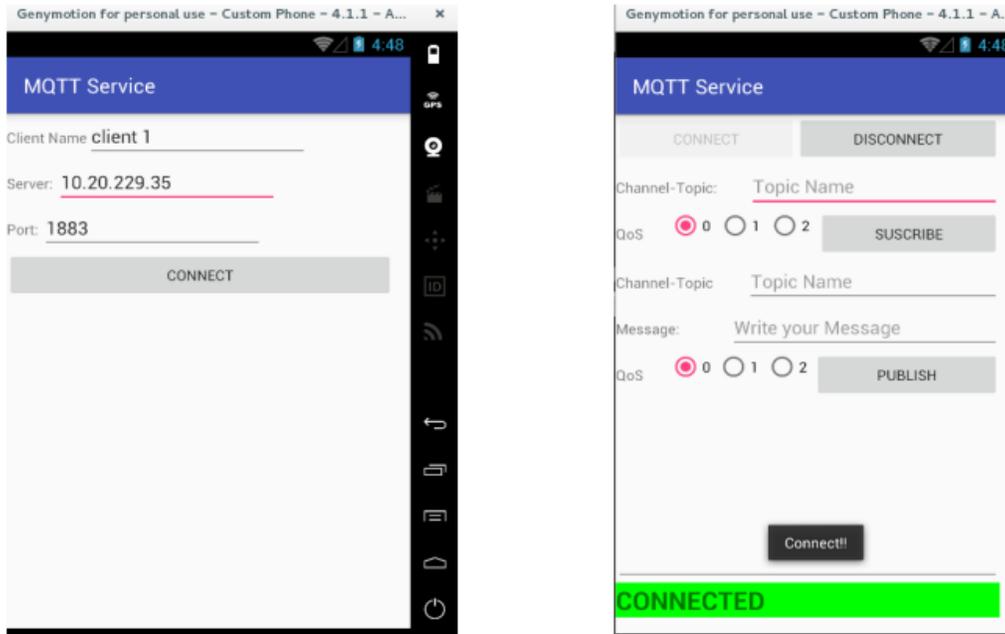
Usefulness test

Case	Event number	Test	Result
1. Connecting the Client to the MQTT Server	1	The client specifies the IP address of the host message broker and the port number assigned by MQTT to verify the status of the connection.	The connection to the MQTT server is successful.
	2	Display the successful connection message to the server	The displayed message indicates that the connection is successful.
2. Subscription to a topic and QoS	1	In the Android app, the subscription parameter and the quality parameter are sent, to press the "Subscribe"	Successful subscription to a topic
	2	Display message indicating subscription to the topic	The message indicates successful subscription
3. Message publishing and QoS	1	Enter the channel where the user has subscribed, write any message and press the "Publish" button, verify that the published message has been sent to the server.	The MQTT server displays the received message
	2	From the server send a reply message indicating that the message has been successfully published	The message indicates that data has been successfully published
4. Storing sent messages in a database	1	All messages sent in each QoS from different clients will be stored	Qo0 and Qo1 May have lost messages Qo2 This level has no loss in sent messages

1. Connecting the client to the MQTT Server. The connection to the server from the application through the service was satisfactory (see *Figure 5*).

Figure 5.

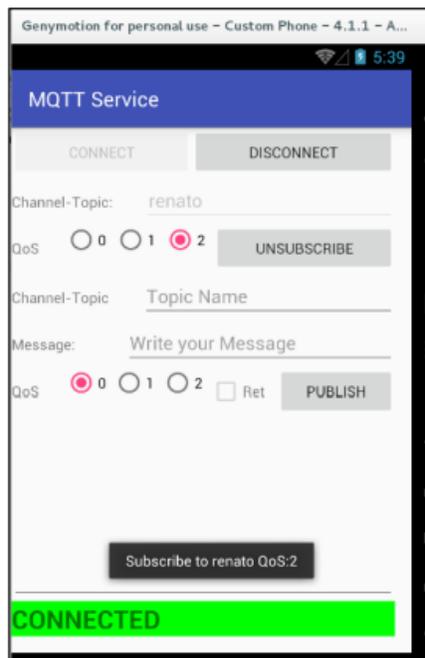
Connecting the Client to the MQTT Server



2. **Subscription to a topic and QoS.** - The subscription to a topic was satisfactory, this is shown in *Figure 6*.

Figure 6.

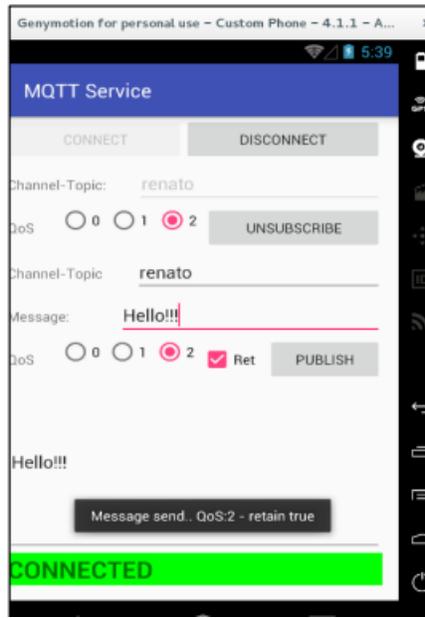
Subscription Interface to the topic



3. Message publishing and QoS. Messages publishing on a topic was satisfactory, see *Figure 7*.

Figure 7.

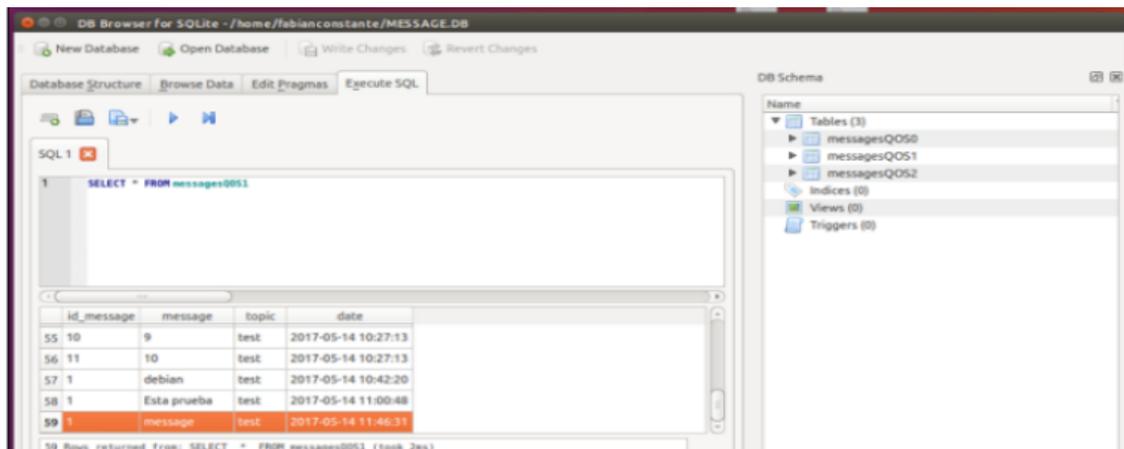
Publication of an MQTT message



4. Storing sent messages in a database. - The user submits, the message is automatically stored in a SQLite Database, this is possible when the .sh program is running. As shown in *Figure 8*.

Figure 8.

Storage of messages sent through the MQTT Service



Conclusions

Can be assured that MQTT is the protocol has used to provide new and revolutionary performance, it opens new areas for messaging use cases. As MQTT specializes in low-bandwidth, high-latency environments, it is considered an ideal protocol for machine-to-machine (M2M) communication.

We specified the design of a pushing notification system and discussed details of key techniques that make the system effective and easy to maintain. This message push system is based on MQTT protocol and makes it possible to push messages to clients in real-time.

From the functionality tests, positive results were obtained, users connected and sent messages correctly and quickly, this on a local server. While in the performance tests it was verified that in QoS1 a message was lost and the reception of the messages with the levels QoS0 and QoS2 were successful. This ensures that messages arrive correctly to users.



References

- Barata, D., Louzada, G., Carreiro, A., & Damasceno, A. (2013). System of acquisition, transmission, storage and visualization of Pulse Oximeter and ECG data using Android and MQTT. *Procedia Technology*, 9, 1265-1272.
- Hwang, H. C., Park, J., & Shon, J. G. (2016). Design and implementation of a reliable message transmission system based on MQTT protocol in IoT. *Wireless Personal Communications*, 91(4), 1765-1777.
- Iyer, S., Bansod, G. v., Praveen Naidu, V., & Garg, S. (2018). Implementation and Evaluation of Lightweight Ciphers in MQTT Environment. 3rd International Conference on Electrical, Electronics, Communication, Computer Technologies, and Optimization Techniques, ICEECCOT 2018, 276–281. <https://doi.org/10.1109/ICEECCOT43722.2018.9001599>
- Katsikeas, S., Fysarakis, K., Miaoudakis, A., Van Bemten, A., Askoxylakis, I., Papaefstathiou, I., & Plemenos, A. (2017). Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol. In *2017 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1193-1200).
<https://doi.org/10.1109/ISCC.2017.8024687>
- Lee, S., Kim, H., Hong, D. K., & Ju, H. (2013). Correlation analysis of MQTT loss and delay according to QoS level. In *The International Conference on Information Networking 2013 (ICOIN)* (pp. 714-717).
- Manh Pham, L., Nguyen, T.-T., & Tran, M.-D. (2019). A Benchmarking Tool for Elastic MQTT Brokers in IoT Applications. *International Journal of Information and Communication Sciences*, 4(4), 59. <https://doi.org/10.11648/J.IJICS.20190404.11>
- Nemetz, S., Schmitt, S., & Freiling, F. (2018). A standardized corpus for SQLite database forensics. *Digital Investigation*, 24, S121–S130. <https://doi.org/10.1016/J.DIIN.2018.01.015>
- Peiji, L., & Yanai, W. (2001). The Application of Push Technology in Mobile Internet *Communications World*, 31, pp. 31 - 32.
- Schiper, A., & Raynal, M. (1996). From group communication to transactions in distributed systems. *Communications of the ACM*, 39(4), 84-87. <https://dl.acm.org/doi/abs/10.1145/227210.227230>
- Shaoyue, H., Xiaodong, X., & Zuyuan, M. (2012). The Application of Active Push Technology in Mobile Collaboration Education [J]. *Modern Education Technology*, 4, 100-103.
- Silva, C., Toasa, R., Martinez, H. D., Veloz, J., & Gallardo, C. (2017). Secure push notification service based on MQTT protocol for mobile platforms. In *XII Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento y Congreso Ecuatoriano en Ingeniería de Software* (pp. 69-84).

Tang, K., Wang, Y., Liu, H., Sheng, Y., Wang, X., & Wei, Z. (2013). Design and implementation of push notification system based on the MQTT protocol. In *International Conference on Information Science and Computer Applications (ISCA 2013)* (pp. 116-119). Atlantis Press.

Zhang, W. M., Zhang, M., Bi, J., & Qin, Z. (2007). Instant messaging: The present and the future. *MINIMICRO SYSTEMS-SHENYANG-*, 28(7), 1162-1168. https://en.cnki.com.cn/Article_en/CJFDTotal-XXWX200707001.htm



Copyright (c) 2022 Cristian Mauricio Gallardo Paredes, Patricia del Rocío Rodríguez Fiallos y Francisco Javier Galora Silva



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)

Automatización con aplicación móvil para agendamiento de cita en peluquerías

Fecha de recepción: 2021-10-25 • Fecha de aceptación: 2021-01-04 • Fecha de publicación: 2022-02-10

Miguel Angel Pino Andrade¹

Investigador independiente, Ecuador

lucariodrago@gmail.com

<https://orcid.org/0000-0002-3044-7765>

Arthur Wilson Chávez Mora²

Investigador independiente, Ecuador

arthurchavez2399@gmail.com

<https://orcid.org/0000-0001-7600-345X>

Juan Daniel Carrion Robles³

Investigador independiente, Ecuador

dann191812@gmail.com

<https://orcid.org/0000-0001-6518-9105>

RESUMEN

El agendamiento de citas en peluquerías es un proceso que la mayoría de los locales no ha automatizado, ya que el servicio se brinda de acuerdo a cómo lleguen los clientes, esto es un problema, ya que puede tomar mucho tiempo en recibir el servicio de peluquería. En este sentido, se propone una aplicación capaz de agendar citas para evitar la espera, con el fin de optimizar el mencionado proceso. Se idea la aplicación desde dos perspectivas, la de usuario; es decir, clientes de la peluquería y la de administrador, el cliente solamente elegirá la hora, día y la sucursal en la cual desea ser atendido, el sistema automáticamente asignará la cita con el peluquero indicado, reservará ese tiempo para que no haya problemas, además de eliminar la espera, a su vez, el próximo cliente acudirá al término de la atención del cliente en turno. El desarrollo de la aplicación se realizará bajo Visual Studio XAMARIN, mediante las directrices de la metodología c, los resultados iniciales

demuestran la efectividad de la aplicación.

PALABRAS CLAVE: Xamarin, Visual Studio, Scrum, Github, Open Source

ABSTRACT

El agendamiento de citas en peluquerías es un proceso que la mayoría de los locales no ha automatizado, ya que el servicio se brinda de acuerdo con como lleguen los clientes, esto es un problema ya que puede tomar mucho tiempo en recibir el servicio de peluquería. En este sentido se propone una aplicación capaz de agendar citas para evitar la espera, con el fin de optimizar el mencionado proceso se idea la aplicación desde dos perspectivas, la de usuario, es decir clientes de la peluquería y la de administrador, el cliente solamente elegirá la hora, el día y la sucursal en la cual desea ser atendido, el sistema automáticamente asignará la cita con el peluquero indicado, reservará ese tiempo para que no haya problemas, además de eliminar la espera, a su vez el próximo cliente acudirá al término de la atención cliente en turno. El desarrollo de la aplicación se realizará bajo Visual Studio XAMARIN, mediante las directrices de la metodología Scrum, los resultados iniciales demuestran la efectividad de la aplicación.

KEYWORDS: Xamarin, Visual Studio, Scrum, Github, Open Source

Introducción

En los últimos años se ha evidenciado un creciente desarrollo de sistemas computacionales e informáticos para automatizar procesos de distintos negocios que se realizan de forma manual, y resolver distintos problemas que afectan a los usuarios, dentro de estas aplicaciones se pueden mencionar aplicaciones web (Nagy et al., 2008), móviles (Reyes et al., 2019), de escritorio (Leung, 2015), entre otras.

Existen distintos procesos que realizan ciertos negocios, que por más sencillos que parezcan, deben ser automatizados para mejorar el servicio al cliente, la literatura demuestra que existen soluciones para distintos tipos de servicio, desde la solicitud de un servicio de autos puerta a puerta (Artieda Guachamin et al., 2021), servicios de compra y venta a domicilio (Reyes Rivera et al., 2020), servicios de localización de mascotas (Burbano Ulloa et al., 2020), servicios de reconocimiento facial (Nuñez & Nuñez, 2020), sistemas de realidad virtual (Florez et al., 2021), pero muy poco se ha considerado el agendamiento de uno de los servicios más utilizados por la mayoría de personas en el mundo, como lo es el servicio de peluquería.

Si bien existen trabajos que mencionan este tipo de negocio, no se identifica un sistema actual en Ecuador que pueda automatizar el proceso de agendamiento de citas en una peluquería, y que se respeten los tiempos de los clientes, inicialmente San Martín Galaz (2005) desarrolla un “prototipo de aplicación informática que permita el registro de la información de clientes para facilitar la toma de decisiones en los procesos de gestión administrativa y financiera de la microempresa Peluquería Nuevo Estilo”.

Por otro lado, Luzardo Cabrera (2014) propone “desarrollar una aplicación Android para pedir cita previa en peluquerías, tomando como ejemplo Peluquerías Naranja, una franquicia de peluquerías canaria, esta aplicación permite al usuario del dispositivo, reservar hora o pedir cita en la peluquería, la cita se reservará siguiendo una serie de pantallas, en las que la información introducida de forma táctil se almacenará posteriormente en una base de datos online, asimismo, la aplicación será capaz de gestionar las citas previas disponibles permitiendo al usuario cancelar su cita previa”.

En Ecuador, la automatización de estos procesos aún es precaria, uno de los autores de este trabajo acudió a recibir un corte de cabello en el local de su confianza, cuando arribó vio que el peluquero estaba con un cliente en turno, mantuvieron una conversación en la cual esclarecieron en el horario que se desocuparía y pudiera acudir para recibir el servicio, sin tener que esperar. El mismo regresó en el horario pactado, finalmente pudo recibir el servicio.

De la situación descrita anteriormente, se visualizó la idea de crear una aplicación móvil con el fin de no tener que esperar o tener que regresar para ser atendido por un peluquero. La idea se fue materializando en la aplicación llamada “Peluquéate”, la misma está diseñada para controlar el tiempo de sus peluqueros y tener a la disposición de todos los clientes, suscritos a la aplicación, con un horario organizado que desemboque en un tiempo menor al empleado, antes del uso de la aplicación para recibir un servicio, lo llamaremos “corte de cabello”.

Metodología

La metodología aplicada en esta investigación es la observación directa, que se define como cuando “el investigador está ahí, en el lugar donde se desarrolla la acción y está preparado para registrar lo que está ocurriendo” (González, 2005), interpretando este concepto al contexto en que se desarrolla el presente estudio, lo que significa que los datos se obtienen directamente desde la fuente; es decir, los distintos negocios de peluquería ubicados en Quito, Ecuador.

En cuanto al desarrollo de software, se utiliza la metodología Scrum, a continuación, se detalla el proceso.

2.1 Modelo tradicional PDCA

El modelo para la elaboración de cada fase del proyecto, siguiendo la metodología Scrum, fue desarrollado por Walter Shewhart (Rigby et al., 2016), en los laboratorios de la empresa Bell y difundido por W. Edwards Deming. El modelo se centra en cuatro pasos: planificar, hacer, verificar, actuar (PDCA, por sus siglas en inglés). Cada acción se compone de una serie de acciones, las mismas que se describen a continuación de acuerdo a cada paso:

2.2 Definición de metodología

Se puede definir la palabra metodología como una disciplina que muestra los pasos a seguir para cada lineamiento que lleva la realización de un proyecto. Los elementos que tiene una metodología son: fases, control y evaluación, documentación, **métodos**, técnicas y herramientas (Montero et al., 2018).

Hay puntos en común de cualquier metodología como lo es el tiempo de aprendizaje, este no debe ser largo. A su vez, el apoyo en forma gráfica es muy importante, también se usa documentación estándar y se trabaja en pro de simplificar el trabajo y aumentar la productividad.

De acuerdo con su naturaleza, las metodologías son orientadas al flujo de información u objetivos, o son híbridas. Mientras que, de acuerdo con el grado de formalismo, pueden ser pesadas o ágiles.

Una metodología ágil se concentra en dar respuestas a falencias que han tenido lugar en las metodologías tradicionales. Se adapta a los requisitos que no se conoce, en lugar de predecirlos; es decir, no trabaja con dichos requisitos, una metodología tradicional sí y los predice, en este sentido Scrum es una metodología ágil (Amaya Balaguera, 2015).

2.3 ¿Qué es SCRUM?

La metodología ágil se caracteriza por trabajar en contextos que tienen incertidumbre, autoorganización, control moderado y transmisión del conocimiento (Sangama Oñate, 2020)

La metodología tiene como base crear ciclos breves para la realización del proyecto, las cuales son llamadas iteraciones, bajo la metodología mencionada se llama “Sprints”, la cual tiene como fases:

- Planificación del *backlog*: aquí se define un documento en el cual se plasman los requisitos del sistema, se mantiene una reunión llamada “Sprint 0”, en la que se definen los objetivos y el trabajo a realizar.
- Seguimiento del sprint: se hacen reuniones para evaluar el avance de las tareas en base a estas tres preguntas: ¿qué se hizo desde la reunión anterior?, ¿cuál es el trabajo hasta la próxima reunión? y ¿qué inconvenientes han surgido y qué hay que solucionar para continuar?
- Revisión del sprint: al finalizar el sprint se hace una reunión para revisar los avances y el incremento generado, además, se presentarán los resultados finales.

También se toman en cuenta los roles que tienen las personas comprometidas con el proyecto, estos pueden ser

- *Product owner*: es un intermediario con el cliente, toma las decisiones de los clientes, se encarga de colocar sus ideas en el *product backlog* y ordenarlas por su prioridad.
- Scrum Master: encargado comprobar la metodología y el modelo para que puedan ser funcionales. Elimina los inconvenientes para el proceso.
- Equipo de desarrollo: está conformado de 5 a 9 personas y tienen autoridad para tomar las decisiones que encaminen al proyecto a su objetivo.

A su vez, las personas no involucradas directamente, que vendrían siendo:

- Usuarios: es el cliente como tal.
- Stakeholders: las personas a las que el programa les producirá un beneficio.
- Managers: toman las decisiones finales participando en la selección de requisitos y objetivos.

También es importante tener en cuenta los elementos de Scrum, tales como:

- Product Backlog: es la lista de necesidades de los clientes.
- Sprint Backlog: lista de tareas que se realizan en un “sprint”.
- Incremento: parte añadida y totalmente operativa en un “sprint”.

2.4 Metodología en “peluquéate”

De acuerdo con las directrices de la metodología Scrum y los elementos en los cuales está basado, se ha decidido utilizar dicha metodología para el desarrollo del proyecto; sin embargo, al ser un proyecto desarrollado por visión e iniciativa de los autores, y no por petición de una

empresa o persona natural para fines comerciales, los roles se han tomado solamente de “Scrum Master” para Arthur Chávez y “equipo de desarrollo”, que está conformado por Miguel Pino y Daniel Carrión. Desde la perspectiva de las personas no involucradas, los roles de “usuarios” y “manager” serán realizadas por los tres nombres mencionados con anterioridad.

