

Arquitectura para la localización de usuarios en aplicaciones de realidad mixta

Architecture for the localization of users in mixed reality applications

José Luis Peralta Luján *

Facultad de Ciencias Físicas y Matemáticas, Universidad Nacional de Trujillo, Av. Juan Pablo II s/n – Ciudad Universitaria, Trujillo, Perú.

* Autor correspondiente: jperalta@unitru.edu.pe (J. Peralta)

DOI: [10.17268/rev.cyt.2022.03.05](https://doi.org/10.17268/rev.cyt.2022.03.05)

RESUMEN

Actualmente gracias a la Realidad Mixta (MR) se está generando una amplia gama de nuevas aplicaciones, las cuales mezclan el mundo real que nos rodea con un mundo virtual generado por un computador, este paradigma tiene como principal objetivo el de sumergir al usuario en un nuevo ambiente, el cual es producto de la fusión antes mencionada, y al que llamamos ambiente de realidad mixta (AmMR). Para lograr esta inmersión es de vital importancia conocer la localización del o los usuarios (especialmente cuando se quiere trabajar en tareas que requieren colaboración entre ellos), y no existiendo aún estudios donde se explore la localización para más de un usuario en estos ambientes, es que el presente trabajo tiene como objetivo el diseño de una arquitectura, para ello se utilizó ATAM (Architecture Tradeoff Analysis Method), el cual permitió la captura de los escenarios de atributos de calidad y la evaluación de esta; así mismo se verificó que los escenarios fueron cubiertos por la arquitectura planteada, obteniendo un tiempo de localización mínimo de 0,10 segundos y máximo de 0,33 segundos, generando alertas de ser necesario en un tiempo menor a un segundo.

Palabras clave: Realidad Mixta; Arquitectura; ATAM.

ABSTRACT

Currently, thanks to Mixed Reality (MR), a wide range of new applications is being generated, which mix the real world that surrounds us with a virtual world generated by a computer. The main objective of this paradigm is to immerse the user in a new environment, which is the product of the aforementioned fusion, and which we call mixed reality environment (AmMR). To achieve this immersion it is vitally important to know the location of the user (s) (especially when you want to work on tasks that require collaboration between them), and there are still no studies where the location for more than one user is explored in these environments. , is that the present work has as objective the design of an architecture, for this ATAM (Architecture Tradeoff Analysis Method) was used, which allowed the capture of the quality attribute scenarios and the evaluation of this; Likewise, it was verified that the scenarios were covered by the proposed architecture, obtaining a minimum location time of 0.10 seconds and a maximum of 0.33 seconds, generating alerts if necessary in less than one second.

Keywords: Mixed Reality; Architecture; ATAM.

1. INTRODUCCIÓN

Actualmente gracias a la Realidad Mixta (MR) se está generando una amplia gama de nuevas aplicaciones, las cuales mezclan el mundo real (ambiente interno) que nos rodea con un mundo virtual generado por un computador (Farshid et al., 2018). Este paradigma tiene como principal objetivo el de sumergir al máximo al usuario en un nuevo ambiente, el cual es producto de la fusión antes mencionada, y al que llamaremos ambiente de realidad mixta (AmMR), pero para lograr ello es de vital importancia conocer la localización del o los usuarios (especialmente cuando se quiere trabajar en tareas que requieren colaboración entre ellos dentro del AmMR) y no existiendo aún estudios donde se explore la localización para más de un usuario en estos ambientes, es que el presente trabajo busca diseñar una arquitectura la cual permita localizar múltiples usuarios en aplicaciones de realidad mixta en ambientes internos.

En la revisión teórica se encontró que existen a nivel mundial muchos estudios que son nombrados como investigaciones hechas en aplicaciones de realidad mixta, como por ejemplo: en programación interactiva de



robots en Ostanin y Klimchik (2018), comunicación de diseño de construcción eléctrica en Chalhoub y Ayer (2018), recorridos turísticos en Nisi et al. (2018), neurocirugía en Incekara et al. (2018), formación docente en Ke et al. (2016) y muchos otros más, pero es necesario tomar en cuenta algunos aspectos los cuales están asociados a la complejidad del problema al cual nos enfrentamos, esto con el fin de determinar y mostrar con claridad la necesidad de la localización de los usuarios en un AmMR.

