

Prototipo de mobile app con MongoDB Realm para gestión de vuelos en American Touring

Jimmy Fernando Castillo Crespín
Maestría en software, cohorte II
Universidad Técnica de Machala
Machala, Ecuador
jfcastilloc_est@utmachala.edu.ec

Abstract—Desde la aparición de los sistemas operativos Android e IOS, las aplicaciones móviles han sido muy utilizadas por las empresas, en especial para el campo del ecommerce debido a su fácil accesibilidad, interconectividad y mejoramiento de la experiencia de usuario por adquirir o utilizar algún producto o servicio. La implementación de mobiles app ofrece ciertos beneficios como rentabilidad, tener mejor posicionamiento, innovación, mayor ventaja competitiva, retención de clientes entre otros beneficios [1] y para el desarrollo de estas aplicaciones móviles existen diversas tecnologías como Ionic, ReactJS, Angular pero estas tecnologías solo cubren la parte de desarrollo más no la interacción con los endpoints de servicios, ni seguridad ni escalabilidad [2], siendo estos aspectos una labor adicional por parte de los desarrolladores y arquitectos de software y actualmente con la inclusión del almacenamiento en la nube nació una nueva alternativa que es MongoDB Realm que ofrece SDKs para el desarrollo ágil de aplicaciones móviles y una interacción directa desde el cliente con MongoDB Atlas y sus endpoints [3].

Por tal motivo, en este artículo se desarrollará un prototipo de aplicación móvil utilizando la arquitectura de MongoDB Realm desarrollando también una pasarela de pagos utilizando mongo para solventar la necesidad de pagos online que requiere la empresa American Touring.

Keywords—*MongoDB Realm, SDK, MongoDB Atlas, escalabilidad, mobile app.*

I. INTRODUCCIÓN

En el año 2017, la empresa Furry Analytics realizó una investigación sobre la usabilidad de las aplicaciones móviles en los usuarios, mostrando resultados interesantes como el tiempo en que los usuarios pasan conectados en sus dispositivos móviles, siendo un promedio de cinco horas diarias, registrándose ganancias de 76 mil millones de dólares de ingresos a las empresas dueñas de estas aplicaciones [4], demostrando la rentabilidad que ofrecen las apps para las empresas. Por tal motivo, la implementación de las aplicaciones móviles en cada uno de los modelos de negocio en las empresas es una inversión primordial para el crecimiento de las mismas.

En la actualidad, muchas empresas optan por utilizar las redes sociales como medio de interacción con sus usuarios [5] o por utilizar algún software existente del mercado, pero existe un alto índice de empresas que no conocen o no se animan a desarrollar e implementar sus propias aplicaciones, dado a los altos costos de desarrollo y mantenimiento del mismo, tal y como lo detalla el autor [6].

Pero las empresas que si se animan a tener sus propias aplicaciones se encuentran con otras problemáticas durante el desarrollo de aplicaciones móviles como son la escalabilidad, costos, seguridad, interoperabilidad y compatibilidad de aplicaciones y aunque actualmente ya existen maneras de solventar estas problemáticas, el costo económico en cuestión

laboral y computacional para darle solución siguen siendo demasiados altos [7].

Una propuesta de solución para la problemática de costos y escalabilidad es MongoDB Realm, su arquitectura simplifica y minimiza el tiempo para desarrollar aplicaciones tanto webs como móviles, esto se debe a que Realm proporciona a los programadores algunos SDK, entre ellos se destacan React, Kotlin y Swift, con las API's necesarias para realizar conexiones directas desde el frontend hacia MongoDB e interactuar con varios de los servicios ofrecidos por esta tecnología como Realm Functions y Realm GraphQL [8].

El objetivo principal de esta investigación es la de desarrollar un prototipo de mobile app para la compra de boletos de avión y para esto se utilizará la arquitectura de MongoDB Realm dado a que las características ofrecidas por esta arquitectura lograrán cumplir con las necesidades de la empresa American Touring que requiere migrar su modelo de negocio hacia las nuevas tecnologías.

El siguiente trabajo está estructurado primeramente por el resumen y la introducción que dan las pautas iniciales sobre que se tratará la investigación, seguido del apartado II denominado marco teórico donde se detalla la teoría con la cual se sustentará la investigación, en el apartado III denominado desarrollo y resultados se presentará la propuesta de solución, finalizando con las conclusiones y bibliografía.

