

ANÁLISIS DE RENDIMIENTO DE LA PC ODROID C2 PARA SU USO EN ESQUEMAS DE CIUDADES INTELIGENTES

Ernesto Bardales Hernández

Universidad Autónoma del Estado de México
ebardalesh.7@gmail.com

Saúl Lazcano Salas

Universidad Autónoma del Estado de México
slazcanos@uaemex.mx

Resumen

Dentro de las llamadas ciudades inteligentes, las redes de sensores para el monitoreo de diversas métricas como calidad del aire, densidad vehicular, entre otros, cobran particular importancia por la información que proporcionan. Estas redes envían sus datos a un punto central de procesamiento para su análisis e interpretación; sin embargo, en muchas ocasiones el envío de estos datos se realiza a través de canales de comunicaciones poco seguros y con baja relación de señal a ruido (*signal-to-noise-ratio*, *SNR*), tales como canales inalámbricos. En el presente trabajo, se propone el uso de la PC Odroid C2 como herramienta para implementar un codificador de convolución como esquema de codificación de canal que permita que los datos transmitidos sean menos susceptibles al ruido, así como para otro tipo de procesos relacionados con el área. Se estudia de la misma forma una tecnología similar: la PC Raspberry Pi versión 3, anexando pruebas de benchmarking de ambas PC.

Palabras Clave: Arduino UNO, ciudades inteligentes, codificación de canal, código de convolución, Odroid C2, Raspberry Pi, redes de sensores.

Abstract

Within the so-called intelligent cities, networks of sensors for the monitoring of various metrics such as air quality, vehicular density, and others, are

particularly important because of the information they provide. These networks send their data to a central processing point for analysis and interpretation; However, this data is often sent through unsafe and low signal-to-noise-ratio (SNR) communications channels, such as wireless channels. In the present work, it is proposed to use the PC Odroid C2 as a tool to implement a convolution encoder as a channel coding scheme that allows the transmitted data to be less susceptible to noise, as well as to other types of processes related to the area. A similar technology is studied in the same way: PC Raspberry Pi version 3, appending benchmarking tests to both PC.

Keywords: *Arduino UNO, Channel encoding, Convolution codes, Odroid C2, Raspberry Pi, Sensor Networks, Smart Cities.*

1. Introducción

El crecimiento de las grandes urbes conlleva una problemática compleja en diversos ámbitos como salud, transporte, trabajo y servicios. El garantizar calidad de vida y sustentabilidad de manera simultánea implica retos cuya solución no es única. En este contexto, una alternativa que ha cobrado particular fuerza en la última década es el llamado paradigma de las ciudades inteligentes, que en esencia, se puede describir como un gran sistema compuesto por pequeños subsistemas, cada uno de los cuales es responsable de una tarea en particular [Movuna, 2016]. El objetivo en una ciudad inteligente es lograr una mejora en la calidad de vida de las personas tomando como punto de partida el uso inteligente, inclusivo y de manera sustentable de las tecnologías de la información [Arroub, 2016].

Dentro de la infraestructura para las ciudades inteligentes, destaca el uso de las tecnologías de la información como apoyo a tareas de monitoreo automatizado en cámaras, sistemas GPS, sensores de camino (detección de baches y objetos) entre otros. Este monitoreo se hace por lo general con apoyo de redes de sensores, que recaban la información deseada y la transmiten a un punto de control para su análisis y toma de decisiones correspondiente [Arroub, 2016], [Benamrou, 2016].

La transmisión de datos puede llevarse a cabo aprovechando la infraestructura existente, como la red eléctrica o empleando canales inalámbricos. Sin embargo, tanto la red eléctrica como los canales inalámbricos presentan el inconveniente de ser canales con una figura de ruido muy agresiva [Tseng, 2014], [Lui, 2014] y en consecuencia, la calidad de los datos en el extremo receptor se puede ver seriamente comprometida en caso de no tener un esquema de codificación de canal adecuado.

En este sentido, desde la publicación del trabajo de Claude E. Shannon en 1948 [Shannon, 1948] hasta la actualidad, se ha realizado un gran esfuerzo en buscar esquemas de codificación de canal que permitan acercarse al límite teórico establecido en dicho trabajo. Uno de los esquemas de codificación de amplia implementación en diversos sistemas son los códigos de convolución [Morelos, 2006], predecesores de los turbo códigos los cuales junto a los códigos LDPC (de sus siglas en inglés Low Density Parity Check), presentan un desempeño muy cercano al límite de Shannon.