El primer aspecto a considerar es la distinción de las aplicaciones de realidad mixta de las de realidad aumentada (AR) y realidad virtual (VR), ya que MR debe estar en una dimensión independiente, la cual está caracterizada por la fusión del mundo real con los hologramas creados por una computadora (Flavián et al., 2018). Bajo esta afirmación, por ejemplo, la investigación de formación docente antes mencionada (Ke et al., 2016) no sería parte de lo que nosotros consideraremos aplicaciones de realidad mixta, ya que básicamente lo que las personas pueden observar en la aplicación es un escenario completamente virtual.

El segundo aspecto es referido a la colaboración y por tanto la interacción entre usuarios del AmMR, Tieto y Turner (2022) son los primeros en hablar de experiencias compartidas en MR; dando importancia a que muchos usuarios pueden colaborar y tomando como referencia una de las preguntas (la primera) que se hacen para ayudar a definir escenarios compartidos, la cual es: ¿Cómo están compartiendo?, es que muestran una de las categorías de una clasificación de manera de compartir llamada Colaboración.

El tercer aspecto a considerar es la posición de los objetos tomando en cuenta la localización del o de los usuarios, debido a que una persona puede considerar a su entorno en sus acciones, por ejemplo, cuando coloca un objeto virtual en un lugar que sea significativo para él (Turner et al., 2022). Es necesario que este objeto se pueda guardar respecto a la posición relativa del usuario, esto porque un desplazamiento de él podría causar un desplazamiento del objeto (Parveau y Adda, 2018). Por ejemplo, si un usuario tiene una caja virtual bajo una mesa, este para verla tendría que inclinarse (Flavián et al., 2018).

Los aspectos antes mencionados muestran la importancia y necesidad que se tiene de determinar la localización de los usuarios en un AmMR, debido a que los objetos virtuales interactuarán con la persona o personas de una forma u otra (tal vez desplazándose respecto a sus movimientos, apareciendo o desapareciendo, etc.) de acuerdo al lugar donde se encuentren en el AmMR. Si a esto le sumamos que en el AmMR ya no solo se podría tener un único usuario si no otros más que requieren interactuar (necesidades de colaboración mostradas anteriormente), será necesario contar con la localización de todos ellos; y si a todo esto le agregamos lo complicado de la estructura del ambiente interno (físico), ambientes que en el mundo real son encontrados en diferentes formas, ya que es casi imposible encontrar dos ambientes idénticos (Tieto y Turner, 2022). Nos encontramos con un escenario difícil de manejar al enfrentarnos al problema de: ¿Cómo localizar usuarios en aplicaciones de realidad mixta?

Por ello el objetivo de este trabajo es diseñar una arquitectura, a partir de componentes arquitectónicos que abstraigan a los componentes físicos que realizan la funcionalidad del sistema sin imponer restricciones de implementación, con el fin de generar una arquitectura de referencia para localizar múltiples usuarios en aplicaciones de realidad mixta.

2. MATERIALES Y MÉTODOS

Se usó un diseño de investigación pre-experimental de un caso de estudio con una sola medición, el muestreo fue no probabilístico, y se eligió un ambiente de realidad mixta en el cual se evaluó el cubrimiento de atributos de calidad por la arquitectura desplegada.

Para la evaluación de la arquitectura se siguieron las fases de ATAM, en la fase 0 se hizo una planificación y se formó un equipo de evaluación; en la fase 1 se presentó ATAM, las pautas del negocio y la arquitectura a evaluar (la propuesta inicial), luego se identificaron las propuestas arquitectónicas, se generó el árbol de utilidad y se analizaron estas propuestas. En la fase 2 se aplicó una lluvia de ideas y se analizó nuevamente las propuestas arquitectónicas para al final presentar los resultados.

Por último, se desplegó la arquitectura mostrada Figura 1, para ello se utilizó una Notebook HP Pavilion (Tabla 2) sobre la cual se ejecutó el software hecho con el lenguaje de programación Visual C++ y la librería gráfica OpenGL, software que permitió a partir de los fundamentos de computación gráfica generar primero un entorno gráfico donde se realizó la virtualización de la ubicación de más de un usuario presente en el entorno, y a partir de los datos obtenidos de las cámaras utilizadas (Tabla 2) se pudo mostrar las posiciones de los usuarios en un mundo tridimensional (Figura 2) tal como se dieron en el mundo real, el software permitió contabilizar el número de localizaciones por segundo debido a que por cada localización tomada se registraba automáticamente el tiempo en milisegundos en la que se daba, esto permitió obtener los datos de la Tabla 6. Con las mediciones realizadas se pueden hacer la verificación del cubrimiento de los escenarios (Tabla 3) por

la arquitectura desplegada. Cabe resaltar que lo descrito anteriormente también permite obtener un despliegue de la arquitectura que puede ser empleado en aplicaciones de realidad aumentada y realidad virtual.

