



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

**EVALUACIÓN DEL RENDIMIENTO DE LA TRANSFERENCIA
DE IMÁGENES ENTRE LOS LENGUAJES DE PROGRAMACIÓN
JAVA Y PYTHON**

ROSMAN JOSÉ PAUCAR CÓRDOVA

**Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo,
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,
como requisito parcial para la obtención del grado de:**

MAGÍSTER EN INTERCONECTIVIDAD DE REDES

RIOBAMBA - ECUADOR

NOVIEMBRE 2018



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación Modalidad Proyectos de Investigación y Desarrollo, denominado: **“EVALUACIÓN DEL RENDIMIENTO DE LA TRANSFERENCIA DE IMÁGENES ENTRE LOS LENGUAJES DE PROGRAMACIÓN JAVA Y PYTHON”**, de responsabilidad del señor Rosman José Paucar Córdova, ha sido minuciosamente revisado y se autoriza su presentación:

Tribunal:

Lic. Pepita Ivonn Alarcón Parra; M.Sc.

PRESIDENTE

Ing. Paúl Xavier Paguay Soxo; M.Sc

DIRECTOR DE TESIS

Ing. Alexandra Buñay Guisñan; M.Sc

MIEMBRO DEL TRIBUNAL

Ing. Juana Karina Arellano Aucancela; M.Sc

MIEMBRO DEL TRIBUNAL

Riobamba Noviembre, 2018

DERECHOS INTELECTUALES

Yo, Rosman José Paucar Córdova, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

ROSMAN JOSÉ PAUCAR CÓRDOVA

No. Cédula 070430202-5

© **2018**, Rosman José Paucar Córdova

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

DECLARACIÓN DE AUTENTICIDAD

Yo, Rosman José Paucar Córdova, declaro que el presente proyecto de investigación, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

Riobamba, noviembre de 2018.

ROSMAN JOSÉ PAUCAR CÓRDOVA

No. Cédula 070430202-5

DEDICATORIA

Dedico este trabajo de titulación a mi familia, debido a que son ellos quienes han sido la motivación fundamental para buscar mi superación profesional, a mi difunto y amado hermano Roberto Fernando Paucar Córdova, a mi bella esposa Paola, a mis hermanas Anita y Mónica, y a mis amados padres que siempre están conmigo cuando más los he necesitado. De igual manera a mis amigos y compañeros de aula, a mis docentes, de forma especial a mi tutor Paul y mis estimadas Karina y Pamela miembros de tribunal de tesis. A todos Uds. por su apoyo para que esta meta se cumpla

Rosman José Paucar Córdova

AGRADECIMIENTO

Agradezco a mi tutor Paúl Paguay, los miembros del tribunal Pamela Buñay y a Karina Arellano, por haberme podido guiar de gran forma para la culminación de la presente investigación.

A los docentes de la maestría por impartirme sus acertados conocimientos. Al personal Directivo y Administrativo de la IPEC por brindarme las debidas facilidades antes y durante el transcurso del programa de maestría.

En general, a todos quienes hicieron posible la realización y culminación exitosa de este proyecto de titulación. A todos ustedes mi más sincero y profundo agradecimiento.

Rosman José Paucar Córdova

INDICE DE CONTENIDO

RESUMEN	xix
CAPITULO I	1
1. INTRODUCCIÓN	1
1.1 Planteamiento del problema	3
1.2 Formulación del problema.....	5
1.3 Sistematización del problema	5
1.4 Justificación de la investigación	5
1.5 Objetivos de la investigación.....	7
1.5.1 <i>Objetivo General</i>	7
1.5.2 <i>Objetivos Específicos</i>	7
1.6 Hipótesis	7
CAPITULO II	8
2. MARCO DE REFERENCIA.....	8
2.1 Antecedentes del problema.....	8
2.2 Raspberry Pi	9
2.2.1 <i>Ventajas y Desventajas de Raspberry Pi</i>	11
2.3 Sistema Operativo Raspbian.....	12
2.4 Red Inalámbrica.....	12
2.5 Redes Inalámbricas en Raspberry.....	13
2.6 Arquitectura de Aplicaciones Cliente – Servidor	14
2.6.1 <i>Ventajas y Desventajas</i>	15
2.7 Sockets.....	15
2.7.1 <i>Funcionamiento de Sockets</i>	16
2.7.2 <i>Lenguajes de Programación</i>	17
2.8 El lenguaje JAVA	17
2.8.1 <i>Características de Java</i>	18
2.8.2 <i>Entornos de Desarrollo</i>	19
2.8.3 <i>Versiones</i>	19

2.8.4	<i>Sockets en Java</i>	20
2.8.5	<i>Funcionamiento de Sockets en Java</i>	20
2.8.6	<i>La Clase Socket en Java</i>	20
2.8.7	<i>La Clase ServerSocket en Java</i>	21
2.9	El lenguaje PYTHON	21
2.9.1	<i>Entornos de Desarrollo</i>	21
2.9.2	<i>Características de PYTHON</i>	22
2.9.3	<i>Versiones</i>	22
2.9.4	<i>Sockets en Python</i>	22
2.9.5	<i>Comparativa entre Java y Python</i>	23
2.10	Rendimiento de los Sistemas	24
2.10.1	<i>Tasa de Transferencia</i>	24
2.10.2	<i>Definición de Tasa de Transferencia</i>	24
2.10.3	<i>Definición de Tasa de Latencia</i>	24
2.10.4	<i>Cálculo de Transferencia de Datos</i>	25
	CAPITULO III	26
3.	DISEÑO DE LA INVESTIGACIÓN	27
3.1	Tipo y diseño de investigación	27
3.1.1	<i>Tipo de investigación</i>	27
3.1.2	<i>Diseño de investigación</i>	27
3.2	Métodos de investigación	28
3.2.1	<i>Método Experimental y de Observación</i>	28
3.3	Enfoque de la investigación.....	28
3.4	Alcance investigativo	28
3.5	Población de estudio	28
3.6	Selección de la muestra	29
3.7	Técnica de recolección de datos primarios y secundarios.	29
3.8	Variables e indicadores.....	30
3.9	Operacionalización de variables	31

3.9.1	<i>Conceptualización de las variables</i>	31
3.9.2	<i>Matriz de consistencia</i>	31
3.10	Procesamiento y Análisis.....	32
3.10.1	<i>Análisis de Indicadores Individual</i>	32
3.10.2	<i>Análisis Multicriterio</i>	33
3.11	Parámetros de Evaluación.....	33
3.11.1	<i>Métricas internas</i>	34
3.11.2	<i>Métricas externas (Consumo de Recursos)</i>	35
3.12	Herramientas.....	36
3.12.1	<i>Herramientas de recolección de información</i>	36
3.12.2	<i>Para tabulación y Análisis Estadísticos</i>	39
3.12.3	<i>Para la comprobación de la Hipótesis</i>	40
3.12.4	<i>Para gestión del proyecto</i>	41
3.13	Planteamiento de fórmulas.	41
3.13.1	<i>Media</i>	42
3.13.2	<i>Mediana</i>	42
3.13.3	<i>Desviación estándar</i>	42
3.13.4	<i>Diferencia Porcentual</i>	42
3.13.5	<i>Análisis Multicriterio Discreto</i>	43
3.14	Escenario	44
3.14.1	<i>Prototipos</i>	45
3.14.2	<i>Hardware cliente/servidor</i>	45
3.14.3	<i>Software utilizado</i>	46
CAPITULO IV		47
4.	RESULTADOS Y DISCUSIÓN	47
4.1	Indicador 1: Métricas Interna.....	47
4.1.1	<i>Índice 1.1: Tiempo de Respuesta</i>	47
4.1.2	<i>Índice 1.2: Tiempo de Procesamiento</i>	49
4.2	Indicador 2: Métricas Externas	50

4.2.1	<i>Índice 2.1: Porcentaje de Uso de CPU</i>	50
4.2.2	<i>Índice 2.1: Porcentaje de Uso de RAM</i>	52
4.2.3	<i>Índice 2.3: Ancho de Banda</i>	53
4.3	Comprobación de la Hipótesis.....	55
CAPITULO V.....		60
5.	DESARROLLO DEL SISTEMA DE VIDEO VIGILANCIA.....	60
5.1	Metodología de Desarrollo SCRUM	60
5.1.1	<i>¿Cómo funciona SCRUM?</i>	60
5.1.2	<i>Componentes</i>	61
5.2	Fase de Planificación	65
5.2.1	<i>Requisitos del Proyecto</i>	65
5.2.2	<i>Roles del Proyecto</i>	66
5.2.3	<i>Product Backlog</i>	66
5.2.4	<i>Sprint backlog (Planificación)</i>	68
5.2.5	Cronograma de actividades.....	69
5.3	Diseño de la solución.....	69
5.3.1	<i>Programación MULTITHILO</i>	69
5.3.2	<i>Sockets</i>	70
5.3.3	<i>Casos de uso</i>	71
5.3.4	<i>BurnDown Chart</i>	72
5.3.5	<i>Historias de usuario</i>	72
5.4	Diagrama de Despliegue.....	74
5.5	Diagrama de Componentes.....	74
CONCLUSIONES.....		75
RECOMENDACIONES.....		77
ANEXOS.....		78

INDICE DE TABLAS

Tabla 1-2:	Ventajas y Desventajas del Modelo Cliente - Servidor	15
Tabla 2-2:	Comparativa de los lenguajes Java y Python.....	21
Tabla 3-3:	Conceptualización de las variables.....	32
Tabla 4-3:	Operacionalización de variables.....	32
Tabla 5-3:	Matriz de consistencia.....	33
Tabla 6-3:	Ponderación de parámetros.....	35
Tabla 7-3:	Estadísticas de tiempos entre Java y PYTHON.....	44
Tabla 8-3:	Matriz de Impacto.....	46
Tabla 9-3:	Detalles Hardware utilizado para las pruebas.....	48
Tabla 10-3:	Características Software	49
Tabla 11-4:	Resultados de Tiempo de Respuesta	50
Tabla 12-4:	Resultados de Tiempo de Procesamiento	53
Tabla 13-4:	Porcentaje de Uso de CPU	55
Tabla 14-4:	Porcentaje de Uso de RAM	56
Tabla 15-4:	Ancho de Banda	58
Tabla 16-4:	Matriz de Impacto.....	60
Tabla 17-5:	Requisitos Funcionales.....	69
Tabla 18-5:	Requisitos No Funcionales	69
Tabla 19-5:	Roles del Proyecto.....	70
Tabla 20-5:	Historias Técnicas.....	71
Tabla 21-5:	Historias de Usuario	71
Tabla 22-5:	Sprint Backlog.....	72
Tabla 23-5:	Historia de Usuario Autenticación	78
Tabla 24-5:	Historia de Usuario conectarse a la cámara.....	78
Tabla 25-5:	Historia de Usuario Grabar.....	78
Tabla 26-5:	Historia de Usuario desactivar cámaras.....	79

INDICE DE FIGURAS

Figura 1-2:	Placa Raspberry Pi	10
Figura 2-2:	Modelos de Raspberry Pi	11
Figura 3-2:	Sistema Operativo Raspbian	12
Figura 4-2:	Modelo Cliente - Servidor	15
Figura 5-2:	Funcionamiento Socket Cliente - Servidor	17
Figura 6-2:	Plataforma Java	20
Figura 7-2:	Transferencia de datos	27
Figura 8-3:	Diferencia entre Tiempo de Procesamiento y Respuesta.....	37
Figura 9-3:	Ejecución del comando ps	40
Figura 10-3:	Herramienta iftop	41
Figura 11-3:	Resultado Software SIAE	43
Figura 12-3:	Pantalla Principal software NAIADE	44
Figura 13-3:	Escenario de prueba	51
Figura 14-3:	Prototipo de Servidor tanto en Java como en Python.....	40
Figura 15-4:	Análisis Estadístico Tiempo de Respuesta.....	53
Figura 16-4:	Análisis Estadístico Tiempo de Procesamiento	55
Figura 17-4:	Análisis Estadístico % Uso de CPU.....	54
Figura 18-4:	Análisis Estadístico % Uso de RAM	58
Figura 19-4:	Análisis Estadístico Ancho de Banda	56
Figura 20-4:	Matriz de Impacto en NAIADE	62
Figura 21-4:	Resultado de la Matriz de Impacto	63
Figura 22-4:	Comparación de pares entre Java y Python	59
Figura 23-4:	Grados de confianza.....	61
Figura 24-5:	Marco de trabajo SCRUM	69
Figura 25-5:	Estados de un hilo	70
Figura 26-5:	Proceso Socket	77
Figura 27-5:	Casos de Uso.....	78
Figura 28-5:	BurnDown Chart	79
Figura 29-5:	Diagrama de Despliegue	74
Figura 30-5:	Diagrama de Componentes	74

ÍNDICE DE GRÁFICOS

Gráfico 1-4:	Tiempo de Respuesta.....	52
Gráfico 2-4:	Tiempo de Porcesamiento	54
Gráfico 3-4:	Porcentaje de Uso de CPU	56
Gráfico 4-4:	Porcentaje de uso de RAM	58
Gráfico 5-4:	Ancho de Banda	60
Gráfico 6-5:	Diagrama de Gantt del proyecto.....	75

ÍNDICE DE ABREVIATURAS

API	APPLICATION PROGRAMMING INTERFACE
CPU	CENTRAL PROCESSOR UNIT
ESPOCH	ESCUELA POLITÉCNICA DEL CHIMBORAZO
	FUNCIONALIDAD, USABILIDAD, CONFIABILIDAD, DESEMPEÑO Y CAPACIDAD
FURPS	DE SOPORTE
HP	HEWLETT PACKARD
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
IEC	INTERNATIONAL ELECTROTECHNICAL COMMISSION
IOT	INTERNET OF THINGS (INTERNET DE LAS COSAS)
IP	PROTOCOLO DE INTERNET
ISO	INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
JDK	JAVA DEVELOPMENT KIT
J2EE	JAVA 2 PLATFORM, ENTERPRISE EDITION
J2SE	JAVA 2, STANDARD EDITION
J2ME	JAVA 2 MICRO EDITION
MIT	MASSACHUSETTS INSTITUTE OF TECHNOLOGY
NIST	NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
RAM	MEMORIA DE ACCESO ALEATORIO
RMI	JAVA REMOTE METHOD INVOCATION
RPC	REMOTE PROCEDURE CALL
RTDBS	REAL-TIME DATA BASE SYSTEM
SI	INFORMATION SYSTEM
SIAE	SISTEMA INTELIGENTE DE ANÁLISIS ESTADÍSTICO
SPEC	ESTÁNDAR PERFORMANCE EVALUATION CORPORATION
SSH	SECURE SHELL
TCP	TRANSMISSION CONTROL PROTOCOL
TIOBE	THE IMPORTANCE OF BEING EARNEST
TLS	TRANSPORT LAYER SECURITY
UDP	USER DATAGRAM PROTOCOL
VLAN	VIRTUAL LOCAL AREA NETWORK
WWW	WORLD WIDE WEB
XML	EXTENSIBLE MARKUP LANGUAGE

RESUMEN

Se realizó el análisis comparativo del rendimiento del proceso de transferencia de imágenes entre los lenguajes libres de programación Java y Python utilizando Internet de las cosas (placa reducida Raspberry Pi), para lo cual se utilizó el modelo propuesto NAIADE, que establece métricas de evaluación, las cuales son de tipo internas (tiempo de respuesta y tiempo de procesamiento) y métricas externas (uso de RAM, uso de CPU y uso de Ancho de Banda). Con estos indicadores, se procedió a trabajar con una muestra de 375 imágenes de una población de 14400 mediante el método estadístico de la Prueba Z, obteniéndose los siguientes resultados: para las métricas internas, tiempo de respuesta se obtuvo 258% a favor de Java, y para el tiempo de procesamiento 13.88% a favor de Java. En cuanto a las métricas externas, el uso de CPU un 86% a favor de Python, uso de RAM un 34% a favor de Python y para el uso de ancho de banda un 2055% a favor de Java. Finalmente, para consolidar los resultados se aplicó el análisis multicriterio con el software de NAIADE mediante una matriz de impacto obteniendo un ranking positivo a favor de Java de 0.7 con respecto a Python, comprobándose de esta manera que si existió una diferencia significativa entre Java y Python. Como propuesta se desarrolló una guía de implementación y especificaciones técnicas de un sistema de videovigilancia basado en Internet de las Cosas IoT en el lenguaje de programación Java, utilizando la metodología de desarrollo ágil SCRUM, para la Dirección Distrital 07D04 Balsas – Marcabelí – Piñas del Ministerio de Educación.

Palabras Clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <REDES>, <LENGUAJES DE PROGRAMACION>, <JAVA> , <PYTHON>, <RENDIMIENTO, SOCKET>, <INTERNET DE LAS COSAS>, <RASPBERRY PI (SOFTWARE – HARDWARE)>, <MÉTRICAS DE EVALUACIÓN>, <NOVEL APPROACH TO IMPRECISE ASSESSMENT AND DECISION ENVIROMENTS (NAIADE)>

ABSTRACT

The comparative analysis of the performance of the image transfer process between the free Java and Python programming languages was carried out using the Internet of things (Raspberry Pi reduced plate), for which the proposed NAIADÉ model was used, that establishes evaluation metrics. Which are internal type (response time and processing time) and external metrics (use of RAM, CPU usage and Bandwidth usage). With these indicators, it was proceeded to work with a sample of 375 images of a population of 14400 by means of statistical method of the Z Test, obtaining the following results: for internal metrics, response time was 258% in favor of Java, and for the processing time 13.88% in favor of Java. As for the external metrics, the use of CPU 86% in favor of Python, use of RAM 34% in favor of Python and for the use of bandwidth 2055% in favor of Java. Finally, to consolidate the results, the multi criteria analysis was applied with the NAIADÉ software through an impact matrix obtaining a positive ranking in favor of Java of 0.7 with respect to Python, proving this way that a significant difference between Java and Python did exist. As a proposal, it was developed a guide of implementation and technical specifications of a video surveillance system based on Internet of Things IoT in the Java programming language, using the methodology of agile development SCRUM, for the District Department 07D04 Balsas-Marcabellí-Piñas of the Ministry of Education.

Key words: <TECHNOLOGY AND ENGIENEERING SCIENCE>, <NETWORKS>, <PROGRAMMING LANGUAGES>, <JAVA>, <PYTHON>, <PERFORMANCE, SOCKET>, <INTERNET OF THINGS>, <RASPBERRY PI (SOFTWARE-HARDWARE)>, <EVALUATION METRICS>, <NOVEL APPROACH TO IMPRECISE ASSESSMENT AND DECISION ENVIRONMENT (NAIADÉ)>.

CAPÍTULO I

1. 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 hasta marzo de 2017 contaba con 17 millones de dispositivos conectados (Pástor, 2018) 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 (Matas, 2016), IoT es una de las grandes tendencias del futuro, con previsiones para el 2020, donde se espera entre cincuenta mil y setenta mil millones dispositivos conectados, cuando la población mundial se estima que sea de 7.6 mil millones.

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, Roquero, 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 (Ramakrishnan, Anbazhagan , Rajan , & Karthik, 2015), ya que de esto dependerá la calidad del Servicio (VIDAL-SILVA, Madariaga, & Solís, 2017). En el mismo sentido, según (Medina, 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.

De igual forma, en la mayoría de sistemas informáticos, existen diversas opciones de lenguajes de programación para el desarrollo de aplicaciones con placas reducidas, habiendo opciones de software propietario y libre, donde los lenguajes de programación de software libre proveen una gran ventaja en cuanto a los costos de implementación y soporte de las comunidades. Según el sitio web (TIOBE, 2018) 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 mas utilizandas en los prototipos para la el desarrollo de sistemas basados en IoT para videovigilancia. Algunos proyectos de este tipo que se pueden citar tenemos por ejemplo el de (Naula, Llanos, & Valdivieso, 2014), (Ferdoush & Li, 2014) que utilizan el lenguaje Python mientras en los de (Adrian-Viorel , Florin, & Valeriu, 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 libre Java o Pyhton 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?

Por otra parte, la Dirección Distrital 07D04 Balsas – Marcabeli – Piñas del Ministerio de Educación, tiene como objetivo garantizar el acceso y calidad de la educación inicial, básica y bachillerato, a los y las habitantes de la parte alta de la Provincia de El Oro. Dentro de los procesos de la Dirección Distrital se realizan actividades como atención a la ciudadanía, en donde diariamente se presta el servicio a las personas en los diferentes departamentos, lo que implica que la seguridad de las instalaciones debe ser adecuada, para ello se requiere a más del personal de seguridad de una infraestructura tecnológica que apoye esta tarea; por lo que un sistema de videovigilancia seria de gran utilidad para la institucion, en donde cuyo funcionamiento garantice un adecuado rendimiento, calidad de imagenes y oportunidad del video, así entonces, posterior a la investigación se determinará la mejor opción para el desarrollo e implementacion de la aplicación en el sistema de videovigilancia para esta Dirección Distrital.

Los capítulos en los cuales se ha dividido el presente trabajo de titulación se detallan a continuación:

CAPÍTULO I, Introducción, en este capítulo se detalla la introducción, problema, objetivos generales y específicos, justificación e hipótesis planteada.

CAPÍTULO II, Marco Referencial, en este capítulo se describen conceptos fundamentales de seguridad de información, criptografía, arquitectura cliente/servidor, plataformas de desarrollo y modelos de calidad de software.

CAPÍTULO III, Diseño de la Investigación, en este capítulo se describe el diseño, tipo, métodos y enfoque de la investigación, la población, variables e indicadores, matriz de consistencia, parámetros y herramientas, el método FURPS, prototipos y software utilizado.

CAPITULO IV, Resultados y Discusión, en este capítulo se detalla la recolección, calculo y análisis de datos en base a indicadores establecidos, se detallara los metodos estadisticos, matriz de consistencia, resultados y la comprobación de la hipótesis.

CAPITULO V, Desarrollo del Sistema, en este capítulo se detalla el desarrollo del aplicativo de respaldos aplicando la metodología SCRUM.

1.1 Planteamiento del problema

En el mundo actual el Internet de las Cosas forma parte de las denominadas “Tecnologías Emergentes”, la misma que tiene diferentes ámbitos de aplicación como: salud, transporte, servicios, seguridad, industria, comercio, etc. En el ámbito de la seguridad tanto a nivel mundial, nacional y de manera local, los sistemas de videovigilancia están siendo muy utilizados, en especial debido a la portabilidad de las Computadoras de Placa Simple. Para cumplir con su propósito, estos sistemas de video vigilancia que hacen uso del Internet de las Cosas (IoT), necesitan enviar imágenes con frecuencia de intervalos cortos para formar el video, que como lo indica Lozano citado por (Pretis, 2003), “Las imágenes en movimiento están formadas por un número determinado de imágenes que con cierta continuidad y velocidad, aproximadamente 25 fotogramas por segundo, producen en el ojo humano la sensación de movimiento”, sin embargo en los sistemas de videovigilancia que deben estar transmitiendo constantemente los estándares cambian como lo mencionan (Axis Communications, 2018) o (SecuraMe, 2018) quienes 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, lo que implica que el sistema

debe contar con un buen rendimiento para la transferencia de imágenes que no declive su funcionamiento en el tiempo, lo cual incidirá en la calidad de servicio del sistema. Por otro lado, las computadoras de placas reducida al contar con capacidades hardware limitadas, requiere que las aplicaciones que en ella se ejecutan hagan un uso eficiente de los recursos que repercute en la eficiencia energética y experiencia de usuario (SIGEA).

