

Evaluación del rendimiento de la transferencia de imágenes en aplicaciones IoT entre los lenguajes de programación Java y Python



Evaluation of image transfer performance in IoT applications between the Java and Python programming languages

Paul Xavier Paguay Soxo.¹, Rosman José Paucar Córdova.², Angel Patricio Flores Orozco.³

Recibido: 14-07-2019 / Revisado: 23-07-2019 / Aceptado: 15-08-2019/ Publicado: 10-09-2019

Abstract

DOI: <https://doi.org/10.33262/cienciadigital.v3i3.4.878>

The purpose of this research project was to compare the performance of the image transfer process using a reduced plate (computer board) between the Java and Python programming languages applied to the video surveillance system implemented within a public institution such as the District Department 07D04 Balsas - Marcabeli - Piñas of the Ministry of Education. For the evaluation, internal and external measures and metrics were established, with a sample of 375 images of a population of 14400, an error of 0.05% resulting in the following, for internal metrics response time was obtained 258% in favor of Java, then in terms of processing time 13.88% in favor of Java, likewise, in terms of external metrics, the use of CPU 86% in favor of Python, the use of RAM, 34% in favor of Python and for Java bandwidth use 20.56 times less KiloBytes than Python, finally applying multicriteria analysis with the NAIADE method it is obtained that with a positive ranking of 0.7 “Java is Better than Python”, so it is evident that it exists a significant difference in the performance of image transfer in Internet-based video surveillance systems of Things between the Java and Python programming languages.

Keywords: Image transfer, Python, Java, performance, video surveillance, reduced plate.

Resumen

El propósito del presente proyecto de investigación fue realizar la comparativa del rendimiento del proceso de transferencia de imágenes utilizando una placa reducida (placa computadora) entre los lenguajes de programación Java y Python aplicado al sistema de video vigilancia implementado dentro de una institución pública como lo es la Dirección Distrital 07D04 Balsas – Marcabeli – Piñas del Ministerio de Educación. Para la evaluación se establecieron medidas y métricas tanto internas como externas, con una muestra de 375 imágenes de una población de 14400, un error de 0.05% dando como resultado los siguiente, para las métricas

¹ Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica. Riobamba, Ecuador, ppaguay@esPOCH.edu.ec

² Universidad Técnica de Machala, Machala, Ecuador. rpaucar@utmachala.edu.ec

³ Escuela Superior Politécnica de Chimborazo, Sede Morona Santiago, Macas, Ecuador, aflores@esPOCH.edu.ec

internas tiempo de respuesta se obtuvo un 258% a favor de Java, luego en cuanto al tiempo de procesamiento 13.88% a favor de Java, así mismo, en cuanto a las métricas externas, el uso de CPU un 86% a favor de Python, el uso de RAM, un 34% a favor de Python y para el ancho de banda Java usa 20.56 veces menos KiloBytes que Python, finalmente aplicando el análisis multicriterio con el método de NAIADDE se obtiene que con un ranking positivo de 0.7 “Java es Mejor que Python”, por lo cual se evidencia que existe una diferencia significativa del rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas entre los lenguajes de programación Java y Python.

Palabras Clave: Transferencia de imágenes, Python, Java, rendimiento, video vigilancia, placa reducida.

INTRODUCCIÓN

A lo largo de la historia se han desarrollado inventos o tecnologías que han cambiado la vida de los seres humanos. Los sistemas de video vigilancia basados en la “Tecnología Emergente” Internet de las Cosas (IoT), son aplicaciones ampliamente utilizadas en la actualidad en las organizaciones de cualquier ámbito, estos sistemas permiten obtener información como imágenes o videos en tiempo real de un lugar a través de una computadora personal o un Smartphone que se conecta a una Computadora de Placa Reducida (en inglés: Single Board Computer o SBC). Una de las opciones que más avances ha tenido en este campo son las placas Raspberry, que para el 2020 habrá 50.000 millones de dispositivos “inteligentes” conectados (Salazar & Silvestre, 2016) y que según (Wai, Jegatheesan, & Chee, 2015) Raspberry puede ser el futuro de las aplicaciones inteligentes (smart) y las comunicaciones cliente - servidor.

Pero en sí ¿Qué es el Internet de las Cosas o por sus siglas en inglés IoT? Según (Evans, 2011), del Internet Business Solutions Group (IBSG) de Cisco, IoT es “sencillamente el punto en el tiempo en el que se conectaron a Internet más cosas u objetos que personas”, lo cual ocurrió entre los años 2008 y 2009, y según estudios (Evans, 2011).

Los medios de comunicación para el envío de información soportados por la Placa Raspberry, dependen de los módulos que se le puedan adaptar, sin embargo, la conexión LAN viene instalada por defecto en su versión Pi B, por otro lado, esta se ve limitada por la necesidad de instalar conexiones de red, por tal motivo la opción de conexión inalámbrica Wi-Fi (Estándar 802.11b) ofrece una ventaja en cuanto a la viabilidad de instalación.