En la Tabla 1 se muestra las técnicas e instrumentos que se utilizaron.

Tabla 1. Técnicas e instrumentos

Técnica	Instrumento
Brainstorming propuesto en Bass et al. (2013)	Documento para árbol de utilidad
Generación del árbol de utilidad propuesto en Kazman et al. (2000)	Documento para descripción detallada de escenarios
Técnicas Basadas en Escenarios propuesto en Bass et al. (2013)	Matriz de prioridad
Observación	Guía de observación

Para el despliegue de la arquitectura como se mencionó antes se utilizó una Notebook HP Pavilion y 2 cámaras Zed con las características que se muestran en la Tabla 2

Tabla 2. Equipos utilizados en el despliegue de la arquitectura

Equipo	Característica
Notebook HP Pavilion 14-ab109la	Microprocesador. Sexta generación del procesador Intel® Core™ i7-6500U a 2,5 GHz, hasta 3,1 GHz con Turbo Boost, dos núcleos. Caché del microprocesador. 4 MB de caché de nivel 3. Memoria. SDRAM DDR3L de 8 GB (1 DIMM). Gráficos de video. NVIDIA GeForce 940M con memoria de video dedicada de 2048 MB. Puertos externos. 2 USB 3.0 SuperSpeed.
Cámara Zed y Cámara Zed mini	6-axis Pose Accuracy. Posición (+/- 1mm) Orientación (0,1 deg). Tasa de actualización de Pose. Hasta 100 Hz.

Nota: Solo se listan las especificaciones que interesan para el despliegue.

3. RESULTADOS Y DISCUSIÓN

Debido a que “muchos objetivos de negocio se manifestarán como requisitos de atributos de calidad” (Bass et al., 2013, p.65) y que “todos los atributos de calidad son potencialmente objetivos del negocio” (Rick Kazman & Bass, 2005, p.6) se tomó como base la categorización de objetivos de negocio hecha por Kazman y Bas la cual permitió proporcionar un enlace entre la utilidad y los escenarios específicos de atributos de calidad en el proceso de creación del árbol de utilidad (Rick Kazman y Bass, 2005).

Permitiendo así generar los escenarios de atributos de calidad que se obtuvieron como resultado de cada objetivo de negocio identificado, ya que “desde la perspectiva de la evaluación ATAM, la cuestión de los objetivos de negocio es doble para dirigir a los escenarios de atributos de calidad y expresar temas de riesgo en términos de amenazas a los objetivos de negocio” (Rick Kazman y Bass, 2005, p.9).

Se utilizó H (alto), M (medio) y L (bajo) para valorar la importancia del negocio (A) y el riesgo técnico (B).

La Tabla 3 muestra los escenarios específicos de atributos de calidad obtenidos en el proceso de creación del árbol de utilidad.

Tabla 3. Árbol de utilidad

		UTILIDAD	
Atributo de Calidad	Refinamiento de Atributos	Escenarios	
		(A, B)	Escenario (#)
Desempeño	Tiempo de Respuesta	(H, M)	En condiciones normales después del inicio del sistema, la obtención de la posición se tarda un tiempo < 1 segundo. (1)
		(H, H)	Bajo carga pesada (2 usuarios en paralelo), la obtención de la posición tarda un tiempo < 1 segundo. (2)
	Latencia	(H, L)	Las alertas críticas generadas por el sistema se muestran en un tiempo < 1 segundo. (3)
		(H, M)	Ante el acceso de más de un usuario simultáneamente el sistema maneja estas peticiones y responde aceptablemente en un tiempo < 3 segundos. (4)

Disponibilidad	Problemas de Hardware	(H, M)	Cuando un componente hardware falla, el sistema alerta y de ser posible reanuda las operaciones normales en un tiempo < 60 segundos. (5)
	Fallas de Software	(M, M)	Se produce un fallo en el sistema este reanuda la operación en un tiempo < 60 segundos. (6)
Usabilidad	Conformidad	(M, M)	Pueden agregarse nuevos usuarios al sistema en un tiempo < 3 minutos. Tales adiciones tienen que seguir algún paso previo de calibración. (7)
	Capacidad de aprendizaje	(M, L)	Un nuevo usuario puede configurar su acceso y estar operando en un tiempo < 5 minutos. (8)
	Retroalimentación	(L, L)	Los eventos críticos son reportados y visibles para el usuario en un tiempo < 5 segundos. (9)

Nota: H (alto), M (medio) y L (bajo) para valorar la importancia del negocio que clasifica el cliente (A) y el riesgo técnico o la dificultad según el arquitecto para lograrlo (B), (#) número de escenario.