Existen algunas herramientas tecnológicas de bajo costo, capaces de realizar tareas requeridas por los esquemas de ciudades inteligentes. Un ejemplo de ellas son las PC de placa reducida con arquitectura ARM.

Las PC analizadas en este trabajo son: la PC Raspberry Pi versión 3, de la cual se han realizado diversos trabajos; por citar algunos se tiene que [González, 2015] menciona el uso de la PC Raspberry Pi versión 2 como servidor de datos, el cual mediante el uso del microcontrolador Arduino conectado en el puerto serial USB de la PC, se encarga de hacer lecturas y recolectar información de sensores y actuadores, almacenándolos en una base de datos para futuros análisis.

Otro trabajo al respecto, es el presentado por [Liscano, 2014] el cual propone mejoras al tránsito vehicular mediante una red de dispositivos electrónicos, diseñando variantes en infraestructura de TICS (Tecnológicas de la Información y Comunicación) enfocadas a entornos de ciudades inteligentes, tomando como base de procesamiento el uso de la PC Raspberry Pi versión 2.

Por otra parte, [Solarte, 2017] generan una propuesta basada en una plataforma que recibe información de redes de sensores diversas. El trabajo de dicha

plataforma es homogeneizar la información recibida, procesarla y ponerla a disposición de los usuarios a través de internet, permitiendo de este modo contar con información en tiempo real vinculada con el entorno, como ejemplo la densidad vehicular o el clima.

Por otro lado, se analiza la PC Odroid C2, la cual es otra tecnología con capacidades similares, incluso superiores a la PC Raspberry Pi; ejemplo de estas son: memoria RAM, CPU, compresión de datos, transferencia por el adaptador de red Ethernet, encriptado mediante SSL (por sus siglas en inglés Secure Sockets Layer), y acceso a almacenamiento de disco.

Actualmente la PC Odroid C2 no ha sido utilizada en esquemas de ciudades inteligentes como lo ha sido la Raspberry Pi, sin embargo al contar con características técnicas superiores, puede ser utilizada bajo este tipo de esquemas.

En el presente trabajo, se analiza la factibilidad de emplear la PC Odroid C2 como plataforma base para la implementación de un codificador de canal de convolución el cual se aplica a los datos provenientes de una red de sensores. El objetivo con lo anterior es brindar a dichos datos robustez para ser transmitidos en canales con SNR bajas, como es el caso de un canal inalámbrico. De igual forma, se compara este mismo esquema de codificación de canal con la PC Raspberry Pi versión 3 obteniendo el desempeño de ambas plataformas. Adicionalmente se realizará a ambas PC pruebas de benchmarking en aspectos relacionados a las ciudades inteligentes, para tener de este modo un punto comparativo base.

2. Métodos

Para realizar la codificación de canal, se selecciona un codificador de convolución como se define en [Morelos, 2006], con una tasa de codificación de $R = \frac{1}{2}$ con polinomios generadores definidos por ecuaciones 1 y 2.

$$P_1(x) = x^2 + x + 1 \quad (1)$$

$$P_2(x) = x^2 + 1 \quad (2)$$

El esquema general del codificador se muestra en la figura 1. Se selecciona este codificador de convolución como referencia ya que a partir del mismo se pueden

implementar esquemas más robustos de codificación de canal tales como los turbo códigos [Berrou, 1993].

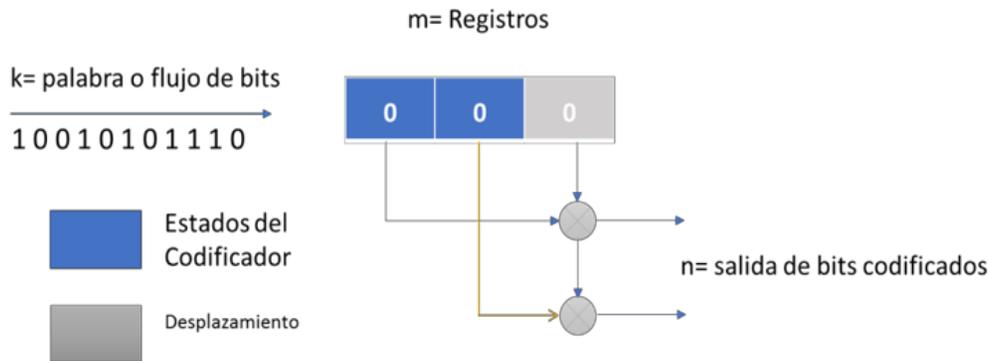


Figura 1 Estructura del codificador convolucional.