Según (Medina, 2014), el rendimiento de un sistema informático, 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.

Para la implementación de los sistemas de video vigilancia, y en el caso particular de las basadas en Internet de las Cosas, se dispone de multitud de lenguajes de programación para el desarrollo, donde los lenguajes libres ofrecen una ventaja en cuanto a costos de implementación y un amplio soporte por parte de la comunidad de desarrolladores a nivel mundial. Entre las opciones de lenguajes libres más utilizadas, popularizadas y con mayor proyección de crecimiento a escala de nivel local, regional y global se encuentran Java y Python, los mismos que según el sitio (TIOBE, 2018) que analiza los lenguajes de programación más utilizados, los posiciona entre los cuatro primeros lugares. Con este antecedente cabe preguntarse, *¿Cuál de los dos lenguajes de programación de software libre más utilizados y popularizados en la videovigilancia a nivel mundial como lo son Java y Python, ofrece un mejor rendimiento para la transferencia de imágenes utilizando Internet de las Cosas?*

Es así, que el presente trabajo se centró en la comparación de los dos lenguajes de programación libres más utilizados y en apogeo en el mundo de la informática y en la videovigilancia, en donde, se determinó experimentalmente y justificando de manera real y documentada, cuál de ellos logró obtener el mejor rendimiento en la transferencia de imágenes utilizando una placa reducida (Raspberry Pi), haciendo uso de la red inalámbrica (802.11ac) de la institución, la cual resulta amigable y oportuna de implementar en el sistema de videovigilancia dentro de la institución pública como lo es la Dirección Distrital 07D04 Balsas – Marcabelí – Piñas del Ministerio de Educación, teniendo en cuenta, que se tomó como área de estudio una de sus principales unidades como lo es el Departamento de Atención Ciudadana, en donde diariamente se atienden a un gran número de usuarios lo que implica que la seguridad de las instalaciones debe ser adecuada, requiriendo a más del personal de seguridad de una infraestructura tecnológica que apoye esta tarea.

1.2 Formulación del problema

¿Cuál de los lenguajes de programación de software libre más popularizados y usados en la programación y videovigilancia, Java o Python ofrece un mejor rendimiento en la transferencia de imágenes en un sistema de videovigilancia basada en Internet de las Cosas?

1.3 Sistematización del problema

- ¿Cuáles son las características de los lenguajes de programación Java y Python junto con sus plataformas necesarias para desarrollar aplicaciones basadas en IoT?
- ¿Cuáles son los parámetros de evaluación y herramientas de medición que permitan la evaluación del rendimiento de la transferencia de imágenes?
- ¿Cuál es el rendimiento de una aplicación desarrollada en Java y en Python en la transferencia de imágenes?
- ¿Cuál es el resultado del análisis comparativo para determinar la herramienta de desarrollo que permita mejor transferencia de imágenes?
- ¿Cómo se implementa el sistema de videovigilancia en la Dirección Distrital 07D04 Balsas – Marcabellí – Piñas del Ministerio de Educación?

1.4 Justificación de la investigación

El Internet de las Cosas dentro de sus diferentes áreas de aplicación cuenta con aplicaciones y variantes de seguridad como lo son los sistemas de videovigilancia, que a su vez son opciones de sistemas que reducen los costos de instalación como lo indica (Shah & Bharadi, 2016), así también el software libre provee una alternativa, económica al momento del desarrollo de sistemas informáticos. Los sistemas de videovigilancia se fundamentan en la transmisión de imágenes o fotogramas, que luego en el servidor son consolidados y utilizados para generar los videos. Esta transmisión y el uso de los recursos que se emplean para su desarrollo, son una tarea constante lo cual toma una relevancia sustancial para el correcto funcionamiento de todo el sistema y que es motivo de la actual investigación. Por otro lado, las plataformas de desarrollo y los lenguajes de programación de los sistemas informáticos empleados en videovigilancia para computadoras de placa simple al igual que para cualquier arquitectura hardware, son cambiantes y cotidianamente

aparecen nuevas versiones mejoradas de las ya existentes, lo cual permite una mejora continua de los servicios software y de hardware.

Con estos antecedentes, la presente investigación tuvo como propósito determinar experimentalmente cuál de los dos lenguajes de programación de software libre más popularizados y usados en el mundo de la informática y en la videovigilancia según (TIOBE, 2018), Java o Python ofrece un mejor rendimiento en la transferencia de imágenes en un sistema de videovigilancia basada en Internet de las Cosas (placa reducida Raspberry Pi).

Para la evaluación del rendimiento se tomó como base las métricas propuestas por (Medina, 2014), que las clasifica en métricas externas e internas, entre las métricas externas están: Número de peticiones atendidas por unidad de tiempo, tiempo medio de respuesta, número medio de errores, y entre las internas: Uso medio de recursos como CPU y RAM. Esta evaluación del rendimiento se analizó en dos escenarios. El primer escenario estableció la implementación de software realizado en lenguaje Java para la transferencia de imágenes, mientras que el segundo escenario estableció la implementación de software realizado en el lenguaje Python. La evaluación de ambos escenarios se realizó bajo las mismas características tanto hardware como software, así como del mismo medio y tipo de transmisión inalámbrica (802.11ac), y bajo la misma arquitectura (Cliente-Servidor).

Al final de la investigación, se determinó estadísticamente la mejor alternativa de software para este tipo de procesos, lo cual incidió en la calidad de video y posterior experiencia de usuario, así como, contribuir con un estudio documental, investigativo y explicativo de cómo funciona y se realiza la transferencia de imágenes en este tipo de escenarios utilizando equipos de las conocidas tecnologías emergentes IoT.

Por otra parte, una vez encontrado el resultado de la comparación de estas plataformas, el lenguaje de programación con mejor rendimiento fue utilizado para aplicarlo en el sistema de videovigilancia basado en IoT, en la Dirección Distrital 07D04 Balsas – Marcabelí – Piñas del Ministerio de Educación, como un aporte del estudio realizado, lo cual contribuye de gran manera para la supervisión de procesos administrativos y de seguridad de la institución, donde laboran diariamente treinta y dos funcionarios distribuidos en trece departamentos.

1.5 Objetivos de la investigación

1.5.1 *Objetivo General*

- Comparar el rendimiento de la transferencia de imágenes en un sistema de video vigilancia basado en Internet de las Cosas entre los lenguajes de programación Java y Python.

1.5.2 *Objetivos Específicos*

- Analizar las características relevantes de los lenguajes Python y Java para el desarrollo de sistemas para transferencia de imágenes basado en Internet de las Cosas.
- Diseñar escenarios de pruebas para la transferencia de imágenes en sistemas con Computadora de Placa Simple.
- Evaluar los resultados de pruebas de rendimiento de la transferencia de imágenes de los escenarios con cada lenguaje Python y Java
- Elaborar una guía de implementación y especificaciones técnicas de un sistema de videovigilancia basado en Internet de las Cosas IoT.

1.6 Hipótesis

El presente trabajo plantea la siguiente hipótesis:

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.

De esta manera, se realizará el estudio de ambas plataformas y su implementación mediante prototipos para poder obtener las conclusiones del caso.

CAPÍTULO II

2. MARCO DE REFERENCIA

2.1 Antecedentes del problema

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

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 del rendimiento entre los dos lenguajes materia de la investigación, no se logró encontrar ningún trabajo investigativo en el ámbito de la transferencia de imágenes utilizando una placa reducida en ningún repositorio local, nacional ni internacional, por lo que resultó atractivo, novedoso y necesario analizar y desarrollar este tema.

2.2 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.

Como se observa en la figura 1-2, el diseño de la Raspberry Pi incluye (Universidad Politécnica de Valencia, 2013):

- Un Chipset Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos Turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía
- Un procesador gráfico (GPU) VideoCore IV
- Un módulo de 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB).

- Un conector de RJ45 conectado a un integrado lan9512 -jzx de SMSC que nos proporciona conectividad a 10/100 Mbps
- 2 buses USB 2.0
- Una Salida analógica de audio estéreo por Jack de 3.5 mm.
- Salida digital de video + audio HDMI
- Salida analógica de video RCA
- Pines de entrada y salida de propósito general
- Conector de alimentación micro USB
- Lector de tarjetas SD

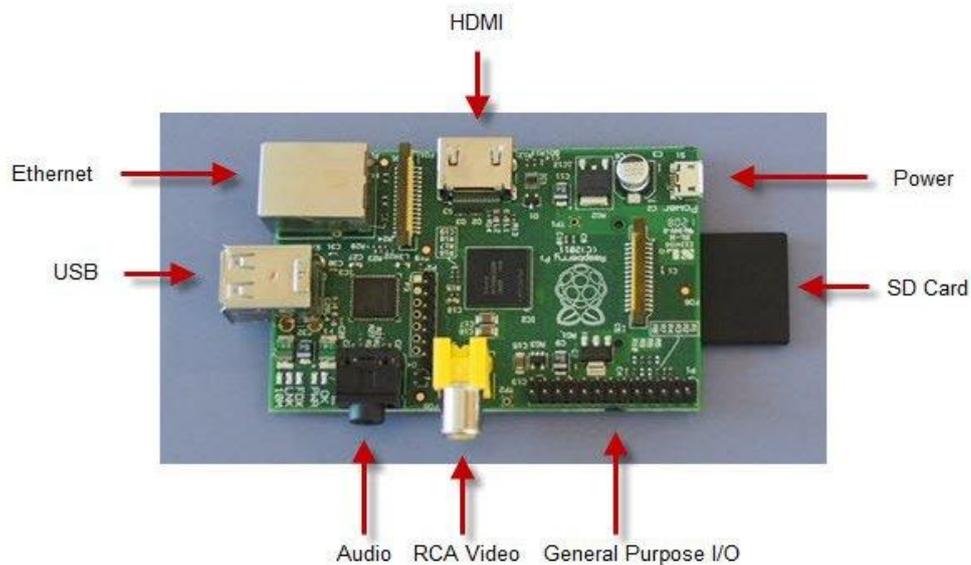


Figura 1-2: Placa Raspberry Pi
Fuente: (Universidad Politécnica de Valencia, 2013)

(Casco, 2014) Señala que “Raspberry Pi tiene todos los componentes esenciales requeridos para correr un sistema operativo. Usa el controlador Broadcom, que es un Soc (System on Chip). Este Soc tiene un poderoso procesador ARM11 que corre a 700 MHz. Esta minicomputadora no trae display, pero puede ser usado con un display HDTV o los estándares de TV NTSC o PAL. Tiene un puerto Ethernet que permite conectarlo a una red. Se pueden cargar sistemas operativos desde

Mac, Windows y Linux. Su capacidad para correr Linux y la accesibilidad a la plataforma mediante una red LAN lo hace una opción perfecta para pequeños servidores web dedicados”.

Entre los principales modelos de Raspberry que se han tenido tenemos los que se describe en la siguiente figura:

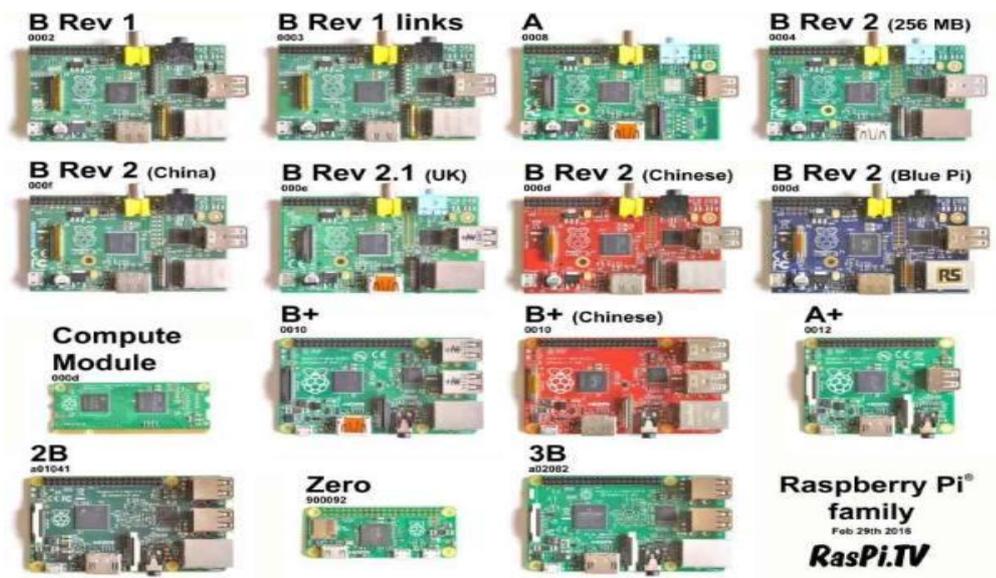


Figura 2-2: Modelos de Raspberry Pi

Fuente: *Raspi.tv*

2.2.1 Ventajas y Desventajas de Raspberry Pi

(Casco, 2014) define: “**Ventajas:** Una pequeña computadora trae toda clase de ventajas. El puerto HDMI puede usarse para conectarse a un televisor y los 2 puertos USB permiten operarlo como a una computadora con teclado y mouse fácilmente. Su procesador gráfico soporta 1080p. También tiene un puerto ethernet para fácil conexión a internet con leves dificultades. Ya que el sistema operativo corre desde una tarjeta SD, este puede cambiarse fácilmente con solo cambiar la tarjeta. Esto es muy útil considerando que se tienen varias opciones para el sistema operativo. Dado su precio, es poderoso y aun así de fácil uso para principiantes. En cuanto a capacidad de expansión, es importante mencionar que se tiene grandes beneficios gracias a la placa que permite la compatibilidad con los shields de Arduino, ya que, sin incluir estas expansiones, el soporte exclusivo para el Pi es muy bajo.

Desventajas: Es muy útil para cualquier proyecto que implique el uso de una computadora, pero a diferencia del Arduino y el BeagleBone, no tiene tantas opciones para interfaces con sensores externos o botones, es decir, orientación a hardware, lo que no lo hace una buena opción para proyectos meramente electrónicos, por ejemplo, en el hogar”

2.3 Sistema Operativo 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.”



Figura 3-2: Sistema Operativo Raspbian

Fuente: (Raspbian)

El Sistema Operativo Raspbian, será con el que más adelante se pondrá a funcionar la Raspberry Pi que se utilizará en el presente trabajo de investigación.

2.4 Red Inalámbrica

Según (Naula, Llanos, & Valdiviezo, 2014): “Las conexiones de los dispositivos, computadoras o terminales pueden ser realizadas mediante redes de comunicaciones inalámbricas (Wireless Network o Wireless Local Area Network, WLAN), entre ellas se destaca la IEEE 802.11, diseñada para ser utilizada en reemplazo de las redes LAN”.

Dentro de las ventajas de las redes wifi son según (Cisco Systems Inc, 2012):

“Accesibilidad: Todos los equipos portátiles y la mayoría de los teléfonos móviles de hoy día vienen equipados con la tecnología Wi-Fi necesaria para conectarse directamente a una LAN inalámbrica. Los empleados pueden acceder de forma segura a sus recursos de red desde cualquier ubicación dentro de su área de cobertura. Generalmente, el área de cobertura es su instalación, aunque se puede ampliar para incluir más de un edificio.

Movilidad: Los empleados pueden permanecer conectados a la red incluso cuando no se encuentren en sus mesas. Los asistentes de una reunión pueden acceder a documentos y aplicaciones. Los vendedores pueden consultar la red para obtener información importante desde cualquier ubicación.

Productividad: El acceso a la información y a las aplicaciones clave de su compañía ayuda a su personal a realizar su trabajo y fomenta la colaboración. Los visitantes (como clientes, contratistas o vendedores) pueden tener acceso de invitado seguro a Internet y a sus datos de empresa.

Fácil configuración: Al no tener que colocar cables físicos en una ubicación, la instalación puede ser más rápida y rentable. Las redes LAN inalámbricas también facilitan la conectividad de red en ubicaciones de difícil acceso, como en un almacén o en una fábrica.

Escalabilidad: Conforme crecen sus operaciones comerciales, puede que necesite ampliar su red rápidamente. Generalmente, las redes inalámbricas se pueden ampliar con el equipo existente, mientras que una red cableada puede necesitar cableado adicional.

Seguridad: El control y la administración del acceso a su red inalámbrica son importantes para su éxito. Los avances en tecnología Wi-Fi proporcionan protecciones de seguridad sólidas para que sus datos sólo estén disponibles para las personas a las que les permita el acceso.

Costos: Con una red inalámbrica puede reducir los costos, ya que se eliminan o se reducen los costos de cableado durante los traslados de cocina, nuevas configuraciones o expansiones”.

2.5 Redes Inalámbricas en Raspberry

Las Raspberry al ser placas reducidas no cuentan con una integración de conexión Wireless, por lo que si se desea esta funcionalidad se debe incorporar mediante una conexión USB, tal como se muestra en la siguiente fotografía:



Fotografía 1-2: Raspberry con conexión a internet y cámara web

Realizado por: Rosman Paucar , 2018

2.6 Arquitectura de Aplicaciones Cliente – Servidor

Según (Fúquene, 2011), “una arquitectura cliente/servidor es un modelo de aplicación distribuida en donde las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un programa cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola máquina, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras”.

Básicamente, la arquitectura cliente-servidor se tienen aplicaciones en un computador que están “conversando” con aplicaciones de otro computador. El cliente solicita servicios y el servidor cubre los requerimientos, por lo que el cómputo cliente-servidor define una arquitectura en la cual la lógica del programa está distribuida entre sistemas cliente y sistemas servidor (Daniela Garcia, Edgar Ríos, 2011)

La comunicación en el modelo cliente-servidor, es de la siguiente forma y estos mensajes se muestran en la figura 4-2.

1. El proceso cliente envía una solicitud a través de la red al proceso servidor.
2. El cliente espera una respuesta.
3. El proceso servidor recibe la solicitud y ejecuta el trabajo que se le pide o busca los datos solicitados.
4. El servidor devuelve una respuesta.

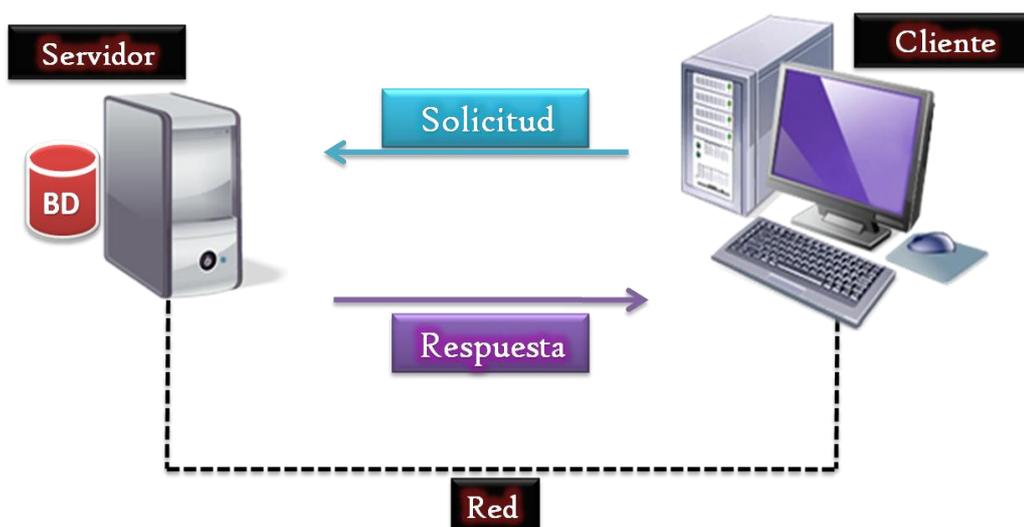


Figura 4-2: Modelo Cliente - Servidor

Fuente: (Daniela Garcia, Edgar Ríos, 2011)

2.6.1 Ventajas y Desventajas

Las principales ventajas y desventajas del esquema Cliente/Servidor tenemos en la tabla 2-1, definidas por (Márquez Bertha & Zulaica José, 2004).

Tabla 2-1: Ventajas y Desventajas del Modelo Cliente - Servidor

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">• Escalabilidad <p>Facilidad de aumentar o suprimir la capacidad de clientes y servidor por separado y el funcionamiento de la red no se verá afectado.</p>	<ul style="list-style-type: none">• Costo elevado <p>Para que este esquema C/S funcione correctamente implica tener un servidor central lo suficientemente potente como para gestionar y compartir recursos con todos los clientes que envían simultáneamente peticiones.</p>
<ul style="list-style-type: none">• Centralización del control <p>Los administradores pueden aplicar controles de seguridad para restringir los accesos a los datos y pueden utilizar mecanismos de rastreo para monitorear los accesos a los datos; toda información que quiere entrar al sistema es revisada y controlada por el servidor en caso de que un programa cliente sea defectuoso o no autorizado no pueda dañar al sistema.</p>	<ul style="list-style-type: none">• Dependencia del servidor <p>En el esquema C/S, toda la red está construida alrededor del servidor, en ocasiones si éste falla puede afectar a toda la infraestructura, por eso es importante tener servidores redundantes con tolerancia a fallos y los sistemas de almacenamiento en modo RAID.</p>
<ul style="list-style-type: none">• Encapsulación <p>Al ser computadores independientes tanto para el cliente como servidor, es posible reparar, actualizar o trasladar un servidor, mientras que los clientes no se verán afectados por ese cambio.</p>	<ul style="list-style-type: none">• Congestión de tráfico <p>Por la gran cantidad de clientes envían peticiones simultáneas al servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor).</p>
<ul style="list-style-type: none">• Tráfico de la red <p>Es reducido debido a que el servidor solamente proporciona la información requerida al cliente, por lo tanto, las estaciones de trabajo ya no procesarán grandes cantidades de información.</p>	<ul style="list-style-type: none">• Mantenimiento <p>Se requiere de personal capacitado para dar el mantenimiento porque implica la interacción de diferentes partes de Hw y Sw, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.</p>

Fuente: (Márquez Bertha & Zulaica José, 2004)

2.7 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.