II. MARCO TEÓRICO

A. MongoDB

El autor [9] lo define como una base de datos orientada a documentos que se caracteriza por su gran escalabilidad y flexibilidad al momento de modelar consultas y otros queries. Dispone de dos ambientes, uno en la nube llamado MongoDB Atlas que dispone de versiones gratuitas y de pago además de ser compartida y dedicada [10] y otro a nivel de servidor orientada más a ser community ofreciendo un alto rendimiento de uso [11].

Mongo al utilizar una estructura de documentos resulta para los desarrolladores una manera muy fácil de aprender y utilizar esta base de datos en sus aplicaciones, dado a que su documentación es muy completa y su comunidad muy amplia [12].

Entre las características más destacables de mongo se encuentran [13]:

- Utiliza Bson para almacenar cualquier tipo de datos en documentos.
- Consultas en tiempo real.
- Utiliza una base de datos distribuida.
- Es de uso gratuito.

- Alta disponibilidad.
- Respaldo integrado de información
- Escalabilidad horizontal
- Ofrece herramientas para la gestión de automatización, monitorización y respaldo.

B. MongoDB Realm

La plataforma de desarrollo propuesta por MongoDB Realm simplifica y minimiza el tiempo para desarrollar aplicaciones tanto webs como móviles, esto se debe a que Realm proporciona a los programadores algunos SDK, entre ellos se destacan React, Kotlin y Swift, con las API's necesarias para realizar conexiones directas desde el frontend hacia MongoDB e interactuar con varios de los servicios ofrecidos por esta tecnología como Realm Functions y Realm GraphQL [8].

La tecnología Realm está conformada por tres capas principales, la primera es conocida como Realm Mobile Database que es una base de datos muy ligera que se encuentra en las aplicaciones clientes, esta capa soluciona uno de los problemas más comunes de las aplicaciones móviles que es el acceso rápido a la información sin la necesidad de llamar webservices e incluso sin tener conexión a internet [14], con respecto a la segunda capa, con Realm Sync se logra esa integración y sincronización con mongo para mantener los datos actualizados entre todos los dispositivos clientes de manera automática para finalmente llevar toda esa información al clústers de MongoDB Atlas [15].

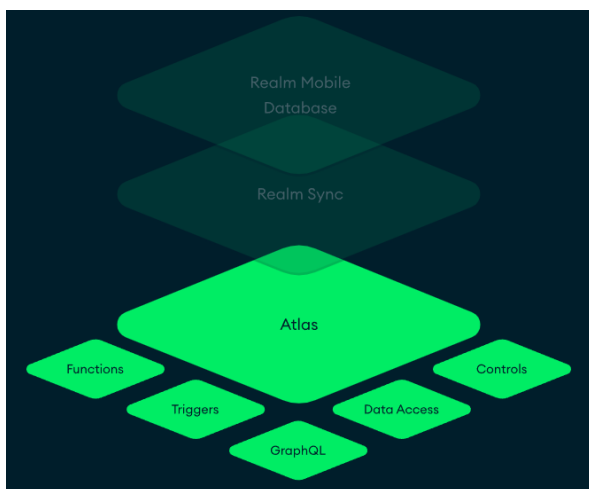


Ilustración 1: Arquitectura de MongoDB Realm
Fuente: [14]

C. MongoDB Atlas

Es el servicio cloud que ofrece MongoDB, nació debido a la gran demanda actual de la utilización de arquitecturas oncloud para aplicaciones, con Atlas se puede gestionar las bases de datos mongo desde cualquier lugar sin la necesidad de instalar herramientas en la computadora, solo basta utilizar cualquier navegador web [16].

Atlas está pensada para ser utilizada en aplicaciones modernas, debido a que utiliza más de 70 regiones de AWS, GCP y Azure, brindándole una gran escalabilidad horizontal de además de una gran automatización, disponibilidad, escalabilidad, seguridad y privacidad [17].

III. DESARROLLO Y RESULTADOS

Se parte inicialmente de los requerimientos funcionales proporcionados por los stakeholders de American Touring:

Tabla 1: Requerimientos funcionales

#	Requerimientos funcionales
1	Mostrar un splashscreen
2	Login con redes sociales de Facebook y Google.
3	Compra de boletos de vuelos.
4	Compra de hospedajes en hoteles.
5	Compra de paquetes (hotel + vuelo)
6	Presentación de ofertas de vuelos.
7	Acreditación de puntos de canjeos.
8	Panel de notificaciones
9	Mapa de locales.