El esquema de configuración para pruebas a emplear se muestra en la figura 2.

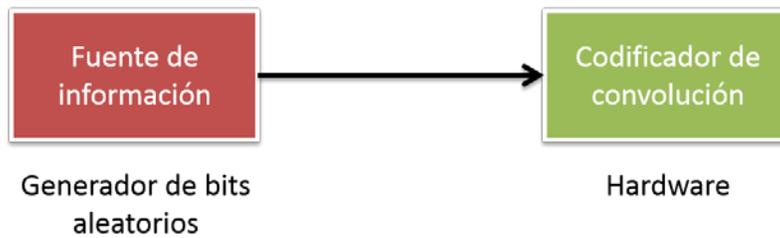


Figura 2 Esquema de transmisión con fuente de información al hardware propuesto.

Como fuente de información se programa un simulador de generador de bits aleatorios a diferentes tasas de transmisión, basado en la plataforma Arduino UNO cuyo algoritmo base se muestra a continuación.

- Inicio
- Iniciando la configuración de transmisión del puerto serial a una tasa de bits determinada.
- Iniciando iteraciones sin fin.
- Imprimiendo en el puerto serial un valor entero aleatorio entre 0 y 1.

Este simulador de generador de bits aleatorios se enlaza al hardware a través del puerto serial de este. La tasa de bits usada se muestra en la tabla 1, es necesario

resaltar que cada valor de bits por segundo correspondiente a la programación de la fuente de información es modificada a diferentes tasas de bits para cada iteración de pruebas.

Tabla 1 Tasas de bits por segundo usadas para el puerto serial del hardware.

Tasa de bits por segundo
1200 bps
2400 bps
4800 bps
9600 bps
19200 bps
38400 bps

Para analizar los niveles de carga de procesamiento del hardware, se somete a pruebas de 4 horas de trabajo continuo con monitoreo cada dos segundos, conforme a las velocidades de transmisión del puerto serial de la tabla 1. El algoritmo para la lectura del bit a ser codificado, se muestra a continuación:

- Inicio
- Iniciar variable identificador del puerto serial y tasa de transmisión
- Iniciando iteraciones sin fin
- Leyendo valor entero del puerto serial mediante variable identificador
- Enviando valor leído al algoritmo del codificador

El diagrama de flujo correspondiente al algoritmo del codificador convolucional de la figura 1 se presenta en la figura 3.

Para analizar la factibilidad de uso de la PC Odroid C2 en un entorno de trabajo real, se analizan tres variables importantes de comportamiento: temperatura, tasa de flujo de bits y porcentaje de carga de CPU. Esta metodología será implementada de la misma forma con la placa Raspberry pi versión 3.

Adicionalmente se usará el software tipo benchmark [Phoronix, 2017] para pruebas a ambas tecnologías, para resaltar las capacidades de la PC Odroid C2 en operaciones a nivel hardware. Los factores sometidos a las pruebas son:

- ✓ Memoria caché
- ✓ Velocidad de RAM
- ✓ Carga de CPU
- ✓ Compresión de datos
- ✓ Algoritmo de encriptado SSL para transporte de datos
- ✓ Velocidad del adaptador de red Ethernet
- ✓ Velocidad de acceso al disco duro