Los sistemas de videovigilancia tienen que permitir visualizar la información en tiempo real por lo tanto requieren del envío constante de información, como lo indican (Leira, Vega, González, & Aracil, 2014) para los casos de servidores streaming, en el que se transfieren por la red de datos, grandes cantidades de fotogramas por segundo, que en lo posterior formarán parte del video resultante. Esto por ende resulta en la necesidad de sistemas que en su estructura apliquen plataformas con un adecuado rendimiento que no declive el servicio en el tiempo, tal como lo indica (Armonk, NY Patente n° 20150220336, 2015), ya que de esto dependerá la calidad del Servicio (Vidal-Silva, Madariaga, & Solís, 2017). En el mismo sentido, según (Medina, Pruebas de rendimiento TIC, 2014), para realizar las pruebas de rendimiento, un criterio de evaluación se basa en métricas externas e internas, entre las métricas externas el mismo establece indicadores como: Número de peticiones atendidas por unidad de tiempo, tiempo medio de respuesta, número medio de errores, y entre las internas están: Uso medio de recursos como CPU y RAM.

Como en la mayoría de sistemas informáticos, existen diversas opciones de lenguajes de programación para el desarrollo aplicaciones con placas reducidas, habiendo opciones

tanto de software propietario y libre, donde los lenguajes de programación de software libre proveen una gran ventaja en cuanto a los costos de implementación. Según el sitio web (TIOBE, 2019) encargada de analizar los lenguajes de programación más utilizados a nivel mundial ubica a los lenguajes de programación Java y Python entre los cuatro primeros puestos, siendo ambas opciones las más utilizadas en los prototipos para la el desarrollo de sistemas basados en IoT. Algunos proyectos de este tipo que se pueden citar tenemos por ejemplo el de (Llanos, Naula, & Valdiviezo, 2014), (Ferdoush & Li, 2014) que utilizan el lenguaje Python mientras en los de (Valeriu , Florin, & Adrian-Viorel , 2014) y (Kryvyy , Serhiy, & Ruslan, 2015) ,utiliza el lenguaje Java para la implementación de soluciones IoT con una placa Raspberry. Con este antecedente se plantea la siguiente interrogante: ¿Cuál de los dos lenguajes de programación Java o Python ofrece un mejor rendimiento en la transferencia de imágenes en un Sistema de Videovigilancia utilizando equipamiento de Internet de las Cosas (IoT) como una placa Raspberry Pi?

La estructura del presente trabajo se detalla a continuación:

- La introducción, problema, objetivos generales y específicos, justificación e hipótesis planteada.
- Marco Referencial, donde se describen conceptos fundamentales de seguridad de información, criptografía y algoritmos criptográficos, arquitectura cliente/servidor, plataformas de desarrollo y modelos de calidad de software
- Diseño de la Investigación, donde se describe el diseño, tipo, métodos y enfoque de la investigación, la población, variables e indicadores, matriz de contingencias, parámetros y herramientas, el método FURPS, prototipos y software utilizado.
- Resultados y discusión, en el cual se detalla la recolección y análisis de datos en base a indicadores establecidos, además se determina el rendimiento total efectivo y la comprobación de la hipótesis.

DESARROLLO

Marco Teórico Referencial

Como antecedentes a la presente investigación se pueden mencionar los siguientes trabajos:

Artículo Científico: “Utilización de la Microcomputadora Raspberry Pi con Capacidad de Comunicación Wi-Fi para la captura de Imágenes mediante Cámara y Almacenamiento de Información en Base de Datos Externa”, presentado por los autores (Naula, Llanos, & Valdiviezo, 2014) en la que concluyen que el prototipo desarrollado puede capturar imágenes de una cámara y enviar a un gestor de base de datos.

La siguiente investigación se titula “Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications”, propuesta por los autores (Li, Ferdoush, 2014), entre los resultados obtenidos es que se desarrolló un sistema de sensores para el envío de información mediante la utilización de la placa Raspberry PI, en la que se concluye que este tipo de sistemas son atractivos debido a que tienen características como: bajo costo, compacto, escalable, fácil de personalizar, fácil de desplegar y mantener.

El artículo titulado “Raspberry Pi as Internet of Things hardware: Performances and Constraints”, de los autores (Maksimović, Vujović, Davidović, Milošević , & Perišić, 2014), en la cual concluyen que “La placa de Raspberry Pi es una opción muy económica, con soporte con un extenso número de periféricos de entrada y salida. La red de comunicación.” Así también que: “Añadiendo el módulo Wi-Fi y proveyendo de acceso a Internet, es posible configurarlo para una comunicación con control remoto, lo cual Raspberry Pi, se hace muy adecuado para aplicaciones con concepto de IoT”.

Por otro lado (Prechelt, 2000) en su artículo “An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl for a search/string-processing program” concluye que existen varias ventajas en las diferentes tareas entre los lenguajes analizados, resaltando la velocidad del lenguaje Perl entre lenguajes de scripts, así también (Cunningham, Subrahmanian, & Westerberg, 2010) en su artículo “User-centered evolutionary software development using Python and Java”, resaltan características como: “Java es un lenguaje de tipo estático verificado que ofrece rendimiento, robustez y modularidad como tal, mientras que Python es un lenguaje de tiempo de ejecución controlado que ofrece prototipado rápido, modificación dinámica en tiempo de ejecución y evaluación retardada”

Con estos antecedentes cabe mencionar que una comparación entre los dos lenguajes materia de la investigación, no se ha encontrado en el ámbito de la transferencia de imágenes utilizando una placa reducida, por lo que es necesario analizar este tema.