2.7.1 Funcionamiento de Sockets

El modelo más básico de los sockets¹ consta de un **servidor** y un **cliente** el proceso se detalla en los siguientes pasos así como en la figura 5-2:

1. Un **servidor** se ejecuta sobre una computadora específica y tiene un *socket* que responde en un *puerto específico*. El servidor únicamente espera, escuchando a través del socket a que un cliente haga una petición.
2. **En el lado del cliente:** el cliente conoce el *nombre de host* o la IP del servidor que se encuentra ejecutando y el *número de puerto* en el cual el servidor está conectado y escuchando. Conociendo esto el cliente realiza una *petición de conexión* con el servidor.

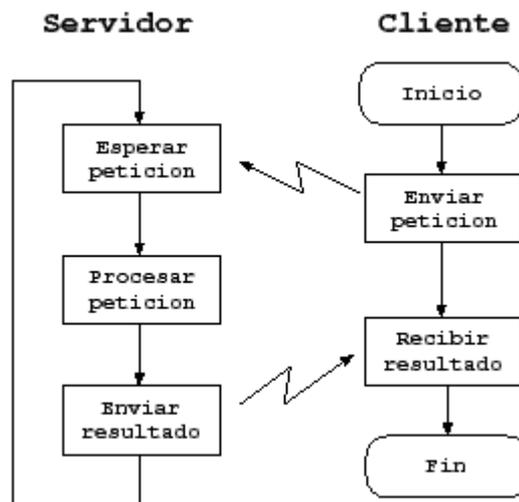


Figura 5-2: Funcionamiento Socket Cliente -Servidor

Fuente: <http://nereida.deioc.ull.es/~cleon/doctorado/doc06/doc06/html/node9.html>

3. El servidor **acepta la conexión**. Además de aceptar, el servidor obtiene un nuevo socket sobre un puerto diferente. Esto se debe a que necesita un *nuevo socket* (y, en consecuencia, un *numero de puerto diferente*) para seguir atendiendo al socket original para peticiones de conexión mientras atiende las necesidades del cliente que se conectó.

¹ Fuente: <http://nereida.deioc.ull.es/~cleon/doctorado/doc06/doc06/html/node9.html>

4. Por la parte del **cliente**, si la conexión es aceptada, un socket se crea de forma satisfactoria y puede usarlo para comunicarse con el servidor. Es importante darse cuenta que el socket en el cliente no está utilizando el número de puerto usado para realizar la petición al servidor. En lugar de éste, el cliente asigna un número de puerto local a la máquina en la cual está siendo ejecutado.
5. Ahora el cliente y el servidor pueden comunicarse escribiendo o leyendo en o desde sus respectivos sockets.
6. Finalmente se cierra o finaliza la conexión.

Los lenguajes de programación Java y Python admiten las comunicaciones de red basadas en socket. El lenguaje Java por medio de la librería `java.net` nos provee dos clases: *Socket* para implementar la conexión desde el lado del cliente y *ServerSocket* que nos permitirá manipular la conexión desde el lado del servidor. Python provee una clase: *Socket* para la conexión desde el lado del cliente y servidor.

2.7.2 Lenguajes de Programación

De acuerdo a los índices de la comunidad de programación TIOBE² actualizado hasta noviembre del 2016, indica la popularidad de los lenguajes de programación, en donde **Java** encabeza el primer lugar y en quinto puesto esta **Python**, con los cuales se puede crear aplicaciones web, móviles y de escritorio. Utiliza una infraestructura de lenguaje común para facilitar la portabilidad del código desde otros lenguajes.

2.8 El lenguaje JAVA

Se diseñó en 1990 por James Gosling, de Sun Microsystems. Inicialmente Java se llamó Oak, aunque tuvo que cambiar de denominación, debido a que dicho nombre ya estaba registrado por otra empresa. Este lenguaje fue diseñado antes del comienzo de la era World Wide Web, puesto que fue diseñado para dispositivos electrónicos de consumo como calculadoras, microondas y la televisión interactiva.

Hoy en día, puede encontrar la tecnología Java en redes y dispositivos que comprenden desde Internet y superordenadores científicos hasta portátiles y teléfonos móviles; desde simuladores de

² **Fuente:** <http://www.tiobe.com/tiobe-index/>

mercado en Wall Street hasta juegos de uso doméstico y tarjetas de crédito: Java está en todas partes (Padilla & Pérez, 2009- 2010).

2.8.1 *Características de Java*

El lenguaje fue diseñado con las siguientes características en mente según (J. Padilla & J. Pérez, 2009- 2010):

- **Simple.** Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos
- **Familiar.** Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de estos.
- **Robusto.** El sistema de Java maneja la memoria de la computadora por ti. No te tienes que preocupar por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique
- **Seguro.** El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje.
- **Interpretado.** Java corre en máquina virtual, por lo tanto, es interpretado.
- **Portable.** Como el código compilado de Java (*byte code*) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.
- **Independiente a la arquitectura.** Al compilar un programa en Java, el código resultante un tipo de código binario conocido como byte code. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma.
- **Multithreaded.** Un lenguaje que soporta múltiples threads es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.
- **Dinámico.** Java no requiere que compile todas las clases de un programa para que este funcione.

2.8.2 Entornos de Desarrollo

En un entorno de desarrollo Java, se distribuyen 3 paquetes de desarrollo (SDK) como se observa en la figura 6-2, según las necesidades de cada programador y lo que quieras programar:

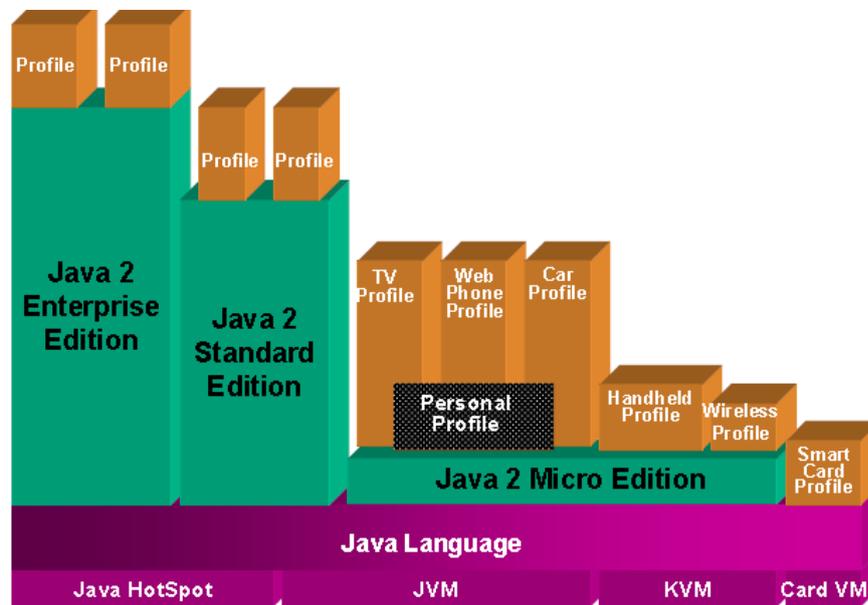


Figura 6-2: Plataformas Java

Fuente: https://www.researchgate.net/figure/Java-Platforms-J2SE-J2EE-J2ME_fig1_268508995

- **J2ME (Java 2 MicroEdition):** Permite hacer todo tipo de aplicaciones para los dispositivos móviles. (ej.: PDAs)
- **J2SE (Java 2 Standard Edition):** Permite hacer todo tipo de aplicaciones (Escritorio con Ventanas, escritorio consola, aplicaciones Web, Applets, etc.)
- **J2EE (Java 2 Enterprise Edition):** Es un paso más que el anterior, agregando funciones empresariales, para el desarrollo de todo tipo de aplicaciones mucho más robustas y complejas (Escritorio con Ventanas, escritorio consola, aplicaciones Web, Applets, etc.), como aplicaciones Distribuidas, Servicios Web, etc.

2.8.3 Versiones

De manera general se puede citar las siguientes versiones:

- JDK 1.0 (23 de enero de 1996)
- JDK 1.1 (19 de febrero de 1997)

- J2SE 1.2 (8 de diciembre de 1998) – denominación Java 2 y el nombre "J2SE" " (Java 2 Platform, Standard Edition).
- 2SE 1.3 (8 de mayo de 2000)
- J2SE 1.4 (6 de febrero de 2002)
- J2SE 5.0 (30 de septiembre de 2004)
- Java SE 6 (11 de diciembre de 2006) - Sun cambió el nombre "J2SE" por Java SE y eliminó el ".0" del número de versión.
- Java SE 7 (28 de julio de 2011)
- Java SE 8 (18 de marzo de 2014)

2.8.4 Sockets en Java

“Los sockets son básicamente formas en las que podemos interconectar 2 (o más) programas mediante el uso de la internet. En java se utilizan para poder crear conexiones utilizando básicamente una IP/hostname y un puerto para establecer la conexión. Para aprender podemos utilizarla para conectar 2 programas por medio de Internet” según (Jorge Villalobos, Uriel Hernandez, 2011).

2.8.5 Funcionamiento de Sockets en Java

(Jorge Villalobos, Uriel Hernandez, 2011), señala que “el modelo más básico de los sockets consta de 2 simples programas, un servidor y un cliente. Básicamente el programa servidor comienza a “escuchar” en un puerto determinado (nosotros lo especificamos), y posteriormente el programa que la hace de “cliente” debe conocer la ip o nombre de dominio/hostname del servidor y el puerto que está escuchando, al saber esto simplemente solicita establecer una conexión con el servidor. Es aquí cuando el servidor acepta esa conexión y se puede decir que estos programas están “conectados”, de este modo pueden intercambiar información”.

2.8.6 La Clase Socket en Java

Según (Neredia, s.f.), la clase Socket del paquete java.net es fácil de usar comparada con la que proporcionan otros lenguajes. Java oculta las complejidades derivadas del establecimiento de la conexión de red y del envío de datos a través de ella. En esencia, el paquete java.net proporciona la misma interfaz de programación que se utiliza cuando se trabaja con archivos.

2.8.7 La Clase ServerSocket en Java

Según (Neredia, s.f.), La clase ServerSocket es la que se utiliza a la hora de crear servidores, al igual que como se ha visto, la clase Socket se utilizaba para crear clientes.

2.9 El lenguaje PYTHON

Según (van Rossum, 2009), Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

2.9.1 Entornos de Desarrollo

Se detallan a continuación los más utilizados por los desarrolladores según (van Rossum, 2009):

- **“Eclipse + Pydev:** Pydev es un plugin que permite a los usuarios usar Eclipse para el desarrollo de Python y Jython, haciendo de Eclipse una IDE de primera clase para desarrollar Python. Este plugin viene con muchas características, como por ejemplo compleción de código, resaltado de sintaxis, analizador de sintaxis, reactor, debug y muchos más.
- **Netbeans (nbpython):** Nbpython es una extensión de Netbeans que permite utilizar esta IDE como entorno para desarrollos en Python, algunas de sus características son el resaltado de sintaxis, compleción de código, soporte para proyectos Python soporte para jython, soporte para pyunit, debugger, administración de versiones, manejo de la librería estándar, ejecución de scripts Python etc.
- **Eric:** Es una IDE para Python y ruby, escrita en Python. Está basada en Qt, integrando el control de edición de cintilla. Está diseñado para ser usable ya sea como editor para pequeños scripts como así también para administración de proyectos profesionales. Eric provee un sistema de plugins que permite ser extendido fácilmente.
- **BeeWare:** Es algo más que una herramienta, de hecho, es una caja de herramientas de ayuda para desarrollar y depurar software en Python. La gran diferencia de BeeWare con un IDE (entorno de desarrollo integrado) es que cada herramienta de esa caja puede usarse de forma

independiente del resto. Cada una puede usarse para hacer pequeñas tareas y todas a la vez por separado sirven para poner en marcha grandes proyectos en Python”

2.9.2 Características de PYTHON

El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, <http://www.python.org/>, y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables. (van Rossum, 2009)

2.9.3 Versiones

Se detallarán a continuación de manera general las más relevantes:

- Versión 0.09 (Febrero de 1991)
- Versión 1.0 (Enero de 1994)
- Versión 1.2 (1995)
- Versión 1.4 (1995, Características nuevas)
- Versión 2.0 (Octubre de 2000)
- Versión 2.1 (2000)
- Versión 2.2 (2001)
- Versión 2.2 (2001)
- Versión 2.6 (2009)
- Versión 3.0 (2010)
- Versión 3.6 (2016)

2.9.4 Sockets en Python

“El socket es un módulo escrito en C (`_socket.so`, `_socket.pyd`, entre otras, dependiendo de la plataforma) con una interfaz de alto nivel desarrollada en Python (`socket.py`). Permite crear clientes y servidores para intercambiar información entre dos o más ordenadores.

La arquitectura y funcionamiento de los sockets en general es un tema aparte, mucho más amplio y complejo, que requeriría de varios artículos y está por fuera de los objetivos de este artículo, que pretende mostrar el funcionamiento básico del módulo en Python”, detallado según (Recursos Python, 2014)

2.9.5 Comparativa entre Java y Python

De manera general se realiza la siguiente tabla 2-2, en la que se realiza una comparativa de los temas más relevante de entre estos dos lenguajes de programación, de modo que según (Gabriel Cuzman):

Tabla 2.2: Comparativa de los lenguajes Java y Python

COMPARATIVA	JAVA	PYTHON
Tipado dinámico vs tipado estático	Tipado Estático. Java le obliga a indicar el tipo de una variable cuando la declara por primera vez y no le permitirá cambiar el tipo más tarde en el programa.	Python usa el Tipado Dinámico, que le permite cambiar el tipo de una variable, reemplazando un entero por una cadena, por ejemplo
Llaves vs Indentación	Java, como la mayoría de otros lenguajes, usa las llaves para definir el principio y el final de cada función o definición de clase.	Python es un poco raro en cuanto que usa indentación para separar el código en bloques. La ventaja de usar indentación es que le obliga a construir su programa en una forma que es fácil de leer, y no habrá ningún error resultante de que falte una llave.
Velocidad vs Portabilidad	Java puede ser usado para crear aplicaciones independientes de la plataforma. Cualquier ordenador o dispositivo móvil que pueda ejecutar una máquina virtual de Java puede ejecutar una aplicación Java Gracias a la popularidad de Java para aplicaciones web y aplicaciones simples de escritorio, la mayoría de los dispositivos ya tienen una máquina virtual de Java instalada El inconveniente de ejecutarse dentro de una máquina virtual es que el programa Java se ejecuta más lentamente que los programas en Python.	Para ejecutar programas Python necesita un compilador que pueda convertir el código en Python a código que su sistema operativo en particular pueda entender.

Fuente: *Herramientas de Desarrollo en Informática. Python*

2.10 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 ejecución en una PC de un conjunto programas de prueba o evaluación (*benchmarks*).

De entre todas las magnitudes medibles de un sistema informático susceptible de ser utilizadas como índices de prestaciones, el *tiempo* en llevar a cabo una actividad determinada representa la más intuitiva y, desde el punto de vista de la manipulación matemática, la menos susceptible de incorporar falsedad.

La corporación **SPEC** (Estándar Performance Evaluation Corporation) propone índices de carácter general, SPECmarks y cuyo significado depende del aspecto en concreto que se esté evaluando. Algunos de estos índices tienen en cuenta *el tiempo de ejecución* de un conjunto de programas y utilizan además algún tipo de normalización y cálculo de medidas para reducir el rendimiento a un único indicador. El tiempo de ejecución de un programa representa la medida exacta del rendimiento de un computador aquel que ejecute la misma cantidad de trabajo en menor tiempo posible será el más rápido (Molero, X.; Ruiz, C.; Rodeño, M., 2004).

2.10.1 Tasa de Transferencia

De manera general este término es utilizado en los sistemas de Red o en la red mismo, por lo que la cantidad de información o datos transmitida aumenta o disminuye dependiendo de la cantidad de usuarios en dicha red. Para poder medir la cantidad de información transmitida se hace uso de dos variantes que son la latencia y la tasa de trasmisión.

2.10.2 Definición de Tasa de Transferencia

También conocida como velocidad de datos o del caudal, es la velocidad a la que los datos se transfieren dentro de un ordenador, entre un ordenador y un dispositivo periférico, o entre redes. (seabrookewindows.com, 2018).

2.10.3 Definición de Tasa de Latencia

Según (Clouding.io, 2017), la “Latencia es el tiempo que tardan los paquetes IP en llegar desde el servidor VPS hasta nuestro ordenador o dispositivo. Representa la diferencia entre ver un dato en tiempo real o con retraso”.

2.10.4 Cálculo de Transferencia de Datos

Según (seabrookewindows.com, 2018), “El cálculo de una tasa de transferencia de datos es simplemente definir el número de bits transferidos en un período determinado de tiempo. Dependiendo de la magnitud del problema y la velocidad de la conexión, la velocidad de transferencia podría ser gigabytes de datos por segundo para una conexión rápida como la fibra óptica o unos pocos miles de bytes por segundo a través de acceso telefónico. Saber qué interfaz se está calculando le ayudará a obtener una respuesta significativa sin necesidad de convertir su respuesta a una escala mayor o menor.

Instrucciones

- Determinar la escala es necesario utilizar para su respuesta. A partir de este punto, determinar si es necesario calcular en una escala de bytes, kilobytes, megabytes o gigabytes.
- Convierte el tamaño del archivo de datos utilizado en su prueba al número de bytes, kilobytes, megabytes o gigabytes. Un kilobyte es de 1024 bytes, un megabyte es de 1024 kilobytes y un gigabyte es de 1024 megabytes. No se limite a multiplicar por 1000 por cada paso hacia arriba.
- Transferir su archivo a través de la interfaz que se está probando y medir la cantidad de tiempo que se necesita para transferir el archivo.
- Divida el tamaño del archivo por el número de segundos que se tarda en transferir el archivo. Esto le dará la velocidad de transferencia de datos, que es el número de bytes por segundo”.

A continuación, se representa gráficamente una imagen de representación de transferencia de datos y ancho de banda (Ver figura 7-2):

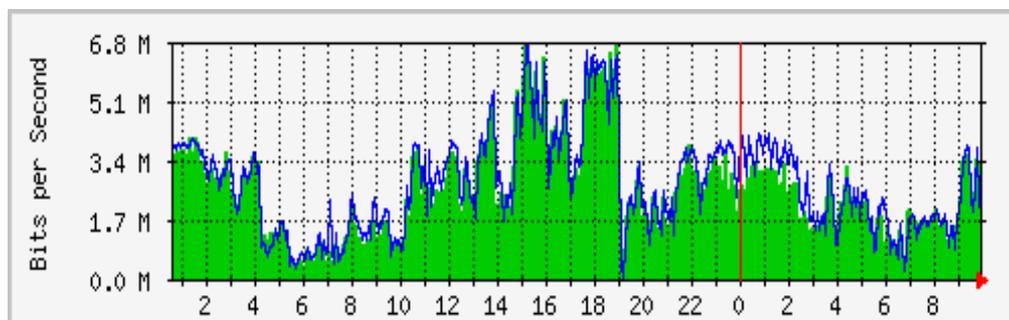


Figura 7-2: Transferencia de datos

Fuente: <https://hostingdiario.com/que-es-la-transferencia-de-datos-y-el-ancho-de-banda/>

De tal manera, que mediante la gráfica 8-2, se puede observar el intervalo y la tasa de transferencia dada en bits por segundos, con lo cual se tiene una breve idea de cómo se da en periodos de tiempos determinados, según (seabrookewindows.com, 2018)

CAPÍTULO III

3. DISEÑO DE LA INVESTIGACIÓN

3.1 Tipo y diseño de investigación

3.1.1 *Tipo de investigación*

La presente investigación se clasifica en dos tipos: Aplicativa y cuasi-experimental.

- ***Aplicativa:*** ya que se basó en conocimientos existentes, derivados de investigaciones previas, dirigidas y aplicados al desarrollo tecnológico del sistema de videovigilancia basado en IoT, en la Dirección Distrital 07D04 Balsas – Marcabelí – Piñas del Ministerio de Educación, para la supervisión de procesos administrativos de la institución, lo cual contribuyó en las tareas de seguridad de la institución, donde laboran diariamente treinta y dos funcionarios distribuidos en trece departamentos.
- ***Experimental:*** ya que estableció escenarios de prueba donde se realizaron las pruebas de rendimiento con ambos lenguajes de programación Java y Python, para al final determinar la plataforma con mejores resultados.

3.1.2 *Diseño de investigación*

El presente trabajo de investigación es de tipo *Cuantitativo*, en el cual se propuso diseñar un sistema de comunicación desarrollado en los lenguajes de programación Java y Python, y probar su rendimiento en el proceso de transferencia de imágenes utilizando una placa reducida para la implementación de un sistema de video vigilancia, lo cual posteriormente se evaluó mediante métodos estadísticos y comparativos.

3.2 Métodos de investigación

3.2.1 Método Experimental y de Observación

En el presente trabajo se utilizaron los siguientes métodos de investigación:

- **Método Científico:** Sirvió para recolectar la información necesaria y adecuada para aplicarla en los escenarios de pruebas, ya que las ideas, conceptos y teorías expuestas en el presente trabajo investigativo son verificables como válidos.
- **Método Deductivo:** Debido que al estudiar en forma general los diferentes tipos de cifrado de datos, manejo de socket y herramientas de medición de rendimiento, se trató de encontrar el más adecuado que contenga las mejores características para la seguridad y envío de la información.
- **Método Comparativo:** Después del estudio general se comparó cada uno de los tipos de cifrado de datos, manejo de socket y herramientas de medición de rendimiento a estudiarlos.

3.3 Enfoque de la investigación

El presente trabajo investigativo tuvo un enfoque cualitativo cuantitativo, para establecer indicadores que permitieron medir a través de valores adjetivos y valores numéricos los resultados obtenidos, para posteriormente ser analizados con métodos estadísticos.

3.4 Alcance investigativo

El estudio tuvo un alcance de investigación explicativo debido que con el análisis de las causas y efectos de la relación entre las variables identificadas llevaron a una conclusión.

3.5 Población de estudio

La población en la presente investigación la conformaron la cantidad de transferencia de imágenes que se realizaron 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, donde se obtuvo una población de 14.400 imágenes a transferir.

3.6 Selección de la muestra

Para encontrar la muestra se aplicó 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%)

$q =$ 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 obtuvo como muestra para el estudio un total de 375 transferencias de imágenes en cada escenario.

3.7 Técnica de recolección de datos primarios y secundarios.

Las técnicas que se utilizaron en la presente investigación fueron:

- *Búsqueda de información:* permitió obtener la información necesaria acerca del objeto de estudio de la investigación para su desarrollo, utilizando las fuentes secundarias disponibles.
- *Pruebas:* permitió realizar las pruebas para analizar el rendimiento de las 2 aplicaciones del sistema.
- *Observación:* permitió determinar resultados de las pruebas realizadas en ambos lenguajes de programación Java y Python.

- *Análisis*: permitió determinar los resultados de la investigación.