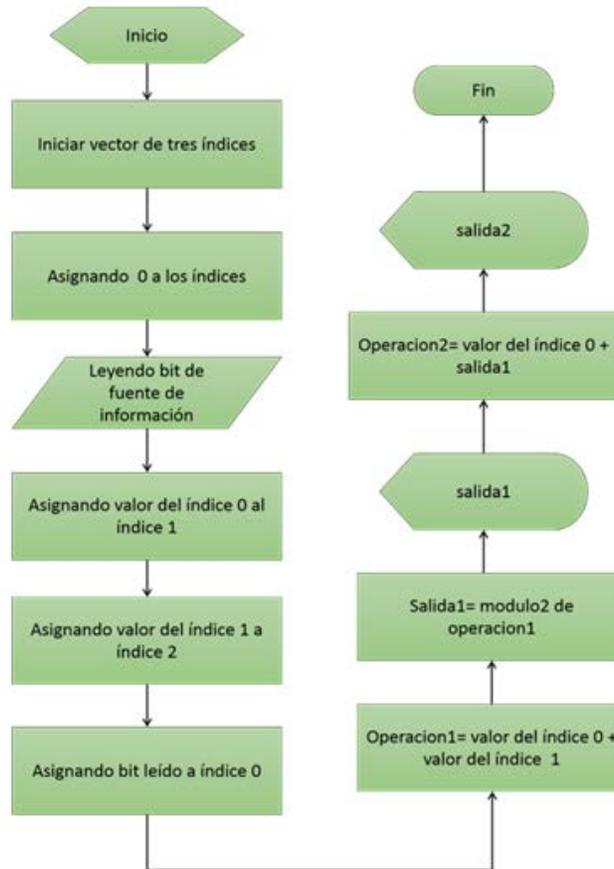


Figura 3 Diagrama de flujo de datos del algoritmo de figura 1.

3. Resultados

- De acuerdo con las pruebas realizadas en ambas tecnologías (con las tasas mostradas en la tabla 1) se obtuvieron los resultados que se muestran en la tabla 2 y tabla 3. Cabe mencionar que la información mostrada hace referencia al promedio de los valores generales.

Tabla 2 Carga de CPU promedio del hardware estudiado.

Tasa de bits	%CPU - Raspberry Pi	%CPU - Odroid C2
1200	1.66	1.41
2400	2.96	2.39
4800	5.65	4.55
9600	9.57	8.33
19200	17.11	14.19
38400	17.3	14.3

Tabla 3 Temperatura de la zona térmica promedio del hardware estudiado.

Tasa de bits	C° Raspberry Pi	C° Odroid C2
1200	37	27.86
2400	45.17	41.13
4800	45.2	42.7
9600	46.46	42.13
19200	46.71	43.63
38400	47	43.2

Ambas tecnologías presentan un comportamiento similar, aunque a tasas de bits altas la PC Odroid C2 comienza a despuntar su desempeño contra la PC Raspberry Pi. En la figura 4 se representa la curva de comportamiento de carga de la CPU con respecto a la tasa de bits determinada a ambas PC.



Figura 4 Representación de la carga de CPU con respecto a la tasa suministrada PC.

En la figura 5 se muestra la curva de comportamiento de temperatura de la zona térmica de ambas PC con respecto a la tasa de bits determinada.

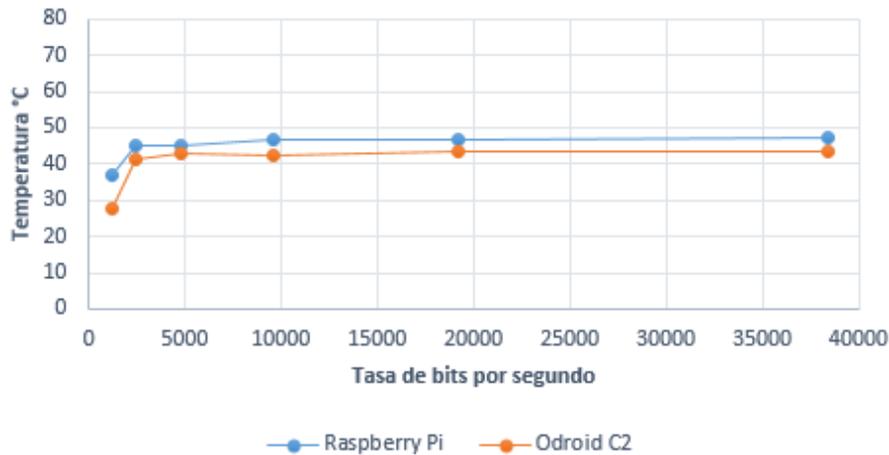


Figura 5 Representación de temperatura con respecto a la tasa suministrada.