- **Raspberry Pi**

Según (Jipsion, 2017) Raspberry Pi, “es un ordenador de placa reducida o (placa única) (SBC) de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.”, este ordenador entre varias ventajas se tiene: el reducido tamaño, lo cual le permite una portabilidad así como el menor consumo de energía necesario para su funcionamiento, en la actualidad la utilización de este dispositivo se ha masificado sobre todo en aplicaciones de domótica, así como en el área del internet de las cosas.

- **Raspbian**

(Raspbian) define a su sistema operativo como: “Es un sistema operativo con licencia libre basado en Debian, optimizado para hardware Raspberry Pi. Un sistema operativo es el conjunto de programas básicos y utilidades que hacen al Raspberry Pi iniciar.”

- **Sockets**

Según (Fúquene, 2011), un socket es “un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Un socket se define como el punto final en una conexión. Los sockets se crean y se utilizan con un sistema de peticiones o de llamadas de función a veces llamados interfaz de programación de aplicación de sockets (Application Programming Interface - API)”.

La mayoría de los lenguajes de programación implementan en sus bibliotecas de clases, métodos aplicando los conceptos de los sockets, hay que recalcar que este método es de nivel más bajo de programación, y sirve de base para las demás implementaciones de frameworks de nivel superior.

- **Rendimiento de los sistemas**

Según (Molero, X.; Ruiz, C.; Rodeño, M., 2004) la tendencia actual dentro del campo de la evaluación de prestaciones se orienta principalmente a utilizar índices que tienen en cuenta el tiempo de respuesta en una PC de un conjunto programas de prueba o evaluación (*benchmarks*).

- **Tiempo de respuesta**

Para (Echeverría, 2016), optimizar los tiempos de respuestas tiene un enorme impacto en la experiencia de los usuarios, navegar de manera fluida y rápida en los sistemas informáticos hace que el usuario se sienta cómodo y se quede más tiempo.

Cuando los tiempos de respuesta del sistema caen más allá de un límite, la interacción fluida desaparece y la eficiencia decae significativamente.

Para el presente estudio, el tiempo de respuesta es el tiempo medido en milisegundos que se necesita para que la petición de video por parte del cliente sea atendida por el servidor.

- **Tiempo de Procesamiento**

El tiempo de procesamiento es el tiempo total que se requiere para completar una transacción, este inicia desde el momento que construye la petición, hasta cuando recibe la respuesta y se procesa la información.

- **Métricas externas**

Para el presente trabajo, el consumo de recursos viene incidido por subíndices como: El porcentaje de Uso de CPU, porcentaje de Uso de RAM, y Por el porcentaje de Uso del Ancho de Banda, se descarta el Uso de Disco por la baja influencia en el proceso de transmisión de los fotogramas.

Marco Metodológico

El presente tipo de trabajo de investigación es cuasi-experimental. En donde se propone diseñar un sistema de comunicación desarrollado en los lenguajes de programación Java y Python y probar su rendimiento del proceso de transferencia de imágenes utilizando una placa reducida para la implementación de un sistema de video vigilancia.

El presente trabajo investigativo tomará un enfoque cualitativo cuantitativo, para establecer indicadores que permitan medir a través de valores adjetivos y valores numéricos los resultados obtenidos, para posteriormente analizarlos con métodos estadísticos

La población en la presente investigación la conforma la cantidad de transferencia de imágenes que se realizarán en una hora, según varias empresas que implementan sistemas de videovigilancia como (Axis Communications, 2018) o (SecuraMe, 2018), establecen que la resolución del video dependerá de la cantidad de fotogramas por segundo (FPS) que se transfieran, en la cual los valores van desde 3 FPS hasta los 25 FPS, *para el presente estudio y luego de pruebas realizadas se ha establecido el valor de 4 FPS con lo cual se garantiza una correcta visualización del video con relación de aspecto(ratio) 16:9, resolución de 360 pixeles y reducción de flicker(parpadeo) de 50Hz, obteniéndose una población de 14.400 imágenes a ser transferidas.*

- **Selección de la muestra**

Para encontrar la muestra se aplicará la siguiente fórmula de muestreo (Aguilar-Barojas, 2005):

$$n = \frac{NZ^2pq}{d^2(N - 1) + Z^2pq}$$

Dónde:

N=Total de la población

$Z_{\alpha} = 1.96$ al cuadrado (Si la seguridad es del 95%)

$p =$ Proporción esperada (50% la más alta)

$q = 1 - p$ (para este caso 50%)

$d =$ Precisión 5%

$$n = \frac{14400 * 1.96^2 * 0.50 * 0.5}{0.05^2(14400 - 1) + 1.96^2 * 0.50 * 0.5}$$

n = 375

Luego de los cálculos se establece como muestra para el estudio un total de 375 transferencias de imágenes a realizar en cada escenario.

- **Análisis de Indicadores Individual**

Para el proceso de análisis de la información recolectada *se utilizó el método estadístico de la Prueba Z*, con un total de 375 imágenes (población desconocida) generando los

resultados mediante el software SIAE 2.0, el mismo que permitió verificar si existe o no una diferencia significativa en cada uno de los indicadores.