Fuente: Stakeholders de American Touring

Una vez establecido cuales son los requerimientos de usuarios, hay que seleccionar con que tecnología se desarrollará la aplicación móvil y para eso MongoDB Realm soporta muchos SDK para esta tarea, pero dado a la ventaja de ser multiplataforma se optará por utilizar el SDK de React Native.

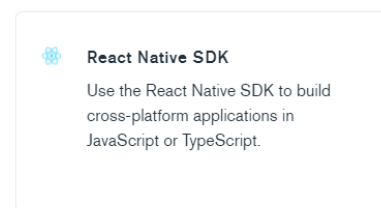


Ilustración 2: SDK a utilizarse en MongoDB Realm
Fuente: Web oficial de MongoDB

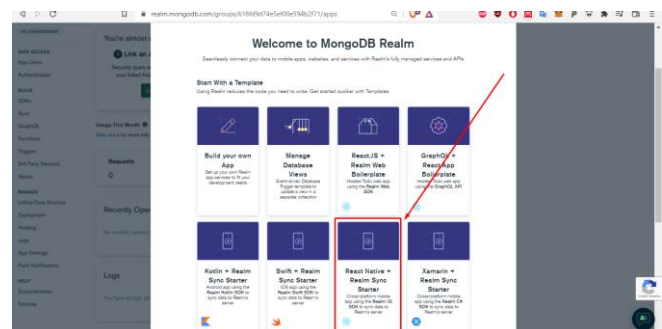


Ilustración 3: Creación de una App React Native en MongoDB Realm
Fuente: Web oficial de MongoDB

Posteriormente, hay que analizar las funcionalidades proporcionadas por Mongo que podrían ser utilizadas para la implementación de la aplicación móvil, siendo seleccionadas las siguientes:

- **MongoDB Realm GraphQL:** se utilizará para acceder de manera más eficiente a los datos proveniente de Atlas.

- **MongoDB Atlas:** hosting utilizado para almacenar la base de datos de mongo donde se consumirá los datos.
- **MongoDB Realm notifications:** se utilizará para notificar al usuario sobre cambios que realice dentro de la aplicación.
- **MongoDB Realm Functions:** se lo utilizará para programar y deployar las funciones (https endpoints) que se utilizará para la comunicación entre los clientes y el servidor.
- **MongoDB Realm Rules:** api Gateway que establecerá los permisos de lectura y escritura de los usuarios a un determinado recurso.
- **Mongo Realm Sync:** componente de mongo que realizará de forma background la sincronización de datos de las aplicaciones clientes hacia el servidor y viceversa.
- **MongoDB Realm Triggers:** módulo que se ejecutará cada vez que se detecte algún cambio en Atlas para posteriormente ser utilizado para el envío de las notificaciones.
- **MongoDB SDK React Native:** módulo utilizado para el desarrollo de la aplicación móvil.
- **MongoDB Realm Users Authentications:** se utilizará este módulo para realizar autenticación con redes sociales.
- **MongoDB Realm Mobile Database:** será utilizado para almacenar datos de manera local y así la aplicación pueda funcionar de manera offline.
- **MongoDB Payments:** pasarela de pagos desarrollado por el equipo de Mongo, será utilizado para las compras de boletos o reservaciones de hoteles dentro de la aplicación.

El resultado de la selección de todas estas funcionalidades se ve reflejado en la ilustración 4 a través de un diagrama de arquitectura C4.

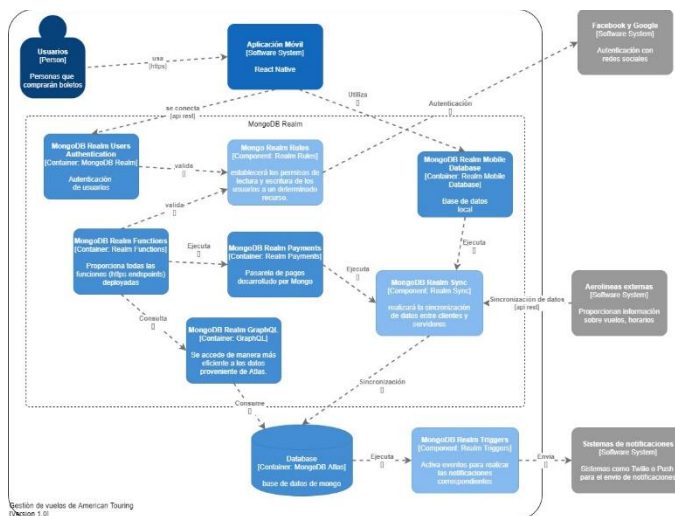


Ilustración 4: Diagrama arquitectónico de la aplicación móvil con Realm
Fuente: Elaboración propia