Resultados

3. 1 Proceso de desarrollo y conceptualización

El gestor utilizado para la base de datos del proyecto es PostgreSQL (Emc2Net, s.f), un gestor *open source* de herramientas avanzadas. Este gestor puede equipararse a otros grandes gestores como Oracle, SQL Management (Microsoft). PostgreSQL tiene dos grandes ventajas, tiene capacidad para trabajar con grandes cantidades de datos y tiene una licencia dual, en caso de que el gestor sea usado para fines comerciales, se debe realizar un pago, si es un proyecto no comercial, no habría problema con su uso y se descarta un posible pago.

Las características que hacen a este gestor de base de datos tan robusto son:

- Alta concurrencia, tienen la capacidad de atender a varios clientes el mismo tiempo y entregar la información sin ningún tipo de bloqueos.
- Soporte para múltiples tipos de datos de manera nativa, a diferencia de otros gestores ofrecen tipos de datos como ip, mac, decimales con precisión configurable, figuras geométricas, entre otros.
- Soporte a triggers, generan acciones cuando se disparan.
- Vistas, se consultan datos de manera diferente a como se almacenan.
- Trabaja con sus datos como si fueran objetos.
- Soporta gran cantidad de lenguajes de programación.

Durante este desarrollo también se hizo la inclusión de Visual Studio, un programa realizado por Microsoft para el desarrollo de plataformas web, móvil y aplicaciones de escritorio. Parte de Visual Studio es Xamarin, y es la plataforma que se va a utilizar para la elaboración de la aplicación (Microsoft, s.f). Xamarin es una plataforma de gran rendimiento para creación de aplicaciones móviles en sistemas operativos de Apple (iOS), Google (Android) y Windows phone con “.net”. La aplicación puede ser desarrollada en Windows o Macintosh.

Por otra parte, Git es un repositorio y sistema de control de versiones, fue desarrollado por Linus Torvalds, la licencia de github es libre. Esta herramienta dispone de herramienta de línea de comandos llamada “git bash” y de escritorio (Blischak et al., 2016).

GitHub es un servicio comercial de alojamiento de repositorios Git, fue creado en el año 2008. GitHub tiene una interfaz web. Permite al usuario crear repositorios vacíos o por clonación de otro repositorio de GitHub, enviar solicitudes de cambio entre repositorios que estén en GitHub, los repositorios creados son de acceso público por defecto, solamente si se contrata la versión

de paga y/o si cumplen ciertos requisitos, se puede tener como privado el repositorio del código. La herramienta proporciona un completo sistema de gestión de comentarios a cada repositorio, cuadro de control con grafos sociales, inclusive una página web propia (Blischak et al., 2016).

Laravel es un *framework* generador de código donde se trabaja con modelo, controlador en donde se implementan rutas para la comunicación de red entre aplicación. Esto facilita la construcción del *backend* de la aplicación móvil “peluquéate”, dado que se manejan múltiples vistas y poder controlar cada una de ella, solo con un lenguaje nativo como Php representaría un incremento en el tiempo de desarrollo y de entrega (u/wjgilmore2014, 2015).

Entendiendo la importancia del manejo de todos los servicios que se requieren para poder implementar una aplicación de consumo a nivel usuario, se ha decidido trabajar en un entorno Linux (Loshin, 2021), en este caso, Ubuntu 20.04 *long time supports*, en lugar de un virtualizador o ejecutor Linux como XAMPP, para esto hay que recordar el IDE de la aplicación es Visual Studio 2022, el mismo que no corre en un servidor Linux, a solución de esto se implementó el uso de la última característica de Windows 10 y 11 llamada WLS – “windows subsystem for linux”.

Esto permite tener todo el Kernal de Linux mediante una terminal que comparte los recursos de Windows, sin tener afectación o choques de red ni de rendimiento.

Esta característica impulsada por Windows, permitió que se trabajara en dos entornos al mismo tiempo de forma armonizada, usando Linux para la gestión de servidores (apache 2) y Windows para el desarrollo de la aplicación.

Php, siendo un lenguaje de programación enfocado al desarrollo web, tiene características especiales que permite la gestión y control de los procesos de consulta entre el *frontend* y la base de datos, por ello se ha decidido utilizarlo como lenguaje principal de gestión *backend*, conectando entre controladores, modelo y, siendo un puente entre la base de datos y la aplicación móvil que actualmente está desarrollada en Xamarin - C# (PHP, s.f).

Como el flujo de desarrollo ideal es poder trabajar en paralelo lo apartado visual con el control, se ha optado por usar Postman para discriminar la necesidad de las vistas al momento del desarrollo del *backend*; es decir, esto permitió asegurar que las conexiones y los controles estén funcionando correctamente, sin la necesidad de depender del *frontend* y retrasar los sprints (Postman, s.f).

Entorno de pruebas de conexión con la base de datos:

Traer los datos (*Get*), registrar (*Post*), actualizar (*Put*), eliminar (*Delete*).

Por su parte, ZeroTier-One es una herramienta que permite crear una red privada virtual, ya que como no se tienen servidores remotos por el costo, se ha decidido montar una laptop como servidor, conectado a esta red virtual y así acceder a la aplicación en cualquier parte del mundo que tenga la aplicación y los permisos de acceso a esta red (Curman, 2021).



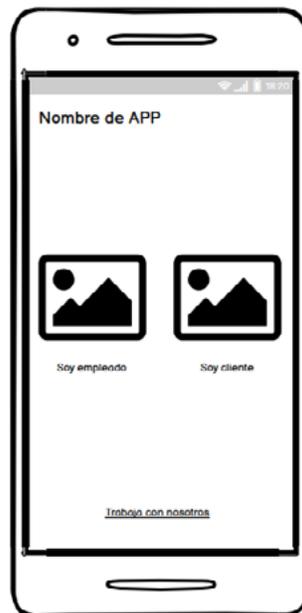
Esto permitió también realizar pruebas desde un teléfono Android físico, ya que ZeroTier es multiplataforma entre Linux, Windows, Android e IOS.

A continuación, se dará a conocer el proceso que se siguió para programar la aplicación. Esto se explicará en “Sprints”, de acuerdo con la metodología elegida para el desarrollo.

- **Sprint Cero:** se tuvo una breve reunión en la cual se dieron a conocer los objetivos, el proceso a realizar y la organización que se daría a lo largo del proyecto. Se definió el gestor de datos con el cual se trabajaría, la plataforma de desarrollo con sus respectivas librerías, el programa para la modulación de datos.
- **Primer Sprint:** se realizó el prototipo de las primeras vistas con Balsamiq (*Figura 1*), en paralelo se realizaba el diseño de la base de datos de las tablas que intervendrían en dichas vistas. El encuentro tuvo lugar digitalmente a través de la herramienta “meet” de Google en la semana del 23 de octubre de 2021.

Figura 1

Prototipo de la primera vista



Fuente: Balsamiq

- **Segundo Sprint:** en el segundo encuentro se siguió con la definición de las tablas de la base de datos, siguiendo las pautas de las vistas (*Figura 2*). El encuentro tuvo lugar digitalmente a través de la herramienta “meet” de Google en la semana del 30 de octubre de 2021.

Figura 2

Prototipo de la segunda vista

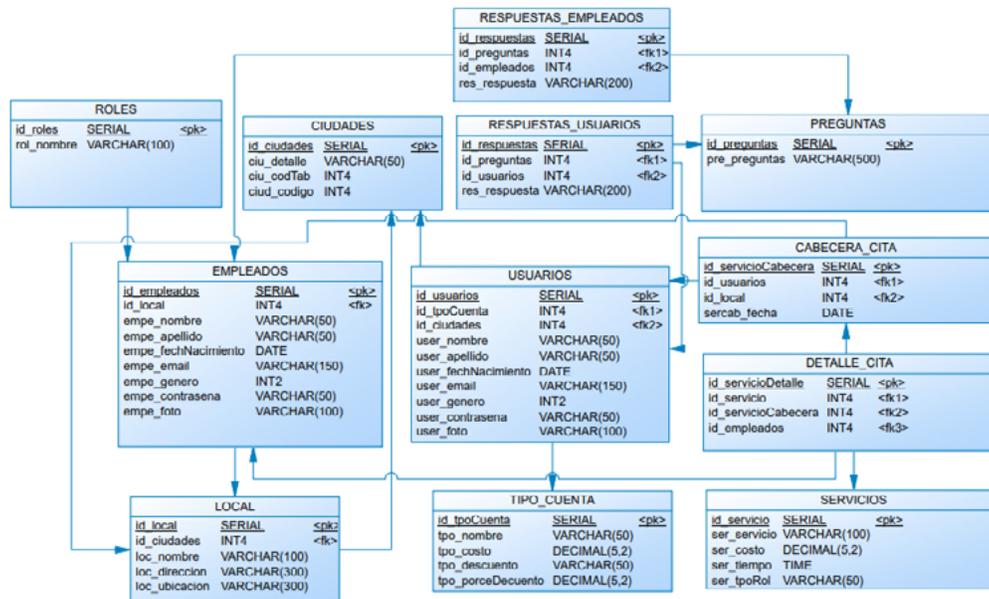


Fuente: Balsamiq

- **Tercer Sprint:** en esta reunión se finalizó el diseño de la base de datos, además del prototipo de todas las vistas para empezar el desarrollo en la plataforma Visual Studio (*Figura 3*). El encuentro tuvo lugar digitalmente a través de la herramienta “meet” de Google en la semana del 06 de noviembre de 2021.

Figura 3

Primera versión de la base de datos

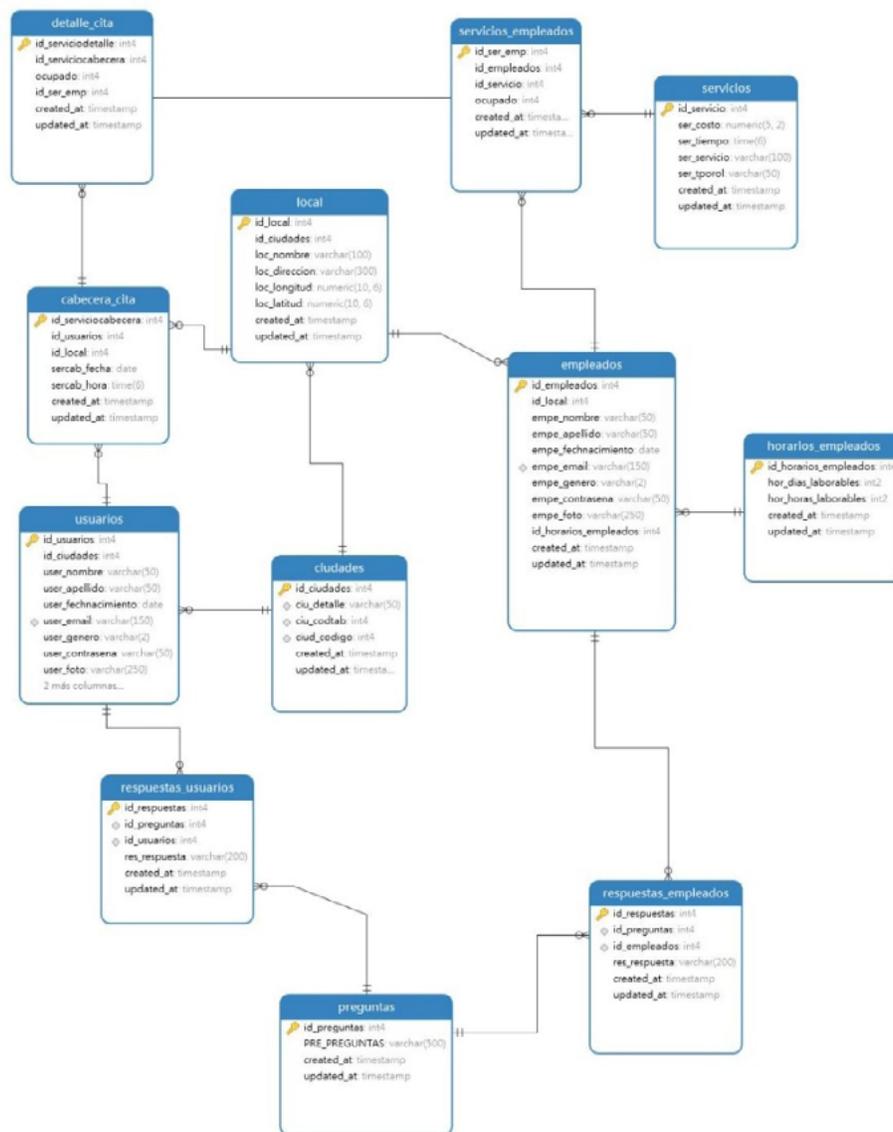


Fuente: Power Designer

- Cuarto Sprint:** desde el tercer Sprint, se realizó el *front end* de la aplicación, se estableció una reunión para definir los detalles del diseño y la conexión entre ventanas en la programación del *back end* y la base de datos desde la computadora de uno de los integrantes del equipo, la computadora actuará como servidor de la aplicación. El encuentro tuvo lugar digitalmente a través de la herramienta “meet” de Google en la semana del 13 de noviembre de 2021.
- Quinto Sprint:** en el penúltimo Sprint se corrigieron algunos detalles en el diseño del *front end* y la base de datos tuvo cambios. Finalmente, en la *Figura 4* se encuentra el diseño físico de la base de datos. El encuentro tuvo lugar digitalmente a través de la herramienta “meet” de Google en la semana del 20 de noviembre de 2021.

Figura 4

Versión definitiva del diseño de la base de datos



Fuente: PostgreSQL

- **Sexto Sprint:** se verificó el producto terminado y completamente funcional. Los pormenores de la aplicación se mostrarán en las gráficas siguientes.

3.2 Proceso de agendamiento

Desde la aplicación, para poder realizar el agendamiento de una cita, se hará lo siguiente:

1. El usuario descarga la aplicación en su dispositivo Android (la aplicación será desarrollada para el sistema operativo en mención, en su fase de prototipo).

2. Se debe registrar en la aplicación. Si está ya registrado, iniciar sesión con sus credenciales.
3. Para registrarse debe especificar su: nombre, apellido, correo, fecha de nacimiento, provincia y ciudad de residencia, género y crear una contraseña (el usuario será el correo electrónico ingresado). Por último, especificar tres preguntas de seguridad con sus respectivas respuestas, éstas servirán para generar una nueva contraseña en el escenario que el cliente la haya olvidado.
4. Una vez iniciada la sesión se podrá visualizar la sección destinada a publicidad y nos otorgará la opción para agendar la cita en la peluquería que se desee.
5. En la parte superior izquierda, se podrá ver tres líneas paralelas horizontales, a través de ésta se direccionará a un menú desplegable que contiene las siguientes opciones: mi perfil, mis citas, agendar citas, acerca de y cerrar sesión. A través de dicho menú también se accederá a la vista para agendar una cita.
6. Dentro de la vista “Nueva cita” se va a seleccionar la sucursal de su conveniencia. En la opción “Ver Mapa” se direccionará a la aplicación “Google Maps” para mostrar la ubicación exacta de la sucursal.
7. En la misma vista seleccionar la hora y la fecha en la que se desea agendar la cita.
8. En la vista siguiente se debe seleccionar el servicio al que se desea acceder y el colaborador con el cual se tendrá la cita. Se puede seleccionar cuantos servicios desee. Después de elegir cada servicio en la parte inferior se mostrará el valor de cada uno.
9. Para finalizar, se direccionará a la vista “Mis citas”, donde se mostrará la cita que se acaba de gestionar.
10. El peluquero recibirá una notificación a la hora de la cita que le mostrará la información detallada.

3.3 Realizado sin la aplicación por parte del usuario

Este escenario está sesgado a las personas que no utilizan la aplicación. El cliente llegará a la peluquería, solicitará una cita y la persona que se encuentre en la recepción de la sucursal agendará la cita mediante la página web, ya que la aplicación administrativa dispone de la opción de agendar citas sin pasar por la aplicación cliente. La cita se agendará a nombre del cliente.

3.4 Pantallas de la aplicación

A continuación, se verán las pantallas de la aplicación. El *splash screen* es una pantalla de carga que permite dar a conocer al usuario sobre la marca del aplicativo entre colores y logotipo (Figura 5).

Figura 5

Pantalla de ingreso a la aplicación



De acuerdo con la *Figura 6*, como en la construcción de la base de datos, se está discriminando los usuarios de los empleados, usando una pantalla de pre login.

Figura 6

Pantalla para escoger el tipo de usuario



La ventana de login permite el ingreso de los usuarios con su cuenta personalizada y en caso de no tener cuenta, se pueden registrar, como se muestra en la *Figura 7*.

Figura 7

Pantalla de cliente para ingresar las credenciales del usuario



Como se observa en la *Figura 8*, en el formulario de registro se ha controlado la selección de ciudades por provincia, esto permite que solo se muestren las ciudades, de acuerdo con la provincia seleccionada por el usuario, ahorrando tiempo de búsqueda.

Figura 8

Pantalla de cliente para el registro en el sistema



The screenshot shows a mobile application interface for 'PELUQUEATE'. At the top, there is a dark blue header with the app's logo and name. Below the header, the title '¡Regístrame!' is displayed. The form contains the following fields: 'Muestra' (with 'Uisrael' selected), 'Fecha de nacimiento:' (with '23/11/2021' entered), an email field (with 'muestra.uisrael@google.com' entered), 'Provincia:' (with 'Guayas' selected), 'Ciudad:' (with 'Seleccione ciudad' as a placeholder), 'Género:' (with 'Seleccione género' as a placeholder), 'Contraseña', and 'Repetir contraseña'. A dark blue button labeled 'SIGUIENTE' is positioned at the bottom right of the form.

Con las preguntas y respuestas se asegura el método de recuperación de contraseña, esto seleccionando tres preguntas de un banco de seis (*Figura 9*).

Figura 9

Pantalla de cliente para completar el registro en el sistema con las preguntas de seguridad

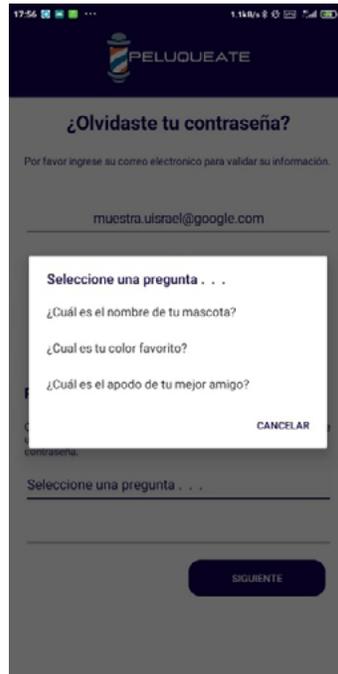


The screenshot shows a mobile application interface for 'PELUQUEATE' titled 'Preguntas de seguridad'. Below the title, there is a paragraph of text: 'Para asegurar tu cuenta es necesario que respondas las siguientes preguntas, así podrás recuperar tu cuenta en caso de que se te haya olvidado la contraseña.' Below this text, there are three dropdown menus, each with the placeholder text 'Seleccione una pregunta . . .'. At the bottom of the screen, there is a dark blue button.

Lo importante de estas vistas es que una vez seleccionado una pregunta (*Figura 10*), no se mostrará más para que escoja otra.

Figura 10

Pantalla de cliente generar una nueva clave ingresando el correo y respondiendo una pregunta



Una vez ingresadas las credenciales se muestra el “Home”, como se ve en la *Figura 11*, el cual fue pensando específicamente para promocionar cada uno de los servicios de las peluquerías.

Figura 11

Pantalla posterior del ingreso exitoso



De acuerdo con la Figura 11, se cuenta con un menú desplegable, donde están las opciones de: mi perfil, mis citas, agendar citas, acerca de y cerrar sesión

Figura 12

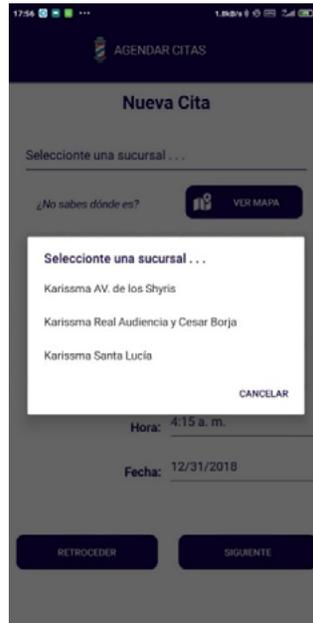
Pantalla que muestra las opciones una vez iniciada sesión



Para agendar una cita se cuenta un listado de todas las sucursales vigentes (*Figura 13*), dependiendo a la provincia y ciudad donde el usuario se había registrado, esto permite acceder al usuario al Google Maps, en caso no conocer cómo llegar.

Figura 13

Pantalla que muestra las pautas para agendar una cita



Como se observa en la *Figura 14*, esta vista lo que hace es mostrar la información registrada de cada usuario, teniendo la opción de modificar ciertos datos, en caso de ser necesario por el usuario, accediendo a la galería o cámara del celular, si es que el usuario desea colocar una foto de perfil, en vez del predeterminado un avatar.

Figura 14

Pantalla que muestra el perfil del cliente y permite editarlo



En caso de ser un empleado, cambiará la forma de presentación de los datos, mostrando, en este caso, la sucursal a la cual el empleado pertenece, esto permite que en un futuro este pueda ser reubicado en cualquier otra sucursal vigente, dentro del sistema

Figura 15

Pantalla que muestra el perfil del empleado y permite editarlo



Conclusiones

El desarrollo móvil multiplataforma representa un gran crecimiento a nivel de mercado, por lo que se puede abarcar un mayor número de dispositivos de diferentes sistemas operativos, permitiendo que la aplicación desarrollada alcance un número considerable de usuarios, que validarán o rechazarán la aplicación.