• **Análisis Multicriterio**

Según (Villacís Cruz, 2005), existen algunos modelos para la elección de un procedimiento de agregación o análisis de los criterios. Entre los básicos se encuentra el método NAIADE (*Novel Approach to Imprecise Assessment and Decision Environments*), el mismo que combina los elementos del modelo de superación, pero agregando funciones continuas y monótonas de credibilidad de la indiferencia y preferencia, para mejorar la calidad de selecciones y rankeos, evitando que pequeñas variaciones de preferencia cambien el orden original de selección. Esto significa que el modelo permite evaluar alternativas de acuerdo a criterios, tomando en cuenta preferencias de distinta intensidad, para evitar que se seleccionen alternativas por diferencias pequeñas en su calificación. El NAIADE, desarrollado por Munda (Munda, 2002) permite también la introducción de datos cualitativos, cuantitativos difusos, determinísticos y estocásticos, encajando así de forma idónea para el presente proyecto ya que se cuenta con varios criterios que inciden en el Rendimiento Total. El software genera un ranking de alternativas las cuales son finitas (método discreto) y no permite dar peso a los criterios, es decir otorgando la igual 'importancia' a los mismos.

• **Evaluación del rendimiento**

La evaluación de rendimiento de sistemas informáticos propuesta por (Medina, 2014) en su artículo “Pruebas de rendimiento”, establece dos tipos de pruebas: internas y externas, las internas que corresponden a tiempos de respuesta y procesamiento, mientras que las externas que guardan relación con el porcentaje de uso de CPU, RAM, y ancho de banda. Esta información se puede observar en la tabla 1.

Tabla 1. Indicadores del rendimiento

| Métrica | Indicador | Unidad de Medida |
|----------|--------------------------|------------------|
| Internas | Tiempo de Respuesta | Milisegundos |
| | Tiempo de Procesamiento | Milisegundos |
| Externas | % Uso de CPU | Porcentaje |
| | % Uso de RAM | Porcentaje |
| | Ancho de Banda Utilizado | KiloBytes |

En varias investigaciones como por ejemplo (Alonso Forcelledo & Cañete González, 2006) y la de (Murillo Morera & Caamaño Plini, 2010) se analizan los indicadores del rendimiento de sistemas como, el uso de CPU y RAM pertenecientes a las métricas externas, sin especificar su pesos de ponderación, así como en la investigación de (Ghodsí, y otros) donde se hace un análisis por separado de tres indicadores, Uso de CPU (M. Externa), Uso de RAM (M. Externa) y Tiempo de Respuesta (M. Interna), y de igual manera no se define ponderaciones. Para la presente investigación de la misma manera no se establece que uno u otro tenga más peso dado el escenario donde se lleva a cabo el experimento, ya que por un lado las métricas internas como las externas son de elevada importancia, la primera por los tiempos que se utiliza para llevar a cabo las tareas y las métricas externas por la cantidad de recursos limitados con los que cuenta una computadora de placa reducida (Raspberry). Para la consolidación de la información se utilizará entonces un método discreto multicriterio con una matriz de impacto (o

evaluación) que puede incluir medidas nítidas, estocásticas o difusas del rendimiento de una alternativa con respecto a un criterio de juicio.

- **Escenario**

El escenario propuesto refleja el entorno en producción donde se ejecutará el sistema de videovigilancia, que es en la Dirección Distrital 07D04 Balsas – Marcabeli – Piñas del Ministerio de Educación, donde se cuenta con una red Wireless (IEEE 802.11b); por lo que la incorporación de uno o varios nuevos dispositivos “inteligentes” como la placa de Raspberry (*haciendo uso de virtudes como tamaño, versatilidad, robustez y conexión inalámbrica*), se une a la red existente sin inconvenientes, evitando así la incorporación de conexiones por medio de cables, lo cual a más de alterar la estética existente del lugar, la incorporación de conexiones y cableado nuevo, canaletas, patch cord, patch panels, ordenadores, etc., garantizándose la transmisión normal por medio inalámbrico, debido a que se cuenta con oficinas separadas por modulares, paredes y techos falsos y ancho de banda de 10 MB por fibra óptica de plan corporativo con compartición 1:1.

Para realizar las pruebas comparativas del rendimiento, se ha diseñado un escenario en donde van a interactuar dos procesos, un Servidor (Raspberry Pi) y un cliente (PC), los mismos que van a estar conectados mediante una red Wireless segmentada, con una velocidad de 4MB, el cual se puede observar gráficamente en la figura 1.

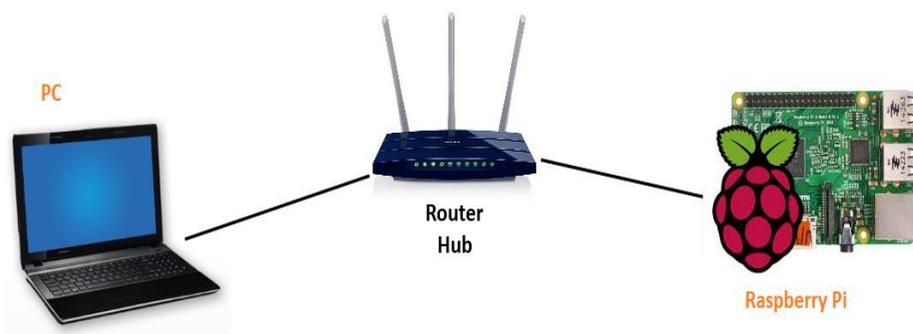


Figura 1. Escenario de pruebas

- **Prototipo**

Los prototipos desarrollados tanto en Java y Python que serán instalados en el Servidor Raspberry Pi contienen las mismas funcionalidades:

- Capturar imágenes mediante una cámara integrada al Raspberry
- Enviar los fotogramas capturados mediante la utilización de sockets a través de la red de datos Wireless.