Finalmente, se diseñó prototipos de pantallas para la aplicación móvil, siendo las siguientes:



Ilustración 5: Splashscreen de la aplicación
Fuente: Elaboración propia

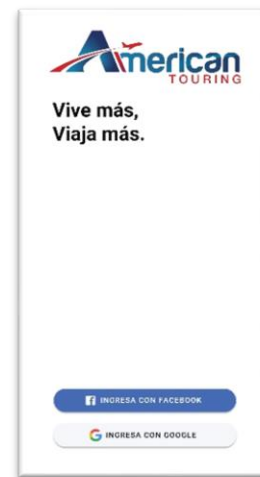


Ilustración 6: Login con redes sociales
Fuente: Elaboración propia

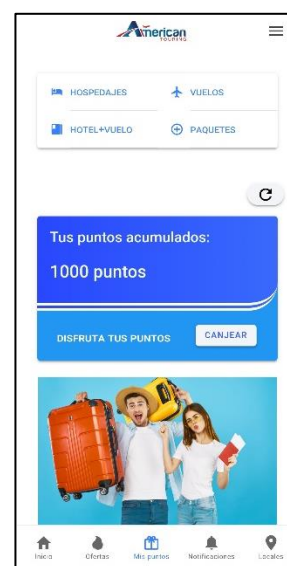


Ilustración 7: Pantalla de inicio
Fuente: Elaboración propia

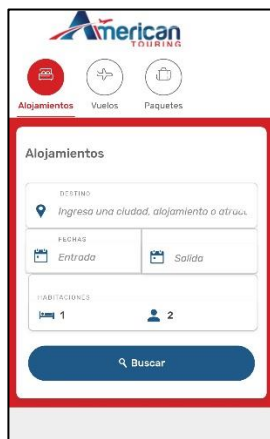


Ilustración 8: Reservación de alojamientos
Fuente: Elaboración propia



Ilustración 9: Compra de boletos de avión
Fuente: Elaboración propia

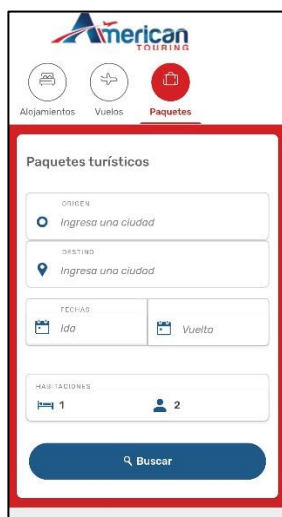


Ilustración 10: Compra de paquetes (hotel+vuelo)
Fuente: Elaboración propia



Ilustración 11: Mapa de locales
Fuente: Elaboración propia

IV. CONCLUSIONES

Al finalizar con esta investigación se logró evidenciar los beneficios que ofrece MongoDB Realm en comparación a las metodologías de desarrollo de aplicaciones móviles tradicionales y además de ofrecer soluciones a varios desafíos que enfrentan los desarrolladores de aplicaciones móviles, como tener siempre información actualizada en los dispositivos clientes ya que las conexiones se pueden perder o los dispositivos se pueden apagar y sincronizar la información en tiempo real es una tarea que Realm Sync lo realiza eficientemente, la creación de Apis con Realm para mantener un mismo esquema de comunicación entre dispositivos clientes, el backend y las bases de datos, programar en distintos sdk para diversos sistemas operativos móviles, encriptación de datos y seguridad, almacenamiento local, funcionamiento sin conexión a internet, interfaz de usuario reactiva.