Xamarin Form, siendo una tecnología *open source* soportada por Microsoft, muestra una increíble capacidad de desarrollo por la cantidad de paquetes y recursos que se pueden utilizar para desarrollar aplicaciones, de acuerdo con la necesidad del cliente o idea de negocio. La programación es una recopilación de varios conceptos afianzados a lo largo de la carrera, como programación orientada a objetos, ing. software y bases de datos, todo con el fin de construir un producto sostenible y con buenas prácticas en el entorno de la tecnología.

La aplicación “Peluquéate” se encuentra en su fase inicial, los resultados experimentales demuestran que los negocios (peluquerías) consolidados, y que manejan cantidad masiva de clientes, se podrían beneficiar con la automatización en su proceso de agendamiento de citas. No obstante, para complementar los resultados de la fase experimental, y conocer si es factible la implementación del software en mención para cada empresa, se debe realizar un estudio de viabilidad.

Referencias

- Amaya Balaguera, Y. D. (2015). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología*, 12(2). <https://doi.org/10.18270/rt.v12i2.1291>
- Artieda Guachamin, C. A., Reza Morales, J. S., & Rojas Cisneros, H. D. (2021). Sistema móvil para la oferta de recorridos con rutas fijas. *REVISTA ODIGOS*, 2(1), 21–40. <https://doi.org/10.35290/ro.v2n1.2021.381>
- Blischak, J. D., Davenport, E. R., & Wilson, G. (2016). A Quick Introduction to Version Control with Git and GitHub. *PLOS Computational Biology*, 12(1), e1004668. <https://doi.org/10.1371/journal.pcbi.1004668>
- Burbano Ulloa, E. A., Constante Molina, M. A., Hidalgo Guamán, L. M., & Moya Chiluiza, F. A. (2020). Prototipo móvil para la geolocalización de mascotas callejeras. *REVISTA ODIGOS*, 1(3), 77–96. <https://doi.org/10.35290/ro.v1n3.2020.372>
- Curman, M. (2021). *Automatizirani sustav prikupljanja podataka o objektima održavanja primjenom IIoT senzora*. [Tesis doctoral, University of Zagreb]. <https://urn.nsk.hr/urn:nbn:hr:235:485182>
- Emc2Net. (s.f). PostgreSQL-ES. <https://e-mc2.net/es/postgresql-es>
- Florez, C., Quevedo, W., Galora, F., & Toasa, R. (2021). Performance of WebGL standard for displaying 3D applications on mobile devices. In *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1-6, IEEE. <https://doi.org/10.23919/CISTI52073.2021.9476391>
- González, E. (2005). La observación directa base para el estudio del espacio local. *Geoenseñanza*, 10(1), 101-105. <https://www.redalyc.org/pdf/360/36010107.pdf>
- Leung, T. (2015). Creating Desktop Applications. In *Visual Studio LightSwitch 2015*, pp. 73–109. Apress. https://doi.org/10.1007/978-1-4842-0766-6_4
- Loshin, P. (octubre, 2021). *Linux operating system*. <https://searchdatacenter.techtarget.com/definition/Linux-operating-system>
- Luzardo Cabrera, A. (2014). *Aplicación Android para pedir cita previa en peluquerías*. [Tesis de grado, Universidad de las Palmas de Gran Canaria]. https://accedacris.ulpgc.es/bitstream/10553/20974/1/0727263_00000_0000.pdf
- Microsoft. (s.f). *Visual studio*. <https://visualstudio.microsoft.com/es/>
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes Revista Multidisciplinaria de Investigación*, 2(17). <https://doi.org/10.31876/RE.V2I17.269>

- Nagy, P. G., Konewko, R., Warnock, M., Bernstein, W., Seagull, J., Xiao, Y., George, I., & Park, A. (2008). Novel, web-based, information-exploration approach for improving operating room logistics and system processes. *Surgical Innovation*, 15(1), 7–16. <https://doi.org/10.1177/1553350608316573>
- Nuñez, A. v., & Nuñez, L. N. (2020). Reconocimiento facial para el encendido automático de vehículos basado en Raspberry Pi. *REVISTA ODIGOS*, 1(2), 53–68. <https://doi.org/10.35290/RO.V1N2.2020.326>
- PHP. (s.f). ¿Qué es PHP? <http://php.net/manual/es/intro-what-is.php>
- Postman. (s.f). API Platform. <https://www.postman.com/>
- Reyes, J., Morales, L., Aldas, D., Reyes, R., & Toasa, R. (2019). Real Time Automatic Andon Alerts for Android Platforms Applied in Footwear Manufacturing. In *Computer and Communication Engineering*, pp. 43–56. https://doi.org/10.1007/978-3-030-12018-4_4
- Reyes Rivera, A., Vallejos Cango., & Quintana García, D. (2020). Nueva realidad: compras en línea y a domicilio. *REVISTA ODIGOS*, 1(3), 47–60. <https://doi.org/10.35290/ro.v1n3.2020.370>
- Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). The Secret History of Agile Innovation. *Harvard Business Review*, 4.
- San Martin Galaz, M. (2005). *Sistema de gestión de clientes para Peluqueria Nuevo Estilo*. [Tesis de grado, Universidad de Talca]. <http://dspace.otalca.cl/handle/1950/5230>
- Sangama **Oñate, A.** (2020). *Metodologías ágiles Scrum, XP, SLeSS, Scrumban, HME, Mobile-D y MASAN empleadas en la industria de dispositivos móviles: Un contraste en favor de la industria del desarrollo móvil*. [Tesis de grado, Universidad Peruana Union] <http://hdl.handle.net/20.500.12840/3906>
- u/wjgilmore2014. (23 de julio de 2015). *How Laravel 5 Prevents SQL Injection, Cross-Site Request Forgery, and Cross-Site Scripting*. EasyLaravelbook. <https://www.easylaravelbook.com/blog/how-laravel-5-prevents-sql-injection-cross-site-request-forgery-and-cross-site-scripting/>

Copyright (c) 2022 Miguel Angel Pino Andrade, Arthur Wilson Chávez Mora y Juan Daniel Carrion Robles



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)



Diseño de aplicación móvil para promoción de productos

Fecha de recepción: 2021-11-27 • Fecha de aceptación: 2022-01-21 • Fecha de publicación: 2022-02-10

Andrea Guissela Pérez Lozada¹

Graficar Publicidad Gramedia, Ecuador

andre_g0397@hotmail.com

<https://orcid.org/0000-0001-7908-091X>

Edison Moreno Jurado²

SRI - Analista de Seguridad, Ecuador

mondeo_eddy33@hotmail.com

<https://orcid.org/0000-0001-6278-7097>

Luis Alberto Pérez Viana³

Telconet Latam, Ecuador

dark200486@gmail.com

<https://orcid.org/0000-0002-9844-2274>

RESUMEN

La idea de este trabajo surge al realizar un análisis de la necesidad que contemplan los emprendedores para dar a conocer sus productos o servicios. De esta forma se logró plantear el objetivo del proyecto que fue desarrollar un prototipo para dispositivos móviles Android, que permita crear un catálogo de productos. Con base en métodos de inferencia, fuentes bibliográficas y encuestas resumidas sobre las preferencias comerciales de los consumidores y las marcas favoritas, tres aspectos que son importantes para los usuarios cuando compran son: direcciones, precios y ubicación, que son los puntos clave para diseñar una aplicación llamada Deals app, el cual brindará información a los usuarios para que puedan filtrar las compras o servicios de acuerdo a sus necesidades y resolver los inconvenientes que impidan una compra satisfactoria, fomentando así el uso de las nuevas tecnologías que brindan un medio de publicidad para comerciantes y negocios y configuran una plataforma de consulta para los consumidores.

PALABRAS CLAVE: producto, comercial, móvil, catálogo, asesor de compras

ABSTRACT

The idea of this work arises from an analysis of the need that entrepreneurs have to publicize their products or services. Thus, the objective of the project was to develop a prototype for Android mobile devices, which allows the creation of a product catalog. Based on inference methods, bibliographic sources and summarized surveys on the commercial preferences of consumers and favorite brands, three aspects that are important for users when shopping are: addresses, prices and location, which are the key points to design an application called Deals app, which will provide information to users so they can filter purchases or services according to their needs and solve the problems that prevent a successful purchase, thus promoting the use of new technologies that provide a means of advertising for merchants and businesses and set up a consultation platform for consumers

KEYWORDS: product, commercial, mobile, catalogue, purchasing advisor

Introducción

Hoy en día, la sociedad enfrenta cada vez más necesidades en educación, sociedad, trabajo y otros campos, lo cual ha generado un gran impacto en las personas, experimentando un gran auge tecnológico que los obliga a capacitarse. En un mundo cada vez más competitivo, las herramientas tecnológicas se han convertido en imprescindibles para mejorar y perfeccionar los procesos de trabajo y comunicación (Piernas, 2014).

Surgen preocupaciones entonces, ¿cuáles son los factores influyentes que pueden conducir a un consumo más eficiente? Las marcas de productos luchan por establecerse en la mente de los consumidores, especialmente de los usuarios de Internet, que tienen una presencia activa en el mercado. Hoy en día hay un crecimiento exponencial de marketing por redes sociales y aplicaciones móviles, el uso de estas ha crecido durante la pandemia (Villacís et al., 2010).

Actualmente, en la literatura existen trabajos relacionados al desarrollo de aplicaciones móviles que solventan problemas cotidianos, aplicaciones para el teletrabajo (Reyes et al., 2020), para registro y búsqueda de mascotas (Burbano et al., 2020), para servicio de taxis (Artieda et al., 2021), turismo (Silva et al., 2018), entre otros, lo que es una base para el desarrollo de nuevas soluciones mediante aplicaciones móviles

En este contexto, las marcas de productos buscan expandir su presencia en línea al llegar a más personas a través de tantos canales en línea como sea posible, como las redes sociales y las aplicaciones, siempre que mantengan el código de la empresa.

Para protegerlos, surge la idea de un modelo de aplicación móvil que permita a los usuarios consultar, registrar y procesar información de productos en tiempo real en cualquier lugar, y les permita mejorar sus compras de zapatos y accesorios, ropa y consumibles, equipos, servicios, brindando información de acuerdo a las necesidades del usuario, ya sea direcciones, precios y/o ubicaciones.

Metodología

El presente proyecto tiene como distintivo el desarrollo de un catálogo como medio viable, incorporando funciones que facilite la publicidad de productos ofertados por comerciales y empresas del Ecuador.

Por lo tanto, se utiliza el método deductivo (Prieto, 2017) en base a las siguientes premisas:

- Los sistemas informáticos permiten automatizar procesos.
- Este proyecto propone el desarrollo de un sistema informático.
- El sistema propuesto automatizara el proceso de publicidad.

Se ha desarrollado una aplicación para dispositivos móviles Android (Verborgh et al., 2013), utilizando servicios web y el formato JSON para interpretar los datos desde una fuente disponible en una base de datos externa.

2.1 Arquitectura de la aplicación

La arquitectura definida en el desarrollo va a ser híbrida, tipo Modelo, Vista, Controlador (MVC) (Díaz & Fernández, 2012), para el teléfono (Ver *Figura 1*), por capas para el modelado de datos (Vásquez, 2010) (Ver *Figura 2*).

Figura 1.

Arquitectura de la aplicación Móvil

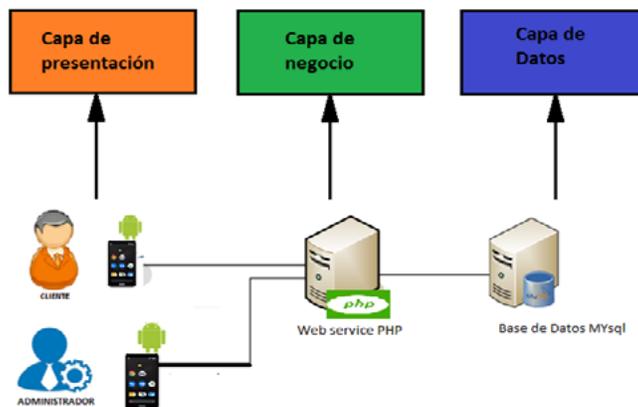
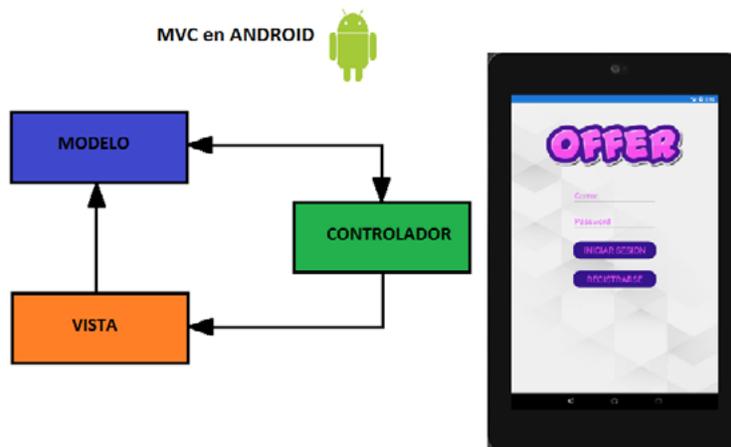


Figura 2.

Arquitectura por capas



Una vez que estén funcionando adecuadamente todos los servicios de la aplicación propuesta se tiene una arquitectura final descrita en la *Figura 3* y *Figura 4*.

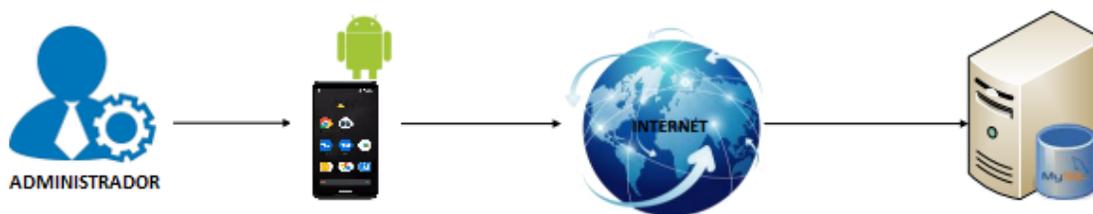
Figura 3.

Arquitectura sistema cliente



Figura 4.

Arquitectura del sistema administrativo



Fuente: Modelado Paint

2.2 Plataformas de desarrollo

Actualmente existen distintas plataformas de desarrollo móvil de tipo nativas, híbridas y multiplataforma, en este trabajo se optó por utilizar Xamarin (s.f). En la *Tabla 1* se detalla las herramientas utilizadas.

Tabla 1.

Descripción de Herramientas Tecnológicas

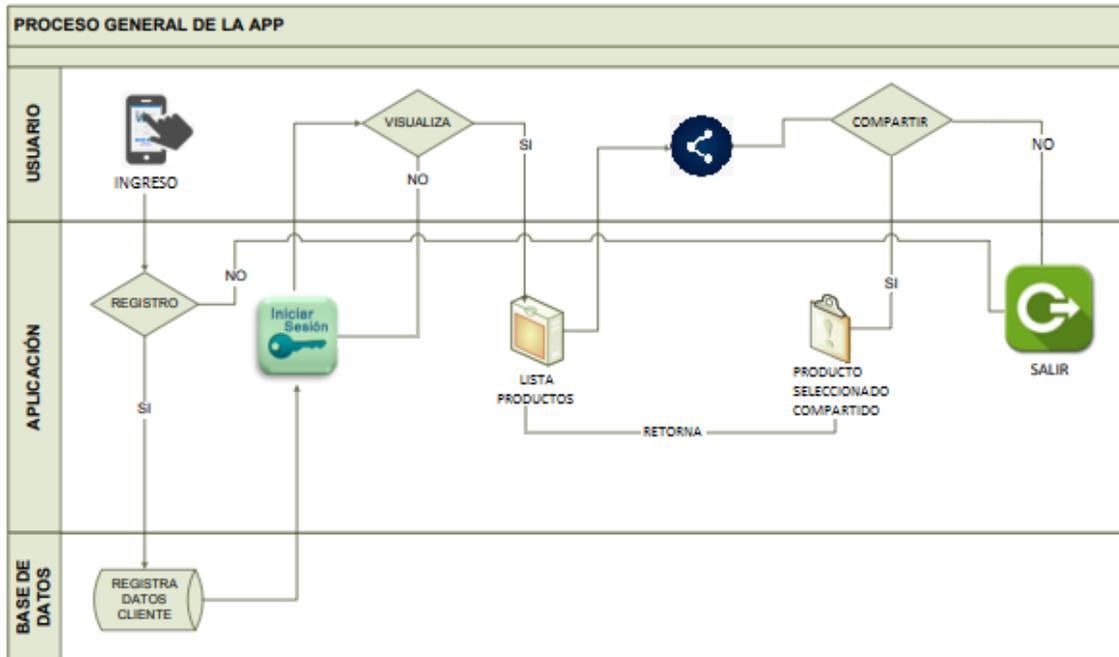
Herramienta	Descripción
XAMARIN	Framework de desarrollo móvil multiplataforma
XAMARIN FORMS	Permite que la APP, sea multiplataforma
MY SQL	Motor de Base de Datos
VISUAL STUDIO.NET	MS Visual Studio 2019. Framework 4.8
REST	Servicios Web

2.3 Flujo en el proceso de aplicación móvil

La *Figura 5* detalla el flujo de acciones que se van a desarrollar en el dispositivo móvil y la *Figura 6* sobre el proceso del administrador.

Figura 5.

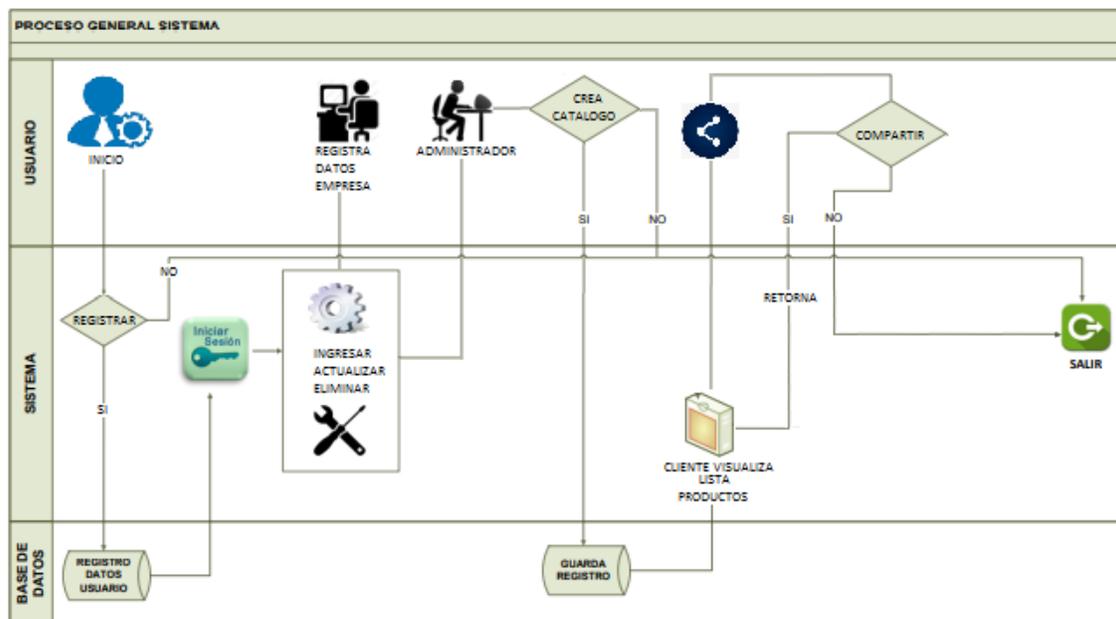
Proceso cliente



Fuente: Modelado End to End

Figura 6.

Proceso administrador



Fuente: Modelado End to End

2.4 Requerimientos

En el desarrollo de aplicaciones informáticas es necesario definir los requerimientos que se deben cumplir, en este sentido la *Tabla 2* y *Tabla 3* describen los requerimientos funcionales y no funcionales.

Tabla 2.

Requerimientos funcionales

Requerimientos Funcionales	
RF - 01	Se permitirá registrar la información de usuarios y los datos de su comercial o empresa.
RF - 02	Acceso por usuario y contraseña
RF - 03	Creación de catálogo de productos.
RF - 04	Se permitirá listar, filtrar la cartera de productos categorizados considerando.
RF - 05	El sistema permitirá insertar, editar, eliminar, exportar archivos del catálogo.
RF - 06	El sistema permitirá anexar redes sociales para compartir archivos.

Tabla 3.