La apariencia del hardware del sistema será como el de la figura 2.



Figura 2. Apariencia del sistema de videovigilancia

En la tabla 2 se detalla el hardware utilizado en el experimento.

Tabla 2. Hardware del sistema

| Características | Cliente | Servidor (Raspberry Pi) |
|-----------------|----------------------------------|--|
| Marca | Dell | Raspberry Pi+ |
| Procesador | Intel i7 | Broadcom BCM2835, Single-Core a 700MHz |
| RAM | 16GB | 512MB |
| Disco Duro | 1TB | Micro SD 4GB |
| Internet | Intel Dual Band Wireless AC 3160 | 150Mbps Wireless N Nano USB Adapter |

En la tabla 3 se observa el software utilizado para las pruebas.

Tabla 3. Software del sistema

| Características | Cliente | Servidor (Raspberry Pi) | |
|-----------------------|-----------|-------------------------|------------|
| | | Java | Python |
| Sistema Operativo | Win10 | Raspbian Kernel 4.14 | |
| Entorno de Desarrollo | Netbeans | Netbeans | BeeWare |
| Plataforma | Java Se 8 | Oracle JDK 1.7 | Python 3.0 |

Análisis de Resultados

A continuación, se realiza el análisis correspondiente al indicador “Métricas Internas”, el mismo que se calcula a partir de los índices Tiempo de Respuesta y Tiempo de Procesamiento que se realizó para el envío de 375 imágenes tanto en Java como en Python.

En la figura 3, se visualiza de manera gráfica la media del tiempo de respuesta en milisegundos de cada proceso.

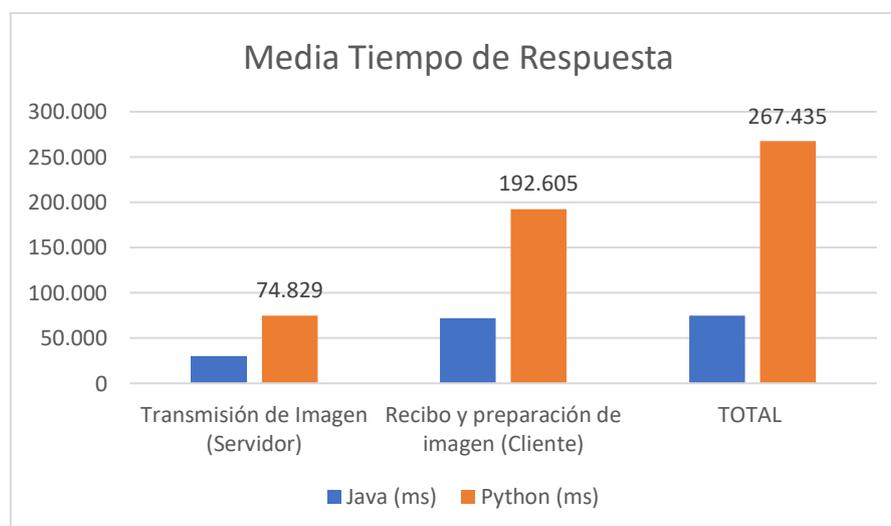


Figura 3. Tiempo de respuesta

En la figura 4, se visualiza de manera gráfica la media del tiempo de procesamiento en milisegundos de cada proceso.

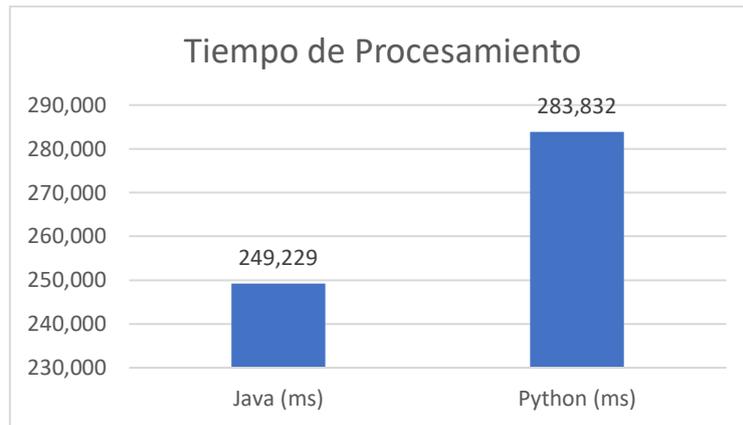


Figura 4. Tiempo de Procesamiento

En la figura 5, se visualiza de manera gráfica la media del porcentaje de uso de CPU de cada proceso.