- Dentro de las pruebas de benchmarking realizadas a ambas PC, se obtuvieron los siguientes resultados. Cabe mencionar que las pruebas están sostenidas a los tiempos predeterminados asignados por el software:
 - ✓ **Tiempo de compresión** de información con la herramienta Gzip figura 6. La PC Odroid C2 destaca con un mejor desempeño por su rapidez de compresión, ya que PC Raspberry Pi se tomó el doble de tiempo para realizar dicha tarea.

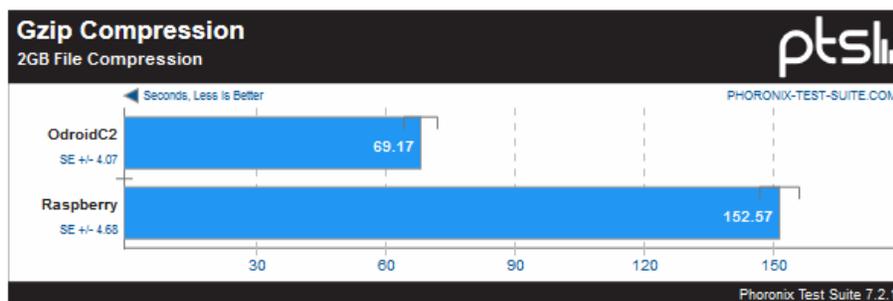


Figura 6 Tiempos de compresión con Raspberry Pi y Odroid C2.

- ✓ **Compresión** de información con la herramienta 7-zip figura 7. En esta prueba se analiza la cantidad de instrucciones por segundo que cada PC puede realizar para una compresión de datos efectiva. La PC Odroid C2 destaca nuevamente al realizar poco más del doble de MIPS (millones de operaciones por segundo).

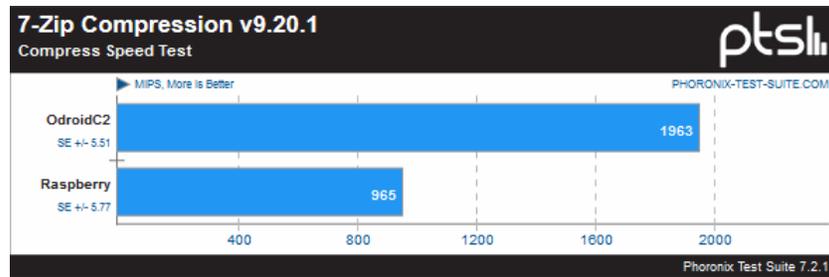


Figura 7 Velocidad de compresión por MIPS.

- ✓ **Velocidad de RAM** promedio de operaciones con valores enteros y punto flotante figura 8 y figura 9. Se analiza la cantidad de MB/s que las operaciones pueden manejar. La PC Odroid C2 muestra ventaja considerable en velocidad de RAM, siendo casi el doble su desempeño.

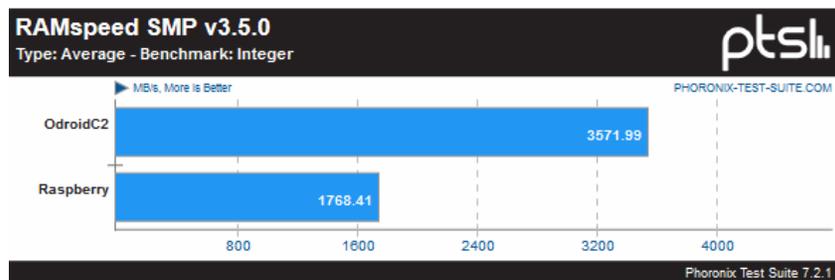


Figura 8 Velocidad de RAM promedio con valores enteros.

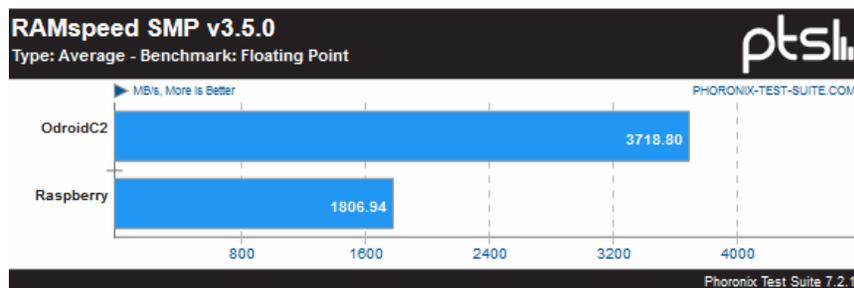


Figura 9 Velocidad de RAM promedio con valores de punto flotante.