Las fuentes que se tomaron como base para esta investigación fueron:

- **Primarias**

- Papers y Revistas científicas
- Investigaciones realizadas
- Libros

- **Secundarias**

- Observaciones
- Textos

3.8 Variables e indicadores

En base a la hipótesis de investigación planteada en el presente trabajo la cual dicta: *Existe una diferencia significativa del rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas (IoT) entre los lenguajes de programación Java y Python.*, se determinaron las siguientes variables:

- **Variable Independiente**

- Lenguaje de programación Java y Python

- **Variable Dependiente**

- Rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas (IoT)

3.9 Operacionalización de variables

3.9.1 Conceptualización de las variables

Teniendo la variable dependiente e independiente se realizó la siguiente generalización de conceptos para desarrollar un sistema de videovigilancia en las plataformas Java o Python y evaluar el rendimiento de ambas en la transferencia de imágenes.

Tabla 3-3: Conceptualización de las variables

VARIABLES	TIPO	CONCEPTO
Lenguaje de programación utilizado (Java o Python)	Independiente	Lenguaje artificial que se utiliza para expresar programas de ordenador (EcuRed, 2018)
Rendimiento de la transferencia de imágenes en sistemas de Videovigilancia basada en Internet de las Cosas (IoT)	Dependiente	Parámetro para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También para validar y verificar otros atributos de la calidad del sistema, tales como el uso de recursos

Realizado por: Paucar Rosman, 2018

Tabla 4-3: Operacionalización de variables

VARIABLES	INDICADORES	INDICES
Independiente Lenguaje de programación utilizado (Java o Python)	• Lenguajes	• Lenguaje de programación Java y Python • Número de imágenes • Tamaño de la imagen
Dependiente Rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas (IoT).	• Métricas Internas	• Tiempo de respuesta • Tiempo de procesamiento
	• Métricas Externas	%Uso de CPU %Uso de RAM %Ancho de Banda

Realizado por: Paucar Rosman, 2018

3.9.2 Matriz de consistencia

Al realizar la matriz de consistencia se establecieron los indicadores e índices que permitieron evaluar la variable tanto independiente y dependiente y obteniendo resultados a partir de las mediciones, este proceso se realizó mediante descripciones de tablas en donde se ingresó el objetivo general, la hipótesis, la variable y su tipo, los indicadores, técnicas e instrumentos de medición de la siguiente manera:

Tabla 5-3: Matriz de consistencia

Formulación del Problema	Objetivo General	Hipótesis General	Variables	Indicadores	Índices	Técnicas	Instrumentos
¿Cuál de los lenguajes de programación de software libre más popularizados y usados en la videovigilancia, Java o Python ofrece un mejor rendimiento de la transferencia de imágenes en un sistema de videovigilancia basada en Internet de las Cosas?	Comparar de rendimiento de la transferencia de imágenes en un sistema de videovigilancia basado en Internet de las Cosas entre los lenguajes de programación Java y Python.	Existe una diferencia significativa del rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas (IoT) entre los lenguajes de programación Java y Python.	Independiente Lenguaje de programación Java y Python	<ul style="list-style-type: none"> • Tipo • Imágenes • Cantidad 	<ul style="list-style-type: none"> • Lenguaje de programación Java y Python • Número de imágenes • Tamaño de la imagen 	<ul style="list-style-type: none"> • Observación 	<ul style="list-style-type: none"> • Netbeans • Sublime text • Hojas de Calculo • Registro de Eventos
			Dependiente Rendimiento de la transferencia de imágenes en sistemas de videovigilancia basada en Internet de las Cosas (IoT)	<ul style="list-style-type: none"> • Métricas Internas • Métricas externas 	<ul style="list-style-type: none"> • Tiempo de respuesta • Tiempo de procesamiento • % Uso de CPU • % Uso de RAM • % Ancho de Banda 	<ul style="list-style-type: none"> • Observación • Análisis • Test • Medición de pruebas 	<ul style="list-style-type: none"> • WireShark, • Network Analyser • Free VMstat • Top Iostat • Sar • Registro de Eventos

Realizado por: Paucar Rosman, 2018

La operacionalización de variables se orientó de acuerdo la evaluación propuesta por (Medina, 2014) en su artículo “Pruebas de rendimiento” donde establece métricas internas y externas.

3.10 Procesamiento y Análisis

3.10.1 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.

3.10.2 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 Decisión 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.

3.11 Parámetros de Evaluación

La evaluación del rendimiento de sistemas informáticos propuesta por (Medina, 2014) en su artículo y libro “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 6-3.

Tabla 6-3: Ponderación de parámetros.

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

Realizado por: Paucar Rosman, 2018

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 (Ghodsi, 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 estableció que uno u otro indicador tenga más peso dado el escenario y las condiciones de pruebas donde se llevó a cabo el experimento es el mismo, ya que por un lado las métricas internas como las externas son de elevada importancia, la primera por los tiempos que se utilizan 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 utilizó un método discreto multicriterio con una matriz de impacto (o evaluación) que incluyó medidas nítidas, estocásticas o difusas del rendimiento de una alternativa con respecto a un criterio de juicio.

3.11.1 Métricas internas

3.11.1.1 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.

3.11.1.2 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.

La diferencia entre tiempo de procesamiento y de tiempo de respuesta se puede representar en la figura 8-3, donde se puede evidenciar que el tiempo de procesamiento es el que involucra el trabajo de captura de imagen y generación de video, mientras que el tiempo de respuesta es el tiempo que transcurre desde que se solicita la imagen hasta que se recibe la misma por parte del servidor.



Figura 8-3: Diferencia entre Tiempo de Procesamiento y Respuesta

Fuente: <http://es.aliexpress.com>

De modo que, el tiempo de procesamiento se refiere al tiempo que se necesita para completar la acción de captura y procesamiento de la imagen para el video, mientras que el tiempo de respuesta al tiempo en que se solicita hasta que se recibe la imagen y está directamente relacionado con la experiencia del usuario.

3.11.2 Métricas externas (Consumo de Recursos)

Para el presente trabajo, el consumo de recursos se compuso 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.

3.11.2.1 Uso de CPU

Para (Heurung, s.f.), el uso de CPU es una medida sobre cuánto procesador general se utiliza en un momento dado. Siendo ahora el estándar los procesadores de varios núcleos, se puede crear un poco de confusión relacionada con los totales. Los procesos se añaden a una cola y después se alimentan a los múltiples núcleos de la CPU para ser procesados en paralelo. Cada núcleo procesa de forma independiente los datos que se le envían, los resultados después se promedian con el resto de núcleos de procesador y la salida general se puntúa desde 0% a 100%.

3.11.2.2 Uso de memoria RAM

La memoria RAM es un tipo de memoria volátil que el sistema utiliza para almacenar los datos que está utilizando durante la ejecución de un determinado programa. Es decir, que cuando se

ejecuta un programa en el ordenador, el sistema carga parte de la información que necesita para ejecutarlo en la memoria RAM, así como los nuevos datos que se generan mientras se realiza las tareas dentro de ese programa, por lo tanto, este indica el porcentaje de memoria que se está utilizando.

3.11.2.3 Ancho de Banda

Según (DEFINICIÓN.DE, s.f.), el concepto de ancho de banda se popularizó en las últimas décadas a partir de la masificación del uso de Internet. En la informática, se conoce como ancho de banda a la cantidad de datos que pueden enviarse y recibirse en el marco de una comunicación. Dicho ancho de banda suele expresarse en bits por segundo o en múltiplos de esta unidad.

3.12 Herramientas

Las pruebas de rendimiento tanto en las métricas internas como externas se ejecutaron con herramientas que en la actualidad son ampliamente utilizadas para recolectar información, así como para procesar dicha información recolectada, mismas que se detallaran en el trascurso de este apartado.

3.12.1 Herramientas de recolección de información

Los datos obtenidos de los experimentos realizados fueron llevados a cabo por herramientas que vienen por defecto en los repositorios del Sistema Operativo Linux, para el presente caso el sistema Raspbian. A continuación, se estudia cada una de ellas:

- **PROTOTIPOS**

Los datos de las métricas internas como, el tiempo de respuesta y el tiempo de procesamiento, fueron capturados mediante programación en cada prototipo, en el cual se capturó el tiempo en milisegundos, para luego realizar las restas respectivas y encontrar los tiempos correspondientes, a continuación, se observa las líneas de código utilizada para capturar el tiempo en milisegundos.

Código Python

```
tiempo_1=int(round(time.time() * 1000))
```

```
tiempo1 = new java.util.Date().getTime();
```

- **PS (PROCESS STATUS)**

Según el sitio (ORACLE, 2011) “El comando ps permite comprobar el estado de los procesos activos en un sistema y mostrar información técnica sobre los procesos. Estos datos son útiles para tareas administrativas, como la determinación de la manera de definir las prioridades del proceso. Según las opciones utilizadas, el comando ps proporciona la siguiente información:

- Estado actual del proceso
- ID de proceso
- ID de proceso principal
- ID de usuario
- Clase de programación
- Prioridad
- Dirección del proceso
- Memoria utilizada
- Tiempo de CPU utilizado”

En la figura 9-2 se observa el resultado de la ejecución del comando ps.

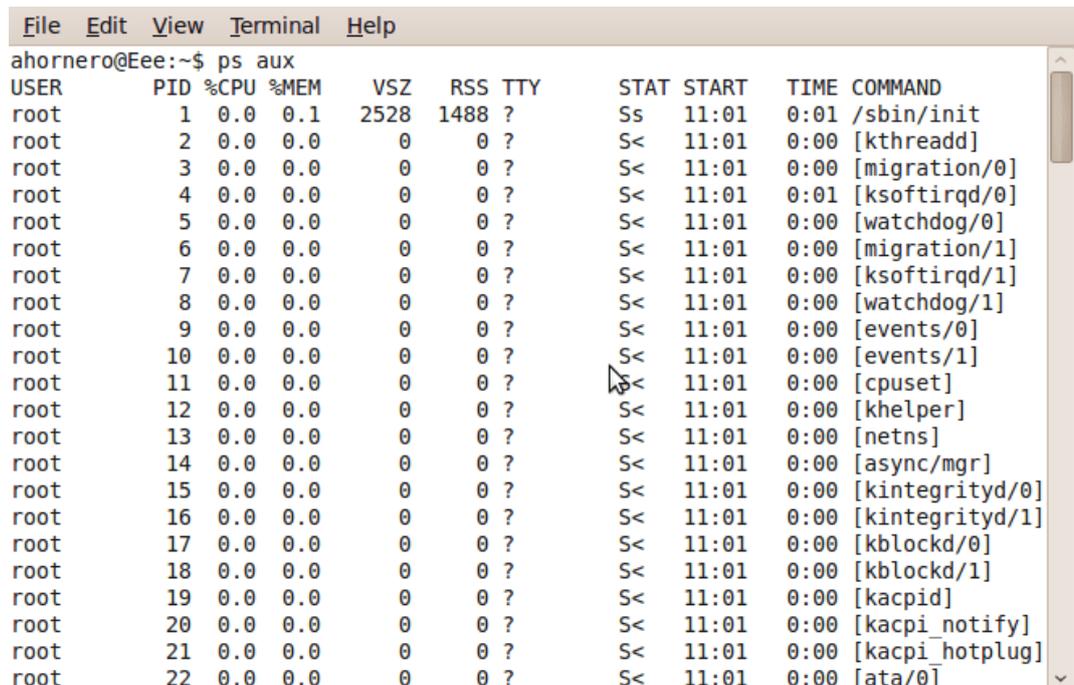


Figura 9-3: Ejecución del comando ps

Fuente: <http://www.linuxhispano.net/2009/12/05/el-comando-ps/>

De tal manera, que el comando ps fue una herramienta de gran uso para poder obtener las diferentes mediciones de las métricas internas en los prototipos desarrollados.

- **IFTOP**

Escucha el tráfico de red en una interfaz de red especificada o en la primera interfaz encontrada, y despliega una tabla con el uso actual del ancho de banda por dos estaciones.

iftop debe ser ejecutado con permisos suficientes para supervisar todo el tráfico de red en la interfaz; es decir debe ser ejecutado como root.

De forma predeterminada, iftop cuenta todos los paquetes IP que pasan por el filtro, y la dirección del paquete se determina de acuerdo con la dirección en que el paquete se mueve a través de la interfaz. Usando la opción -F es posible obtener iftop para mostrar los paquetes que entran y salen de una red dada.

Por ejemplo:

```
# iftop -F 192.0.0.0/255.0.0.0
```

Con iftop se analiza los paquetes que entran y salen de la red 192.168.1.*, a emplearse en los prototipos, un ejemplo de su ejecución y aplicación se la puede observar en la figura 10-3 de la red en producción

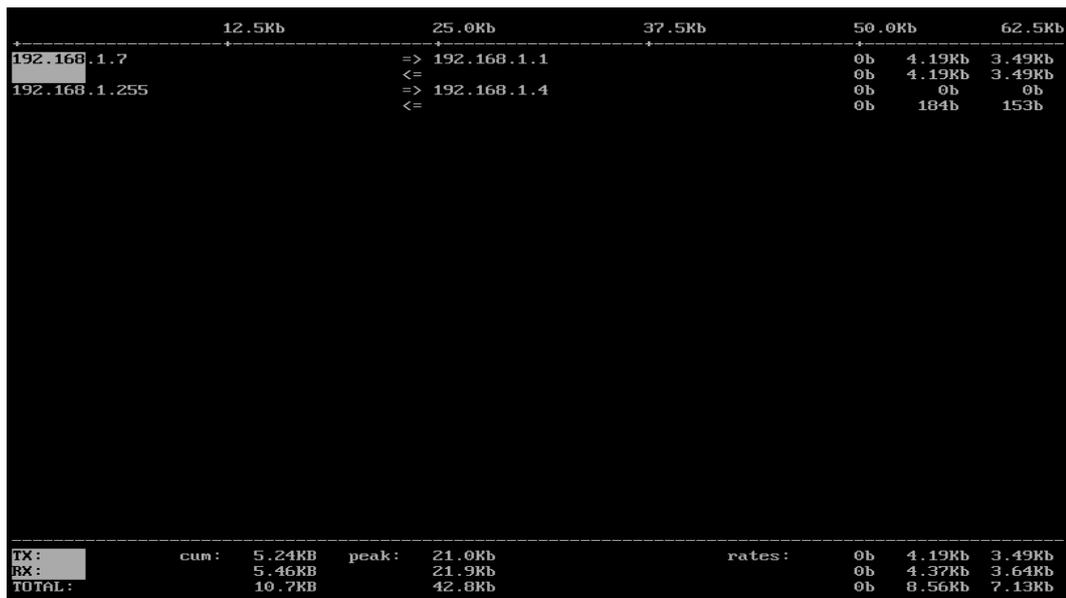


Figura 10-3: Herramienta iftop

Realizado por: Paucar Rosman, 2018

- **WATCH**

Según (man7, 2018) el comando “watch”, “ejecuta el comando repetidamente, mostrando su salida y errores. Esto le permite ver la salida del programa cambiar con el tiempo. Por defecto, el comando se ejecuta cada 2 segundos y el reloj se ejecutará hasta que se interrumpa”.

Este comando es integrado en conjunto con el comando ps para obtener la información tanto de CPU como de RAM, lo cual facilitó la obtención de los valores de las métricas externas utilizadas en los dos prototipos a desarrollarse.

3.12.2 Para tabulación y Análisis Estadísticos

Una vez recolectada la información esta fue tabulada, “limpiada” y posteriormente analizada, utilizando métodos estadísticos que permitieron validar los resultados obtenidos, en este trabajo se utilizaron las siguientes herramientas y utilitarios:

- **Hoja de Cálculo Excel**

Una hoja de cálculo es una aplicación, que permite manipular datos numéricos y alfanuméricos dispuestos en forma de tablas compuestas por celdas. La celda es la unidad básica de información en la hoja de cálculo, donde se insertan los valores y las fórmulas que realizan los cálculos (Rubert, s.f.).

Según (López Noriega, Lagunes Huerta, & Herrera Sánchez, 2006), “Desde sus inicios las Hojas de Cálculo lograron un éxito rotundo debido a que su uso en los problemas cotidianos y reales en el mundo empresarial representaba una oportunidad de llevar esos problemas reales y representarlos a través de números y fórmulas”. En el presente trabajo se utilizó con el fin de tabular los datos y realizar una “limpieza” de los mismos. Se hizo uso de hojas de cálculo para realizar mediciones y comparaciones de medias, medianas, desviaciones estándar y cálculos estadísticos de los resultados de cada prototipo.

- **SIAE**

Software desarrollado por los doctores Narciza Salazar y Alonso Álvares, que permite realizar el análisis estadístico de datos a través del ingreso de información como muestra, desviación estándar, número de la población o muestra y % de error. El resultado se muestra

en una gráfica de la Campana de Gauss como se observa en la figura 3-4, este software se utilizó para graficar los resultados de los análisis estadísticos para cada indicador. El SIAE sirvió para realizar el cálculo estadístico de cada indicador correspondiente a las métricas internas y externas y en empleo del método estadístico de la “Prueba Z”.

A continuación, se muestra una figura de cómo más adelante se representan gráficamente estas comparaciones:

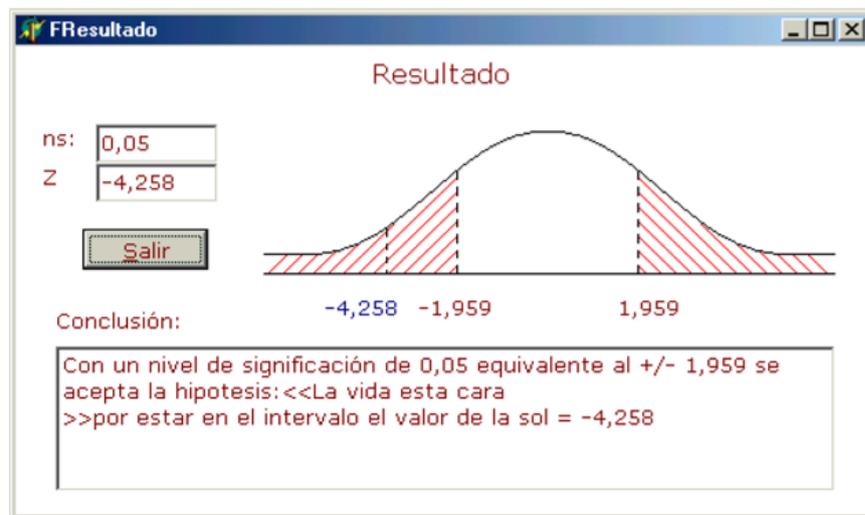


Figura 11-3: Resultado Software SIAE

Realizado por: Paucar Rosman, 2018

3.12.3 Para la comprobación de la Hipótesis

Se hizo uso de la herramienta llamada NAIADE, la cual es un software para análisis multicriterio (MCA) que fue desarrollado por Giuseppe Munda, en cooperación entre la Universidad Autónoma de Barcelona y el Centro Común de Investigación de la CE (en Ispra).

Es un método discreto multicriterio cuya matriz de impacto (o evaluación) puede incluir medidas nítidas, estocásticas o difusas del rendimiento de una alternativa con respecto a un criterio de juicio, por lo que es muy flexible para las aplicaciones del mundo real. Esto lo hace particularmente útil para estudios donde se incluyen parámetros con varios grados de precisión de las variables consideradas. En la figura 12-3 se observa la pantalla principal del software.

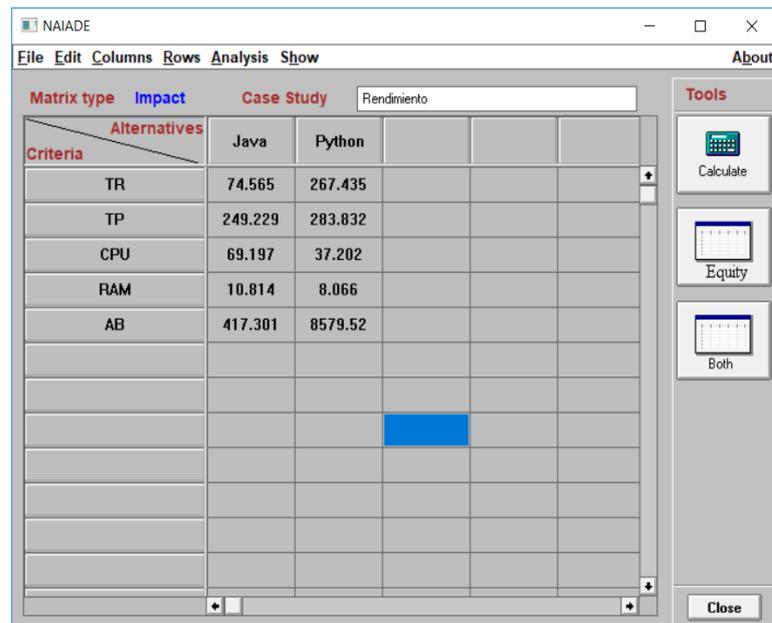


Figura 12-3: Pantalla Principal software NIAIDE

Realizado por: Paucar Rosman, 2018

3.12.4 Para gestión del proyecto

Para la gestión del proyecto se utilizó el software Project que según (WIKIPEDIA, 2018), “es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.”

3.13 Planteamiento de fórmulas.

Las fórmulas que se utilizaron para las comparaciones estadísticas de cada indicador fueron: media, mediana y desviación estándar, definidas por (Matus), para los que se desarrolló la siguiente tabla de recolección de información:

Tabla 7-3: Estadísticas de tiempos entre Java y PYTHON

Estadísticas	Java	PYTHON
Media (ms)		
Mediana (ms)		
Desviación Estándar (ms)		

Realizado por: Paucar Rosman, 2018

3.13.1 *Media*

La medida de localización más importante es la media, o valor promedio, de una variable. La media proporciona una medida de localización central de los datos. Si los datos son datos de una muestra, la media se denota; si los datos son datos de una población, la media se denota con la letra griega μ . (Ver Fórmula 1)

Fórmula 1: Promedio o Media

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$$

3.13.2 *Mediana*

Según (Anderson, Sweeney, & Williams, 2008) La mediana es otra medida de localización central. Es el valor de en medio en los datos ordenados de menor a mayor (en forma ascendente). Cuando tiene un número impar de observaciones, la mediana es el valor de en medio. Cuando la cantidad de observaciones es par, no hay un número en medio. En este caso, se sigue una convención y la mediana es definida como el promedio de las dos observaciones de en medio.

Fórmula 2: Mediana

$$Me = L_{i-1} + \frac{\frac{N}{2} - N_{i-1}}{n_i} a_i$$

3.13.3 *Desviación estándar*

La desviación estándar es la medida más útil de la variación de los datos. En los ejemplos hemos podido observar que la dispersión de un conjunto de datos es pequeña si los datos se reúnen muy cerca de la media y es grande si estos se dispersan ampliamente en torno de la media.