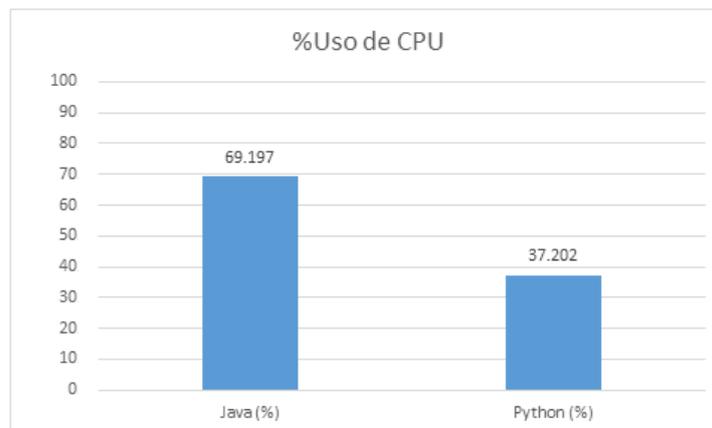


Figura 5. Porcentaje de uso de CPU

En la figura 6, se visualiza de manera gráfica la media del porcentaje de uso de RAM de cada proceso.

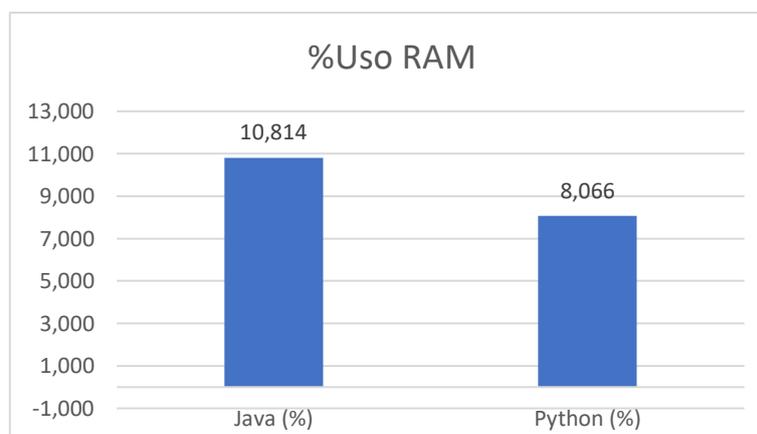


Figura 6. Porcentaje de uso de RAM

En la figura 7, se visualiza de manera gráfica la media del Ancho de Banda de cada proceso.

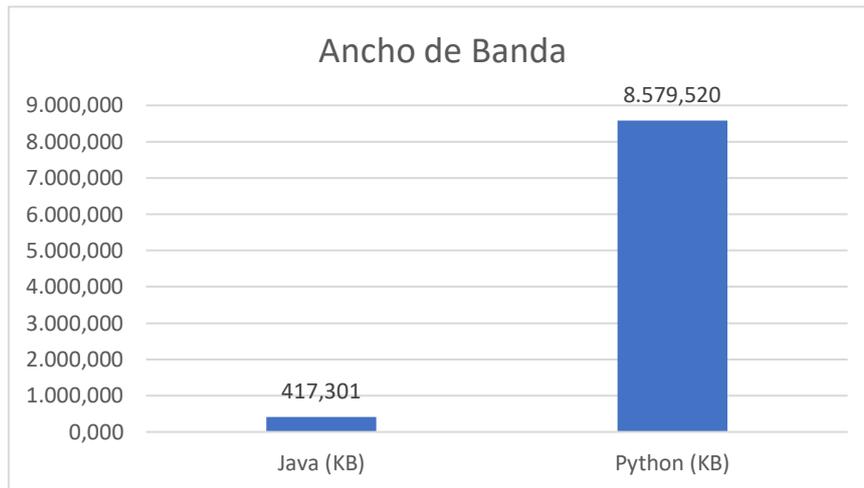


Figura 7. Uso de ancho de banda

Para comprobar si existe o no diferencia entre los dos lenguajes de programación a la hora de ejecutar el trabajo de transferencia de imágenes entre una placa computadora y un servidor, se establecieron dos hipótesis, la hipótesis alternativa y nula.

- **Ha=** Existe una diferencia significativa del rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas entre los lenguajes de programación Java y Python.
- **Ho=** NO Existe una diferencia significativa del rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas entre los lenguajes de programación Java y Python.

Utilizando el método NAIADe de Análisis Multicriterio, se establece la siguiente Matriz de Impacto que se observa en la tabla 4.

Tabla 4. Matriz de impacto

| Dimensiones y Variables | Unidad de Medida | Tipo | Objetivo | Java | Python |
|-------------------------|------------------|--------------|-----------|---------|---------|
| Tiempo de Respuesta | Milisegundos | Cuantitativa | Minimizar | 74,565 | 267,435 |
| Tiempo de Procesamiento | Milisegundos | Cuantitativa | Minimizar | 249,229 | 283,832 |
| % Uso de CPU | Porcentaje | Cuantitativa | Minimizar | 69,197 | 37,202 |
| % Uso de RAM | Porcentaje | Cuantitativa | Minimizar | 10,814 | 8,066 |
| Ancho de Banda | Kilobytes | Cuantitativa | Minimizar | 417,301 | 8579,52 |

Fuente: Trabajo de campo

Elaborado por: Los autores

La matriz de impacto se ingresó en el software NAIADe como se observa en la figura 8.