- ✓ **Velocidad de transmisión del adaptador de red Ethernet** figura 10. La PC Odroid C2 tuvo un tiempo de transferencia de información sobresaliente, al transmitir en menor tiempo con respecto a PC Raspberry Pi.

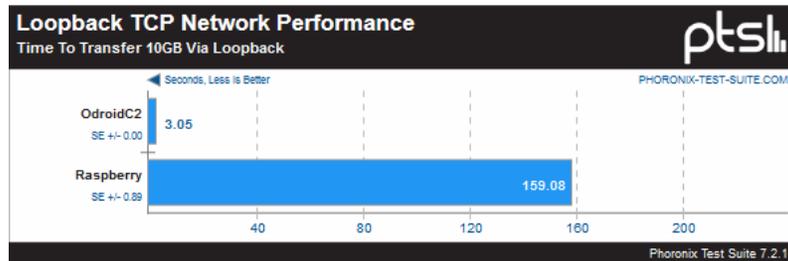


Figura 10 Velocidad de adaptador de red.

- ✓ **Velocidad de escritura del medio de almacenamiento** figura 11 en diferentes sectores. La PC Odroid C2 presentó de manera considerable una mejor tasa de escritura de información a diferencia de la PC Raspberry Pi.

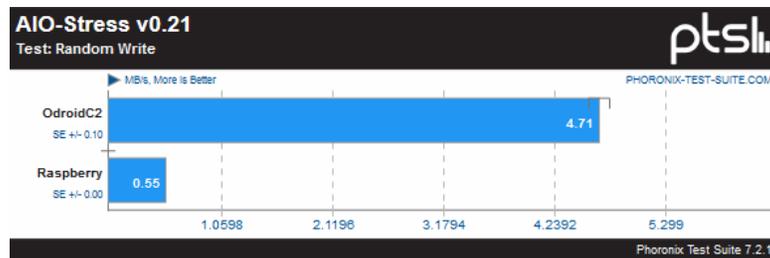


Figura 11 Velocidad de escritura en el medio de almacenamiento.

- ✓ **Velocidad de memoria caché medido en MB/s para diversas operaciones** figura 12. La memoria caché de la PC Odroid C2 obtuvo la ventaja al operar a más del doble de velocidad, que la tecnología competidora.

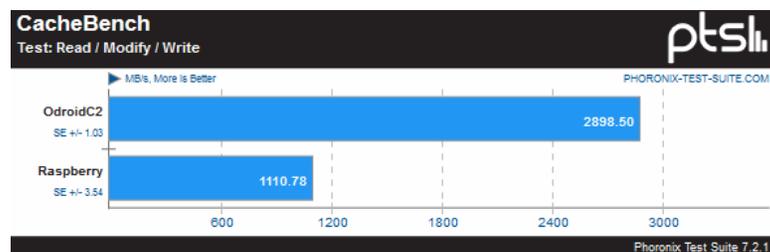


Figura 12 Velocidad de memoria caché.

- ✓ **Velocidad de codificación de video MPEG** figura 13. La PC Odroid C2 obtuvo una mejor marca de tiempo en codificación de video, frente

a la PC Raspberry Pi, la cual se tomó poco más del doble de tiempo que la PC Odroid C2 en operación de codificación.

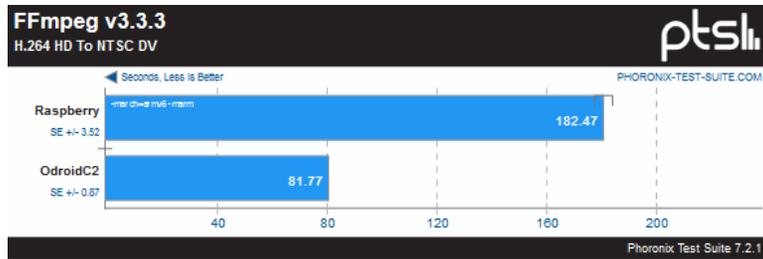


Figura 13 Tiempos de codificación de video.