Requerimientos No Funcionales

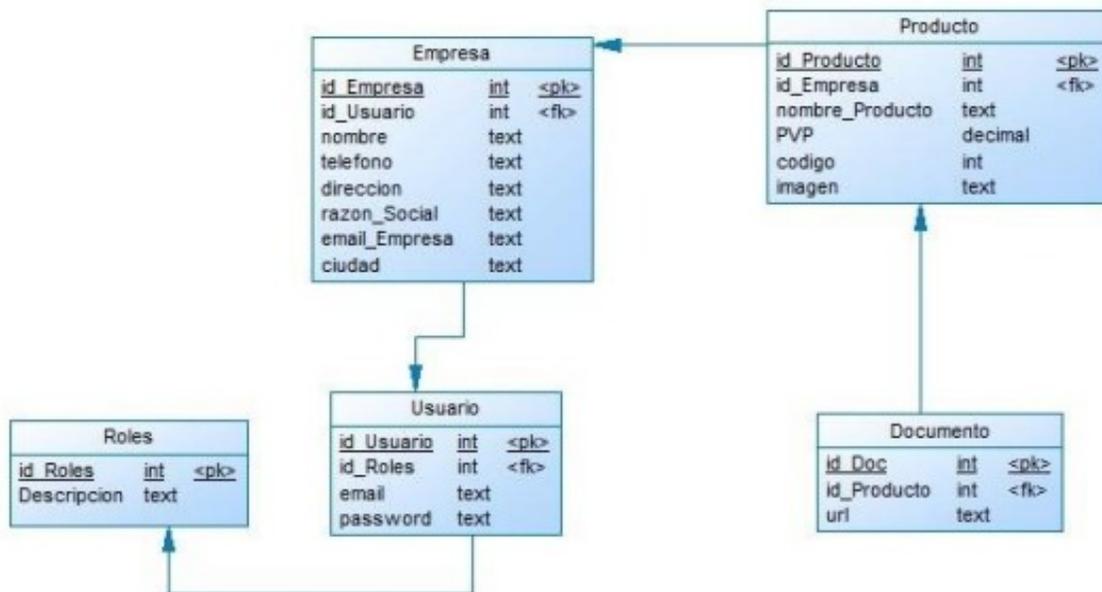
Requerimientos No Funcionales	
RNF - 01	Se configurará para que funcione correctamente en la franja más vasta viable de dispositivos Android, desde su versión más baja.
RNF – 02	Inicialmente funcionara en Android, posterior para iOS
RNF - 03	La aplicación proporcionará mensajes de ayuda, advertencias y errores.
RNF – 04	El acceso los datos por roles de usuario.
RNF – 05	El sistema tendrá robustez para garantizar que la aplicación esté siempre disponible.
RNF - 06	El sistema se concentrará en el uso intuitivo (facilidad de uso).

2.5 Diagrama UML

Para representar los esquemas del programa se utiliza herramientas de modelado UML (Ahmad et al., 2019), tal cual se obtuvo una abstracción de las propiedades primordiales de la aplicación, en la *Figura 7* se grafica.

Figura 7.

Diagrama UML



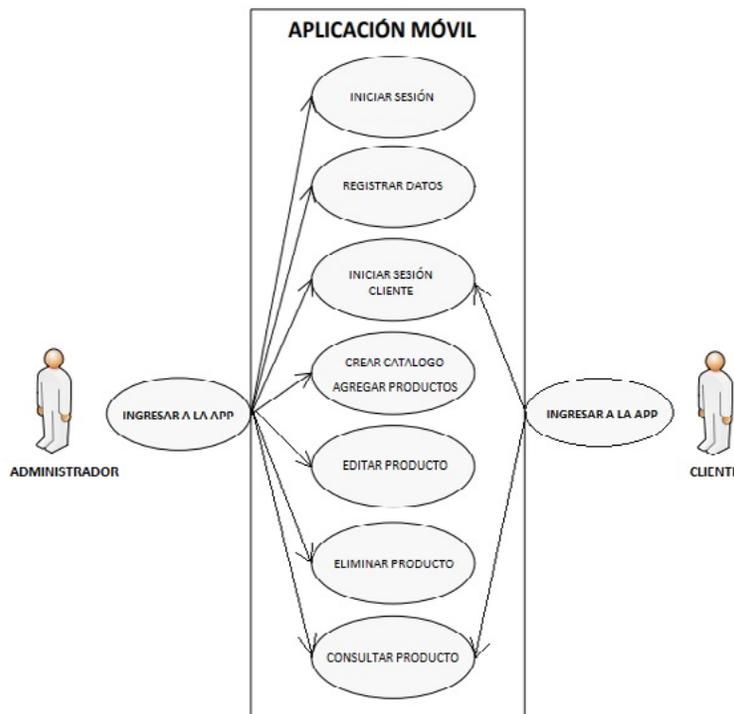
Fuente: Power Designer

2.6 Caso de uso

La *Figura 8* muestra la relación de actores con la aplicación y el sistema.

Figura 8.

Caso de uso



La *Tabla 4* y *5* detallan los casos de uso de cada actor, es decir para el administrador y los usuarios

Tabla 4.

Descripción de diagrama caso de uso administrador

Caso de uso	Diagrama de la aplicación
Actor	Vendedor
Descripción	El vendedor debe iniciar sesión para registrar los datos de su empresa y realizar acciones como ingresar un producto, modificar, eliminar, vía aplicación móvil.
Pre-condición	Acceso a internet
Actividades	Ingreso al sistema con usuario y contraseña. Presentación lista de catálogo de productos con imagen y precio. Insertar datos: cargar datos de productos. Modificar datos: editar datos de productos. Eliminar datos: eliminar datos de productos. Compartir catálogo: Compartir catálogo por medio de redes sociales.

Tabla 5.

Descripción de diagrama caso de uso cliente

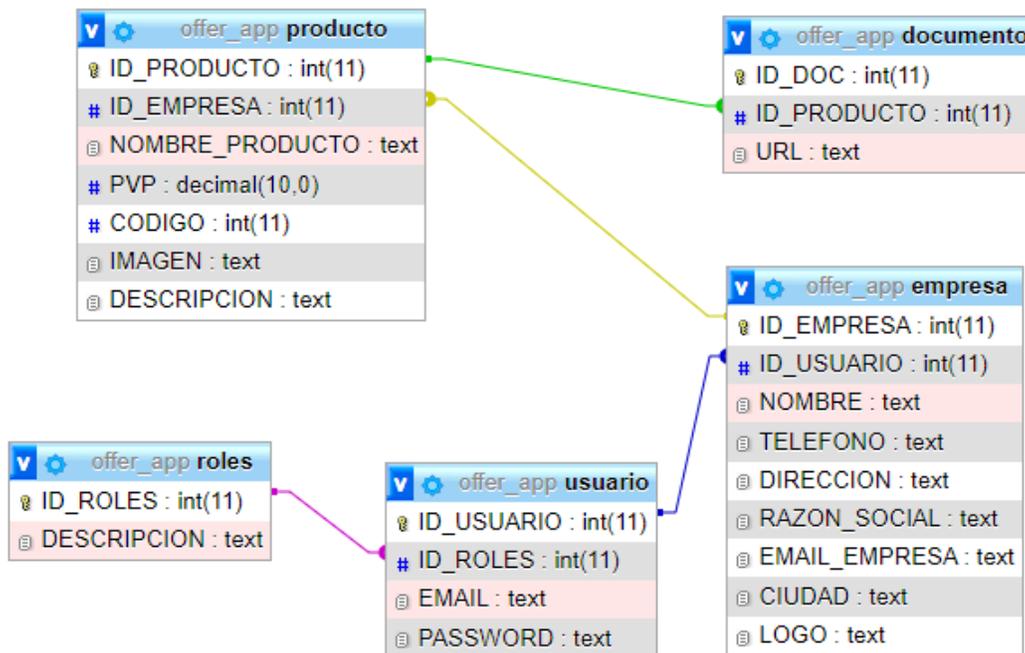
Caso de Uso	Diagrama de la aplicación
Actor	Usuario
Descripción	Debe registrar sus datos para iniciar sesión y acceder al sistema
Pre-condición	Acceso a internet
Actividades	Acceso al sistema por usuario y contraseña Presentación lista de catálogo de productos con imagen y precio. Compartir catálogo: Compartir catálogo por medio de redes sociales.
Flujo alternativo	Control de intentos de acceso al sistema.

2.7. Diagrama de base de datos

Al ser un sistema pequeño, se utilizan 5 tablas relacionadas para el flujo de los datos, la *Figura 9* muestra el modelo entidad relación de la base de datos. En este trabajo se utilizó el gestor de base de datos MYSQL (s.f).

Figura 9.

Base de datos

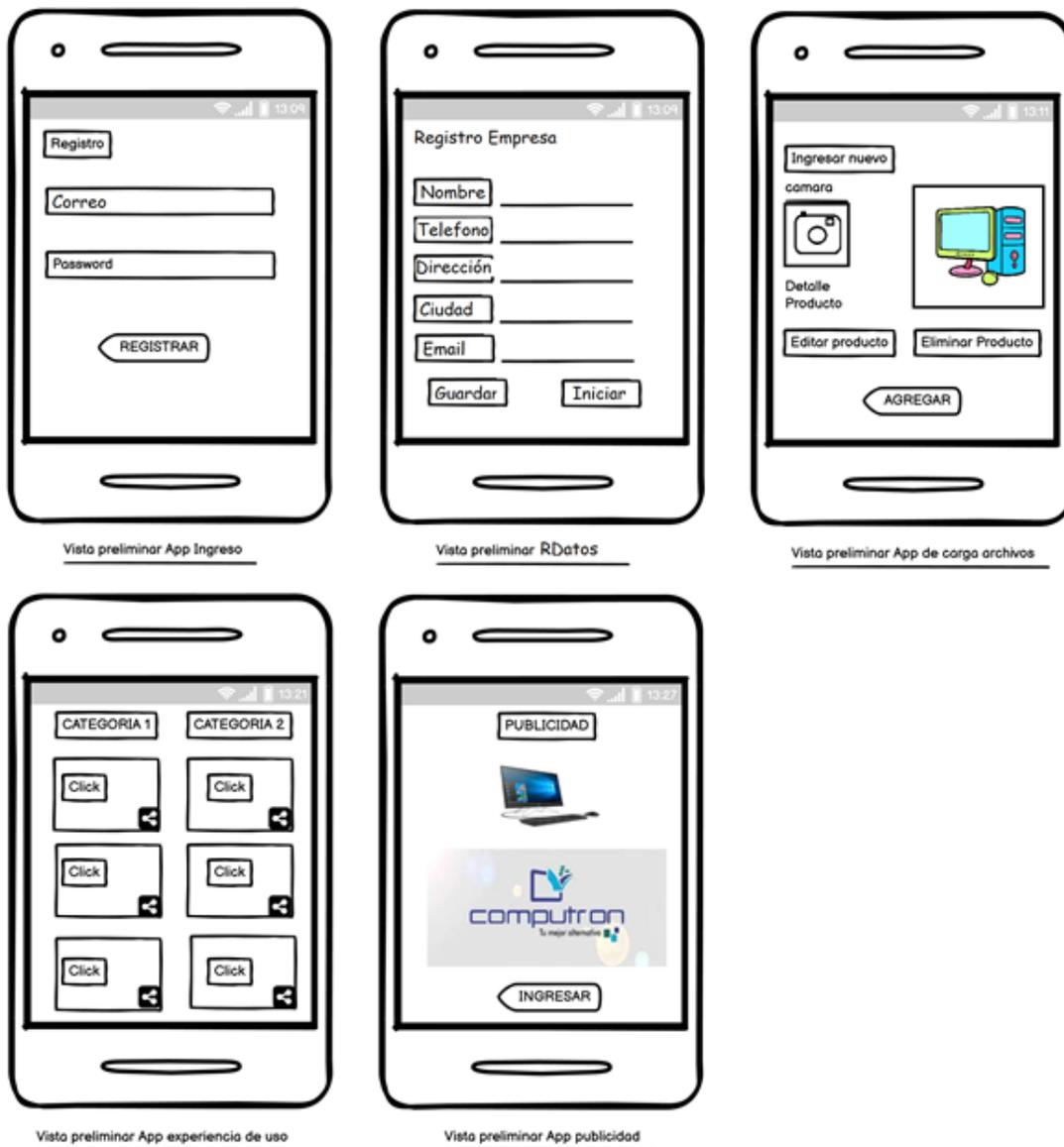


2.8 Prototipado

A continuación, se presenta el diseño preliminar de las pantallas en la herramienta Balsamiq Mockups (Balsamiq, s.f). La *Figura 10* muestra las pantallas iniciales del sistema definidas en la fase de recolección de información.

Figura 10.

Prototipado

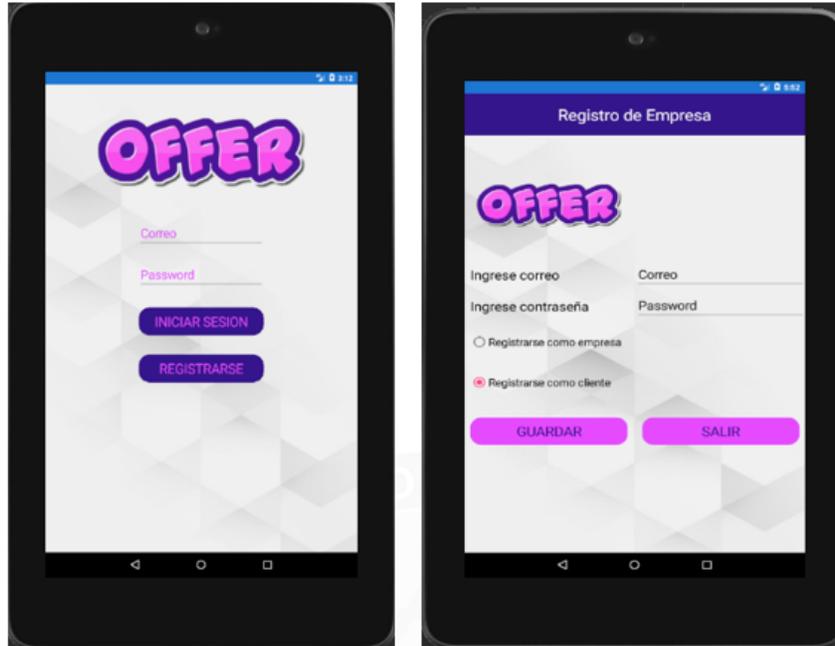


2.9 Descripción interfaz de la App

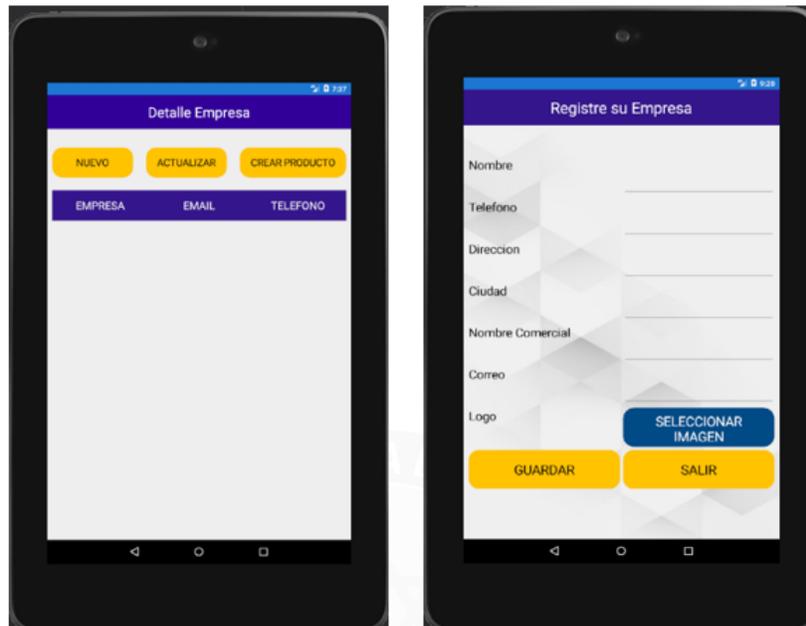
La pantalla inicial del sistema tiene 2 botones comenzar sesión, registrarse y en la parte preeminente se muestra el logo de la aplicación (Ver *Figura 11*).

Figura 11.

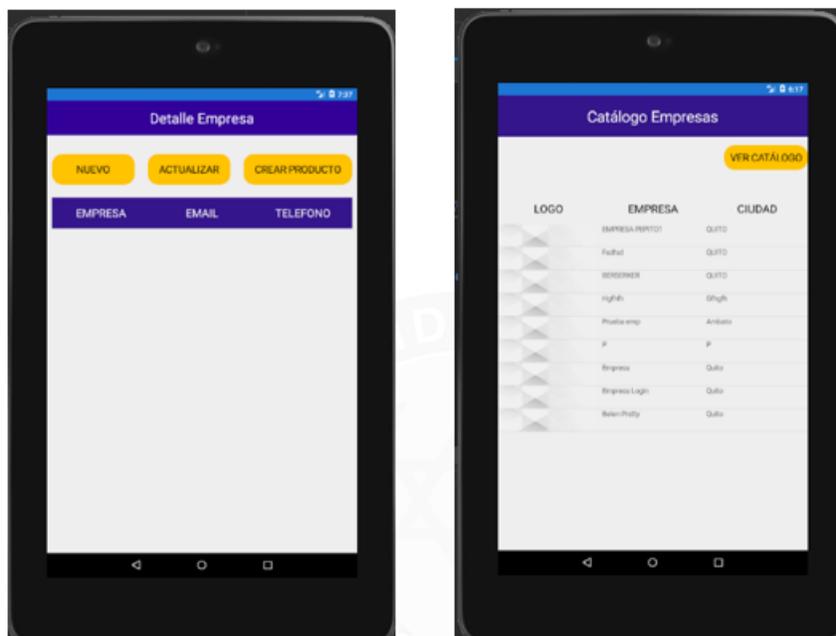
Descripción Interfaz



Una vez que el usuario accede al sistema, tiene la opción de registrar su empresa para poder promocionarla, esto se muestra en la *Figura 12*.

Figura 12.*Registrar, Empresa*

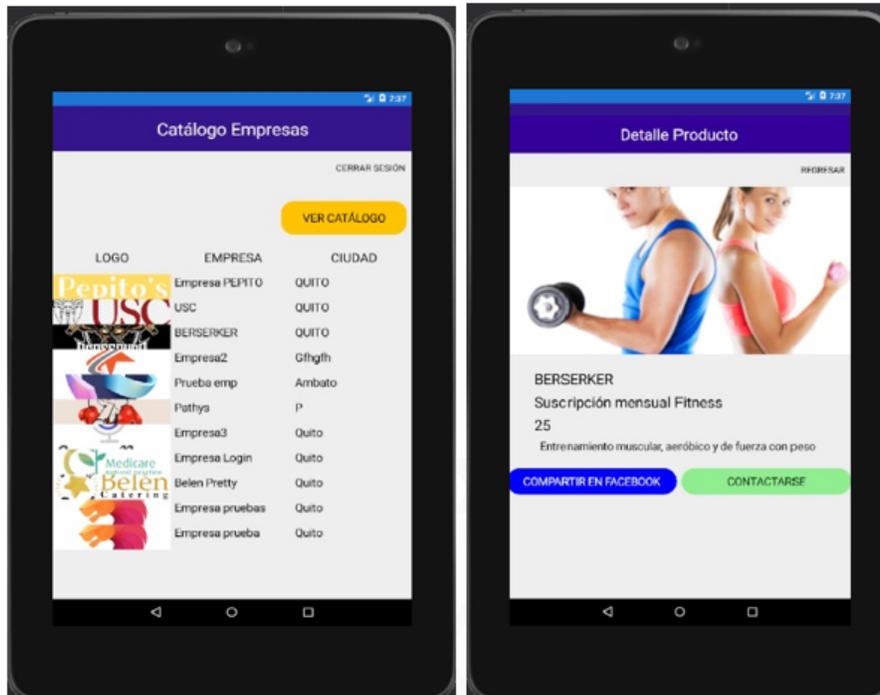
Cuando se registra la empresa, aparece en el listado de empresas que están promocionándose, ver la *Figura 13*.

Figura 13.*Crear, Actualizar, Eliminar Catalogo*

Finalmente, en la *Figura 14* se muestran los detalles de la empresa y el producto que promocionan.

Figura 14.

Ver Catalogo, Seleccionar un Producto



Resultados

Una vez finalizada la fase de codificación, se realizaron las pruebas respectivas, tomando en cuenta los aspectos siguientes:

- Síntesis sobre los comentarios de usuarios al momento de utilizar la aplicación, tomando en cuenta una autenticación correcta y una autenticación incorrecta.
- Comprobación funcional en diferentes teléfonos con requisitos mínimos para la instalación de la aplicación.

A continuación, se muestra un test que tiene como objetivo evaluar la percepción del usuario al usar la aplicación Offer App y analizar 5 puntos críticos que determinarán:

Tabla 6.

Criterios a evaluar

Criterios de evaluación	
Aspecto	Descripción

Contenido	De acuerdo a la normativa definida por la empresa
Facilidad de Aprendizaje	Resultados al usar la aplicación por primera vez.
Accesibilidad	Consideraciones Visuales
Satisfacción	Grado de Satisfacción del usuario
Eficiencia	Adaptación de usuarios frente al uso de la aplicación .

Tabla 7.*Prueba autenticación*

Pruebas de Usabilidad	
Descripción	Una vez instalada la aplicación aparece una ventana de “Inicio de Sesión”, en el que ingresa su correo y contraseña internamente se validan estos datos.
Condiciones de ejecución	El usuario debe estar registrado en el sistema
Entrada	El usuario cliente ingresa su correo, contraseña y presiona “Iniciar Sesión” ingresa a ventana, ver catálogos. El usuario administrador presiona “Registrarse” ingresa a ventana de Registro Administrador e ingresa su correo, contraseña para registro.
Resultado Esperado	Tras el ingreso de correo y contraseña, si el proceso ha sido correcto en la base de datos se registra y el usuario puede utilizar la aplicación.
Evaluación de la prueba	Prueba satisfactoria.

- **Participantes:** se tomará una muestra de 20 personas negociantes. La prueba de usabilidad se llevará a cabo bajo la supervisión de los desarrolladores.
- Criterio de aceptación o rechazo:
 - **Errores leves:** confusión del usuario con algún botón
 - **Errores medios:** falta de personalización para presentar datos.
 - **Errores críticos:** la sincronización no garantiza que la información de la base de datos sea igual a la del dispositivo móvil, se almacena información errónea con inconsistencia de datos, el proceso de actualización no es satisfactorio.