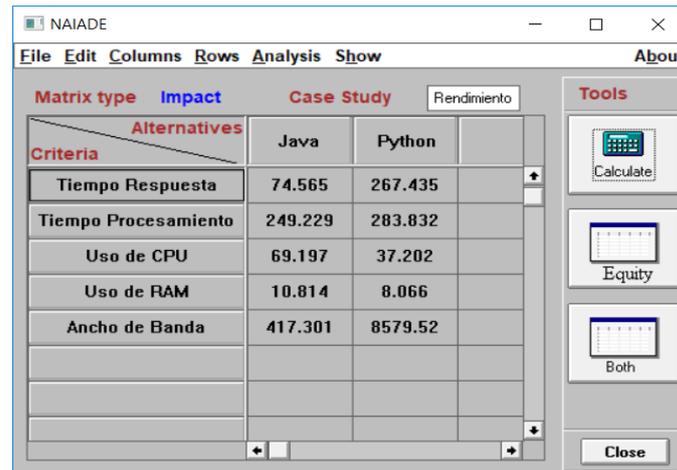


Figura 8. Matriz de impacto en el Software NAIADE

Una vez ejecutado el ejercicio, los resultados que arroja NAIADE con un alfa (α) de 0,5; se obtiene un ranking Φ^+ de 0.7 selecciona al lenguaje Java como mejor que Python. En la figura 9 se puede observar el resultado de la matriz de impacto.

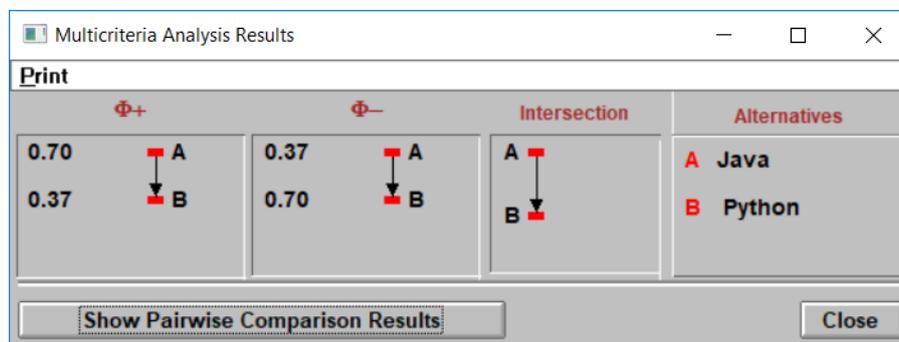


Figura 9. Resultado de la Matriz de Impacto

Se puede observar que con un grado de confianza de 0.6744 (Ver figura 10) Java es “Mejor que Python en cuanto al rendimiento en la transferencia de imágenes”

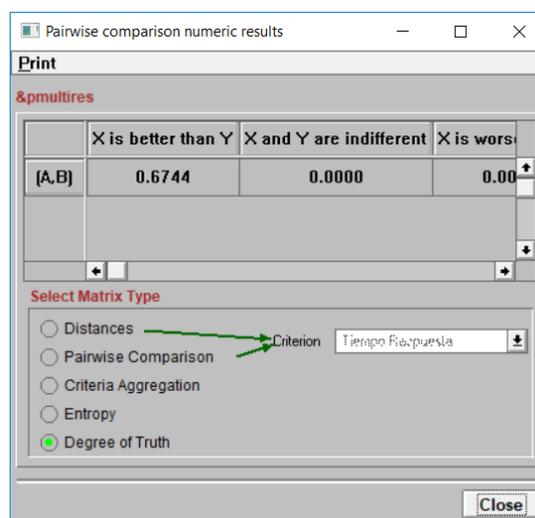


Figura 10. Grados de verdad

El resultado concuerda con la hipótesis alternativa (Ha) del estudio “Existe una diferencia significativa del rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas entre los lenguajes de programación Java y Python.”, en referencia a que, de acuerdo con los indicadores introducidos y bajo los parámetros del modelo, la evaluación dice que en términos de rendimiento en la transferencia de imágenes en el sistema de videovigilancia el lenguaje Java tuvo mejor desempeño que el lenguaje Python.

CONCLUSIONES

- Los lenguajes de programación Java y Python por varios años se han encontrado en el listado de los más utilizados a nivel mundial (BV, 2016), gracias a sus características como uso libre, seguridad, robustez, escalabilidad, multiplataforma, lo cual los hace atractivos para los desarrolladores de sistemas informáticos, no solo en el campo aplicaciones web, escritorio o móvil sino también como en el presente trabajo en el ámbito de la videovigilancia por medio del uso de dispositivos inteligentes basados en el Internet de las Cosas, a su vez se encontró una diversa documentación y recursos como las librerías *sarxos*, *webcam*, *Webcam* (En Java) y *Pygame* (En Python) que permitieron llevar a cabo el desarrollo de los prototipos.
- Luego del análisis correspondiente se obtuvo que, para las métricas internas tiempo de respuesta se obtuvo una diferencia porcentual de 258.6% a favor de Java, luego en cuanto a la tiempo de procesamiento 13.88% a favor de Java, así mismo, en cuanto a las métricas externas, el uso de CPU un 86% a favor de Python, el uso de RAM, un 34% a favor de Python y para el ancho de banda un Java ocupa 20.56 veces menos KB que Python, finalmente aplicando el análisis multicriterio con el método de NAIADSE se obtiene que con un ranking positivo de 0.7 “Java es Mejor que Python”.
- Es importante recalcar que hubo diferentes resultados en la evaluación de cada indicador, por lo cual será indispensable definir adecuadamente las prioridades que permitan determinar la mejor opción a la hora de decidir por uno u otro lenguaje para la implementación