- ✓ **Rendimiento de CPU** para cálculos de punto flotante figura 14. La PC Odroid C2 muestra ventaja al operar en tiempo menor que la tecnología competidora, ya que la PC Raspberry Pi se tomó más del doble de tiempo para realizar sus respectivas operaciones.

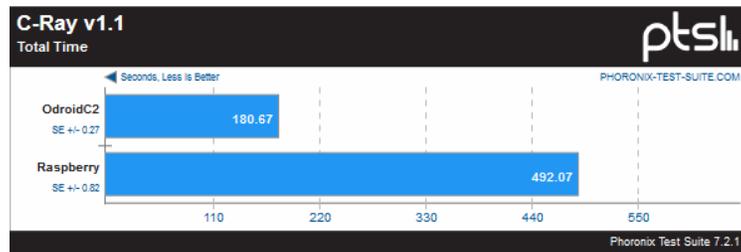


Figura 14 Tiempo de cálculo de punto flotante en CPU.

- ✓ **Rendimiento de encriptado** mediante el protocolo SSL con firma digital RSA-4096 medido en símbolos/segundo, figura 15. La PC Odroid C2 mantiene ventaja en capacidad de encriptado, por lo que la PC Raspberry Pi se queda por debajo de la mitad.

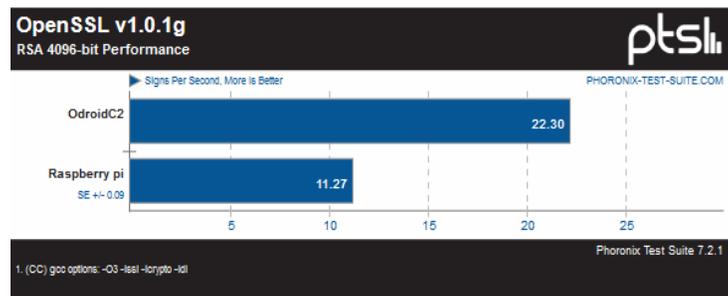


Figura 15 Rendimiento de encriptado mediante SSL.

4. Discusión

A partir de los resultados mostrados en la sección anterior, se puede observar que para tasas de datos de hasta 38400 bps, ambas plataformas de hardware presentan un comportamiento muy estable, y sobre todo confiable. Sin embargo, se debe resaltar que la PC Odroid C2 consume menos recursos de CPU en relación con la PC Raspberry Pi versión 3.

Tema de particular interés es la temperatura, pues la PC Odroid C2 incorpora en su arquitectura un disipador de calor que le permite tener un comportamiento mucho más estable, lo cual resulta en una ventaja significativa si se desea conectar dispositivos diversos. La demanda de corriente tiene relación directa con el incremento de temperatura en cualquier plataforma.

Adicionalmente, la PC Odroid C2 incorpora un mecanismo de protección en caso de que la temperatura de trabajo exceda los límites de operación, consistente en disminuir las operaciones realizadas por el procesador hasta que la temperatura regrese a los umbrales de operación [Roy, 2015].

Con los resultados obtenidos de las pruebas de benchmarking realizadas a ambas PC, y tomando como referencia algunas de las tareas que se llevan a cabo dentro de una infraestructura de una ciudad inteligente, se observa que la PC Odroid C2 presenta ventaja de potencia de operaciones, como es el caso del encriptado de datos mediante el protocolo SSL, del cual la PC Odroid C2 tuvo un desempeño mayor con respecto a la Raspberry Pi. Asimismo, la transferencia de datos mediante el puerto de red Ethernet de la PC Odroid C2 es mucho más veloz, al incorporar un puerto Gigabit Ethernet con lo que puede transferir una cantidad de información mayor en menor tiempo, obteniendo de esa manera un mejor uso de los recursos de la PC Odroid C2.

Otro factor importante dentro de las tareas de una ciudad inteligente es el análisis de compresión de datos. En este sentido, al transmitir información por un medio de comunicación inalámbrico es necesario considerar que mientras mejor compactados vayan los datos, se hará uso de un menor ancho de banda, dejando espacio suficiente para que más canales de comunicación ejecuten otros procesos. La PC Odroid C2 presentó un nivel superior de compresión de datos al

operar a más MIPS (millones de instrucciones por segundo) en menor cantidad de tiempo que la tecnología competidora.