Conclusiones

Luego de hacer toda la investigación pertinente de la App, se puede deducir que los individuos permanecen abiertas al uso del Internet en dispositivos móviles y las novedosas tecnologías, así sea por gusto o por necesidad, por ende, el brindar una aplicación móvil, con propiedades diferenciadoras como tiempos y movimientos, que son recursos principales en la actualidad, dichos realizan de Offer App una aplicación distinta en un sector de veloz incremento y con alta competitividad.

Offer App se convierte en un plan innovador de apoyo publicitario, al comerciante, al empresario y al consumidor, debido a que es algo con lo cual el desarrollo del mercado del Ecuador necesita para asegurar su aumento.

Referente a la App tiene 2 instancias relevantes, la primera es que al inicio va a ser gratuita y la segunda su diseño e implementación va a ser agradable, simple de usar debido a las tácticas de la arquitectura planteada.

Uno de los enfoques con más auge en los últimos años es el marketing digital, el cual permitió que las empresas potencien su marca usando canales de repartición como son las redes sociales, por ende, Offer App va conforme a estas novedosas tendencias en el desarrollo de herramientas tecnológicas.

Referencias

- Ahmad, T., Iqbal, J., Ashraf, A., Truscan, D., & Porres, I. (2019). Model-based testing using UML activity diagrams: A systematic mapping study. *Computer Science Review*, 33, 98–112. <https://doi.org/10.1016/j.cosrev.2019.07.001>
- Artieda, C., Reza, J., & Rojas, H. (2021). Sistema móvil para la oferta de recorridos con rutas fijas. *REVISTA ODIGOS*, 2(1), 21–40. <https://doi.org/10.35290/ro.v2n1.2021.381>
- Balsamiq. (s.f). Rapid, Effective and Fun Wireframing Software. <https://balsamiq.com/>
- Burbano, E., Constante, M., Hidalgo, L., & Moya, F. (2020). Prototipo móvil para la geolocalización de mascotas callejeras. *REVISTA ODIGOS*, 1(3), 77–96. <https://doi.org/10.35290/ro.v1n3.2020.372>
- Díaz, Y., & Fernández, Y. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, 11(2). <https://www.revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>
- MySQL. (s.f). MySQL 5.7 Reference Manual : 1.3.1 What is MySQL? <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- Piernas, R. (24 de abril de 2014). ¿Qué es el Engagement en las Redes Sociales? Bilnea. <https://bilnea.com/que-es-engagement-en-redes-sociales/>
- Prieto, B. (2017). El uso de los métodos deductivo e inductivo para aumentar la eficiencia del procesamiento de adquisición de evidencias digitales. *Cuadernos de contabilidad*, 18(46), 56-82. <https://doi.org/10.11144/javeriana.cc18-46.umdi>
- Reyes, A., Vallejos, M. de L., & Quintana, D. (2020). Nueva realidad: compras en línea y a domicilio. *REVISTA ODIGOS*, 1(3), 47–60. <https://doi.org/10.35290/ro.v1n3.2020.370>
- Silva, C. A., Toasa, R., Guevara, J., Martinez, H. D., & Vargas, J. (2018). Mobile Application to Encourage Local Tourism with Context-Aware Computing. In *2018 International Conference on Information Technology & Systems, ICITS'18*, 796–803. https://doi.org/10.1007/978-3-319-73450-7_75
- Vásquez, F. A. (2010). Programación por capas. *Paradigmas*, 2(1), 13–29.
- Verborgh, R., Harth, A., Maleshkova, M., Stadtmüller, S., Steiner, T., Taheriyani, M., & Van de Walle, R. (2014). Survey of semantic description of REST APIs. In *REST: Advanced Research Topics and Practical Applications* (pp. 69-89). Springer. https://doi.org/10.1007/978-1-4614-9299-3_5

Villacís, B., Carrillo, D., & Martínez, A. (2010). Estadística demográfica en el Ecuador: diagnóstico y propuesta. *Instituto Nacional de Estadística y Censos*, 1-74. <https://www.ecuadorencifras.gob.ec/wp-content/descargas/Libros/Demografia/documentofinal1.pdf>

Xamarin. (s.f). What is Xamarin? - I Microsoft Docs.

Copyright (c) 2022 Andrea Guissela Pérez Lozada, Edison Moreno Jurado y Luis Alberto Pérez Viana



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)



“Second Mind” una aplicación móvil para el almacenamiento de memorias del usuario

Fecha de recepción: 2021-10-28 • Fecha de aceptación: 2021-12-14 • Fecha de publicación: 2022-02-10

Roberth Alexander Almache Vallejos¹

Comando Conjunto de las Fuerzas Armadas del Ecuador

roberth9501@hotmail.com

<https://orcid.org/0000-0002-5947-9576>

Jorge Ramiro Castro Chavez²

Int Food Services Corp, Ecuador

jorgesuzuky666@gmail.com

<https://orcid.org/0000-0003-4500-5940>

Diego Alejandro González Calderon³

Rankmi, Chile

dgonzalez.jim@gmail.com

<https://orcid.org/0000-0003-0121-867X>

Johanna Daniela Ruano Cardenas⁴

Maint.SA, Ecuador

danyruanoc13@gmail.com

<https://orcid.org/0000-0002-7766-2809>

RESUMEN

Actualmente, a nivel profesional y personal se maneja una gran cantidad de información de cualquier ámbito, sea laboral, escolar, familiar, salud, etc., y por el avance tecnológico ya no se llevan registros manuales, sino un almacenamiento digital, ubicándolo en varios lugares, tales como correos, notas rápidas o archivos de texto, que limitan la obtención inmediata de información a recordar. En este sentido, la presente investigación tiene como objetivo diseñar un prototipo móvil

llamado “Second Mind”, que permite al usuario almacenar información (notas, tips, conceptos, rutinas, ayudas técnicas, etc) en el dispositivo móvil, donde se puede buscar y tener accesibilidad a datos importantes o relevantes, permitiendo manejar su información de forma efectiva, inmediata y organizarla en base a su necesidad, con esto se le permitirá accionar sin dificultad en cualquier lugar que se desempeñe con sus registros personales.

PALABRAS CLAVE: digital, app, almacenamiento, tecnológico, dispositivo

ABSTRACT

Currently, at a professional and personal level, a large amount of information is handled in any field, be it work, school, family, health, etc., and due to technological advances, manual records are no longer kept, but digital storage, locating it in various places, such as emails, quick notes or text files, which limit the immediate collection of information to be remembered. In this sense, the present research aims to design a mobile prototype called “Second Mind”, which allows the user to store information (notes, tips, concepts, routines, technical aids, etc) in the mobile device, where you can search and have accessibility to important or relevant data, allowing you to manage your information effectively, immediately and organize it based on your need, with this will allow you to act without difficulty anywhere you perform with your personal records.

KEYWORDS: digital, app, storage, technological, device

Introducción

En el día a día, para poder comunicarnos y realizar incluso distintas acciones, se necesita de información o datos que faciliten las actividades diarias. Se ha visto que en generaciones anteriores la tecnología no era muy accesible y manejaban su información en libros, libretas y registros manuales, poco a poco la tendencia ha ido cambiando y se ha permitido digitalizar la información, facilitando incluso hasta la escritura de la misma. Se observa actualmente que incluso con este avance, y con varios aplicativos que se tiene, como por ejemplo en Google el Dropbox, MEGA, MediaFire, Google Drive, que se lanzó en el año 2012, así como también el mismo bloc de notas, o el correo electrónico (Martín-Barbero, 2015), existen limitantes al obtener los datos deseados en el momento justo que se requiere.

La información muchas veces se almacena en varios lugares y registros, de hecho, sea por capacidad, disponibilidad, o netamente por falta de una aplicación como repositorio que ayude a almacenarla, ha dificultado la disponibilidad, ya que con frecuencia se necesita traer un dato importante a la mente, provocando retrasar o posponer ciertas actividades, sea a nivel personal, comercial, de ingeniería, e incluso a nivel escolar, se suelen suspender actividades académicas por no disponer de información inmediata que permita organizar al usuario y trabajar de forma productiva.

En base a dicha necesidad, se ha visto como mejoramiento la creación de un prototipo de aplicación móvil en Xamarin forms, manejando una base de datos SQLite y con el IDE de desarrollo Visual Studio .Net 2019, desarrollada por versiones en GitHub (B, 2021), la misma que permitirá al usuario almacenar información, sean textos, imágenes, links en su dispositivo con un buscador que con solo ingresar una palabra clave se obtendrá la información, manteniendo disponibilidad continua e inmediata.

Con lo antes expuesto, este desarrollo está dirigido a cualquier persona que maneje y disponga de un dispositivo móvil Android, y tenga la necesidad de registrar datos y tener memorias almacenadas para su desempeño, siendo un aplicativo muy útil y novedoso donde puede llegar a muchos usuarios, con tan solo aprovechar la facilidad móvil (Dominguez, 2018). Este desarrollo es un aporte a la sociedad que puede llegar a cualquier parte y sin distinción, a tal punto de que a un niño se le facilite almacenar información escolar y a un adulto memorias difíciles de recordar y comprender con tan solo una búsqueda en su dispositivo.

Metodología

La implementación de una metodología en un desarrollo de software proporciona grandes beneficios al proyecto, mejorando la calidad y productividad dentro del entorno de trabajo, para este caso en específico se utilizó la observación directa y la metodología Kanban.

De acuerdo con Marshall y Rossman (1989), citados en Kawulich (2005), la observación es "la descripción sistemática de eventos, comportamientos y artefactos en el escenario social elegido

para ser estudiado” (p.79), en este sentido, el uso de esta técnica permitió tener un panorama más amplio en cuanto a la descripción de las situaciones que se presentaron.

A su vez, es importante me mencionar la definición de la metodología aplicada. “La palabra Kanban viene del japonés y traducida literalmente quiere decir tarjeta con signos o señal visual. El tablero más básico de Kanban está compuesto por tres columnas: “Por hacer”, “En proceso” y “Hecho”. Si se aplica bien, y funciona correctamente, sirve como una fuente de información, ya que demuestra dónde están los cuellos de botella en el proceso y qué es lo que impide que el flujo de trabajo sea continuo e ininterrumpido”, (Kanbanize, s.f).

2.1 Fases de ciclo de producción

Kanban está basado en el desarrollo incremental, esto quiere decir que el trabajo a desarrollar se divide en diversas fases, con esto lo que se busca es acelerar el proceso del desarrollo del software; estas fases son puestas en las columnas que correspondan. En estas tarjetas no se describe la totalidad de la tarea a realizar solamente, se da una descripción general del entorno de lo que se quiere, para que el equipo de trabajo conozca la carga que posee; en estas tarjetas se describen si esta tarea depende de otras (Gilibets, 2020).

En la siguiente *Tabla 1* se visualizará la lista de requerimientos presentados para este proyecto.

Tabla 1.

Requerimientos del prototipo

Requerimientos funcionales		
Identificación	Nombre del requerimiento	Descripción del requerimiento
RF01	Recursos de prototipo móvil "Second Mind"	Para el prototipo se debe emular desde un celular. El acceso al aplicativo mantendrá un login.
RF02	Menú de opción de prototipo móvil	La aplicación tendrá dos botones, uno de registro para ingreso de datos y otro para consulta de la información.
RF03	Registro de información de prototipo móvil	El usuario al seleccionar Registro podrá ingresar y grabar los datos como: textos, imágenes, links en la prototipo y una palabra clave para recordar.
RF04	Consulta de información en prototipo "Second Mind"	El usuario al seleccionar en pantalla "Recordar" se habilitará un buscador, al tipear una palabra clave podrá mostrar su información almacenada en "Second Mind".
RF05	Eliminar Información en prototipo móvil	El usuario podrá eliminar información registrada que ya no desee mantener almacenada.

Requerimientos no funcionales

Identificación	Nombre del requerimiento	Descripción del requerimiento
RNF01	Interfaz del sistema.	El prototipo presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios.
RNF02	Cámara para prototipo móvil	Se requiere uso de cámara propia del dispositivo móvil.
Identificación	Nombre del requerimiento	Descripción del requerimiento
RNF03	Almacenamiento en prototipo móvil	La memoria permitirá que la búsqueda sea altamente inmediata, los datos se almacenan en SQLITE y las imágenes se almacenarán en Amazon para AW S3 "repoawsxamarin".
RNF04	Desempeño de prototipo móvil	Se garantizará a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a la misma.

2.2 Requerimientos de software

- GitHub. - Es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso (González, 2021).

- Visual Studio .Net 2019.- Lenguaje de programación diseñado para crear una gran variedad de aplicaciones que se ejecutan en .NET. Este lenguaje con seguridad de tipo es eficaz y están orientado a objetos. Se basan en .NET Compiler Platform “Roslyn”, que proporciona API de análisis de código sofisticadas y es de código abierto en GitHub (Microsoft, s.f).
- Xamarin Forms. - Es un marco de interfaz de usuario de código abierto que permite a los desarrolladores crear aplicaciones de Xamarin.Android, Xamarin.iOS y Windows desde una única base de código compartida.
- BD SQLite. - Es un ligero motor de bases de datos de código abierto, que se caracteriza por mantener el almacenamiento de información persistente de forma sencilla (SQLite,2018). A diferencia de otros sistemas gestores de bases de datos como MySQL, SQL Server y Oracle DB, SQLite tiene las siguientes ventajas:
 - No requiere el soporte de un servidor, SQLite no ejecuta un proceso para administrar la información, si no que implementa un conjunto de librerías encargadas de la gestión.
 - No necesita configuración, libera al programador de todo tipo de configuraciones de puertos, tamaños, ubicaciones, etc.
 - Usa un archivo para el esquema, crea un archivo para el esquema completo de una base de datos, lo que permite ahorrarse preocupaciones de seguridad.
 - Es de código abierto: está disponible al dominio público de los desarrolladores, al igual que sus archivos de compilación e instrucciones de escalabilidad (Develou, s.f).

Es por eso que SQLite es una tecnología cómoda para los dispositivos móviles. Su simplicidad, rapidez y usabilidad permiten un desarrollo muy amigable.

- Xamarin Essentials. - Es un paquete compatible con Microsoft que se conecta a varias características nativas para el usuario y mejora con cada versión, en este caso se usó para las imágenes y links (Ruiz, 2017).
- AWS SDK. - Para .NET simplifica el uso de los servicios de AWS porque ofrece un conjunto de bibliotecas que a los desarrolladores que usan .NET les resultan lógicas y familiares (Madhuri & Madhuri, 2016). Todos los AWS SDK ofrecen compatibilidad con el ciclo de vida de las API, como administración de credenciales, reintentos, cálculo de referencias de datos y serialización. AWS SDK para .NET también admite abstracciones de nivel superior, como S3 Transfer Utility, Cognito Identity Provider y AWS DyanamoDB Session State Provider. Visite el GitHub de aws/dotnet y encuentre bibliotecas y herramientas para .NET en AWS (Amazon, s.f).

2.3 Restricciones

Tomando en cuenta que el proyecto tiene como propósito un prototipo funcional, se tiene en cuenta las siguientes restricciones:

- Para el prototipo no se podrá usar dispositivos móviles que tengan una versión inferior de API 22.
- El prototipo será compatible desde la versión 5.1 o posterior.
- El dispositivo que utilice el prototipo debe tener conexión a Internet para almacenar las imágenes en el repositorio de Amazon AWS.
- En el prototipo no se podrán almacenar videos.

A continuación, en la *Tabla 2* se muestran las personas involucradas en el desarrollo del proyecto con sus role asignados:

Tabla 2.

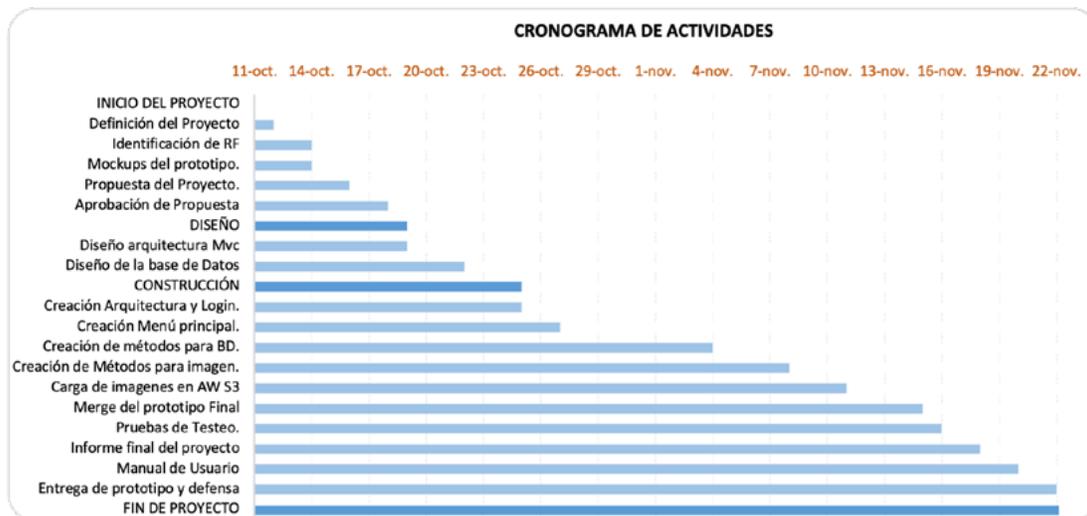
Responsables y roles en la metodología Kaban

Personas/Instituciones	Roles
Mg. Renato Toasa/Universidad Israel	Stakeholder
Diego González	Service Request Manager.
Jorge Castro/Robert Almache/Johanna Ruano	Service Delivery Manager.

A su vez, se estructura un cronograma de actividades como base, para el desarrollo del proyecto que se muestra a continuación en la *Figura 1*.

Figura 1.

Cronograma de actividades



2.4 Fases del proyecto

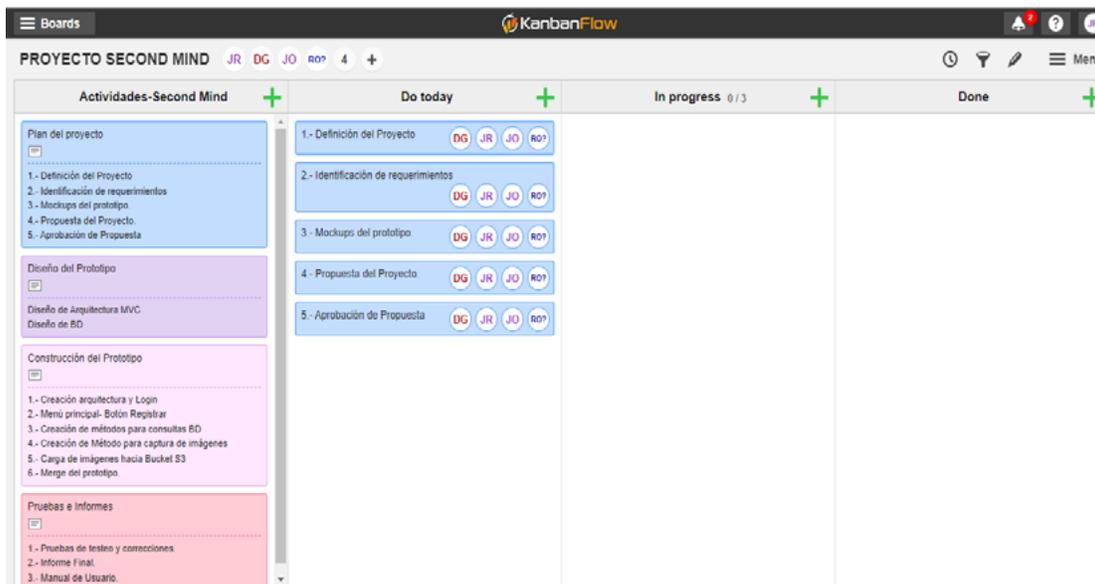
- **Fase 1.- Inicio**

En esta fase desarrollada del 11 al 19 de octubre se definieron los lineamientos del proyecto, el tema, la identificación de requerimientos, *mockups* del prototipo y gestión de actividades para el desarrollo.

En la siguiente *Figura 2* se muestra el tablero Kanban, definido por actividades del proyecto, tareas por hacer, tareas en progreso y tareas finalizadas.

Figura 2.

Planificación e inicio actividades Kanban

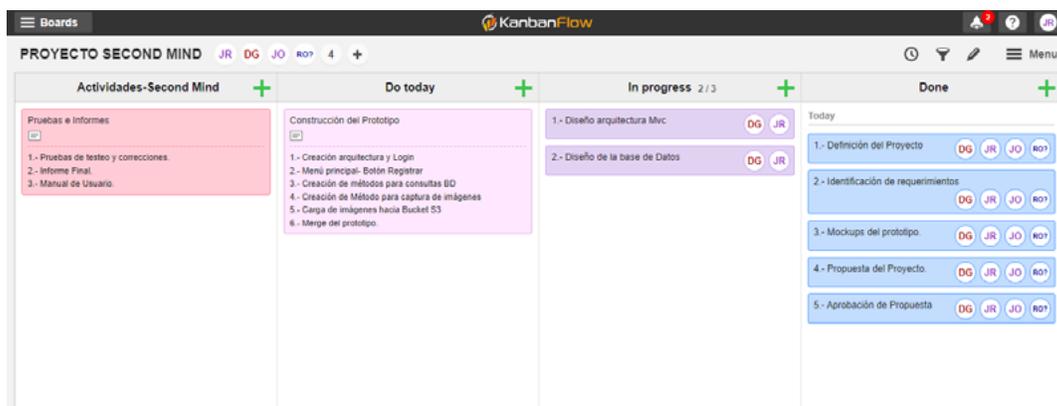


- **Fase 2.- Diseño del prototipo**

En esta fase, distribuida del 19 al 23 de octubre, se procedió con el diseño y desarrollo del prototipo, en base a los requerimientos funcionales y no funcionales identificados.