A partir de la valoración que se hizo de estos escenarios se obtuvo una matriz de prioridad la cual se muestra en la Tabla 4

Tabla 4. Matriz de prioridad

Importancia del Negocio	Riesgo Técnico		
	L	M	H
L	9	-	-
M	8	6, 7	-
H	3	1, 4, 5	2

Nota: H (alto), M (medio) y L (bajo), los números indican los escenarios obtenidos en el árbol de utilidad.

Se obtuvo el siguiente diseño de arquitectura, el cual está estructurado por capas como se muestra en la Figura 1.

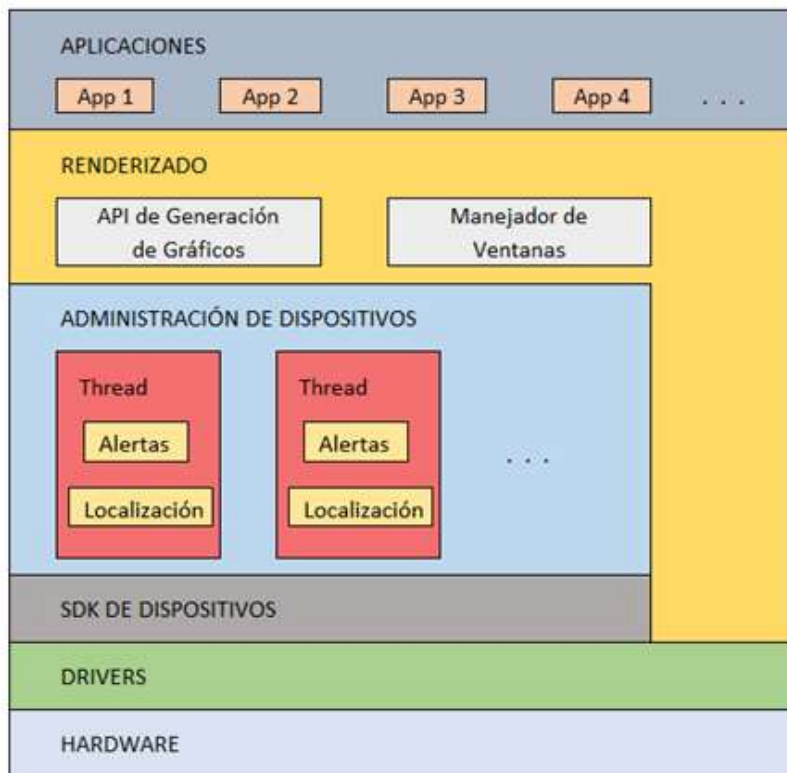


Figura 1. Diseño de la arquitectura.

En esta arquitectura se tienen las siguientes capas:

Hardware. Encapsula los componentes físicos que son necesarios para el procesamiento como una computadora, GPU, puertos USB, etc., así como los dispositivos necesarios para la captura de la localización y captura de imagen del mundo real.

Drivers. Los dispositivos necesitan software que les permitan realizar un correcto funcionamiento y que nos permita enviar y recibir datos a un dispositivo, son los fabricantes los que se encargan de distribuir y desde luego actualizar ellos, por lo tanto, dependen de los dispositivos hardware que se elegirán para una implementación específica.

SDK de Dispositivos. Un Kit de Desarrollo de Software (SDK) permite encapsular muchas de las tareas específicas que tienen los dispositivos en abstracciones de alto nivel que permiten que los desarrolladores puedan disminuir el tiempo de desarrollo de soluciones, es decir proporcionan una API (Interfaz de programación de aplicaciones).

Administración de Dispositivos. Es la parte más importante de la arquitectura debido a que aquí se tiene que administrar a más de un usuario, por ello es necesario generar un Thread para cada uno de ellos, cada hilo realizará de forma independiente la captura de la localización, generación de alertas de ser necesario y captura de imágenes del mundo real.