Fórmula 3: Desviación Estándar

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n}}$$

3.13.4 *Diferencia Porcentual*

La diferencia porcentual entre un valor y otro se calcula dividiendo el valor mayor sobre el menor (Molero, Juiz, & Rodeño, 2004).

Fórmula 4: Variación Porcentual

$$VP = \frac{V2}{V1}$$

Donde:

V1 = Representa el valor menor

V2 = Representa el valor mayor

3.13.5 Análisis Multicriterio Discreto

El problema de multicriterio discreto se puede describir de la siguiente manera: A es un conjunto finito de N acciones factibles (o alternativas); M es el número de diferentes puntos de vista o criterios de evaluación g_m , $i=1,2, \dots, M$ considerados relevantes en un problema específico, donde se evalúa que la acción a es mejor que la acción b (ambas pertenecientes al conjunto A) de acuerdo con el m-ésimo punto de vista si $g_m(a) > g_m(b)$. De esta forma, un problema de decisión puede representarse en forma de tabla o matriz (como el presentado en la Tabla 8-3). La principal ventaja de los modelos multicriterios, es que permiten considerar una gran cantidad de datos, relaciones y objetivos que generalmente están presentes en un problema específico del mundo real, de modo que el problema en cuestión se puede estudiar de forma multidimensional. Por otro lado, una acción a puede ser mejor que una acción b según un criterio y peor según otra, por lo tanto, en general, no existe una solución que optimice todos los criterios simultáneamente y, por lo tanto, se deben encontrar soluciones de compromiso (Munda, 2006).

Tabla 8-I: Matriz de Impacto

Alternativa Criterio	Alternativa 1	Alternativa 2	Alternativa N
Criterio 1				
Criterio 2				
....				
Criterio M				

Fuente: (Munda, 2006)

El parámetro Alpha (α) es usado para expresar el requerimiento mínimo de credibilidad indexado. Solo aquellos criterios que están por encima del umbral α continuarán positivamente en la agregación. El índice de intensidad $J_i^*(a, b)$ de preferencia * (donde * se define por $\gg, >, \sim, =, <$ y \ll) de la alternativa **a** versus **b** es definida de la siguiente forma (Villacís Cruz, 2005):

Fórmula 5: Índice de Intensidad

$$\mu^*(a, b) = \frac{\sum_{m=1}^M \max(\mu^*(a, b)_m - \alpha, 0)}{\sum_{m=1}^M |\mu^*(a, b)_m - \alpha|}$$

3.14 Escenario

El escenario propuesto refleja el entorno en producción donde se ejecutó el sistema de videovigilancia, que es en la Dirección Distrital 07D04 Balsas – Marcabelí – Piñas del Ministerio de Educación, que 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 unió 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, match panels, ordenadores, etc., garantizando así la trasmisión normal por medio inalámbrico, debido a que cuenta con oficinas separadas por modulares, paredes y techos falsos y un ancho de banda de 10 MB por fibra óptica de plan corporativo con compartición 1:1.

Para realizar las pruebas comparativas del rendimiento, fue diseñado un escenario en donde interactuaron dos procesos, un Servidor (Raspberry Pi) y un cliente (PC), los mismos que estuvieron conectados mediante una red Wireless (802.11ac) segmentada, con velocidad de 4MB, lo cual se puede observar gráficamente en la figura 13-3:

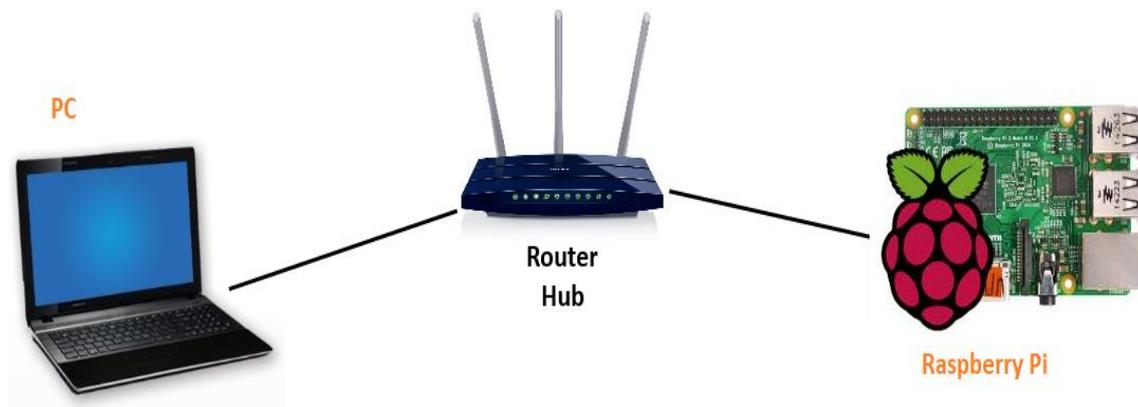


Figura 13-3: Escenario de pruebas

Realizado por: Paucar Rosman, 2018

3.14.1 *Prototipos*

Los prototipos desarrollados tanto en Java y Python fueron instalados en el Servidor Raspberry Pi, que 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 es el de la figura 14-3:



Figura 14-3: Prototipo de Servidor tanto en Java como en Python

Realizado por: Paucar Rosman, 2018

3.14.2 *Hardware cliente/servidor*

En la tabla 9-3 se detallan las características del hardware utilizado en el escenario planteado.

Tabla 9-3: Detalles Hardware utilizado para las pruebas

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

Realizado por: Paucar Rosman, 2018

3.14.3 Software utilizado

En la tabla 10-3 se muestra el software utilizado para el desarrollo de los dos prototipos Java y Python.

Tabla 10-3: Características Software

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

Realizado por: Paucar Rosman, 2018

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

Los datos obtenidos de cada indicador fueron recolectados utilizando las herramientas que se especificaron en el apartado 3.13.1, y que para mayor detalle se encuentran en el Anexo C1.

A continuación, se presentan los resultados consolidados.

4.1 Indicador 1: Métricas Interna

A continuación, se realiza el análisis correspondiente al indicador “Métricas Internas”, el mismo que se calculó a partir de los índices Tiempo de Respuesta y Tiempo de Procesamiento.

4.1.1 Índice 1.1: Tiempo de Respuesta

En la tabla 11-4 se detalla la media (\bar{X}) y desviación estándar (σ) de los datos obtenidos del tiempo de respuesta tanto para JAVA y PYTHON:

Tabla 11-4: Resultados de Tiempo de Respuesta

	Java (ms)		Python (ms)	
	\bar{X}	σ	\bar{X}	σ
Transmisión de Imagen (Servidor)	2.9867	3.014	74.829	5.9
Recibo y preparación de imagen (Cliente)	71.579	53.585	192.605	5.896
TOTAL	74.565	53.693	267.435	11.682

Realizado por: Paucar Rosman, 2018

En el gráfico 1-4, se visualiza de manera gráfica la media del tiempo de respuesta dado en milisegundos de cada proceso que se realizó para el envío de 375 imágenes tanto en Java como en Python.

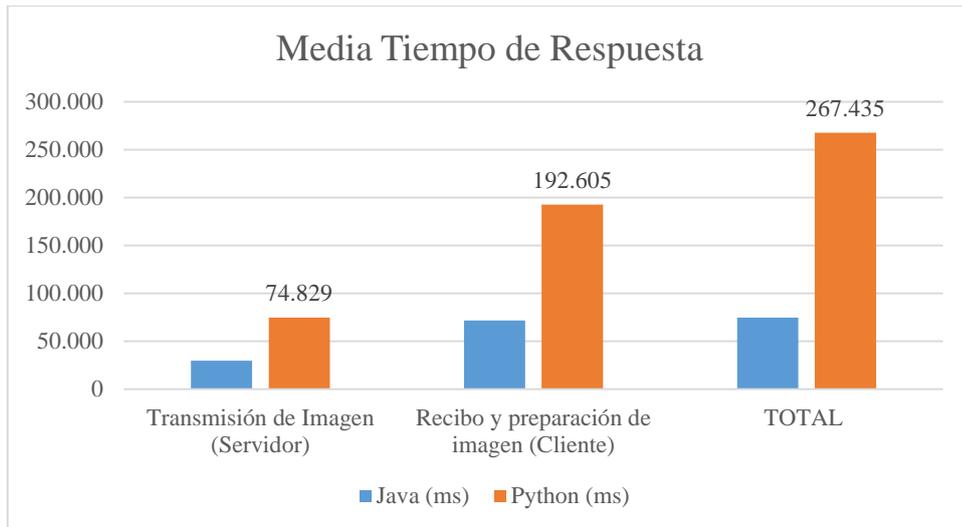


Gráfico 1-4: Tiempo de Respuesta

Realizado por: Paucar Rosman, 2018

Realizando un análisis de medias para encontrar si existió o no una diferencia significativa, utilizando el sistema SIAE, con un error estándar de 5%, mediante la prueba Z, se obtuvo que: “Si existe una diferencia significativa en el tiempo de respuesta entre Java y Python”, este resultado se puede observar en la figura 15-4.

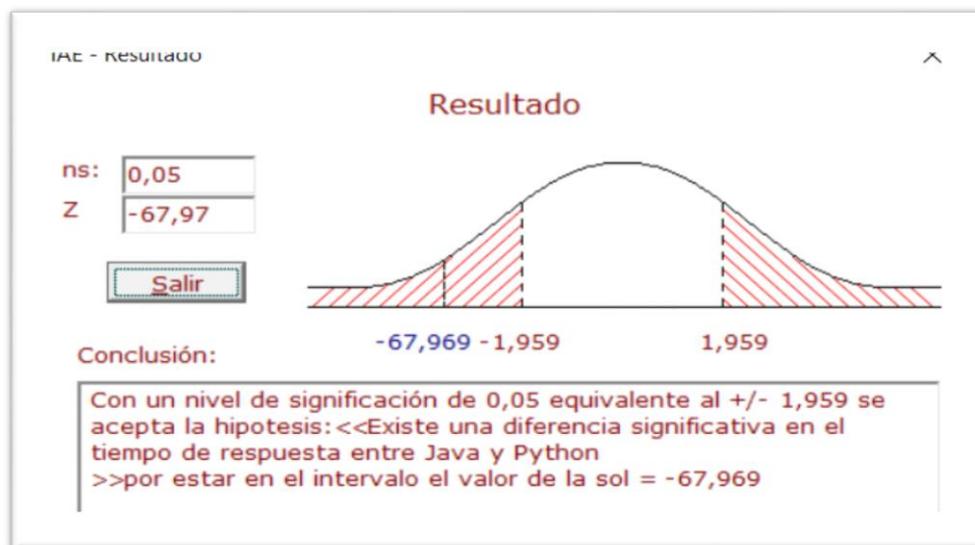


Figura 15-4: Análisis Estadístico Tiempo de Respuesta

Realizado por: Paucar Rosman, 2018

Conclusión:

Para el índice de tiempo de respuesta se concluye que, Java obtuvo un mejor tiempo de respuesta al ser menor que el que requiere Python, y aplicando la fórmula de diferencia porcentual del

apartado 3.14.4, se obtuvo que, en términos porcentuales Java es 3.586 más rápido que Python es decir unos 258.6%.

4.1.2 Índice 1.2: Tiempo de Procesamiento

En la tabla 12-4 se detalla la media (\bar{X}) y desviación estándar (σ) de los datos obtenidos del tiempo de procesamiento tanto para JAVA y PYTHON:

Tabla 12-4: Resultados de Tiempo de Procesamiento

	Java (ms)		Python (ms)	
	\bar{X}	σ	\bar{X}	σ
Tiempo de Procesamiento	249,229	86,060	283,832	11,731

Realizado por: Paucar Rosman, 2018

En el gráfico 2:4, se visualiza de manera gráfica la media del tiempo de procesamiento en milisegundos de cada proceso que se realizó para el envío de 375 imágenes tanto en Java como en Python.

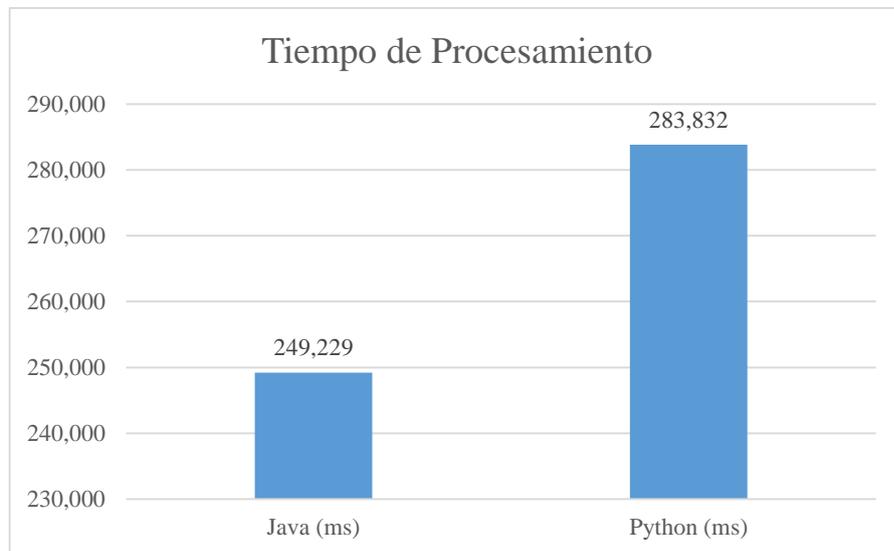


Gráfico 2:4: Tiempo de Procesamiento

Realizado por: Paucar Rosman, 2018

Realizando un análisis de medias para encontrar si existió o no una diferencia significativa, utilizando el sistema SIAE, con un error estándar de 5%, mediante la prueba Z, se obtuvo que: “Si existe una diferencia significativa en el tiempo de procesamiento Java y Python”, este resultado se puede observar en la figura 16-4.

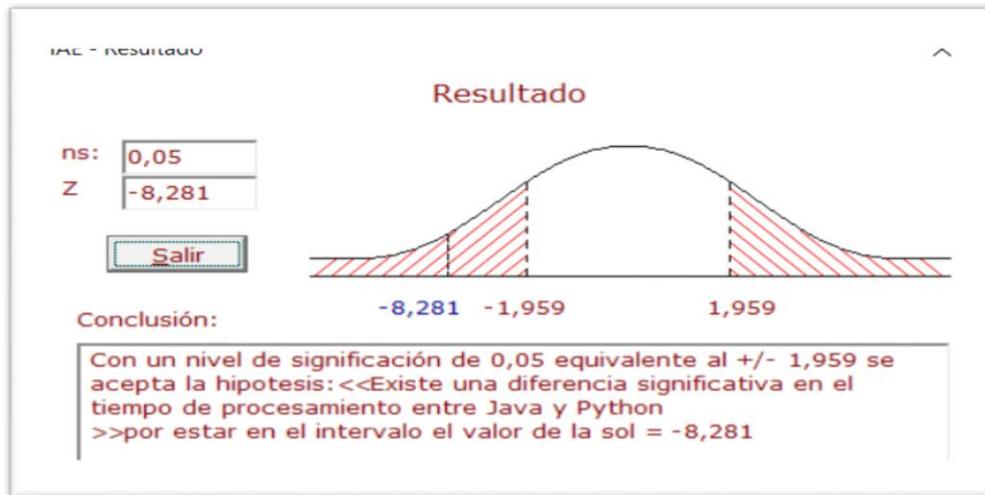


Figura 16-4: Análisis Estadístico Tiempo de Procesamiento

Realizado por: Paucar Rosman, 2018

Conclusión:

Para el índice de tiempo de respuesta se concluye que, Java obtuvo un mejor tiempo de procesamiento al ser menor que el que requiere Python, aplicando la fórmula de diferencia porcentual del apartado 3.14.4, se obtuvo que, en términos porcentuales Java es 1.1388 más rápido que Python es decir unos 13.88% más rápido.

4.2 Indicador 2: Métricas Externas

A continuación, se realiza el análisis correspondiente al indicador “Métricas Externas”, el mismo que se calculó a partir de los índices Porcentaje de Utilización del CPU, Porcentaje de Utilización de la Memoria RAM y Uso de Ancho de Banda.

4.2.1 Índice 2.1: Porcentaje de Uso de CPU

Con la utilización de las herramientas PS y WATCH, en la tabla 13-4 se detalla la media (\bar{X}) y desviación estándar (σ) de los datos obtenidos del porcentaje de uso del CPU tanto para JAVA y PYTHON:

Tabla 13-4: Porcentaje de Uso de CPU

	Java (%)		Python (%)	
	\bar{X}	σ	\bar{X}	Σ
%Uso de CPU	69,197	0,637	37,202	0,638

Realizado por: Paucar Rosman, 2018

En el gráfico 3-4, se visualiza de manera gráfica la media del porcentaje de uso de CPU de cada proceso que se realizó para el envío de 375 imágenes tanto en Java como en Python.

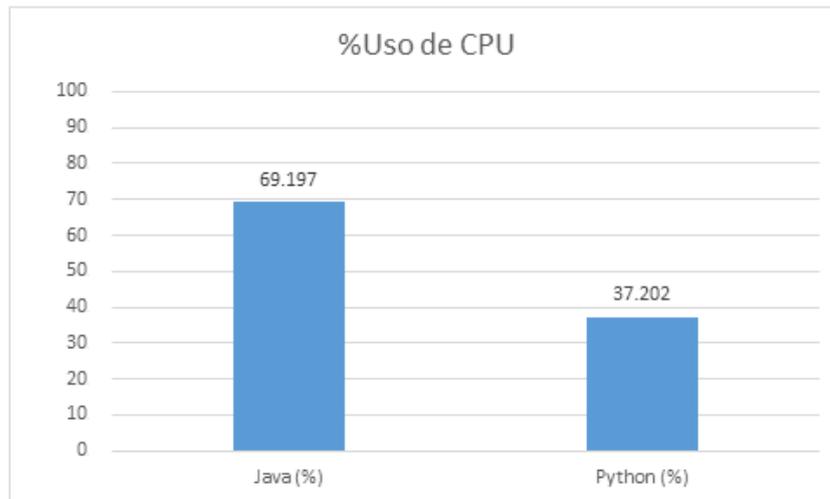


Gráfico 3-4: Porcentaje de Uso de CPU

Realizado por: Paucar Rosman, 2018

Realizando un análisis de medias para encontrar si existió o no una diferencia significativa, utilizando el sistema SIAE, con un error estándar de 5%, mediante la prueba Z, se obtuvo que: “Si existe una diferencia significativa en el porcentaje de uso de CPU entre Java y Python”, este resultado se puede observar en la figura 17-2.

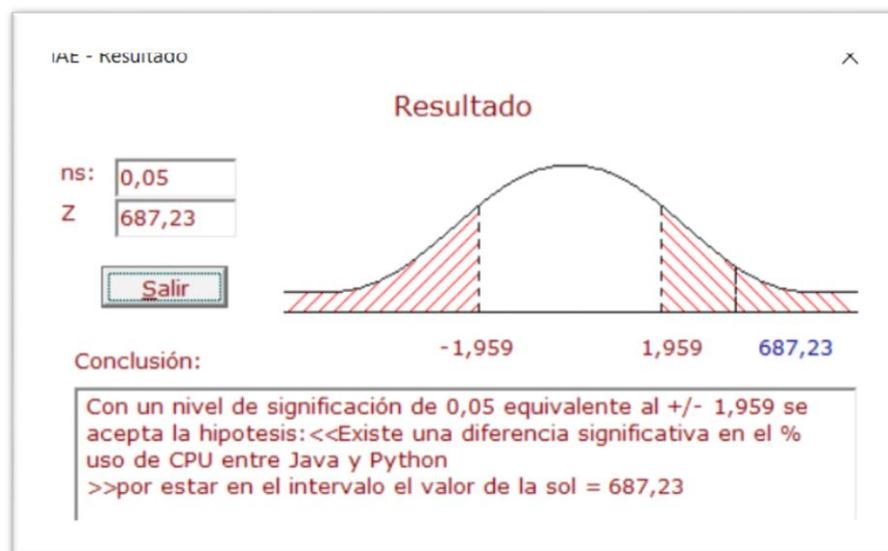


Figura 17-2: Análisis Estadístico % Uso de CPU

Realizado por: Paucar Rosman, 2018

Conclusión:

Para el índice del porcentaje de uso de CPU se concluyó que, Python obtuvo un mejor porcentaje de uso al ser menor que el que requiere Java, aplicando la fórmula de diferencia porcentual del apartado 3.14.4, se obtuvo que, en términos porcentuales Python utiliza un 1.86 menos porcentaje de procesador que Java es decir 86% menos.

4.2.2 Índice 2.1: Porcentaje de Uso de RAM

Con la utilización de las herramientas PS y WATCH, en la tabla 14-4 se detalla la media (\bar{X}) y desviación estándar (σ) de los datos obtenidos del porcentaje de uso del RAM tanto para JAVA y PYTHON:

Tabla 14-4: Porcentaje de Uso de RAM

	Java (%)		Python (%)	
	\bar{X}	σ	\bar{X}	σ
%Uso RAM	10,814	0,287	8,066	0,060

Realizado por: Paucar Rosman, 2018

En el gráfico 4-4, se visualiza de manera gráfica la media del porcentaje de uso de RAM de cada proceso que se realizó para el envío de 375 imágenes tanto en Java como en Python.

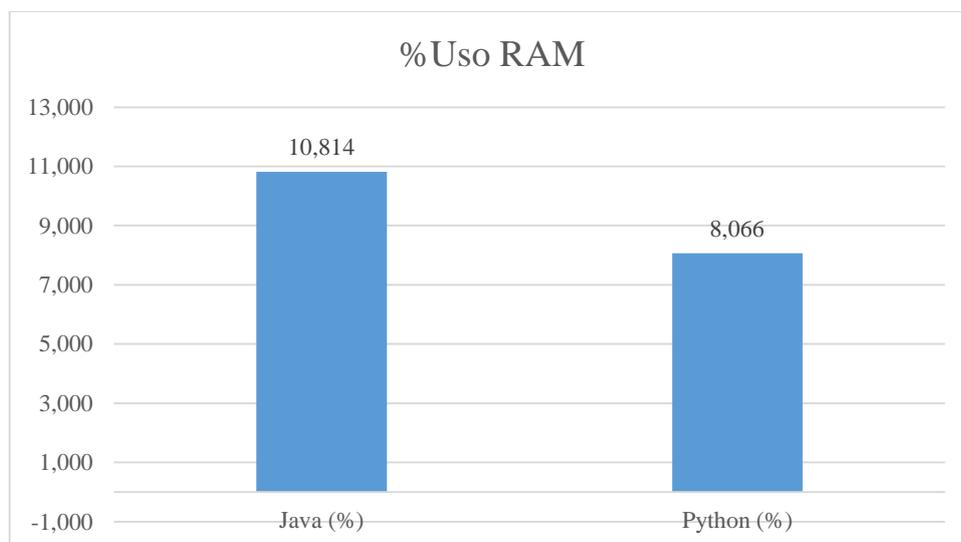


Gráfico 4:4: Porcentaje de Uso de RAM

Realizando un análisis de medias para encontrar si existió o no una diferencia significativa, utilizando el sistema SIAE, con un error estándar de 5%, mediante la prueba Z, se obtuvo que: “Si existe una diferencia significativa en el uso de RAM entre Java y Python”, este resultado se puede observar en la figura 18-4.