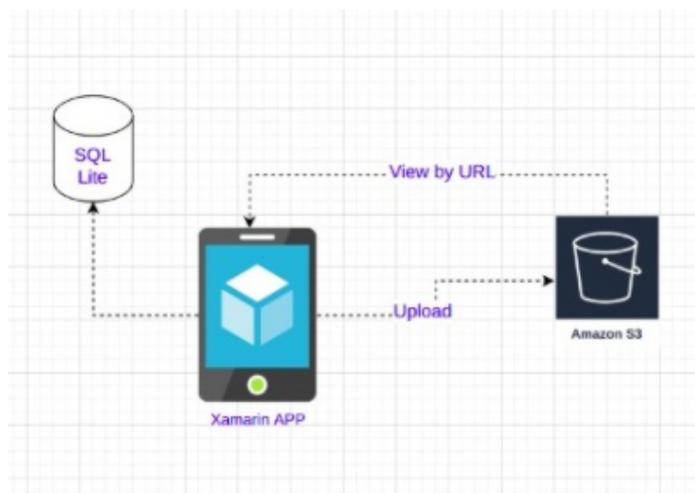
En ella se registra y se ejecutan las siguientes actividades que se visualizan en la *Figura 3*.

Figura 3.
 Tablero fase 2



En la siguiente *Figura 4* se muestra el diseño de la arquitectura MVC

Figura 4.
 Arquitectura Mvc "Second Mind"

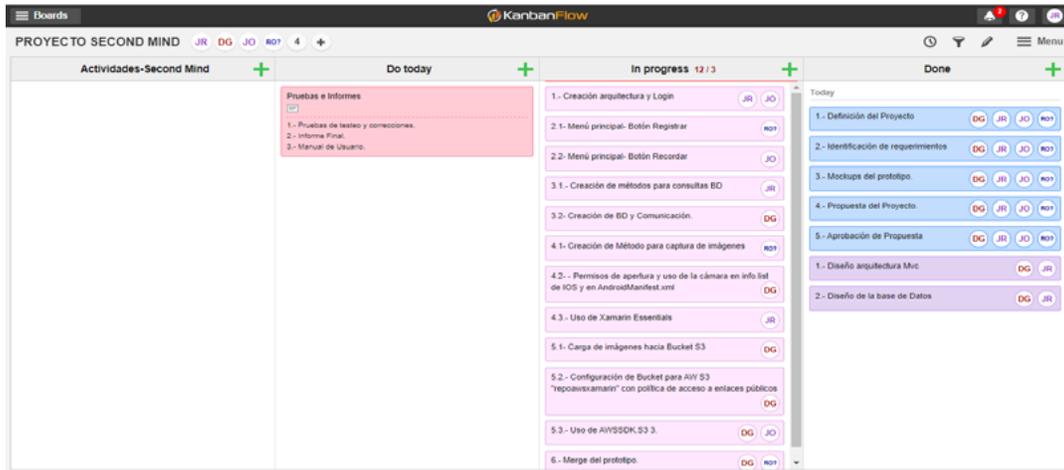


• **Fase 3.- Construcción del prototipo**

En esta fase, distribuida del 25 de octubre al 15 de noviembre, se procedió con la creación de Login, menú principal donde consta el botón Registrar/Consultar, métodos para BD e imagen, cargas de imágenes y Merge. En la *Figura 5* se observa el tablero correspondiente.

Figura 5.

Tablero fase 3

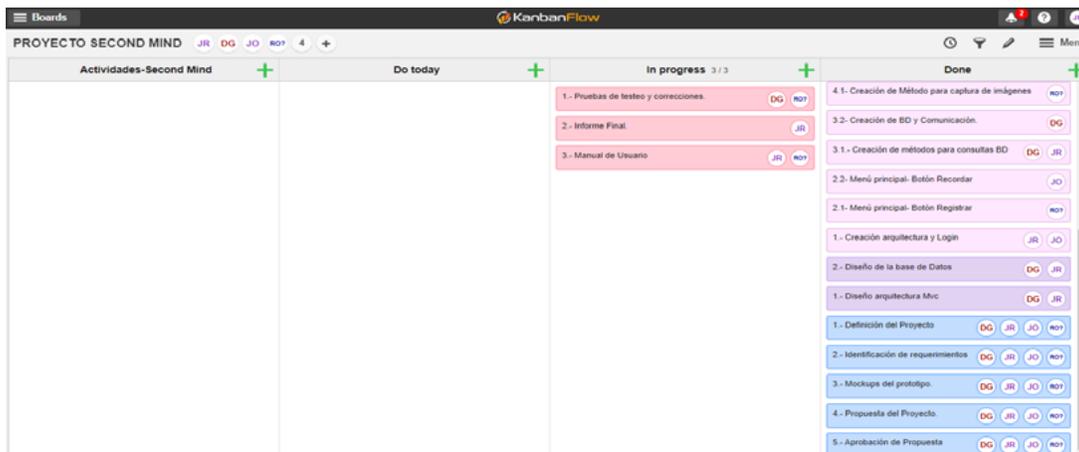


- **Fase 4.- Pruebas e informes**

En esta fase, distribuida del 15 de noviembre al 21 de noviembre, se realizaron las pruebas de testeo y correcciones del prototipo, el informe final y el manual de usuario (*Figura 6*).

Figura 6.

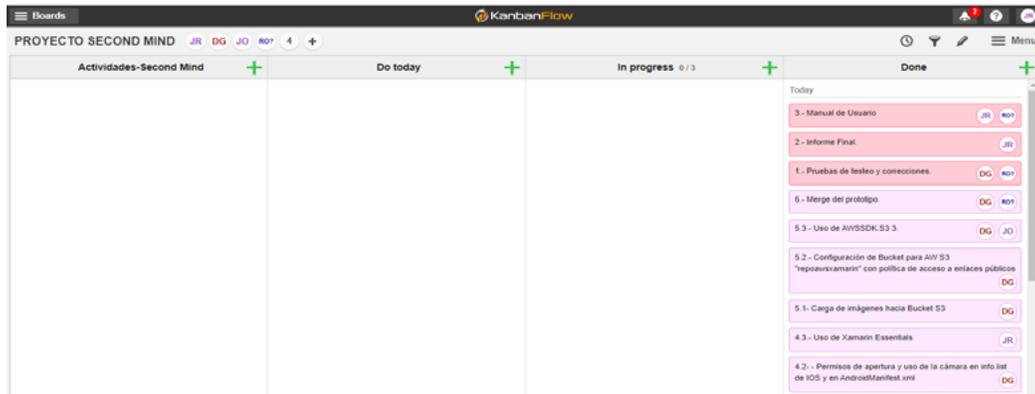
Tablero fase 4



De esta forma, en las fechas estipuladas en el cronograma se da por concluidas las actividades (*Figura 7*), usando la metodología desarrollada por intervalos y con responsables por cada actividad o tarea.

Figura 7.

Fin actividades en Tablero Kanban



Resultados

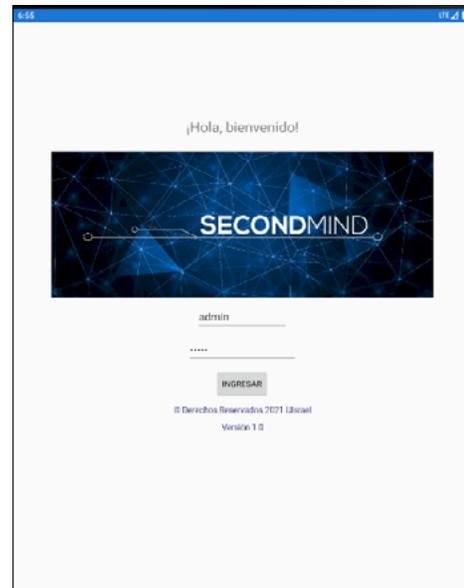
Para el estudio se tiene como parte de los resultados la funcionalidad del prototipo móvil que cumple con las especificaciones planteadas y aprobadas, el mismo se ve reflejado y demostrado con las pruebas funcionales, que a continuación se detallan:

En las *Figura 8* y *Figura 9* se observa la pantalla principal y *login* único (administrador).

Figura 8.
Login de User



Figura 9.
Pantalla Inicial



En la *Figura 10* se muestra el menú con los dos botones (Registra/Recuerda) y en la *Figura 11* se visualiza la pantalla para ingreso de información.

Figura 10.
Menú Principal

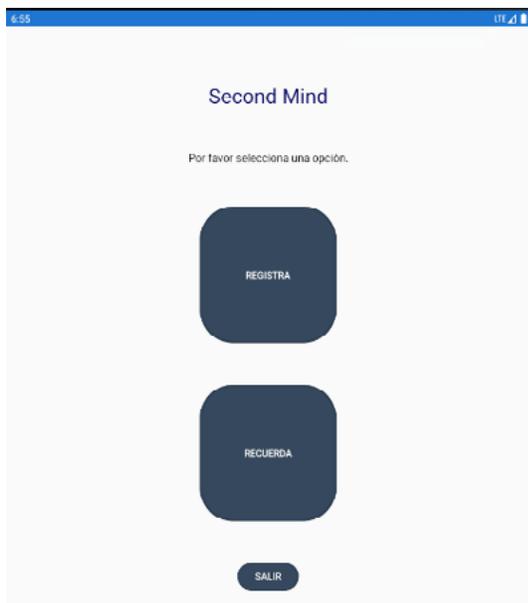
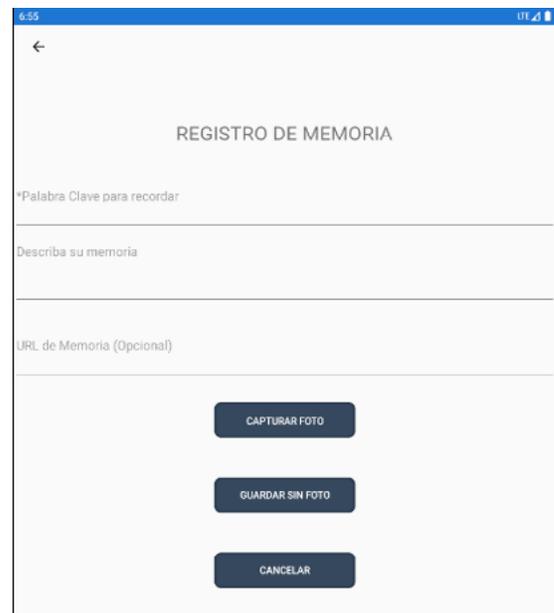


Figura 11.
Pantalla registro de memoria



En la *Figura 12* se ingresa los datos y en la *Figura 13* se muestra la pantalla al seleccionar captura de imagen a ser registrada con los respectivos botones de control.

Figura 12.

Ingreso de información



Figura 13.

Pantalla para capturar imagen



En la *Figura 14* se muestra en pantalla la imagen capturada con cámara de teléfono y en la *Figura 15* se presenta el aviso para botón subir y guardar.

Figura 14.

Pantalla de Imagen Capturada

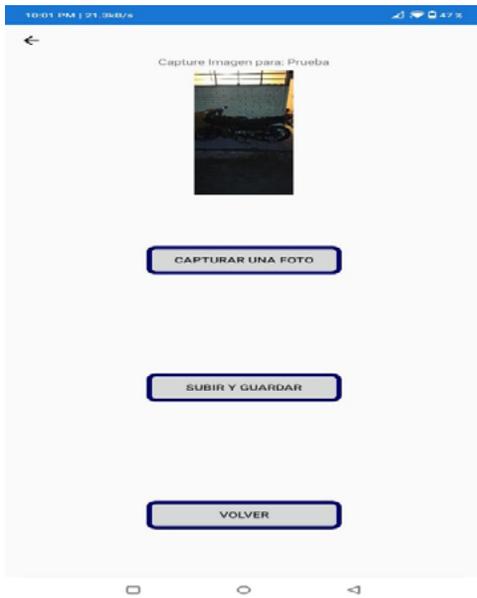
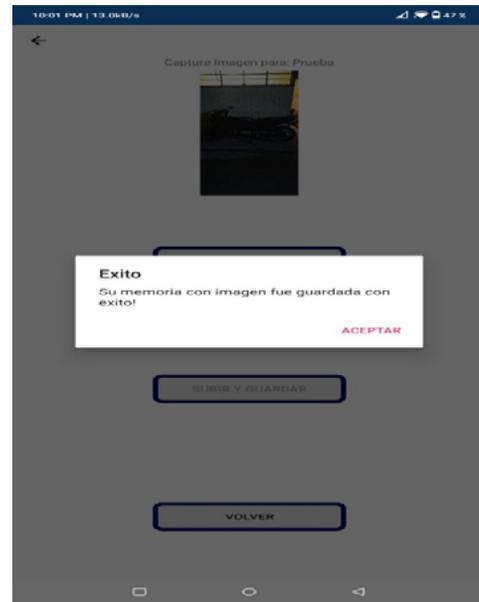


Figura 15.

Aviso por registro de imagen



En la *Figura 16* y *Figura 17* se muestra la pantalla al seleccionar Recordar, donde se visualizan las memorias almacenadas y se consulta con la palabra clave.

Figura 16.

Acceso de botón recordar



Figura 17.

Pantalla consulta con palabra clave



En la *Figura 18* y *Figura 19* se muestra la consulta hecha por el usuario, donde se visualiza la descripción, el texto y el enlace con el linkeo correspondiente.

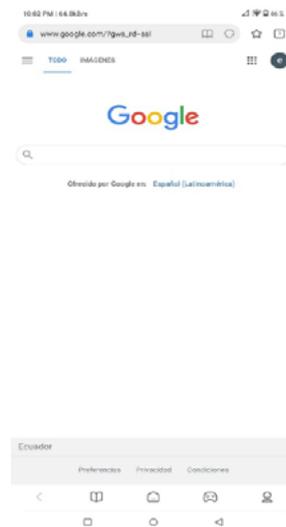
Figura 18.

Resultado de consulta por descripción



Figura 19.

Funcionalidad de acceso a url



En la *Figura 20* y *Figura 21* se muestra la consulta hecha por el usuario, donde se visualiza la descripción, el texto y la imagen almacenada en repositorio, correspondiente a ese registro.

Figura 20.

Resultado de consulta por descripción

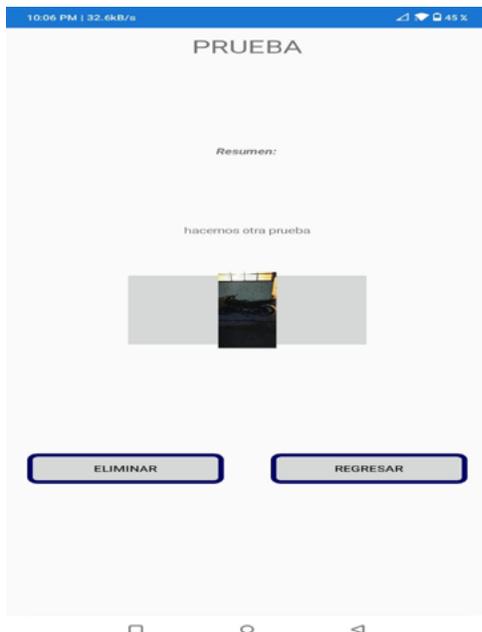


Figura 21.

Visualización de imagen en repositorio



En la *Figura 22* se puede evidenciar la funcionalidad de eliminación de un registro en aplicativo.

Figura 22.

Botón eliminar y mensaje de advertencia



Finalmente, en las siguientes *Figura 23*, *Figura 24* y *Figura 25* se muestran las advertencias en caso de errores de ingreso, falta de información para registro y errores en ingreso de la URL.

Figura 23.

Error al ingresar

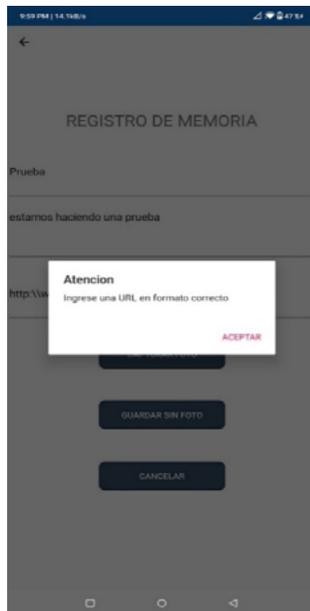


Figura 24.

Error al guardar sin información

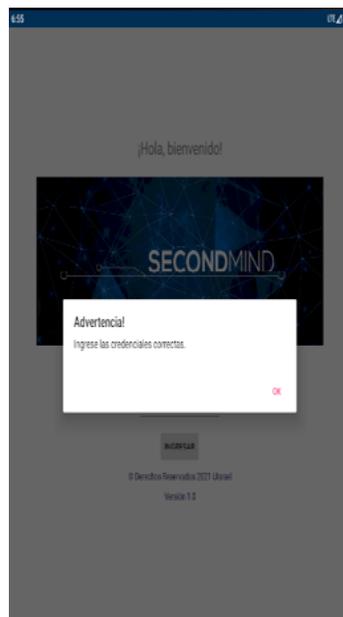
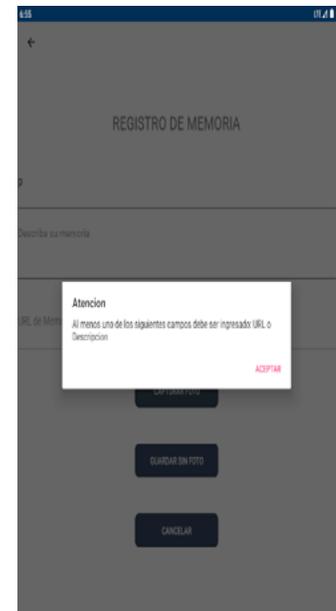


Figura 25.

Error de URL



Conclusiones

Para el proyecto fue una parte primordial identificar la necesidad del usuario, así como también los requerimientos para que el prototipo sea efectivo, "Second Mind" es un aplicativo interactivo que permitirá registrar y almacenar información a corto o largo plazo, evitando así un almacenamiento caduco y tedioso, como es el uso innecesario de Google, o correos electrónicos para reenviarse y mantener la información, archivos en Pc que sean limitantes de acceder. Con tan solo el uso de esta app móvil, que tiene una interfaz amigable, se podrá disponer de textos, tutoriales y grabar incluso una URL o imágenes, usando su propia cámara del dispositivo, así traerá y recordará cualquier información que necesite de forma inmediata.

El almacenamiento de la información del prototipo se realizó de forma híbrida, la parte de texto en SQLite, por lo que la ubicación del archivo se especifica en la biblioteca y puede estar en su directorio de inicio en la carpeta del programa de invocación, y la imagen se almacena en el repositorio Amazon S3, por el tamaño que estas disponen, así el manejo de la persistencia de información fue exclusivo para el prototipo.

Para el desarrollo se utilizó Xamarin en Leguaje Visual Basic 2019, con una base SQLite y un repositorio Amazon S3 (proporciona almacenamiento de objetos a través de una interfaz de servicio web), como control de versiones el GitHub para agilizar la creación del prototipo y la metodología Kanban, para concluir dentro de los tiempos establecidos el desarrollo.

Finalmente, se realizaron las pruebas correspondientes, que permitieron verificar la funcionalidad del prototipo y que se cumplan los requerimientos funcionales y no funcionales, levantados al inicio del proyecto.



Referencias

- Amazon,W. (s.f). SDK de AWS para .NET. <https://aws.amazon.com/es/sdk-for-net/>
- B, G. (08 de marzo de 2021). ¿Qué es GitHub y Cómo Usarlo? Hostinger Tutoriales. <https://www.hostinger.es/tutoriales/que-es-github>
- Develou. (s.f). Tutorial De Bases De Datos SQLite En Aplicaciones Android. <https://www.develou.com/android-sqlite-bases-de-datos/>
- Dominguez, D, L. (26 de febrero del 2018). La revolución móvil (enfocada a las Apps). <https://www.delaradominquez.es/apps-la-revolucion-movil/#page-content>
- Gilibets, L. (11 de noviembre del 2020). *Qué es la metodología Kanban y cómo utilizarla*. IEBS. <https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/>
- Kanbanize. (s.f). Qué es Kanban: Definición, Características y Ventajas. <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
- Kawulich, B. (2005). Participant Observation as a Data Collection Method. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 6(2). <https://doi.org/10.17169/fqs-6.2.466>
- Madhuri, T., & Sowjanya, P. (2016). Microsoft Azure v/s Amazon AWS cloud services: A comparative study. *International Journal of Innovative Research in Science, Engineering and Technology*, https://www.academia.edu/download/53932691/Microsoft_Azure_vs_Amazon_AWS_Cloud.pdf
- Madhuri, T., & Sowjanya, P. (2016). Microsoft Azure v/s Amazon AWS cloud services: A comparative study. *International Journal of Innovative Research in Science, Engineering and Technology*, 5(3), 3904-3907. DOI:10.15680/IJIRSET.2016.0503098
- Martín-Barbero, J. (2015). *Comunicación masiva: discurso y poder* (Vol. 1). Ediciones Ciespal.
- Microsoft. (s.f). Desarrollo de aplicaciones .NET. <https://visualstudio.microsoft.com/es/vs/features/net-development/>
- Ruiz, J. (18 de diciembre de 2017). [Xamarin.Forms] Uso de SQLite, múltiples tablas, relaciones y operaciones en cascada. <https://javiersuarezruiz.wordpress.com/2017/12/18/xamarin-forms-uso-de-sqlite-multiples-tablas-relaciones-y-operaciones-en-cascada/>
- SQLite, DB (09 de septiembre de 2018). SQLite Database search in Xamarin Forms <https://stackoverflow.com/questions/52241191/sqlite-database-search-in-xamarin-forms>

Copyright (c) 2022 Roberth Alexander Almache Vallejos, Jorge Ramiro Castro Chavez, Diego Alejandro González Calderon y Johanna Daniela Ruano Cardenas



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)

Analizando Internet de las Cosas y la nube informática

Fecha de recepción: 2021-11-01 • Fecha de aceptación: 2022-01-06 • Fecha de publicación: 2022-02-10

Darwin Stalin Ramírez Supe¹

Ministerio de Agricultura y Ganadería

stalin_ramirez@yahoo.es

<https://orcid.org/0000-0003-0568-6489>

Estefanía de las Mercedes Zurita Meza²

Universidad Técnica de Ambato

tefamerce92@gmail.com

<https://orcid.org/0000-0001-5509-0152>

Francisco Javier Galora Silva³

Universidad Internacional de la Rioja, España

francisco.galora049@comunidadunir.net

<https://orcid.org/0000-0002-5464-5336>

RESUMEN

Este artículo presenta un estudio de 2 nuevas tecnologías que han ido creciendo a pasos agigantados en los últimos años, una de ellas es el Internet de las Cosas (IoT), el cual permite a los usuarios conectar miles de millones de máquinas inteligentes e intercambiar información, monitorear y controlar servicios, como sistemas de automatización del hogar interconectados, atención médica, agricultura, monitoreo de seguridad, redes eléctricas o servicios críticos. Mientras que la otra es el Cloud Computing, que se refiere al acceso, configuración y operación de recursos en ubicaciones remotas.