REFERENCIAS BIBLIOGRÁFICAS

- Aguilar-Barojas, S. (2005). Fórmulas para el cálculo de la muestra en investigaciones de salud. *redalyc.org*.
- Alonso Forcelledo, J., & Cañete González, C. (2006). Influencia en el rendimiento de un sistema de la velocidad de CPU y cantidad de memoria. *uniovi*.
- Axis Communications. (2018). *Axis.com*. Obtenido de <https://www.axis.com/customer-story/76>
- BV, T. s. (2016). *Índice de TIOBE para noviembre de 2016*. Obtenido de <http://www.tiobe.com/tiobe-index/>
- Clouding.io. (13 de Marzo de 2017). *Latencia y velocidad de conexión*. Obtenido de <https://clouding.io/blog/latencia-y-velocidad-de-conexion/>
- Cunningham, Subrahmanian, & Westerberg. (2010).

- Echeverría, D. (2016). Tiempo de Respuestas y Experiencia de Usuario. *Genexus Consulting*, PP. 231-234.
- Evans, D. (2011). *Internet de las cosas: Cómo la próxima evolución de Internet lo cambia todo*. Cisco Internet Business Solutions Group (IBSG).
- Ferdoush, S., & Li, X. (2014). Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications. *ElSeiver*, pp. 103-110.
- Fúquene. (2011).
- Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., & Stoica, I. (s.f.). Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. *Usenix*.
- Jipsion. (2017). *Seguridad en redes y seguridad en la información*. Obtenido de http://improvet.cvut.cz/project/download/C2ES/Seguridad_de_Red_e_Informacion.pdf
- Kannan, R., Mani, A., Ravindran, R., & Subbian, K. (2015). *Armonk, NY Patente n° 20150220336*.
- Kryvyy, R., Serhiy, T., & Ruslan, G. (2015). Features home automation system development based Raspberry Pi using Java ME SDK. *IEEE Xplore*.
- Leira, R., Vega, C., González, I., & Aracil, J. (2014). HPSEngine: Motor de alto rendimiento y baja latencia para el procesamiento distribuido en tiempo real. *research.cvega.es*.
- Li, Ferdoush. (2014).
- Llanos, R., Naula, M., & Valdiviezo, C. (2014). Utilización de la minicomputadora raspberry pi con capacidad de comunicación wi-fi para la captura de imágenes mediante cámara y almacenamiento de información en base de datos externa. *Dspace ESPOL*.
- Maksimović, Vujović, Davidović, Milošević, & Perišić. (2014).
- Medina, J. (2014). *Pruebas de rendimiento TIC*. Murcia: Kindle.
- Molero, X.; Ruiz, C.; Rodeño, M. (2004). *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*. España: Pearson Prentice Hall.
- Murillo Morera, J., & Caamaño Plini, S. (2010). Implementación de un servidor FTP utilizando el modelo cliente servidor mediante el uso de sockets en lenguaje C Unix con el fin de mejorar los tiempos de respuesta en la red. *Uniciencia*, pp. 83-89.
- Naula, Llanos, & Valdiviezo. (2014).
- Prechelt. (2000).
- Raspbian. (s.f.).
- Salazar, J., & Silvestre, S. (2016). *Internet de las cosas*. Czech Republic: České vysoké učení technické v Praze.

- SecuraMe. (2018). *http://www.securamente.com.* Obtenido de *http://www.securamente.com/que-son-los-fps-en-cctv/*
- TIOBE. (jul de 2019). *TIOBE Index*. Obtenido de *https://www.tiobe.com/tiobe-index/*
- Valeriu, M., Florin, S., & Adrian-Viorel, D. (2014). Control system for video advertising based on Raspberry Pi. *IEEE Xplore*.
- Vidal-Silva, C., Madariaga, E., & Solís, R. (2017). Estudio Piloto de la Importancia del Rendimiento, Seguridad y Fiabilidad en el Proceso de Desarrollo de Software en Chile. *Información Tecnológica*, pp. 95-106.
- Villacís Cruz, B. (2005). *La crisis del oro azul: Un análisis de la sustentabilidad del agua en la ciudad de Quito*. Quito: Flacso.
- Wai, C., Jegatheesan, J., & Chee, S. (2015). Exploring IOT Application Using Raspberry Pi. *International Journal of Computer Networks and Applications*, pp. 27-34.

PARA CITAR EL ARTÍCULO INDEXADO.

Paguay Soxo, P., Paucar Córdova, R., & Flores Orozco, A. (2019). Evaluación del rendimiento de la transferencia de imágenes en aplicaciones IoT entre los lenguajes de programación Java y Python. *Ciencia Digital*, 3(3.4.), 327-341.
<https://doi.org/10.33262/cienciadigital.v3i3.4.878>



El artículo que se publica es de exclusiva responsabilidad de los autores y no necesariamente reflejan el pensamiento de la **Revista Ciencia Digital**.

El artículo queda en propiedad de la revista y, por tanto, su publicación parcial y/o total en otro medio tiene que ser autorizado por el director de la **Revista Ciencia Digital**.