Ante las notables diferencias entre ambas tecnologías se puede observar que la PC Odroid C2 muestra mejor desempeño en las tareas a las que fue sometida.

5. Conclusiones

A partir de los resultados obtenidos, se observa que la PC Odroid C2 presenta un comportamiento en desempeño superior a otras plataformas similares, concretamente la PC Raspberry Pi versión 3.

Se destaca la capacidad de la PC Odroid C2 para realizar diversas tareas de alta importancia como lo es la codificación de canal, la compresión de datos, entre otros, y de este modo garantizar que los datos a ser transmitidos llevan robustez en caso de errores en la transmisión de los mismos y por otra parte, los datos no viajan expuestos sobre el medio de comunicación elegido.

Otras plataformas como la PC Raspberry Pi, presentan un desempeño aceptable en tareas individuales y paralelas, pero a su vez presentan tiempos de respuesta más elevados al momento de implementar tareas más complejas, por lo cual se limita el uso de la misma en entornos de mayor demanda de procesamiento.

Llevando estas características al entorno de ciudades inteligentes, emplear la PC Odroid C2 se vuelve una tarea muy atractiva, dadas las características y desempeño analizados. Esta plataforma es una herramienta con prestaciones muy sobresalientes, haciendo muy atractivo su uso en este entorno. Queda como trabajo a futuro, implementar esquemas prototipo como:

- ✓ Análisis de algoritmos de compresión de imágenes y video, pensando en reducir la cantidad de datos a ser enviados y procesados.
- ✓ Análisis de esquemas de codificación de canal más robustos, como turbo codificación y esquemas de perforado, aptos para canales inalámbricos.
- ✓ Análisis de complejidad de algoritmos de cifrado, particularmente cifrado de curvas elípticas que permiten un cifrado muy robusto de los datos a un costo razonable de complejidad.

6. Bibliografía y Referencias

- [1] A. Arroub, B. Zahi, E. Sabir & M. Sadik, A literature review on Smart Cities: Paradigms, opportunities and open problems, International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 2016.
- [2] B. Benamrou, B. Mohamed, A. S. Bernoussi & O. Mustapha, Ranking models of smart cities, de 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, 2016.
- [3] C. Berrou, A. Glavieux & P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: Turbo codes, Proc. IEEE Int. Conf. Commun, 1064-1070, 1993.
- [4] C. E. Shannon, A Mathematical Theory of Communication, The Bell System Technical Journal, vol. 27, 379–423, 623–656, 1948.
- [5] D. F. Tseng, F. G. Mengistu, Y. S. Han, A. M. Mulatu & T. R. Tsai, Robust Turbo Decoding in a Markov Gaussian Channel, in IEEE Wireless Communications Letters, vol. 3, nº 6, 633-636, 2014.
- [6] E. González, Red de sensores, Internet de las cosas, Tesis de grado, Escola Tècnica Superior d'Enginyeria Informàtica, Universitat Politècnica de València, 2015.
- [7] J. Muvuna, T. Boutaleb, S. B. Mickovski & K. J. Baker, Systems engineering approach to desing and modelling of smart cities, de International Conference for Students on Applied Enginnering (ICSAE), Newcastle upon Tyne, UK , 2016.
- [8] Phoronix, Phoronix test suite: <https://www.phoronix-test-suite.com/>. último acceso: 15 Agosto 2017.
- [9] R. H. Morelos, The Art of Correcting Coding, San Jose Francisco University USA: WILEY, 2006.
- [10] R. Roy & V. Bommakanti, User Manual Odroid C2, South Korea: Hardkernel, 2015.
- [11] T. A Liscano, D. Montoya, Mejoramiento de la movilidad y el tránsito en la ciudad de Santiago de Cali a través de la planeación y diseño de dos

servicios basados en TIC, Tesis de grado, Facultad de ingeniería departamento académico de tecnologías de información y comunicaciones, Universidad ICESI, 2014-2015.

- [12] Y. Liu, Z. Tan, H. Hu, L. J. Cimini & G. Y. Li, Channer Estimation for OFDM, IEEE Communications Surver & Tutorials, vol. 16, nº 4, 1891-1908, 2014.
- [13] Z. Solarte , L. Peña, D. Almario y C. Loaiza, Urbaneyes: plataforma de la gestion de información de ciudad, El hombre y la maquina, nº 48, 54-61, 2017.