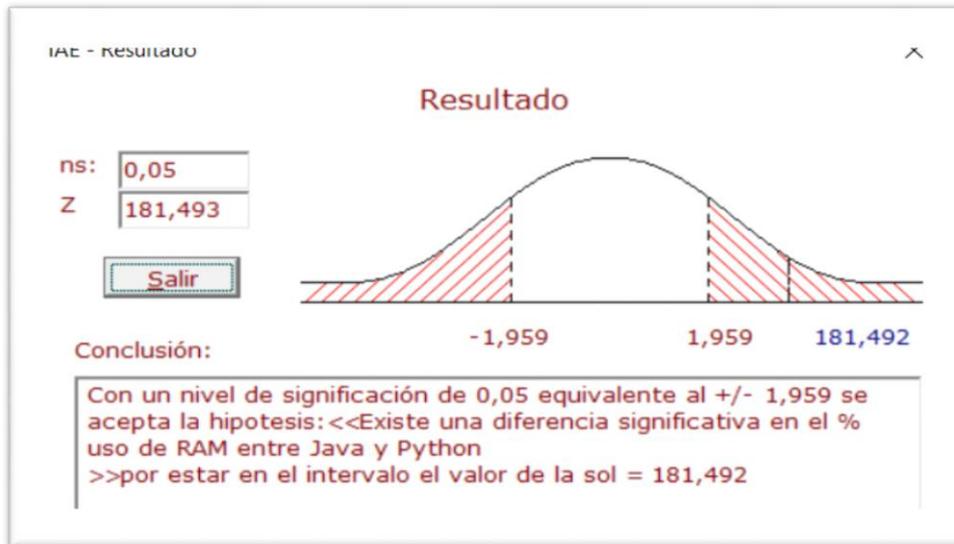


Figura 18-4: Análisis Estadístico % Uso de RAM

Realizado por: Paucar Rosman, 2018

Conclusión:

Para el índice de porcentaje de uso de RAM se concluyó que, Python obtuvo un mejor porcentaje de uso al ser menor que el que requiere Java, aplicando la fórmula de diferencia porcentual del apartado 3.14.4, se obtuvo que, en términos porcentuales Python utiliza un 1.34 menos porcentaje de uso de RAM que Java es decir unos 34% menos.

4.2.3 Índice 2.3: Ancho de Banda

Con la utilización de las herramientas IFTOP y WATCH, en la tabla 15-4 se detalla la media (\bar{X}) y desviación estándar (σ) de los datos obtenidos en Kilo Bytes en lo referente al Ancho de Banda tanto para JAVA y PYTHON.

Tabla 15-4: Ancho de Banda

	Java (KB)		Python (KB)	
	\bar{X}	σ	\bar{X}	σ

Ancho de Banda	417,301	202,276	8.579,520	1.834,499
-----------------------	----------------	----------------	------------------	------------------

Realizado por: Paucar Rosman 2018.

En el gráfico 5-4, se visualiza de manera gráfica la media del Ancho de Banda de cada proceso que se realizó para el envío de 375 imágenes tanto en Java como en Python.

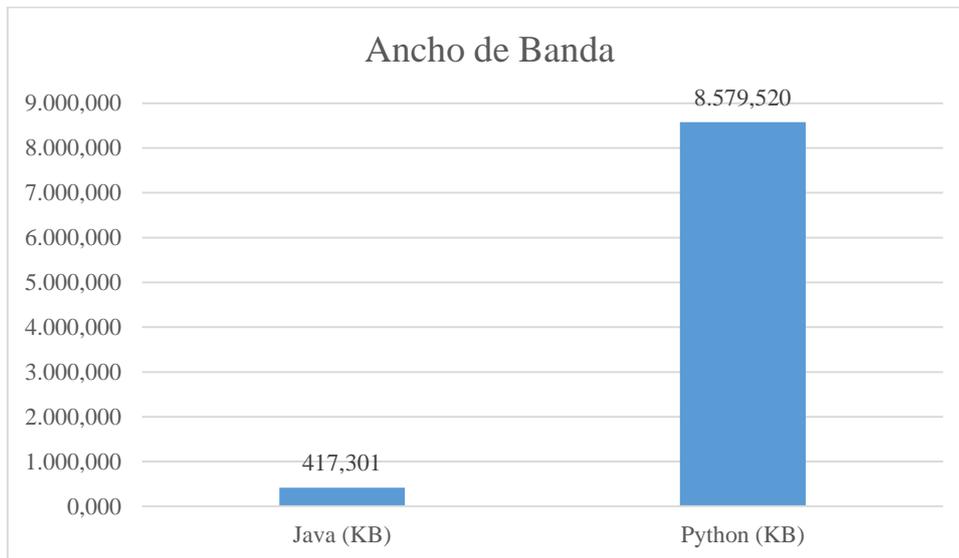


Gráfico 5-4: Ancho de Banda

Realizado por: Paucar Rosman, 2018

Realizando un análisis de medias para encontrar si existió o no una diferencia significativa, utilizando el sistema SIAE, con un error estándar de 5%, mediante la prueba Z, se obtuvo que: “Si existe una diferencia significativa en el Ancho de Banda entre Java y Python”, este resultado se puede observar en la figura 19-4.

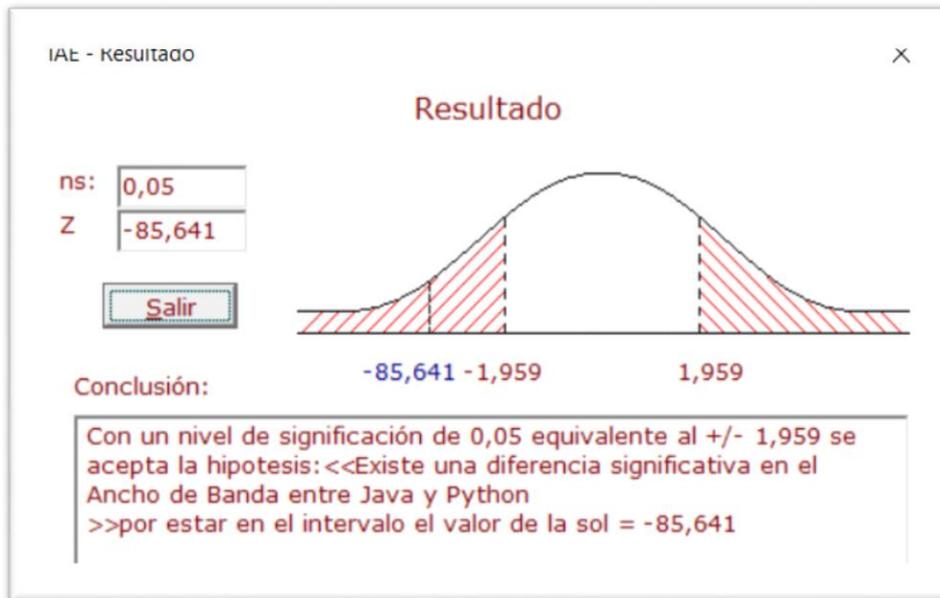


Figura 19-4: Análisis Estadístico Ancho de Banda

Realizado por: Paucar Rosman, 2018

Conclusión:

Para el índice de Ancho de Banda se puede concluir que, Java obtuvo un mejor Ancho de Banda al ser menor que el que requiere Python, aplicando la fórmula de diferencia porcentual del apartado 3.14.4, se obtuvo que, en términos porcentuales Java utiliza un 20.559 menos Ancho de Banda que Python es decir unos 2055% menos.

4.3 Comprobación de la Hipótesis

Para la comprobación de la hipótesis partimos de la hipótesis del apartado 1.6.

- 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 estableció la siguiente Matriz de Impacto que se observa en la tabla 16-4.

Tabla 16-4: 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

Realizado por: Paucar Rosman, 2018

La matriz de impacto se ingresó en el software NAIADE como se observa en la figura 20-4.

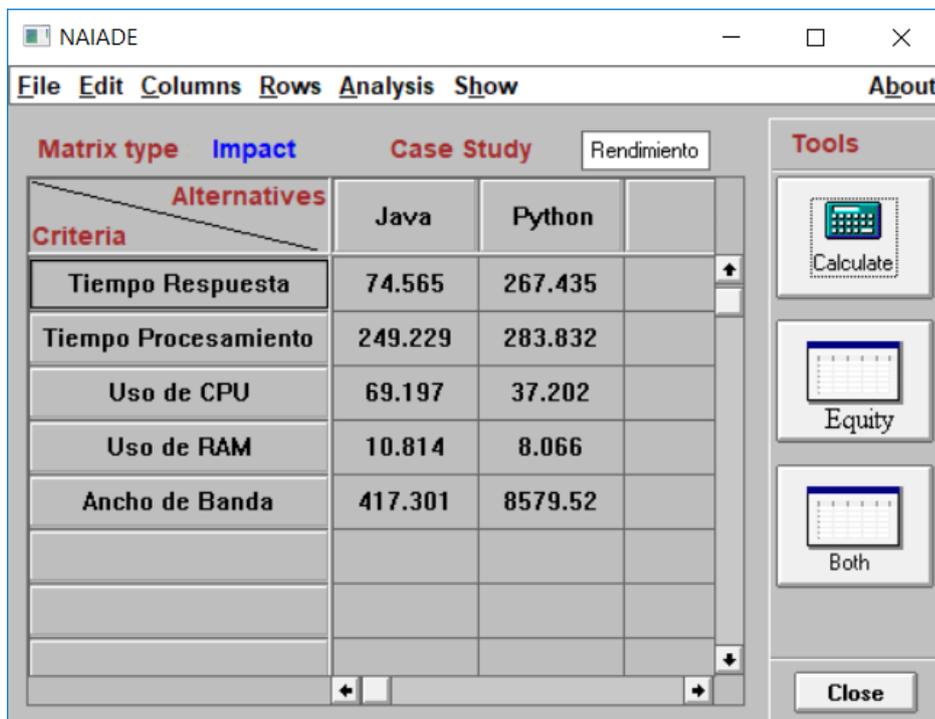


Figura 20-4: Matriz de Impacto en NAIADE

Realizado por: Paucar Rosman, 2018

Una vez ejecutado el ejercicio, los resultados que arrojó el software NAIADE con un alfa (α) de 0,5; se obtiene un ranking Φ^+ de 0.7 que selecciona al lenguaje Java como mejor que Python. En la figura 21-4 se puede observar el resultado de la matriz de impacto.

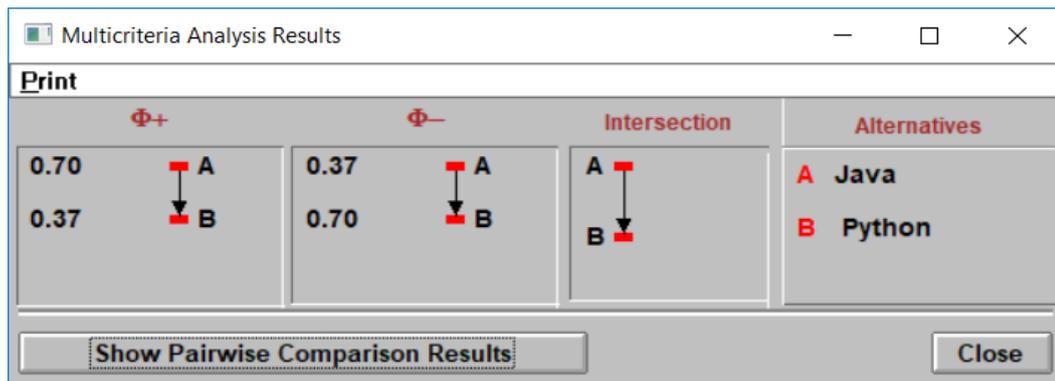


Figura 21-4: Resultado de la Matriz de Impacto

Realizado por: Paucar Rosman, 2018

De modo que el resultado concuerda con la hipótesis 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.

Por otro lado, también fue interesante analizar la comparación por cada criterio que ofrece el software, visualizándose gráficamente en la siguiente figura:

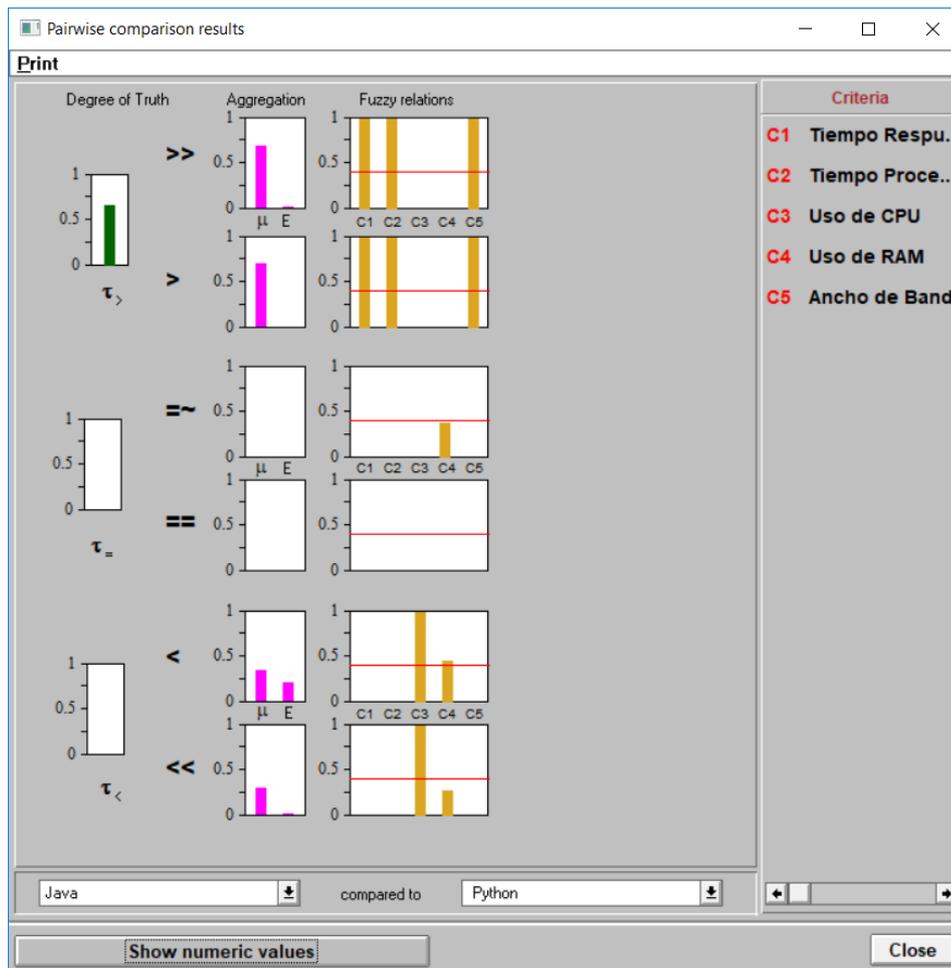


Figura 22-4: Comparación de pares entre Java y Python

Realizado por: Paucar Rosman, 2018

En la figura 23-4 se puede observar la comparación en cada uno de los indicadores y cómo aportaron al resultado final, en primer lugar, de las 6 opciones de comparación que ofrece NAIADE:

- (>>) “Mucho mejor”
- (>) “Mejor”
- (≈) “Aproximadamente Igual”
- (=) “Igual”
- (<) “Peor”
- (<<) “Mucho peor”

Se puede observar que con un grado de confianza de 67.44 (Ver figura 23-4) Java es “Mejor que Python en cuanto al rendimiento en la transferencia de imágenes” la misma que viene resaltada con la barra de color verde.

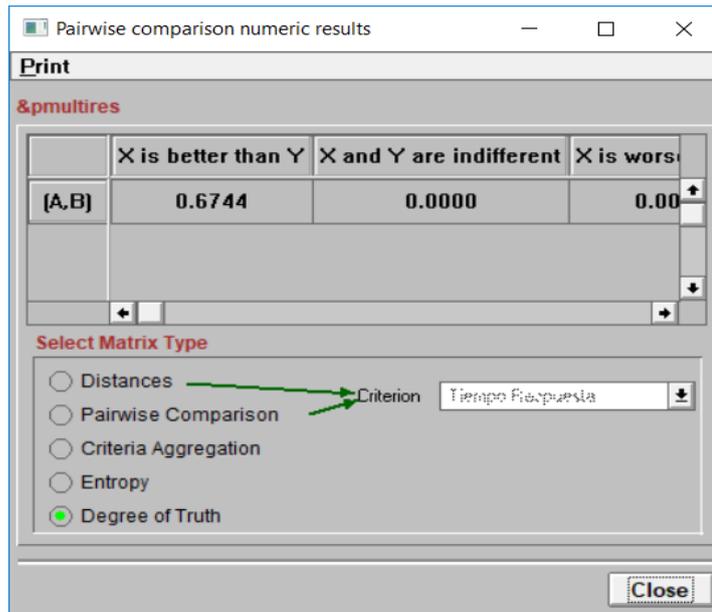


Figura 23-4: Grados de Confianza

Realizado por: Paucar Rosman, 2018

Posteriormente en la misma figura 22-4 con barras de color violeta se denota que hubo un cambio significativo en los valores de los criterios los cuales se detallan en las barras de color amarillo como es el caso del criterio 1 (Tiempo de Respuesta), criterio 2 (Tiempo de Procesamiento) y criterio 5 (Ancho de Banda). Por otro lado, para el criterio 3 (Uso de CPU) y 4 (Uso de RAM) se observa que Python es mejor que Java.

CAPÍTULO V

5. DESARROLLO DEL SISTEMA DE VIDEO VIGILANCIA

5.1 Metodología de Desarrollo SCRUM

Según (Rubin, 2012), SCRUM, Scrum se basa en un pequeño conjunto de valores centrales, principios y prácticas (colectivamente, el marco de Scrum). Las organizaciones que usan Scrum deben adoptar el marco de Scrum en su totalidad, tal vez no a través de toda la organización, todo de una vez, pero ciertamente dentro de los equipos iniciales que usarán Scrum. Aceptar todo Scrum no significa, sin embargo, que las organizaciones deben implementar Scrum de acuerdo con algunas reglas rígidas, fórmula única para todos. Más bien, significa que las organizaciones siempre deben ser fieles al marco de trabajo de Scrum al tiempo que eligen una combinación adecuada de enfoques para sus implementaciones de Scrum.

Scrum combina los valores, principios y prácticas de Scrum con un conjunto de probado y verdadero. Algunos de estos enfoques serán apropiados para su situación; otros no. Cualquier enfoque deberá ser inspeccionado y adaptado a sus circunstancias únicas.

5.1.1 *¿Cómo funciona SCRUM?*

Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de estas. En la figura 24-5 se observa todo el marco de trabajo de SCRUM.

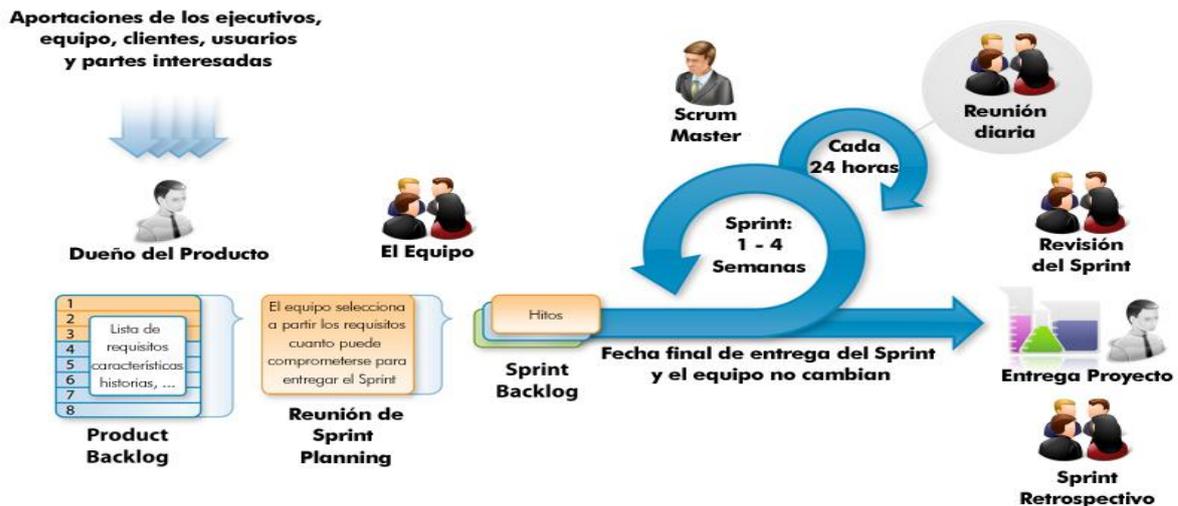


Figura 24-5: Marco de trabajo SCRUM

Fuente: iswugextremeprog.wordpress.com

Según (Trigas), SCRUM se compone de 5 fases, concepto, especulación, exploración, revisión y cierre.

En el concepto, se define en forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.

En la especulación, se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo de producto, tales como costes y agendas. Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno.

En la exploración, se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.

En la revisión, el equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado

En el cierre, se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados "mantenimiento", que hará que el producto final se acerque al producto final deseado.

5.1.2 Componentes

SCRUM está compuesto por los siguientes componentes: Reuniones / Ceremonias, Roles y Artefactos.

5.1.2.1 Reuniones:

Las reuniones que establece SCRUM son: Planificación, Reuniones Diarias, Revisión y Retrospectiva.

En la Reunión de Planificación como su nombre lo indica permite establecer un cronograma de liberaciones de acuerdo a la prioridad y estimación definido, en esta reunión deben participar el patrocinador (product owner) del proyecto y el equipo de trabajo.

La reunión diaria permite hacer un seguimiento del proyecto, en esta se plantean tres preguntas:

- ¿Qué hice ayer?
- ¿Qué voy a hacer hoy?
- ¿Qué problemas han surgido y hay que solucionar para continuar

La reunión de revisión del sprint permite verificar el incremento realizado hasta la fecha, en esta reunión deben estar presentes todos los interesados, donde se corroboran los objetivos establecidos del sprint.

La reunión de retrospectiva es la que participan todo el equipo de trabajo, para establecer los aciertos y errores, mantener o potenciar los primeros y mitigar los segundos.

5.1.2.2 Roles:

Se dividen en dos grupos, los “Cerdos” (Comprometidos) y las “Gallinas” (Interesados), que surge de un chiste sobre la intención de poner un restaurant.