Se propone hacer un análisis de qué es y qué ofrece el IoT en los diferentes sensores que existen, de igual forma, analizar los servicios que ofrece el Cloud Computing, y qué resultados se obtendrán si se integran estas 2 nuevas tecnologías, permitiendo que trabajen juntos, logrando

con ello la generación de nuevas aplicaciones, nuevas áreas de investigación y finalmente lo que a todas las personas les interesa, cómo facilitar la vida humana mediante el uso de estas tecnologías. Concluyendo que la combinación entre el IoT y Cloud Computing podrían construir una superpotencia de TI creará tecnologías que actualmente son inimaginables y nuevos campos de investigación que beneficiará a la humanidad.

PALABRAS CLAVE: Cloud computing, laas, Paas, Saas, Internet de las cosas, sensores

ABSTRACT

This article presents a study of 2 new technologies that have been growing by leaps and bounds in recent years, one of them is the Internet of Things (IoT), which allows users to connect billions of smart machines and exchange information, monitor and control services, such as interconnected home automation systems, health care, agriculture, security monitoring, power grids or critical services. The other is Cloud Computing, which refers to the access, configuration and operation of resources in remote locations.

It is proposed to make an analysis of what IoT is and what it offers in the different sensors that exist, likewise, analyze the services offered by Cloud Computing, and what results will be obtained if these 2 new technologies are integrated, allowing them to work together, thereby achieving the generation of new applications, new areas of research and finally what everyone is interested in, how to facilitate human life through the use of these technologies. Concluding that the combination of IoT and Cloud Computing could build an IT superpower that will create technologies that are currently unimaginable and new fields of research that will benefit humanity.

KEYWORDS: Cloud computing, laas, Paas, Saas, Internet of things, Ssensors

Introducción

Con el gran avance tecnológico que se viene dando en los últimos años, se han desarrollado nuevas tecnologías en el área de sistemas informáticos, comunicaciones, robótica, realidad virtual, entre otros, por lo que es muy importante explorar la integración de estas diferentes áreas de trabajo. Como parte de los avances en los sistemas informáticos y las comunicaciones, estos cuentan con tecnologías importantes como la computación en la nube y el Internet de las Cosas (IoT), se puede asegurar que al integrar estas tecnologías se pueden desarrollar nuevos sistemas que pueden facilitar y mejorar la calidad de vida de las personas (Hwang et al., 2013).

El IoT permite a los usuarios conectar miles de millones de máquinas inteligentes e intercambiar información, monitorear y controlar servicios, como sistemas de automatización del hogar interconectados, atención médica, agricultura, monitoreo de seguridad, redes eléctricas o servicios críticos. Controlar la infraestructura y controlar el IoT son los próximos métodos contemporáneos. Entre ellos, a través de la digitalización dinámica del sistema físico que se prepara, para brindar servicios de valor agregado para dispositivos móviles, la frontera entre el entorno artificial y el entorno real se reduce continuamente. El IoT se ejecuta mediante un software especialmente diseñado, que se puede realizar detectando, controlando y cambiando el estado de las cosas (Aazam et al., 2014).

Este descubrimiento que ha generado un gran avance tecnológico para la humanidad, debe convertirse en una serie de objetos interconectados en un mundo conexo, tanto así que permita a los cirujanos realizar operaciones de forma remota, a los usuarios a evaluar sus hogares y proveedores de energía. Debido al uso diversificado de la tecnología IoT, el número de dispositivos inteligentes conectados a IoT ha proliferado y se espera que supere los 30 millones para finales de 2022. Con el aumento de los elementos de dispositivos heterogéneos asociados con el IoT y la generación de datos, será difícil para el IoT independiente de aplicar de forma eficaz potencia y ancho de banda a las tareas. Desde este punto de vista, se prevé la integración de la computación en nube y el IoT (Alzakholi et al., 2020), cuando la nube está conectada al IoT, se genera contenido multimedia.

Dado que el valor multimedia consume más capacidad de procesamiento, espacio, almacenamiento y programación de recursos, será fundamental administrarlo de manera efectiva y realizar una administración eficaz de los recursos en la nube. De lo contrario, los servicios de IoT con tareas críticas requieren un procesamiento y una capacidad de respuesta elevados. Recientemente, el IoT y la computación en la nube se han estudiado y aplicado constantemente en varios campos, ya que pueden proporcionar un nuevo método para la percepción inteligente y la conexión M2M (incluyendo hombre a hombre, máquina y máquina a máquina), uso bajo demanda e intercambio eficiente de recursos (Dores et al., 2014).

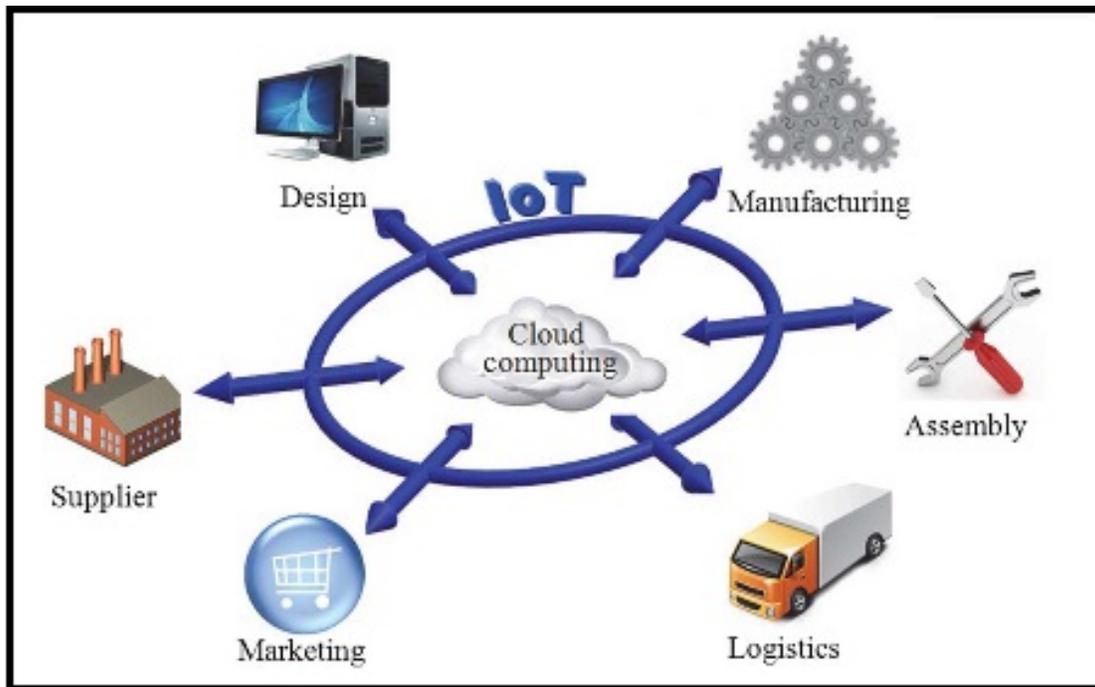
Cuando estas tecnologías funcionan individualmente generan grandes resultados y nuevos campos de investigación, pero si estas dos tecnologías trabajan juntas, los resultados esperados serían asombrosos y se podrían generar soluciones inimaginables que beneficiarían al campo de la ciencia y la investigación.



En particular, la computación en la nube cobrará cada vez más importancia a través de la ejecución de aplicaciones y un modelo de computación de clase mundial, mientras que la cada vez más ubicua IoT será un protagonista tecnológico desde diferentes frentes, incluida la computación en todas partes o la última tecnología, impresoras de última generación, entre otras tecnologías relacionadas con el universo de IoT. La *Figura 1* muestra una idea de cómo funcionarían la computación en la nube y el IoT.

Figura 1.

IoT y computación en la nube



Una de las investigaciones que ha estado trabajando y que incluye estas dos nuevas tecnologías, ha generado el artículo *IoT y Cloud Computing en Automatización de Sistemas de Modelado de Ensamblaje* de los autores Chengen et al. (2014), en el que se propone ayudar a que un sistema de modelado de ensamblajes convencional evolucione hacia un sistema avanzado que sea capaz de hacer frente a la complejidad y los cambios automáticamente.

Además, en el artículo *Traer IoT y Cloud Computing hacia Pervasive Healthcare* de los autores Charalampos Doukas e Ilias Maglogiannis, se presenta una plataforma basada en Cloud Computing para la gestión de sensores de salud móviles y portátiles, demostrando así el paradigma IoT aplicado a el cuidado de la salud, ya que las aplicaciones sanitarias integradas que utilizan redes de sensores corporales generan una gran cantidad de datos que necesitan ser gestionados y almacenados para su procesamiento y uso futuro (Doukas y Maglogiannis, 2012).

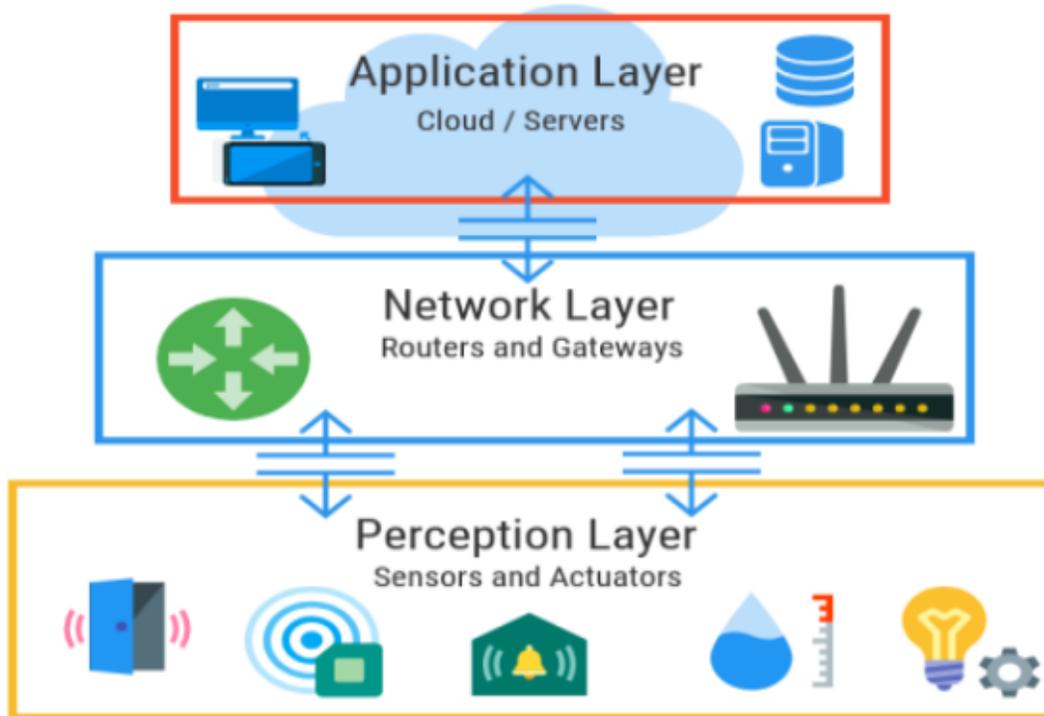
Existen pocos estudios sobre la integración de estas dos importantes nuevas tecnologías, por lo que es necesario realizar un análisis que permita conocer cada una de estas nuevas tecnologías y que se conseguirá haciendo que funcionen juntas.

El presente trabajo está organizado de la siguiente manera. En la sección 2 se habla en detalle de lo que es el IoT, en la sección 3 se aborda todo lo relacionado con el Cloud Computing y se hace una integración de estas nuevas tecnologías, luego se presentan las conclusiones y trabajos futuros, y finalmente se detallan las referencias utilizadas en esta investigación.

Metodología

Los servicios de computación en la nube se implementan en muchos campos relacionados con el IoT, incluido el procesamiento de datos del genoma, la enseñanza y el aprendizaje, los servicios para pymes, los métodos de aprendizaje electrónico, la realidad aumentada, la fabricación, recuperación de emergencias, las ciudades inteligentes, etc., la investigación forense remota, hotelería, negocios, gobierno electrónico y gestión de recursos humanos, Internet del automóvil, entre otros (Perera et al., 2014).

Los desafíos de la computación en nube y el IoT en un entorno de aplicación único y separado ha crecido a pasos agigantados y se han generado nuevos escenarios de estudio para los investigadores (Gongjun et al., 2014). La principal dificultad para estudiar cómo integrar IoT y la computación en la nube es un descubrimiento incierto. La computación en la nube y IoT se han extendido a nivel mundial y su expansión ha sido rápida en los últimos años (Zhang et al., 2010). Cuando se combinan entre sí, las características que muestran serán muy buenas, son muy especiales e importantes entre sí (Aazam et al., 2014). Los investigadores organizaron una serie de aplicaciones relacionadas con la coordinación de la nube y el IoT para desarrollar y acumular datos con la ayuda de capacidades informáticas y de almacenamiento en la nube. Las capas de conocimiento de la *Figura 2* se explican a continuación. Es importante mencionar que las capas de aplicación, red y detección están interconectadas.

Figura 2.*Cloud-IoT arquitectura*

2.1 Internet de las cosas (IoT)

El Internet de las cosas (IoT) es un nuevo paradigma que se está convirtiendo rápidamente en la piedra angular de las comunicaciones inalámbricas modernas. La idea básica detrás de este concepto es la presencia a nuestro alrededor de diferentes cosas como identificación por radiofrecuencia (RFID), sensores, actuadores, teléfonos celulares, etc., que, al cooperar entre sí, pueden lograr objetivos comunes (Atzori et al., 2010).

Sin duda, el punto fuerte de la IoT es el alto impacto que tendrá en diversos aspectos de la vida cotidiana y el comportamiento de las personas. Desde el punto de vista de un usuario privado, los efectos más obvios de la introducción de IoT serán visibles, tanto en el lugar de trabajo, como en el hogar. En este contexto, la domótica, la vida asistida, la ciber salud y la mejora del aprendizaje son solo algunos ejemplos de posibles escenarios de aplicación, en los que el nuevo paradigma jugará un papel protagonista en un futuro próximo. Del mismo modo, desde la perspectiva de los negocios usuarios, las consecuencias más aparentes serán igualmente visibles en campos como la automatización y la fabricación industrial, la logística, la gestión de procesos y empresas, el transporte inteligente de personas y mercancías (Gubbi et al., 2013).

Las múltiples definiciones de IoT que se pueden rastrear en la comunidad de investigadores dan testimonio del gran interés en el tema de IoT y de la vivacidad de los debates al respecto. Al navegar por la literatura, un lector interesado puede experimentar una dificultad real para

Plataforma como servicio (PaaS). El tercer enfoque es proporcionar la aplicación como servicio y se denomina Software como servicio (SaaS) (Emeras et al., 2019). La *Figura 4* presenta la idea principal de estos servicios.

Figura 4.

Servicios de computación en la nube



2.2.1 Infraestructura como servicio IaaS

IaaS es la entrega de hardware como servicio. Es una evolución del alojamiento tradicional que no requiere ningún compromiso a largo plazo y permite a los usuarios aprovisionar recursos bajo demanda. El proveedor de IaaS hace muy poca administración, además de mantener el centro de datos operativo, los usuarios deben implementar y administrar los servicios de software ellos mismos, tal como lo harían en su propio centro de datos. Amazon Web Services Elastic Compute Cloud (EC2) y Secure Storage Service (S3) son ejemplos de ofertas de IaaS (Rajan, 2012).

2.2.2 Plataforma como servicio PaaS

PaaS proporciona una plataforma informática que utiliza la infraestructura de la nube. Tiene todas las aplicaciones que normalmente requiere el cliente implementado en él. Por lo tanto, el cliente no tiene que pasar por las molestias de comprar e instalar el software y el hardware necesarios para ello. A través de este servicio, los desarrolladores pueden obtener todos los sistemas y entornos necesarios para el ciclo de vida del software, ya sea en desarrollo, prueba, despliegue y alojamiento de aplicaciones web. Ejemplos clave son GAE, Azure de Microsoft (Jadeja y Modi, 2012).

2.2.3 Software como servicio SaaS

SaaS es un modelo en el que un proveedor de servicios aloja las solicitudes y las pone a disposición de los clientes a través de una red, normalmente Internet. SaaS se está convirtiendo en un modelo de entrega cada vez más prevalente a medida que maduran las tecnologías

subyacentes que respaldan los servicios web y la arquitectura orientada a servicios (SOA) y los nuevos enfoques de desarrollo se vuelven populares. SaaS también se asocia a menudo con un modelo de licencia de suscripción de pago por uso. Las aplicaciones SaaS también deben poder interactuar con otros datos y otras aplicaciones en una variedad igualmente amplia de entornos y plataformas. SaaS está estrechamente relacionado con otros modelos de prestación de servicios que se ha descrito. SaaS se implementa en la mayoría de los casos para proporcionar funcionalidad de software comercial a clientes empresariales a bajo costo, al tiempo que les permite obtener los mismos beneficios del software con licencia comercial y operado internamente sin la complejidad asociada de instalación, administración, soporte y licencias (Godse y Mulik, 2009).

Resultados

3.1 IoT y cloud computing

La computación en la nube y el Internet de las cosas (IoT), dos tecnologías completamente diferentes, se han convertido en parte de nuestras vidas. Se espera que su adopción y uso generalizado crezca aún más, convirtiéndolos en componentes importantes de Internet en el futuro.

Esencialmente, la *cloud* actúa como capa intermedia entre las cosas y las aplicaciones, donde esconde toda la complejidad y las funcionalidades necesarias para implementar estas últimas. Este marco tendrá un impacto en el futuro desarrollo de aplicaciones, donde la recopilación, el procesamiento y la transmisión de información, producirá nuevos desafíos que deben abordarse, también en un entorno de múltiples nubes (Aguzzi et al., 2013), la *Tabla 1* muestra la complementariedad e integración de la nube y la IoT.

Tabla 1.

Complementariedad e integración de la nube y la IOT

Internet de las cosas (IoT)	Cloud Computing
Ubicuo, cosas colocadas en todas partes	omnipresente, recursos utilizables desde todas partes
Cosas del mundo real	Recursos virtuales
Capacidades computacionales limitadas	Capacidades computacionales virtualmente ilimitadas
Almacenamiento limitado o sin capacidad de almacenamiento	capacidades de almacenamiento virtualmente ilimitadas
Internet como punto de convergencia	Internet para la prestación de servicios
Fuente de un Big Data	Beans para gestionar un Big Data

Las aplicaciones que son posibles o mejoradas significativamente gracias a las tecnologías como IoT y *Cloud* son:

1. **Smart healthcare.** IoT y las tecnologías multimedia han hecho su entrada en el campo de la salud gracias a la vida ambiental asistida y la telemedicina. Los dispositivos inteligentes, Internet móvil y los servicios en la nube, contribuyen a la innovación continua y sistemática de la asistencia sanitaria y permiten servicios médicos ubicuos rentables, eficientes, oportunos y de alta calidad (HU et al., 2017).
2. **Smart City.** IoT puede brindar un entorno común para servicios de *Smart City* orientados al futuro (por ejemplo, representaciones 3D a través de sensores RFID y geotiquetado) y exponiendo la información de manera uniforme. Varias soluciones propuestas recientemente sugieren utilizar arquitecturas de nube para permitir el descubrimiento, la conexión y la integración de sensores y actuadores, creando así plataformas capaces de proporcionar y soportar conectividad ubicua y aplicaciones en tiempo real para ciudades inteligentes (Mitton et al., 2012).
3. **Smart home y smart metering.** IoT tiene una gran aplicación en entornos domésticos, donde los dispositivos integrados heterogéneos permiten la automatización de actividades internas comunes. En este escenario, la nube es el mejor candidato para crear aplicaciones flexibles con solo unas pocas líneas de código, lo que hace que la automatización del hogar sea una tarea trivial (Kamilaris et al., 2011). Para permitir que una variedad de hogares inteligentes unifamiliares independientes accedan a servicios reutilizables a través de Internet, la solución resultante debe satisfacer tres requisitos cruciales: interconexión de red interna (es decir, cada dispositivo digital en el hogar inteligente debe poder interconectarse con cualquier otro), control remoto inteligente (es decir, los electrodomésticos y servicios en el hogar inteligente deben ser gestionados de manera inteligente por cualquier dispositivo, desde cualquier lugar) y automatización (los electrodomésticos interconectados dentro del hogar deben implementar sus funciones mediante la vinculación a los servicios proporcionados por la nube orientada al hogar inteligente) (Wu He et al., 2014).
4. **Video vigilancia.** La videovigilancia inteligente se ha convertido en una herramienta de la mayor importancia para varias aplicaciones relacionadas con la seguridad. Como alternativa a los sistemas de gestión autónomos internos, los análisis de video complejos requieren soluciones basadas en la nube para satisfacer adecuadamente los requisitos de almacenamiento (por ejemplo, los medios almacenados son centralmente seguros, tolerantes a fallas, bajo demanda, escalables y accesibles a alta velocidad) y procesamiento (por ejemplo, procesamiento de video, algoritmos de visión por computadora y módulos de reconocimiento de patrones para extraer conocimiento de las escenas). Las soluciones propuestas almacenan y administran de manera inteligente el contenido de video proveniente de cámaras (IP y analógicas) y lo entregan de manera eficiente a múltiples dispositivos de usuario a través de Internet, mediante la distribución de las tareas de procesamiento a través de los recursos del servidor físico bajo demanda, de una manera equilibrada de carga y tolerante a fallas (Held et al., 2012).
5. **Movilidad inteligente y automotriz.** Como tecnología emergente, se espera que IoT ofrezca soluciones prometedoras para transformar los sistemas de transporte y los

servicios de automóviles (es decir, sistemas de transporte inteligentes, ITS). La integración de tecnologías en la nube con WSN, RFID, redes satelitales y otras tecnologías de transporte inteligente representa una oportunidad prometedora para abordar los principales desafíos actuales. Se puede desarrollar e implementar una nueva generación de nubes de datos vehiculares basadas en IoT para brindar muchos beneficios comerciales, como aumentar la seguridad vial, reducir la congestión vial, administrar el tráfico y recomendar el mantenimiento o reparación de los automóviles (Wu He et al., 2014).