Renderizado. Es necesario generar un nuevo ambiente que mezcle las imágenes capturadas del mundo real junto con los objetos virtuales para ello se requiere una API de generación de gráficos, la cual nos permita la generación de gráficos por computadora y acceso a la GPU para que este proceso sea mucho más rápido, también es necesario elegir un manejador de ventanas con el cual trabaje esta API y que se encargue de gestionar algunos eventos hechos en la interfaz del sistema.

Aplicaciones. Aquí es donde se agrupan todas las implementaciones, es decir las soluciones a problemas que los usuarios podrían especificar en el futuro, entonces cada aplicación (App) dependerá del caso específico que se plantee.

Al desplegar la arquitectura diseñada se generó un entorno gráfico para poder visualizar las posiciones de los usuarios en el ambiente tal como se muestra en la Figura 2.

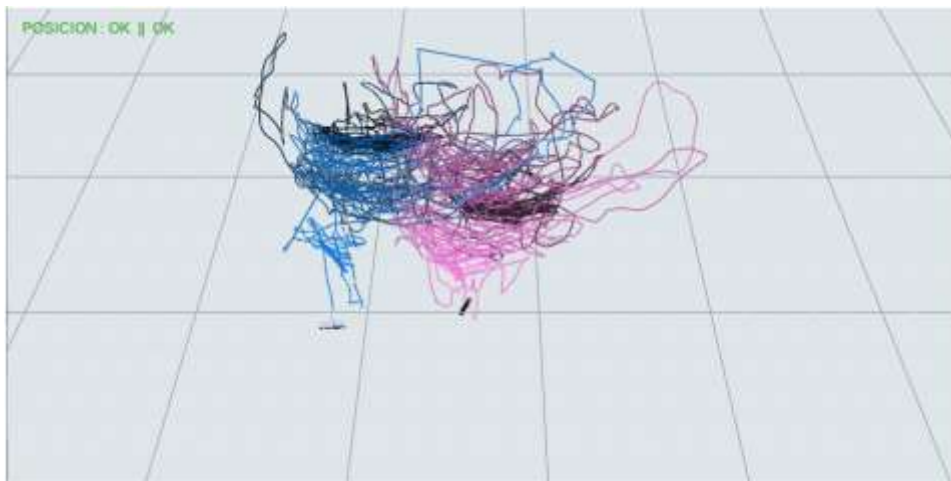


Figura 2. Visualización de la localización de dos usuarios en el ambiente.

Se realizó la captura de datos durante 16,5 minutos obteniendo más de 6000 mediciones de localización en cada uno de los usuarios como se muestra en la Tabla 5.

Tabla 5. Mediciones de localización hechas para los usuarios

Usuario	Número de mediciones
Usuario 1	6105
Usuario 2	6458
Total	12563

Nota: Mediciones hechas durante 16,5 minutos

En la Figura 3 y Figura 4 se puede observar de forma gráfica la cantidad de mediciones de localización obtenidas en cada segundo en ambos usuarios.

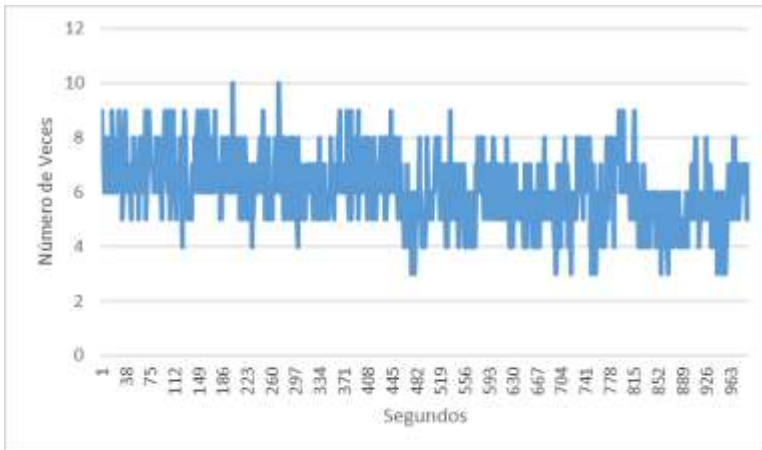


Figura 3. Cantidad de localizaciones por segundo del usuario 1.