Comprometidos

- Product Owner
Se podría decir que el Product Owner es el experto en el producto. Debe conocer y ser capaz de explicar la lógica de negocio del mismo hasta sus últimos detalles y es la persona que actúa como puente o intermediario entre el equipo de desarrollo y el cliente, es por ello que este puesto suele ser ocupado por consultores de negocio o, en ocasiones, perfiles relacionados con el mundo comercial.
- Scrum Master

Se tiende a pensar que el Scrum Master es el Project Manager moderno, cuando en realidad el Scrum Master es tan sólo el encargado de hacer entender Scrum a todos los implicados y velar por su cumplimiento.

- Team

Está formado por el conjunto de profesionales que cada Sprint trabaja de forma colaborativa para desarrollar un incremento de producto completo y potencialmente entregable. (becominganagilearchitect.com, 2015)

Interesados

- Usuarios

Es el destinatario final del producto

- Stakeholders

Las personas a las que el proyecto les producirá un beneficio. Participan durante la revisión del sprint.

- Managers

Toma las decisiones finales participando en la selección de los objetivos y de los requisitos (Trigas).

5.1.2.3 Artefactos:

Los artefactos definidos por SCRUM son: Product Backlog, Sprint Backlog, Sprint, Burn Down Chart.

Lista de Objetivos (Product Backlog)

La lista de objetivos/requisitos priorizada representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista (con la ayuda del Facilitador y del equipo, quien proporciona el coste estimado de completar cada requisito). Este debe reflejar las expectativas del cliente, esta lista permite involucrarle en la dirección de los resultados del producto o proyecto.

Iteración (Sprint)

Cuando el cliente solicita una entrega de los objetivos/requisitos completados hasta ese momento, el equipo puede necesitar añadir una iteración de entrega, más corta que las iteraciones habituales,

donde realizar alguna tarea que no ha sido necesaria o posible hasta el momento de la entrega final y acabar de corregir defectos detectados en la última demostración.

Lista de Iteraciones (Sprint Backlog)

Lista de tareas que el equipo elabora en la reunión de planificación de la iteración (Sprint planning) como plan para completar los objetivos/requisitos seleccionados para la iteración y que se compromete a demostrar al cliente al finalizar la iteración, en forma de incremento de producto preparado para ser entregado.

Esta lista permite ver las tareas donde el equipo está teniendo problemas y no avanza, con lo que le permite tomar decisiones al respecto.

Para cada uno de los objetivos/requisitos se muestran sus tareas, el esfuerzo pendiente para finalizarlas y la auto asignación que han hecho los miembros del equipo.

Tablero de tareas (Story Board)

La lista de objetivos a completar en la iteración (Product Backlog Items) se puede gestionar mediante un tablón de tareas (Scrum Taskboard). Al lado de cada objetivo se ponen las tareas necesarias para completarlo, en forma de post-its, y se van moviendo hacia la derecha para cambiarlas de estado (pendientes de iniciar, en progreso, hechas). Para cada miembro del equipo se puede utilizar adhesivos de colores más pequeños sobre cada tarea, de manera que se pueda ver en qué tareas está trabajando cada cual.

El equipo elabora esta lista de tareas en la segunda parte de la reunión de planificación de la iteración. La lista va evolucionando (nuevas tareas, cambios, estado, esfuerzo pendiente, ...) a medida que la iteración avanza, especialmente durante la reunión diaria de sincronización.

Este tablón o cuadro de mandos actúa como radiador de información tanto para el equipo como para cualquier otra persona relacionada con el proyecto

El progreso de la iteración y su velocidad con respecto a tareas u horas pendientes se muestra mediante un gráfico de trabajo pendiente (Burndown chart).

Gráficos de Trabajo (BurnDown Chart)

Un gráfico de trabajo pendiente a lo largo del tiempo muestra la velocidad a la que se está completando los objetivos/requisitos. Permite extrapolar si el Equipo podrá completar el trabajo en el tiempo estimado.

Se pueden utilizar los siguientes gráficos de esfuerzo pendiente:

- Días pendientes para completar los requisitos del producto o proyecto (product burndown chart), realizado a partir de la lista de requisitos priorizada (Product Backlog).
- Horas pendientes para completar las tareas de la iteración (sprint burndown chart), realizado a partir de la lista de tareas de la iteración (Iteration Backlog).

Este tipo de gráfico permite realizar diversas simulaciones: ver cómo se aplazan las fechas de entrega si se le añaden requisitos, ver cómo se avanzan si se le quitan requisitos o se añade otro equipo, etc. (Hiebaum, 2015)

5.2 Fase de Planificación

En esta fase se realiza una planificación del proyecto, estableciendo prioridades, y fechas de entrega.

5.2.1 Requisitos del Proyecto

Los requerimientos de un sistema informático se clasifican en funcionales y no funcionales, a continuación, se detallan cada uno de ellos en las tablas 17-5 y 18-5.

Tabla 17-5: Requisitos Funcionales

Nro.	Descripción
RQF1	El sistema debe permitir autenticar al usuario en el sistema de videovigilancia
RQF2	El sistema debe permitir conectarse a una cámara de video vigilancia
RQF3	El sistema debe permitir grabar un video del monitoreo en un tiempo determinado
RQF4	El sistema debe permitir desactivar una cámara de videovigilancia
RQF5	Instalación del Sistema Operativo Raspbian

Realizado por: Paucar Rosman, 2018

Tabla 18-5: Requisitos No Funcionales

Nro.	Descripción
RQNF1	El sistema debe permitir acceder con seguridad al sistema de videovigilancia
RQNF2	El sistema debe permitir conexiones desde cualquier lugar de la red
RQNF3	El sistema debe ofrecer un video con calidad mínima para observar lo que sucede en el entorno que está siendo monitoreado.
RQNF4	El sistema debe ser fácil de usar a los usuarios
RQNF5	El sistema debe ser portable, en caso de cambio de lugares de los dispositivos.

Realizado por: Paucar Rosman, 2018

5.2.2 Roles del Proyecto

Los roles definidos para el proyecto son los siguientes que se muestran en la tabla 19-5.

Tabla 19-5: Roles del Proyecto

Nombre	Rol	Descripción
Ing. Paúl Paguay	Scrum Master	Encargado de Coordinar el proyecto
Ing. Rosman Paucar	Team	Encargado del desarrollo del proyecto
	Product Owner	Encargado de validar y aprobar el proyecto
	Usuario	Encargado de probar la funcionalidad del proyecto

Realizado por: Paucar Rosman, 2018

5.2.3 Product Backlog

El Product Backlog del proyecto se lo ha dividido en dos partes una para las historias técnicas y otra para las historias de usuario que se llevarán a cabo en el proyecto.

5.2.3.1 Product Backlog Historias Técnicas

Las historias técnicas, también conocidas como elementos no-funcionales, o como se quiera llamarlos. Son cosas que deben hacerse pero que no son un entregable ni están directamente relacionadas con ninguna historia específica, y no son de valor inmediato para el Dueño de Producto. (Kniberg, 2007)

Las historias técnicas para el proyecto se pueden observar en la tabla 20-5.

Tabla 20-5: Historias Técnicas

HISTORIAS TÉCNICAS			
Nro.	Descripción	Prioridad (0-10)	Esfuerzo (Horas)
HT01	Documento de Análisis de Requerimientos	10	30
HT02	Instalar el servidor para almacenamiento de videos	10	20
HT03	Diseño Global de la Solución	10	80
HT04	Desarrollo de la capa de acceso a datos	10	10
HT05	Instalación del Sistema de Control de Versiones (Subversión)	10	5
HT06	Diseño conceptual de la solución	10	30
HT07	Documentación del sistema	10	50
HT08	Instalación del Sistema Operativo Raspbian	8	10
HT09	Instalación de las plataformas de desarrollo	8	20
HT10	Configuración de dispositivos de red y webcam en la Placa Computadora (Raspberry)	8	20

Realizado por: Paucar Rosman, 2018

5.2.3.2 Product Backlog Historias de Usuario

Una historia de usuario representa una necesidad de negocio que puede ser implementada en un sprint y aporta valor al producto. Al final del Sprint la historia añade una nueva funcionalidad o característica al producto y puede ser candidata para pasar a producción. (Acosta, 2017).

Las historias de usuario han sido definidas y se puede visualizar en la tabla 21-5

Tabla 21-5: Historias de Usuario

HISTORIAS DE USUARIO			
Nro.	Descripción	Prioridad (0-10)	Esfuerzo (Horas)
HU01	Como usuario del sistema, deseo autenticarme en el sistema de videovigilancia	10	5
HU02	Como usuario del sistema deseo conectarme a la cámara de seguridad	10	60
HU03	Como usuario del sistema deseo grabar el video sobre el monitoreo de la cámara activa	8	30
HU04	Como usuario deseo desactivar las cámaras de seguridad	4	20

Realizado por: Paucar Rosman, 2018

5.2.4 *Sprint backlog (Planificación)*

El Product Backlog posterior a ser establecido, sirve de insumo para crear el Sprint Backlog, el mismo que es un artefacto para planificar las entregas a lo largo del proyecto, la misma que se observa en la tabla 22-5.

Tabla 22-5: Sprint Backlog

SPRINT	NRO.	Esfuerzo (Horas)	RESPONSABLE	FECHAS	
				INICIO	FIN
1	HT01	30	Team	01 Mar 2018	30 Mar 2018
	HT02	20	Team		
	HT03	80	Team		
	HT04	10	Team		
2	HT05	5	Team	02 Abr 2018	30 Abr 2018
	HT06	30	Team		
	HT07	50	Team		
	HT08	10	Team		
	HT09	20	Team		
	HT10	20	Team		
3	HU1	5	Team	01 May 2018	22 May 2018
	HU2	60	Team		
	HU3	30	Team		
	HU4	20	Team		
	H05	10	Team		

Realizado por: Paucar Rosman, 2018

5.2.5 Cronograma de actividades

En base al Sprint Backlog establecido, se procedió a generar el cronograma de actividades utilizando el software Project en su versión 2016, como se observa en el gráfico siguiente:

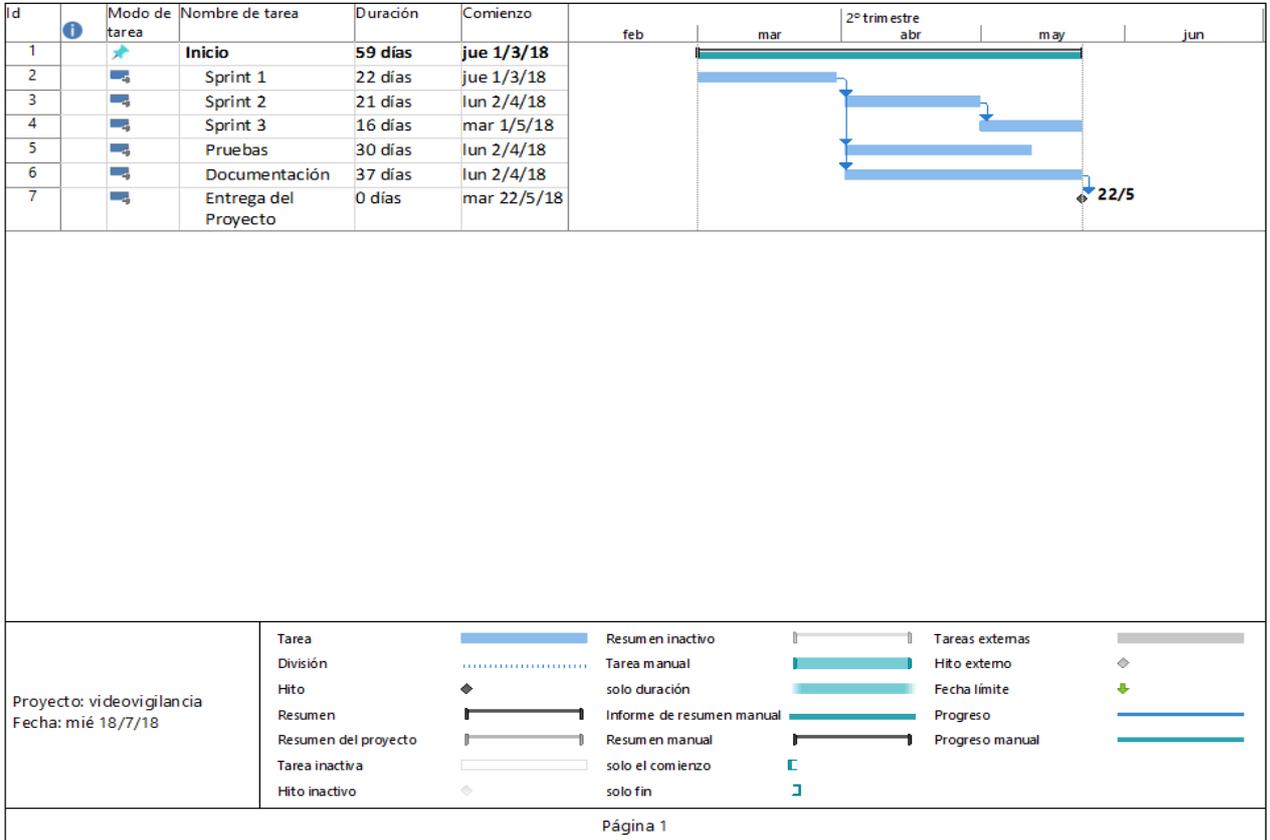


Grafico 6-5: Diagrama Gantt del proyecto

Realizado por: Paucar Rosman, 2018

5.3 Diseño de la solución

5.3.1 Programación MULTITHILO

Según (Tutorialspoint, s.f.), un programa de subprocessos múltiples contiene dos o más partes que pueden ejecutarse al mismo tiempo y cada parte puede manejar una tarea diferente al mismo tiempo, haciendo un uso óptimo de los recursos disponibles, especialmente cuando su computadora tiene varias CPU.

Por definición, la multitarea es cuando múltiples procesos comparten recursos de procesamiento comunes, como una CPU. Multi-threading amplía la idea de la multitarea a aplicaciones en las que puede subdividir operaciones específicas en una sola aplicación en hilos individuales. Cada uno de los hilos se puede ejecutar en paralelo. El sistema operativo divide el tiempo de

procesamiento no solo entre diferentes aplicaciones, sino también entre cada subproceso dentro de una aplicación.

Multi-threading le permite escribir de manera que varias actividades pueden continuar simultáneamente en el mismo programa.

Los estados de un hilo se pueden observar en la figura 25-5.

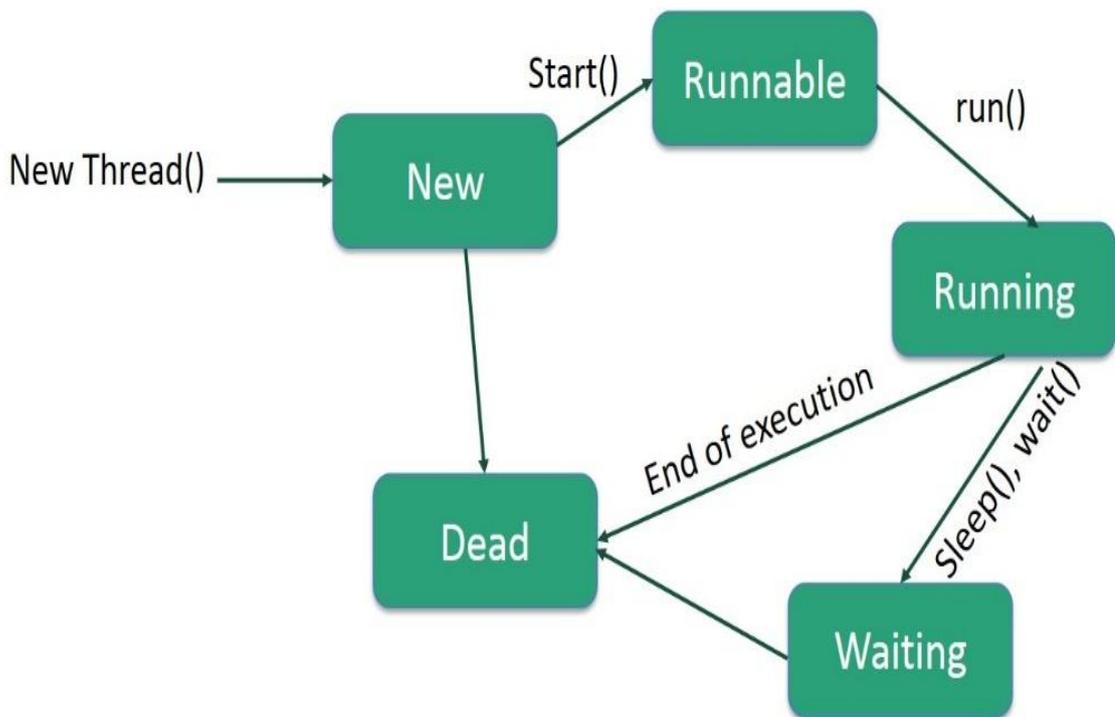


Figura 25-5: Estados de un hilo

Realizado por: Paucar Rosman, 2018

5.3.2 Sockets

Según, (MasAdelante.com, s.f.), un socket (enchufe), 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 (API, application programming interface).

Un socket es también una dirección de Internet, combinando una dirección IP (la dirección numérica única de cuatro partes que identifica a un ordenador particular en Internet) y un número

de puerto (el número que identifica una aplicación de Internet particular, como FTP, Gopher, o WWW).

La secuencia de pasos por las que pasa un socket se puede observar en la siguiente figura 26-5.

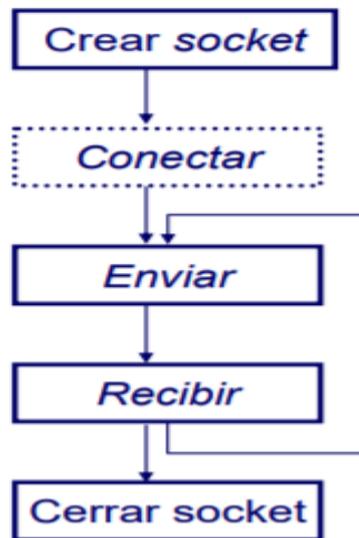


Figura 26-2: Proceso Socket

Fuente: (García)

5.3.3 Casos de uso

En la figura 27-5 se observa los casos de uso que conforma el sistema informático.

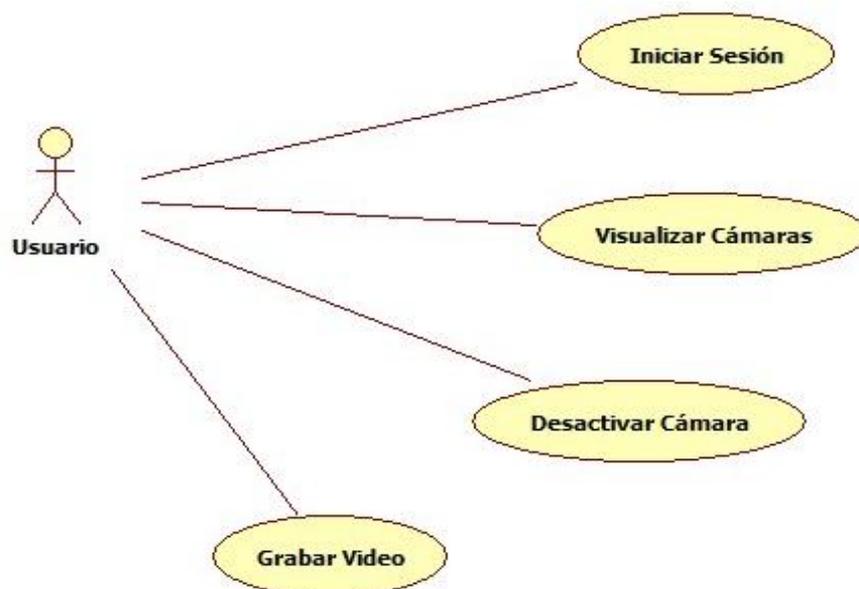


Figura 27-5: Casos de Uso

Realizado por: Paucar Rosman, 2018

5.3.4 BurnDown Chart

En la figura 28-5 se puede observar el monitoreo del proyecto mediante la gráfica BurnDown Chart.