3.2 Desafíos Cloud-IoT

La capa intermedia entre los objetos y las aplicaciones es el almacenamiento en la nube, que oculta matices y funciones. Todos reconocemos que Internet de las Cosas es una red de artefactos enlazados, y en estos objetos están implicadas varias aplicaciones. Los problemas de cada aplicación son únicos, pero generalmente pertenecen a categorías similares. Para resolver estos desafíos se debe prestar más atención a los desafíos de seguridad y evaluar las consecuencias de las nuevas tecnologías. Después de integrar la nube y el Internet de las Cosas, la gente ha estado preocupada por la desconfianza y comprensión de los proveedores de la nube, sobre la ubicación física de la información detallada transmitida a la nube, a través de diferentes protocolos de IoT. Hay varias preguntas sobre el sistema de almacenamiento de servicios en la nube de múltiples inquilinos. La información de los consumidores múltiples se almacena en una instalación, lo que destruirá la confidencialidad y el proceso de fuga de información confidencial. Debido a la desconfianza en los proveedores de servicios en la nube, esta forma de vulnerabilidad se considera una amenaza interna y es, con mucho, uno de los problemas más imprevistos en la industria de TI. Los desafíos clave de Cloud-IoT se explican a continuación.

A. Seguridad. Los datos seguros de IoT se colocan en la nube para su procesamiento y recuperación. Esto implica el cifrado de datos y la seguridad de los datos que se envían o almacenan en repositorios basados en la nube, durante el acceso y uso de la nube. El grado de falta de información de computación en la nube hace que los propietarios de los datos no sean conscientes de la ubicación física de sus datos. Hoy en día, los datos están estrechamente relacionados con todo lo que nos rodea, por lo que la seguridad de los datos en el paradigma Cloud-IoT es el tema principal.

B. Rendimiento computacional y almacenamiento. Los planes de rendimiento de almacenamiento y computación que incluyen el uso de dispositivos IoT basados en la nube requieren requisitos de objetivos de alto rendimiento. Dado que los dispositivos IoT basados en la nube funcionan en muchas aplicaciones, es difícil cumplir con estas especificaciones en todos los entornos.

C. Fiabilidad. Los dispositivos de IoT dependen de la nube para proporcionar trabajo para aplicaciones de tiempo crítico, y el efecto reflejará directamente el resultado del programa. Por ejemplo, en el campo de los instrumentos quirúrgicos para automóviles o la seguridad.

D. Almacenamiento de Big Data. Alrededor de 2025 se proporcionarán casi 50 mil millones de dispositivos IoT, y estos ingresos se convertirán en un obstáculo importante para que los proveedores de servicios en la nube accedan a los datos de forma rápida y segura.

E. Mantenimiento. Con base en el conocimiento aprendido en las secciones anteriores, se necesitan tecnologías y planes extremadamente eficientes para rastrear y administrar la protección y la eficiencia en el entorno de la nube, para satisfacer la demanda de hasta 50 mil millones de dispositivos IoT.

F. Computación al límite. Las restricciones de latencia, las restricciones de movilidad y la implementación de IoT distribuida de Geodis, requieren que la nube responda de inmediato. Por lo tanto, la computación en el borde es un compromiso entre la computación clásica y la computación en la nube. Aunque está más cerca de la implementación, es difícil fusionarse porque requiere conocimiento de la ubicación.

G. Dispositivos de IoT asistidos por el usuario. En una implementación de IoT de este tipo, los usuarios deben incluir detalles y beneficios para compensar su participación en la comunicación. En el arduo desafío desde que empezaron a funcionar los factores sociales, el consumidor ha hecho aportes en su contexto.

H. Interacción con los dispositivos. Los sistemas de Cloud IoT generalmente requieren información de varios dispositivos para su procesamiento e implementación. En este caso, las especificaciones, como el espacio de almacenamiento y la potencia informática basada en la nube, pueden resultar difíciles.

Conclusiones

Como parte de las conclusiones del presente artículo, se determina que la combinación de Internet de las cosas (IoT) y la computación en la nube, así como los servicios de desarrollo, podrían construir una superpotencia de TI que creará tecnologías que actualmente son inimaginables, y nuevos campos de investigación que beneficiará a toda la humanidad, sin importar si están involucrados en el mundo de la tecnología o trabajan en otra zona.

Tanto el IoT como el Cloud Computing son tecnologías que irán creciendo y evolucionando en el futuro, ya que hoy en día uno de los mayores obstáculos es el canal de comunicación entre ellos, por que a mayor demanda de peticiones, se necesita mayor ancho de banda, por ende, con la llegada de las conexiones 5G, permitirá el crecimiento y desarrollo de las tecnologías habladas en el documento presente.

Con este análisis se podrá tener una idea clara de cómo funciona el Internet de las cosas y el Cloud Computing, de esta forma, en el futuro se puede realizar un estudio que proponga utilizar estas 2 tecnologías y que beneficie a la sociedad, esto no será una tarea fácil, pero con un poco de esfuerzo e inversión se puede lograr.

Referencias

- Aazam, M., Khan, I., Alsaffar, A. A. y Huh, E.-N. (2014). Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. En *2014 11th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE. <https://doi.org/10.1109/ibcast.2014.6778179>
- Aguzzi, S., Bradshaw, D., Canning, M., Cansfield, M., Carter, P., Cattaneo, G., & Stevens, R. (2013). Definition of a research and innovation policy leveraging cloud computing and IoT combination. *Final Report, European Commission, SMART, 37*, 2013.
- Alzakholi, O., Haji, L., Shukur, H., Zebari, R., Abas, S. y Sadeeq, M. (2020). Comparison Among Cloud Technologies and Cloud Performance. *Journal of Applied Science and Technology Trends, 1(2)*, 40–47. <https://doi.org/10.38094/jastt1219>
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. y Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM, 53(4)*, 50–58. <https://doi.org/10.1145/1721654.1721672>
- Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks, 54(15)*, 2787-2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Chengen Wang, Zhuming Bi y Li Da Xu. (2014). IoT and Cloud Computing in Automation of Assembly Modeling Systems. *IEEE Transactions on Industrial Informatics, 10(2)*, 1426–1434. <https://doi.org/10.1109/tii.2014.2300346>
- Dores, C., Reis, L. P., & Lopes, N. V. (2014). Internet of things and cloud computing. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-4). IEEE. DOI: [10.1109/CISTI.2014.6877071](https://doi.org/10.1109/CISTI.2014.6877071)
- Doukas, C., & Maglogiannis, I. (2012). Bringing IoT and cloud computing towards pervasive healthcare. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (pp. 922-926). IEEE. DOI: [10.1109/IMIS.2012.26](https://doi.org/10.1109/IMIS.2012.26)
- Emeras, J., Varrette, S., Plugaru, V. y Bouvry, P. (2019). Amazon Elastic Compute Cloud (EC2) versus In-House HPC Platform: A Cost Analysis. *IEEE Transactions on Cloud Computing, 7(2)*, 456–468. <https://doi.org/10.1109/tcc.2016.2628371>
- Godse, M., & Mulik, S. (2009,). An approach for selecting software-as-a-service (SaaS) product. In *2009 IEEE International Conference on Cloud Computing* (pp. 155-158). IEEE.
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems, 29(7)*, 1645-1660. <https://doi.org/10.1016/j.future.2013.01.010>

- Held, C., Krumm, J., Markel, P. y Schenke, R. P. (2012). Intelligent Video Surveillance. *Computer*, 45(3), 83–84. <https://doi.org/10.1109/mc.2012.97>
- Hu, Q.-H., Zhang, S., & Han, Y. (2017). Research of Smart Home Service Platform Based on Mobile Cloud. *DES-tech Transactions on Engineering and Technology Research*, (sste). <https://doi.org/10.12783/dtetr/sste2016/6503>
- Hwang, K., Dongarra, J. y Fox, G. C. (2013). *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann.
- Jadeja, Y., & Modi, K. (2012). Cloud computing-concepts, architecture and challenges. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)* (pp. 877-880). IEEE.
- Kamilaris, A., Pitsillides, A., & Trifa, V. (2011). The smart home meets the web of things. *International Journal of Ad Hoc and Ubiquitous Computing*, 7(3), 145-154. <https://doi.org/10.1504/ijahuc.2011.040115>
- Mitton, N., Papavassiliou, S., Puliafito, A., & Trivedi, K. S. (2012). Combining Cloud and sensors in a smart city environment. *EURASIP journal on Wireless Communications and Networking*, 2012(1), 1-10. <https://doi.org/10.1186/1687-1499-2012-247>
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2013). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1), 414-454. <https://doi.org/10.1109/surv.2013.042313.00197>
- Rajan, R. A. P. (2012). Evolution of Cloud Storage as Cloud Computing Infrastructure Service. *IOSR Journal of Computer Engineering*, 1(1), 38–45. <https://doi.org/10.9790/0661-0113845>
- Wu He, Gongjun Yan., & Li Da Xu. (2014). Developing Vehicular Data Cloud Services in the IoT Environment. In *IEEE Transactions on Industrial Informatics*, 10(2), 1587–1595. <https://doi.org/10.1109/tii.2014.2299233>
- Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1), 7–18. <https://doi.org/10.1007/s13174-010-0007-6>

Copyright (c) 2022 Darwin Stalin Ramírez Supe, Estefanía de las Mercedes Zurita Meza y Francisco Javier Galora Silva



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)

NORMAS DE PUBLICACIÓN REVISTA ODIGOS

La **REVISTA ODIGOS** es una publicación científica de la Universidad Tecnológica Israel, cuya difusión es trianual: febrero, junio y octubre.

El propósito es publicar, en español e inglés, trabajos de investigación científica y desarrollo tecnológico vinculados a las Ciencias de la Ingeniería y Exactas (ciencia y tecnología, computación, física, matemática, telecomunicaciones, electrónica).

Está dirigida a docentes, investigadores y profesionales que estén interesados en la actualización y el seguimiento de los procesos de investigación científico-tecnológica en esta área del conocimiento. Es de acceso abierto y gratuito, e incluye artículos originales de investigación, ensayos y reseñas.

Es importante acotar que las evaluaciones se hacen con pares a doble ciego para garantizar la objetividad y la calidad de las publicaciones.

1. ALCANCE Y POLÍTICA

Las aportaciones tienen que ser originales y no haber sido publicados previamente o estar en proceso de revisión de otro medio.

Estas pueden ser mediante:

- **Artículos:** trabajos de naturaleza teórica y empírica con una extensión de entre 12 y 16 páginas, incluyendo título, resúmenes, descriptores, tablas y referencias.

La estructura a seguir es la siguiente: título, autor (es), institución, correo electrónico de cada autor, código Orcid (<https://orcid.org/>), resumen, palabras clave, abstract (no se aceptarán traducciones provenientes de traductores automáticos), keywords, introducción, metodología, resultados, conclusiones y referencias bibliográficas.

- **Ensayos:** son revisiones exhaustivas del estado de la cuestión de un tema de investigación reciente y actual justificado mediante la búsqueda sistemática de autores que traten sobre esa problemática. Para esta sección se aceptan trabajos con un máximo de entre 12 y 16 páginas, incluyendo título, resúmenes, descriptores, tablas y referencias.

La estructura a seguir es la siguiente: título, autor (es), institución, correo electrónico de cada autor, código Orcid (<https://orcid.org/>), resumen, palabras clave, abstract (no se

aceptarán traducciones provenientes de traductores automáticos), keywords, introducción, desarrollo, debate/discusión/conclusiones y referencias bibliográficas.

- **Reseñas:** consiste en la valoración crítica de un autor, un libro u obra artística en la que se realice una evaluación o crítica constructiva. Tiene una extensión de máximo 12 páginas incluyendo título, resúmenes, descriptores, tablas y referencias.

La estructura a seguir es la siguiente: título, autor (es), correo electrónico de cada autor, código Orcid (<https://orcid.org/>), resumen, palabras clave, abstract (no se aceptarán traducciones provenientes de traductores automáticos debido a su baja calidad), keywords, introducción, desarrollo, debate/discusión/conclusiones y referencias bibliográficas.

2. PROCESO EDITORIAL

Se informa a los autores que los trabajos que se publicaran deben respetar el formato de la plantilla establecida y ser enviados exclusivamente por el OJS (Open Journal System): <https://revista.uisrael.edu.ec/index.php/ro>, por esa vía se manejará el proceso de estimación/desestimación y de aceptación/rechazo, así como en caso de aceptación, el proceso de revisión.

En el período máximo de 30 días, a partir de la recepción de cada trabajo, los autores recibirán una notificación. En caso de que el manuscrito presente deficiencias formales o no se incluya en el focus temático de la publicación, el Editor principal o Director Científico desestimarán formal o temáticamente el trabajo sin opción de reclamo por parte del autor. Por el contrario, si presenta carencias formales superficiales, se devolverá al autor para su corrección antes del inicio del proceso de evaluación. Para ello se establecen las siguientes categorías: **aceptado, aceptado con cambios menores, aceptado con cambios mayores, rechazado.**

Se solicita a los autores que una vez recibida la resolución por parte del Editor de la Revista o del Director Científico envíen el documento corregido en no más de 30 días para una segunda revisión, salvo a aquellos autores a quienes se ha notificado su documento como rechazado.

Los manuscritos serán evaluados científicamente, de forma anónima por pares expertos en la temática, con el fin de garantizar la objetividad e independencia de la Revista.

Los criterios de valoración para la aceptación/rechazo de los trabajos por parte del Consejo Editor son los siguientes:

- Actualidad y novedad.
- Relevancia y significación: avance del conocimiento científico.
- Originalidad.
- Fiabilidad y validez científica: calidad metodológica contrastada.
- Organización (coherencia lógica y presentación formal).
- Coautorías y grado de internacionalización de la propuesta y del equipo.



- Presentación: buena redacción.

3. PRESENTACIÓN Y ESTRUCTURA DE ORIGINALES

Los trabajos se presentarán en tipo de letra Times New Roman 12, interlineado simple, con alineación a la izquierda y sin tabuladores ni retornos de carros entre párrafos. Solo se separan con un retorno los grandes bloques (autor, título, resúmenes, descriptores, créditos y epígrafes). Los trabajos se presentan en Word para PC. Las normas de citas y bibliografía se basan en APA 7ma edición.

A continuación, se detalla en profundidad como debe desarrollarse el texto académico:

- Nombre y apellidos completos de cada uno de los autores por orden de prelación, el número deberá estar justificado por el tema, su complejidad y su extensión, siendo 4 el máximo. Junto a los nombres ha de seguir la institución, correo electrónico de cada autor y código ORCID.
- Resumen en español con un máximo de 200 palabras, donde se describirá de forma concisa el motivo y el objetivo de la investigación, la metodología empleada, los resultados más destacados y principales conclusiones, con la siguiente estructura: justificación del tema, objetivos, metodología del estudio, resultados y conclusiones. Ha de estar escrito de manera impersonal en tercera persona: “El presente trabajo se analizó...”.
- Abstract en inglés con un máximo de 200 palabras. Para su elaboración, al igual que para el título y los keywords, no se admite el empleo de traductores automáticos. Los revisores analizan también este factor al valorar el trabajo
- De 4-6 palabras clave en español/ 4-6 keywords en inglés.
- Introducción: debe incluir los fundamentos y el propósito del estudio, utilizando citas bibliográficas, así como la revisión de la literatura más significativa proveniente de fuentes válidas y de calidad académica.
- Metodología: Será presentado con la precisión que sea conveniente para que el lector comprenda y confirme el desarrollo de la investigación. Se describirá el enfoque metodológico adoptado, la población y muestra, así como las técnicas seleccionadas.
- Resultados: se realizará una exposición de la información recabada durante el proceso de investigación. En caso de ser necesario los resultados se expondrán en figuras o/y tablas (**Ver plantilla de estilo**).
- Conclusiones: resumirá los hallazgos, relacionando las propias observaciones con otros estudios de interés, señalando aportaciones y limitaciones sin reiterar datos ya comentados en otros apartados.
- Referencias bibliográficas: Las citas bibliográficas deben reseñarse en forma de referencias al texto. No debe incluirse bibliografía no citada en el texto. El número de referencias bibliográficas deben ser como mínimo 12 y máximo 20, cantidad necesaria para contextualizar el marco teórico, la metodología usada y los resultados de investigación. Se

presentarán alfabéticamente por el apellido primero del autor (agregando el segundo solo en caso de que el primero sea de uso muy común, y unido con guion). Debe usarse la norma APA 7ma edición.

- Apoyo financiero (opcional): El Council Science Editors recomienda a los autor/es especificar la fuente de financiación de la investigación. Se considerarán prioritarios los trabajos con aval de proyectos competitivos nacionales e internacionales. En todo caso, para la valoración científica del manuscrito, este debe ir anonimizado con XXXX solo para su evaluación inicial, a fin de no identificar autores y equipos de investigación, que deben ser explicitados posteriormente en el manuscrito final.

4. DERECHOS DE AUTOR

Los autores que participen de los procesos de evaluación y publicación de sus ediciones conservan sus derechos de autor, cediendo a la revista el derecho a la primera publicación, tal como establecen las condiciones de reconocimiento en la licencia [Creative Commons Reconocimiento 4.0 Internacional](#) (CC BY), donde los autores autorizan el libre acceso a sus obras, permitiendo que los lectores copien, distribuyan y transmitan por diversos medios, garantizando una amplia difusión del conocimiento científico publicado.

5. LISTA DE COMPROBACIÓN PARA ENVÍOS

Los investigadores deberán llenar en el OJS la lista de comprobación para envíos. En caso de que no cumpla uno de los requisitos, el autor no podrá subir el archivo. Por ello es necesario que se revisen los siguientes parámetros antes de enviar el documento.

- El envío no ha sido publicado previamente ni se ha sometido a consideración por ninguna otra revista (o se ha proporcionado una explicación al respecto en los comentarios al editor/a).
- El archivo de envío está en formato Microsoft Word.
- Siempre que sea posible, se proporcionan direcciones URL para las referencias.
- El texto alineado a la izquierda con tiene interlineado sencillo; letra Times New Roman, 12 puntos de tamaño de fuente.
- El texto se adhiere a los requisitos estilísticos y bibliográficos resumidos en las [Directrices para autores](#).
- Si se envía a una sección evaluada por pares de la revista, deben seguirse las instrucciones en asegurar una evaluación anónima.

6. PRÁCTICAS DESHONESTAS: PLAGIO Y FRAUDE CIENTÍFICO

En el caso de que haya algún tipo de infracción contra los derechos de la propiedad intelectual, las acciones y procedimientos que se deriven de esa situación serán responsabilidad de los autores/as. En tal sentido, cabe mencionar las siguientes infracciones graves:



- **Plagio:** consiste en copiar ideas u obras de otros y presentarlas como propias, como por ejemplo el adoptar palabras o ideas de otros autores sin el debido reconocimiento, no emplear las comillas en una cita literal, dar información errónea sobre la verdadera fuente de la cita, el parafraseo de una fuente sin mencionarla, el parafraseo abusivo, incluso si se menciona la fuente.
- **Fraude científico:** consiste en la elaboración, falsificación u omisión de información, datos, así como la publicación duplicada de una misma obra y los conflictos de autoría. **CITACIÓN Y REFERENCIAS BIBLIOGRÁFICAS** el sistema de citación y referencias bibliográficas se ajustarán a las American Psychological Association (Normas APA, 7^a. edición).
- Se respetará de forma tácita el orden de los autores que figure en el documento original enviado.





Uisrael - 2022

Francisco Pizarro E4-142 y Marieta de Veintimilla
Teléfono: (593) 2 255-5741
rodigos@uisrael.edu.ec
Quito - Ecuador