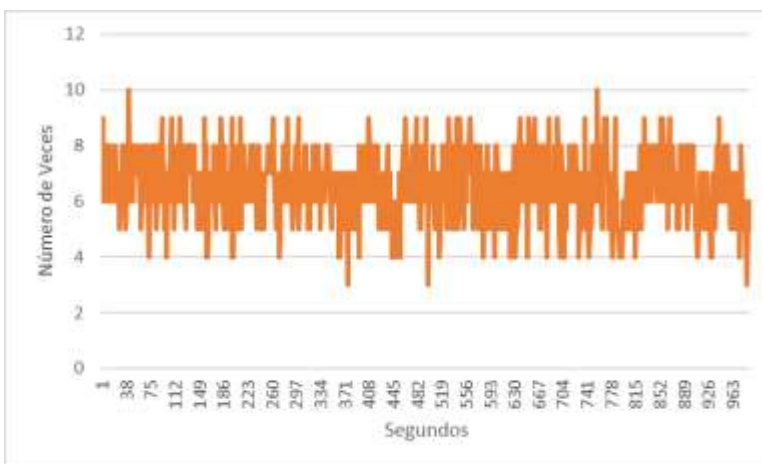


Figura 4. Cantidad de localizaciones por segundo del usuario 2.

Como se puede ver en la Figura 3 y Figura 4 en las mediciones hechas para el usuario 1 y usuario 2 se obtiene en cada segundo varias veces la localización de cada uno de ellos, estos resultados se capturaron al mismo tiempo es decir trabajando con los 2 usuarios de forma paralela; aquí se puede ver que como mínimo se pudo llegar a obtener 3 veces por segundo y como máximo 10 veces por segundo la localización de ellos, esto debido a que la administración de cada usuario se hizo en un hilo independiente tal como se muestra en el diseño de la arquitectura (capa administración de dispositivos) lo cual permitió trabajar con más de un usuario al mismo tiempo.

A partir de estas mediciones se obtuvo el promedio de veces por segundo en que se realiza la localización de los usuarios y el tiempo mínimo y máximo en que se realizó la localización de cada uno de ellos (Tabla 6).

Tabla 6. Mediciones de localización hechas para los usuarios

Usuario	CVMín	CVMáx	PV	TMin	TMáx
Usuario 1	3	10	6,17	0,10	0,33
Usuario 2	3	10	6,52	0,10	0,33

Nota: CVMín y CVMáx son la cantidad mínima y máxima (respectivamente) de localizaciones que se pueden obtener en un segundo; PV es el promedio de veces por segundo en el que se obtuvo la localización y TMin y TMáx son el tiempo mínimo y máximo (respectivamente) en segundos en el que se puede obtener una localización.

En la Figura 5 y Figura 6 se puede observar los resultados obtenidos producto de la verificación de las alertas que se generaron, para poder probar esto se forzó a fallos para ver cómo podrían afectar estas a la obtención de la localización.



Figura 5. Alertas generadas en el usuario 1.



Figura 6. Alertas generadas en el usuario 2.

Cuando se verificó las alertas (Figura 5 y Figura 6) los resultados mostraron que en cada segundo donde se dieron estas en ninguno de los usuarios se ve afectado el número mínimo de veces en el que se pudo obtener la localización, debido a que se encuentra como mínimo 5 veces la localización en algunos de estos segundos, incluso se observa por ejemplo en el usuario 1 (Figura 5) que en el segundo 141 se pudo obtener hasta 40 alertas y 6 veces la localización en ese mismo segundo.

4. CONCLUSIONES

El trabajo tuvo como objetivo diseñar una arquitectura la cual permita localizar múltiples usuarios en aplicaciones de realidad mixta y se logró realizar ello (Figura 1), luego se hizo un despliegue de la misma porque era necesario verificar el cubrimiento de los escenarios (Tabla 3), en ellos se puede encontrar que se planteó, por ejemplo, que en condiciones normales después del inicio del sistema, la obtención de la posición se tarde un tiempo menor a un segundo, que bajo carga pesada (dos usuarios en paralelo), la obtención de la posición tarde también un tiempo menor a segundo y que ante el acceso de más de un usuario simultáneamente el sistema maneje estas peticiones y responda aceptablemente en un tiempo menor a tres segundos; por ello lo más importante es el tiempo máximo de localización que se obtuvo, 0,33 segundos, lo cual cumple con lo solicitado. Llegando incluso a obtener la localización de cada uno de los usuarios en algún momento hasta en diez veces por segundo.