Todos los desafíos mencionados anteriormente se los pueden resolver de manera aislada, pero esto presenta un esfuerzo adicional ya que se requiere buscar y seleccionar diferentes librerías o frameworks y muchas de las veces la combinación de múltiples librerías ocasiona problemas al momento de probarlas en otros sistemas operativos, una librería puede funcionar perfectamente en Android, pero no funcionar en IOS, al utilizar MongoDB Realm esta problemática se soluciona.

A pesar de que en esta investigación solamente se elaboró un prototipo de desarrollo e implementación de una app, solo basta con observar los beneficios ofrecidos por la tecnología de MongoDB Realm para darse cuenta que el tiempo de desarrollo y costos de aplicaciones móviles son relativamente más rápidas y de bajo costo en comparación con otras tecnologías existentes en el mercado.

REFERENCES

- [1] B. Lim, Y. Xie and E. Haruvy, "The impact of mobile app adoption on physical and online channels," *Journal of Retailing*, 2021.

- [2] S. Gao, L. Liu, Y. Liu, H. Liu and Y. Wang, "API recommendation for the development of Android App features based on the knowledge mined from App stores," *Science of Computer Programming*, vol. 202, 2021.
- [3] J. J. Davis, M. Hirschhorn and J. Schvimer, "eXtreme Modelling in Practice," *PVLDB (Proceedings of the VLDB Endowment)*, vol. 13, no. 9, pp. 1346-1358, 2020.
- [4] B. Thiagaraj, Akalya and J. Joicy, "Mobile Data Analytics: An Overview of Tools," *International Journal of Scientific Research in Computer Science Applications and Management Studies*, vol. 8, no. 2, 2019.
- [5] R. Imtiaz, S. Q. u. A. Kazmi, M. Amjad and A. Aziz, "The impact of social network marketing on consumer purchase intention in Pakistan: A study on female apparel," *Management Science Letters*, vol. 9, no. 7, pp. 1093-1104, 2019.
- [6] E. Osmanbegović, M. Suljić and H. Agić, "A REVIEW OF ESTIMATION OF SOFTWARE PRODUCTS DEVELOPMENT COSTS," *Ekonomski vjesnik : Review of Contemporary Entrepreneurship, Business, and Economic Issues*, vol. 30, no. 1, pp. 195-207, 2017.
- [7] S. Bilgaiyan, S. Sagnika, S. Mishra and M. Das, "A Systematic Review on Software Cost Estimation in Agile Software Development," *Journal of Engineering Science and Technology Review*, vol. 10, no. 4, pp. 51-64, 2017.
- [8] D. Matkedar, R. Fatima, S. Danish, S. T. Mahvesh and F. Khanam, "Prototype for Cooked Food Quality Monitoring System using Application Based IOT Integrated System," *International Journal of Research Publication and Reviews*, vol. 2, no. 8, pp. 367-371, 2021.
- [9] B. Jose and S. Abraham, "Exploring the merits of nosql: A study based on mongodb," *International Conference on Networks & Advances in Computational Technologies (NetACT)*, pp. 266-271, 2017.
- [10] C. Huang, "Data Consistency Properties of Document Store as a Service (DSaaS): Using MongoDB Atlas as an Example," *Technology Conference on Performance Evaluation and Benchmarking*, 2019.
- [11] B. Jose and S. Abraham, "Performance analysis of NoSQL and relational databases with MongoDB and MySQL," *Materials Today: Proceedings*, vol. 24, no. 3, pp. 2036-2043, 2020.
- [12] J. Kumar and V. Garg, "Security analysis of unstructured data in NOSQL MongoDB database," *International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, pp. 300-305, 2017.
- [13] MongoDB, "Que es Mongo?," 10 11 2021. [Online]. Available: <https://www.mongodb.com/es/what-is-mongodb>. [Accessed 03 12 2021].
- [14] Mongo, "MongoDB Realm," 2021. [Online]. Available: <https://www.mongodb.com/realm>. [Accessed 03 12 2021].
- [15] A. K. Samanta and N. Chaki, "Performance Monitoring of MongoDB on Varied Cluster Configuration: An Experimental Approach," *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 525-530, 2021.
- [16] A. K. Samanta and N. Chaki, "Performance Monitoring of MongoDB on Varied Cluster Configuration: An Experimental Approach," *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 525-530, 2021.
- [17] Mongo, "MongoDB Atlas," 2021. [Online]. Available: <https://www.mongodb.com/es/cloud>. [Accessed 03 12 2021].