Al requerir trabajar con más de un usuario, una parte fundamental que necesitó la arquitectura fue la de administración de dispositivos (Figura 1) ya que se requería realizar una gestión de cada uno de ellos de forma independiente, por esto para realizar la localización la arquitectura manejó a cada uno en un hilo lo cual permitió que ellos realicen su trabajo de forma individual, obteniéndose un tiempo mínimo de localización de 0,10 segundos en cada uno de ellos. Además, se determina que no existe dependencia entre el número de

localizaciones que se hacen de un usuario y otro, es decir, que el aumento de veces por segundo en el que se realizó la localización en un usuario no causó la disminución de captura de localización en el otro. También, se puede concluir que por las características de esta parte de la arquitectura la agregación de más usuarios no afectará los tiempos de localización.

Se puede ver en los escenarios (Tabla 3) la necesidad de poder manejar alertas debido a que por ejemplo se solicita que las alertas críticas generadas por el sistema se muestren en un tiempo menor a 1 segundo y que cuando un componente hardware falle, el sistema alerte y de ser posible reanude las operaciones normales en un tiempo menor a 60 segundos; por ello la arquitectura desplegada logró generar alertas en un tiempo menor a un segundo incluso forzando al sistema a fallas se pudo determinar que puede generar un máximo de 40 alertas por segundo, esto sin afectar el número de veces en que se realiza la localización de un usuario, es decir, no se ve afectado el tiempo máximo de localización (0,33 segundos) logrando no interferir con otros escenarios.

Con todos los resultados obtenidos en el trabajo se puede determinar que la arquitectura cubre los escenarios generados en el árbol de utilidad permitiendo así localizar a los usuarios. Además, se puede concluir que esta arquitectura también puede servir para aplicaciones de realidad virtual y de realidad aumentada.

REFERENCIAS BIBLIOGRÁFICAS

- Bass, L.; Clements, P.; Kazman, R. 2013. Software architecture in practice. 3ra Edición. Addison-Wesley. Massachussets, United States. 589 pp.
- Chalhoub, J.; Ayer, S. 2018. Using mixed reality for electrical construction design communication. *Automation in Construction* 86: 1-10.
- Farshid, M.; Paschen, J.; Eriksson, T.; Kietzmann, J. 2018. Go boldly!: Explore augmented reality (AR), virtual reality (VR), and mixed reality (MR) for business. *Business Horizons* 61: 657-663.
- Flavián, C.; Ibáñez, S.; Orús, C. 2019. The impact of virtual, augmented and mixed reality technologies on the customer experience. *Journal of Business Research* 100: 547-560.
- Incekara, F.; Smits, M.; Dirven, C.; Vincent, A. 2018. Clinical feasibility of a wearable mixed-reality device in neurosurgery. *World Neurosurgery* 118: 422-427.
- Kazman, R.; Bass, L. 2005. Categorizing business goals for software architectures. Software Engineering Institute. Disponible en: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=7751>.
- Kazman, R.; Klein, M.; Clements, P. 2000. ATAM: Method for architecture evaluation. Software Engineering Institute. Disponible en: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=5177>.
- Ke, F.; Lee, S.; Xu, X. 2016. Teaching training in a mixed-reality integrated learning environment. *Computers in Human Behavior* 62: 212-220.
- Nisi, V.; Dionisio, M.; Barreto, M.; Nunes, N. 2018. A mixed reality neighborhood tour: Understanding visitor experience and perceptions. *Entertainment Computing* 27: 89-100.
- Ostanin, M.; Klimchik, A. 2018. Interactive robot programing using mixed reality. *IFAC-PapersOnLine* 51: 50-55.
- Parveau, M.; Adda, M. 2018. 3iVClass: A new classification method for virtual, augmented and mixed realities. *Procedia Computer Science* 141: 263-270.
- Tieto, V.; Turner, A. 2022. Experiencias compartidas en realidad mixta. Microsoft Docs. Disponible en: <https://docs.microsoft.com/es-mx/windows/mixed-reality/shared-experiences-in-mixed-reality>.
- Turner, A.; Arya, H.; Tieto, V.; Coulter D. 2022. Coordinate systems. Microsoft Docs. Disponible en: <https://docs.microsoft.com/en-us/windows/mixed-reality/coordinate-systems>.