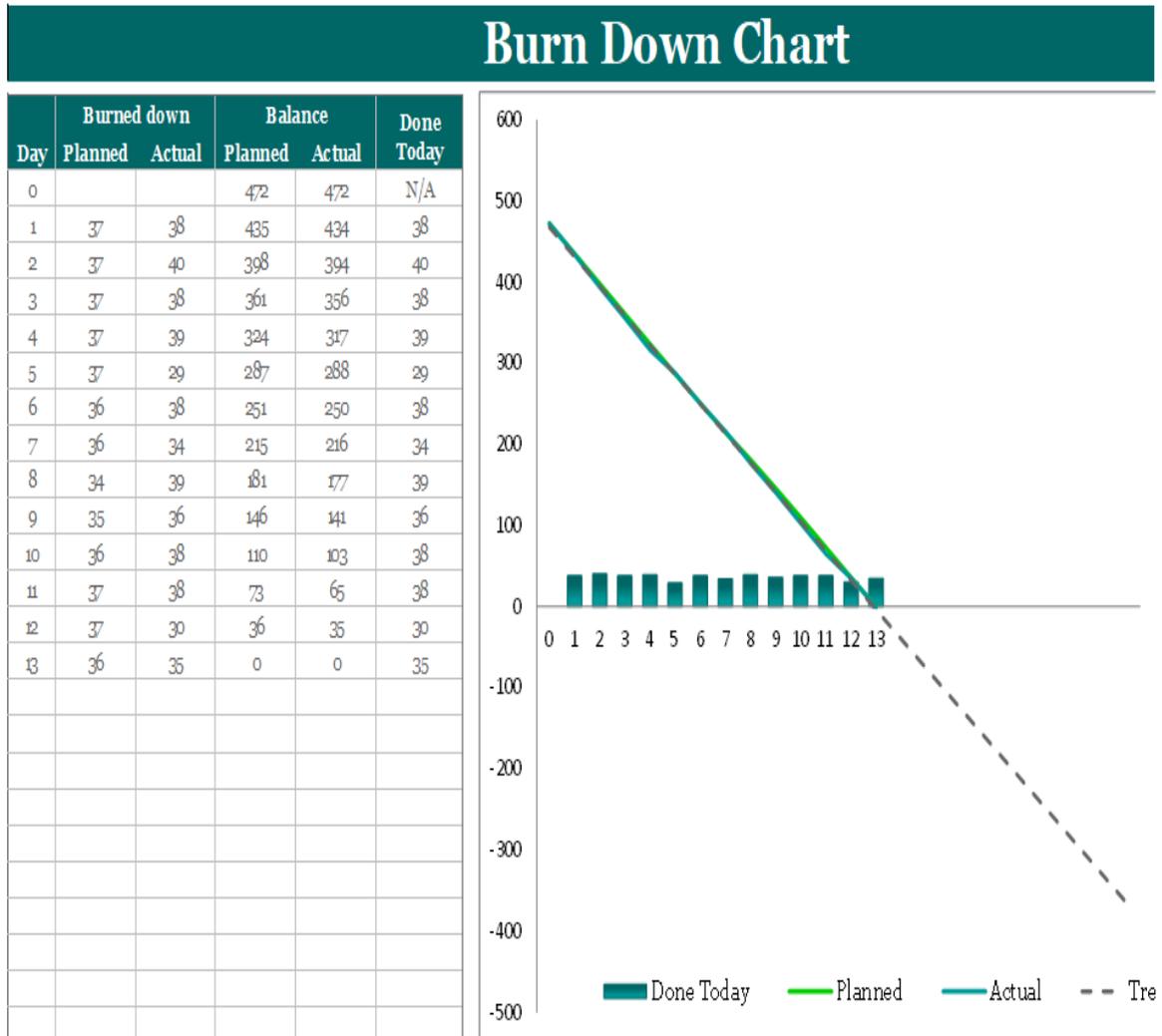


Figura 28-2: BurnDown Chart

Realizado por: Paucar Rosman, 2018

5.3.5 Historias de usuario

A continuación, se especifican las historias de usuario más importantes del sistema de videovigilancia, como se observa en las tablas 23-5, 24-5 y 25-5:

Tabla 23-5: Historia de Usuario Autenticación

HISTORIA DE USUARIO	
ID: HU01	Nombre de la Historia: Autenticación de Usuario
Responsable: Team	Sprint: 03
Fecha Inicio: 01/05/2018	Fecha Fin: 22/05/2018
Descripción: COMO usuario DESEO autenticarme en el sistema	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Ingreso de credenciales • Verificación de credenciales • Acceso al sistema 	

Realizado por: Paucar Rosman, 2018

Tabla 24-5: Historia de Usuario conectarse a la cámara

HISTORIA DE USUARIO	
ID: HU02	Nombre de la Historia: Conectar a Cámara
Responsable: Team	Sprint: 03
Fecha Inicio: 01/05/2018	Fecha Fin: 22/05/2018
Descripción: COMO usuario DESEO conectarme a la cámara de seguridad	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Ingreso al sistema • Visualizar las cámaras de seguridad 	

Realizado por: Paucar Rosman, 2018

Tabla 25-5: Historia de Usuario Grabar

HISTORIA DE USUARIO	
ID: HU02	Nombre de la Historia: Grabar Video
Responsable: Team	Sprint: 03
Fecha Inicio: 01/05/2018	Fecha Fin: 22/05/2018
Descripción: COMO usuario DESEO grabar el video sobre el monitoreo	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Ingreso al sistema • Escoger la cámara a descargar el video • Escoger lugar para guardar • Descargar el video 	

Realizado por: Paucar Rosman, 2018

Tabla 26-5: Historia de Usuario desactivar cámaras

HISTORIA DE USUARIO	
ID: HU02	Nombre de la Historia: Desactivar cámaras
Responsable: Team	Sprint: 03
Fecha Inicio: 01/05/2018	Fecha Fin: 22/05/2018
Descripción: COMO usuario DESEO desactivar las cámaras de seguridad	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Ingreso al sistema • Escoger la cámara a descargar el video • Desactivar la cámara 	

Realizado por: Paucar Rosman, 2018

5.4 Diagrama de Despliegue

En la figura 29-5 se observa el modelo de implementación de la solución



Figura 29-5: Diagrama de Despliegue

Realizado por: Paucar Rosman, 2018

5.5 Diagrama de Componentes

En la figura 30-5 se observa el diagrama de componentes que contendrá la solución.

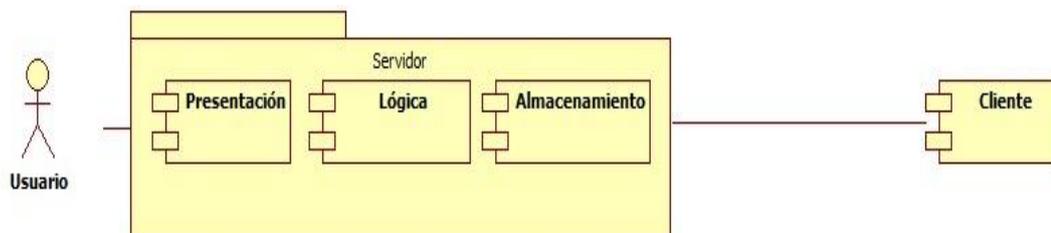


Figura 30-5: Diagrama de Componentes

Realizado por: Paucar Rosman, 2018

El proceso del cliente (Raspberry) será un aplicativo de consola que se iniciará al encender la Placa de Raspberry mientras que el Servidor que receptorá las imágenes para crear el video que en lo posterior será mostrado al usuario.

CONCLUSIONES

El desarrollo de la presente evaluación del rendimiento de la transferencia de imágenes entre los lenguajes de programación Java y Python, devolvió las siguientes conclusiones:

- Los lenguajes de programación Java y Python *son de uso libre*, lo cual facilitó su descarga, instalación y uso en esta investigación, de igual manera, al *ser multiplataforma*, se transformó en una gran ventaja ya que con ello se permitió la implementación de los aplicativos tanto de los servidores como de los clientes en los sistemas operáticos Windows 10 y Raspbian respectivamente. Por otra parte, con las características de *portabilidad, seguridad y robustez* que poseen, se permitió la instalación en los dispositivos utilizados (laptop y Raspberry Pi), conexión mediante sockets, sumándose a que se encontró una amplia y completa documentación de aplicaciones relacionadas al ámbito de la videovigilancia y uso de dispositivos inteligentes, lo cual junto con la ayuda de foros de programación, repositorios y recursos como las librerías *sarxos.webcam.Webcam* (en Java) y *Pygame* (en Python) permitieron llevar a cabo el correcto desarrollo de los prototipos para las respectivas pruebas.
- Se desarrolló dos escenarios de pruebas, uno para la plataforma Python y otro para la plataforma Java, donde su implementación y posterior comunicación entre el Cliente y el Servidor (*en una computadora de placa reducida Raspberry*), se desarrollaron bajo las mismas condiciones (iguales: clientes, servidores, conexión inalámbrica, ancho de banda, relación de radio, resolución, y flicker), lo cual podría cambiar ligeramente si se sujeta a nuevos requerimientos y condiciones como también nuevo hardware y software.
- 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* y en cuanto al *tiempo de procesamiento 13.88% a favor de Java.*, en lo referente a las métricas externas, se obtuvo que *el uso de CPU arrojó un 86% a favor de Python, el uso de RAM, un 34% a favor de Python*, y para el *ancho de banda Java ocupa 20.56 veces menos KB que Python*. Finalmente, y aplicando el análisis multicriterio con el método de NAIADE se obtiene que con *un ranking positivo de 0.7 “Java es Mejor que Python”*, por lo cual se acepta la hipótesis alternativa y se rechaza la nula.
- La implementación del sistema de videovigilancia basado en la Internet de las Cosas, permitió desarrollar un sistema en la plataforma Java, el cual obtiene información en tiempo real, así como de integrarse a la red inalámbrica con facilidad y sin afectar los recursos existentes, posibilitando con un estudio más profundo incorporar características que los dispositivos

comunes no cuentan como envío de información adicional a través de los diferentes canales de comunicación cifrada como alertas, mensajes, incorporación de características de domótica, entre otros.

RECOMENDACIONES

Producto del presente trabajo se obtienen las siguientes recomendaciones:

- Para la instalación de los sistemas operativos en computadoras de placa reducida (Raspberry) es recomendable verificar previamente la compatibilidad entre la versión de la placa con el Sistema Operativo a instalar para que tenga un rendimiento adecuado, y de esta manera no afecte a la calidad del video y la experiencia del usuario sea mejor.
- Las cámaras web que se incorporan a los sistemas de videovigilancia en instituciones y lugares con infraestructura existente, deben ser de alta calidad para que las imágenes pueden ofrecer una adecuada visualización y experiencia del usuario final, así también si los sistemas van a ser instalados en edificaciones ya terminadas es recomendable utilizar comunicación Wireless segmentadas, evitando de esta manera cableados que a más de afectar la estética del lugar, hacen necesario la incorporación de canaletas, patch cord, patch panels, organizadores, etc., lo cual hará incurrir es gastos para su adquisición e instalación respectivamente; compensándose las bondades que ofrecen las conexiones cableadas con equipos para conexiones inalámbricas que ofrezcan incorporación como por ejemplo de streaming, calidad de servicio, altas velocidades de trasmisión, entre otros (802.11Q).
- Se recomienda tomar como guía la presente investigación, y realizar nuevas investigaciones en el mismo campo del Internet de las Cosas e incorporar más detalles como podrían ser análisis de la calidad de servicio, incorporación de características de domótica, experiencia de usuario, entre otros.

BIBLIOGRAFÍA

- Acosta, D.** (Dic de 2017). *Scrum con Jira – Relación entre Épicas, Historias y Tareas Técnicas*.
Obtenido de <http://diegoacosta.net>: <http://diegoacosta.net/blog/2017/12/10/scrum-con-jira-relacion-entre-epicas-historias-y-tareas-tecnicas/>
- Alonso Forcelledo, J., & Cañete González, C.** (2006). Influencia en el rendimiento de un sistema de la velocidad de CPU y cantidad de memoria. *uniovi*.
- Anderson, D., Sweeney, D., & Williams, T.** (2008). *Estadística para la Administración y Economía*. México D. F.: Cengage Learning™.
- Arias, F. G.** (2012). *El Proyecto de Investigación. Introducción a la Metodología Científica*. (6ta ed.). Caracas: Episteme.
- Axis Communications.** (2018). *Axis.com*. Obtenido de <https://www.axis.com/customer-story/76>
- Bakos, Y. y Treacy, M.** (Junio de 1986). Information technology and corporate strategy: A research perspective. En M. Quarterly (Ed.). MIS Quarterly. Obtenido de <http://dx.doi.org/10.2307/249029>
- becominganagilearchitect.com.** (29 de Jun de 2015). *becominganagilearchitect.com*. Obtenido de El Equipo Scrum. Scrum Master, Product Owner y Developer Team: <http://www.becominganagilearchitect.com/scrum-roles-scrum-master-product-owner-desarrolladores>
- BV, T. s.** (2016). *Índice de TIOBE para noviembre de 2016*. Obtenido de Caro, R., López, V., & Miñana, G. <http://www.tiobe.com/tiobe-index/>
- Caro, R., López, V., & Miñana, G.** (20 de Febrero de 2013). *La ingeniería de fiabilidad de sistemas informáticos a través de EMSI*. Obtenido de <http://journal.poligran.edu.co/index.php/elementos/article/view/417/394>
- Casco, S. M.** (2014). Raspberry Pi, Arduino y Beaglebone Black. 4,7.
- 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).

Daniela Garcia, Edgar Ríos. (2011).

Daniela Garcia, Edgar Ríos. (Abril de 2011). *Canal Cifrado para Comunicación Cliente/Servidor*. Obtenido de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2817/Tesis.pdf?sequence=1>

DEFINICIÓN.DE. (s.f.). *definicion.de*. Obtenido de <https://definicion.de/ancho-de-banda/>

Dromey. (1996). *Cornering the Chimera*. (I. Software, Ed.) Recuperado el 22 de junio de 2017, de <http://www.computer.org/software/so1996/s1033abs.htm>

E. Camacho, F. Cardeso, G. Nuñez. (Abril de 2004). *Arquitecturas de Software: Guías de Estudio*. Obtenido de <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>

Echeverría, D. (2016). Tiempo de Respuestas y Experiencia de Usuario. *Genexus Consulting*, PP. 231-234.

EcuRed. (13 de ago de 2018). *Lenguaje de Programación*. Obtenido de https://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n

Fúquene. (2011).

Gabriel Cuzman. (s.f.). *Herramientas de Desarrollo en Informática. Python*.

García, C. (s.f.). *PROGRAMACIÓN CON SOCKETS*. Madrid: Departamento de Ingeniería Telemática - Carlos III de Madrid.

Ghodsí, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., & Stoica, I. (s.f.). Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. *Usenix*.

Gomez, J. (2014). Investigación Descriptiva. *Magazine*, 10. Recuperado el 7 de Noviembre de 2016

Hall, S. (s.f.). *¿Qué es un programa SPSS?* Recuperado el 13 de Jul de 2018, de Techlandia.com:
https://techlandia.com/programa-spss-sobre_48697/

Héctor Corrales S., Carlos Cilleruelo R. & Alejandro Cuevas N. (2014). *Criptografía y Métodos de Cifrado*. Obtenido de
<http://www3.uah.es/libretics/concurso2014/files2014/Trabajos/Criptografia%20y%20Metodos%20de%20Cifrado.pdf>

Hernández, C. L. (mayo de 2006). *Herramientas y Lenguajes de Programación*. (U. d. Laguna, Ed.) Obtenido de
<http://nereida.deioc.uill.es/~cleon/doctorado/doc06/doc06/html/node9.html>

Heurung, A. (s.f.). *Uso de CPU y tiempo de procesador*. Obtenido de techlandia.com:
https://techlandia.com/cpu-procesador-info_199144/

Hiebaum, J. (6 de Abr de 2015). <http://manualdelgamedesigner.blogspot.com>. Obtenido de
Método SCRUM: <http://manualdelgamedesigner.blogspot.com/2015/04/metodo-scrum.html>

J. Padilla & J. Pérez. (2009-2010). *Comparativa JSEE vs .NET*. Obtenido de
<http://joseperezlozano.com/wp-content/uploads/2010/05/J2EEOPuntoNetVersionWeb.pdf>

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

Jorge Villalobos, Uriel Hernandez. (7 de Abril de 2011). Obtenido de
<http://codigoprogramacion.com/?s=Socket+en+Java#.WwshzkgvzIU>

Kioskea. (2014). *Criptografía*. Obtenido de <http://es.ccm.net/contents/129-criptografia>

Kniberg, H. (2007). *Scrum and XP from the Trenches*. USA: C4Media.

Li, Ferdoush. (2014).

- López Noriega, M., Lagunes Huerta, C., & Herrera Sánchez, S.** (2006). Excel como una herramienta asequible en la enseñanza de la estadística. *Redalyc.org*, 7(1). Recuperado el 13 de Jul de 2018, de <http://www.redalyc.org/html/2010/201021084007/>
- M. Preetha & M. Nithya.** (June 2013). A study and performance of RSA algorithm. *IJCSMC*. Obtenido de <http://www.ijcsmc.com/docs/papers/June2013/V2I6201330.pdf>
- Maksimović, Vujović, Davidović, Milošević, & Perišić.** (2014).
- man7. (03 de Mar de 2018). *man7.org*. Obtenido de <http://man7.org/linux/man-pages/man1/watch.1.html>
- María Begoña R. & Patricia Bazán.** (s.f.). *Java y .NET comparación de paradigmas*. Obtenido de http://linti.unlp.edu.ar/uploads/docs/java_y__net_comparacion_de_paradigmas.pdf
- Márquez Bertha & Zulaica José.** (12 de Enero de 2004). *Colección de Tesis Digitales*. (UDLAP, Ed.) Recuperado el 24 de junio de 2017, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/
- MasAdelante.com.** (s.f.). *¿Que es un Socket? - Definición de Socket*. Obtenido de <http://www.masadelante.com/faqs/socket>
- Matus, J.** (s.f.). *Estadística Descriptiva e Inferencial I*.
- Medina, J.** (2014). *Pruebas de rendimiento TIC*. Murcia: SG6.
- Molero, X., Juiz, C., & Rodeño, M. J.** (2004). *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*. University of the Balearic Islands: ResearchGate.
- Molero, X.; Ruiz, C.; Rodeño, M.** (2004). *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*. España: Pearson Prentice Hall.
- Morales, M.** (2003). *Compresión de Datos + Criptografía*. Obtenido de <http://www.tamps.cinvestav.mx/~mmorales/documents/Compre-Cripto.pdf>
- Moreno Mateos, A.** (Junio de 2016). Obtenido de http://oa.upm.es/44701/1/TFG_ALEJANDRO_MORENO_MATEOS.pdf

- Munda, G.** (2006). A NAIADE based approach for sustainability benchmarking. *International Journal of Environmental Technology and Management* , pp. 65-78.
- 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).
- Neredia.** (s.f.). *Herramientas y Lenguajes de Programación 05-06*. Obtenido de <http://nereida.deioc.ull.es/~cleon/doctorado/doc06/doc06/html/node1.html>
- Oliveira, H. D.** (10 de junio de 2012). *Burndown Chart*. Obtenido de <http://agilizandando.blogspot.com/2012/06/burndown-chart-quemando-tareas.html>
- ORACLE.** (2011). Guía de administración del sistema: administración avanzada. Obtenido de https://docs.oracle.com/cd/E24842_01/html/E23086/sprocess-1.html
- Padilla & Pérez.** (2009-2010).
- Palacio, J.** (2006). *El modelo Scrum*. Recuperado el 24 de octubre de 2017, de http://www.academia.edu/13762309/El_modelo_Scrum
- Pereira, B., Ayaach, F., Quintero, H., Granadillo, I., & Bustamante, J.** (s.f.). *Métricas de Calidad de Software*.
- Prechelt.** (2000).
- Pressman, R.** (2002). *Ingeniería de Software: Un Enfoque Práctico* (Quinta Edición ed.). McGraw Hill.
- Pressman, R. S.** (2010). *Ingeniería de Software: Un enfoque práctico* (Séptima Edición ed.). México: McGraw Hill. Obtenido de <http://eva.sepyc.gob.mx:8383/greenstone3/sites/localsite/collect/ciencia1/index/assoc/HASH015f/ceb375c1.dir/33040073.pdf;jsessionid=3B1AFBB6E4A987B547DFE49F5BA9594C>

Rai, A., Patnayakuni, R. y Seth, N. (2006). Firm performance impacts of digitally enabled supply chain integration capabilities. *MIS Quarterly*. Obtenido de https://www.scopus.com/record/display.uri?eid=2-s2.0-32044456370&origin=inward&txGid=4DBA70BA42A9F39233ECE32335307AFF.ws_nAw8kcdt7IPYLO0V48gA%3a2

Raspbian. (s.f.).

Recursos Python. (17 de Mayo de 2014). *socket – Establecer una conexión TCP, calcular su latencia (ping) e intercambiar información*. Obtenido de <https://recursospython.com/guias-y-manuales/socket-conexion-cliente-servidor-tcp/>

Robert K. Abbott, Hector Garcia-Molina. (September de 1992). Scheduling Real-Time Transactions: A Performance Evaluation. *ACM Transactions on Database Systems*, 17(3), 513-560. Obtenido de <http://dl.acm.org/citation.cfm?id=132276>

Rodríguez, M. (2005). *Programación Concurrente*. (U. d. Canaria, Ed.) Obtenido de <https://www2.ulpgc.es/hege/almacen/download/20/20233/tema1.pdf>

Rojas M., Rosas V., Roaro R. & Puebla J. (2012). *Programación Paralela y Concurrente*. Obtenido de <http://facturapp.weebly.com/sockets.html>

Rubert, D. (s.f.). *Excel: Introducción a las Hojas de Cálculo*. Obtenido de Universitat per a Majors: <http://www3.uji.es/~vrubert/unimajors/2011-iiim/sesion15/#1>

Rubin, K. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Michigan: Pearson.

S. Oukili and S. Bri. (2015). FPGA implementation of Data Encryption Standard. Casablanca: 27th International Conference on Microelectronics (ICM). Obtenido de <http://ieeexplore.ieee.org/document/7438004/>

seabrookewindows.com. (28 de Abril de 2018). *Cómo calcular la tarifa de transferencia de datos*. Obtenido de <http://www.seabrookewindows.com/ZPlbz8ZM1/>

seabrookewindows.com. (4 de Junio de 2018). *Definición de "tasa de transferencia de datos"*. Obtenido de <http://www.seabrookewindows.com/eQL8k8gQ8/>

- SecuraMe. (2018). *http://www.securamente.com*. Obtenido de <http://www.securamente.com/que-son-los-fps-en-cctv/>
- Soriano, M. (s.f.). *Seguridad en redes y seguridad en la información*. Obtenido de http://improvet.cvut.cz/project/download/C2ES/Seguridad_de_Red_e_Informacion.pdf
- Stallings, W. (2006). *Cryptography and Network Security* (5ta. ed.). Boston, Columbus, Indianapolis, New York: Prentice Hall. Obtenido de <http://fac.ksu.edu.sa/sites/default/files/cryptography-network-security-5th-edition.pdf>
- Stefan Waner y Steven R. Costenoble. (julio de 2013). *Intervalos de confianza: Varios temas en línea para Matemáticas Finitas*. Obtenido de <https://www.zweigmedia.com/MundoReal/finitetopic1/confint.html>
- Tangient LLC. (2015). *Programación de redes de Telecomunicaciones*. Obtenido de <https://programacionderedesdetelecomunicaciones.wikispaces.com/Programacion+de+Sockets>
- Tenouk. (s.f.). *NETWORK PROGRAMMING*. Obtenido de <http://www.tenouk.com/Module39a.html>
- Trigas, M. (s.f.). *Gestión de Proyectos Informáticos Metodología SCRUM*. Obtenido de [s3.amazonaws.com:
https://s3.amazonaws.com/academia.edu.documents/39164786/mtrigasTFC0612memoria_1.pdf?](https://s3.amazonaws.com/academia.edu.documents/39164786/mtrigasTFC0612memoria_1.pdf?)
- Tutorialspoint. (s.f.). *tutorialspoint.com*. Obtenido de https://www.tutorialspoint.com/java/java_multithreading.htm
- Universidad Politécnica de Valencia. (2013).
- van Rossum. (2009). *PYTHON*.
- Vera Delgado, Rafael Palacios. (enero-febrero de 2006). *Introducción a la Criptografía: tipos de algoritmos*. Obtenido de https://www.icaei.es/contenidos/publicaciones/anales_get.php?id=1210

Villa Piray, D. M. (15 de Diciembre de 2014). *Análisis del rendimiento entre framework Java Server Face (JDF) y JBOSS-SEAM. Caso práctico en el sistema socioeconómico.* Obtenido de <http://dspace.espoch.edu.ec/handle/123456789/3615>

Villacís Cruz, B. (2005). *La crisis del oro azul: Un análisis de la sustentabilidad del agua en la ciudad de Quito.* Quito: Flacso.

WIKIPEDIA. (18 de Jun de 2018). *Wikipedia.org.* Obtenido de Microsoft Project: https://es.wikipedia.org/wiki/Microsoft_Project

Ynzunza, C. y Izar, J. (2013). Efecto de las estrategias competitivas y los recursos y capacidades orientados al mercado sobre el crecimiento de las organizaciones. *Contaduría y Administración.* Obtenido de [http://dx.doi.org/10.1016/S0186-1042\(13\)71202-6](http://dx.doi.org/10.1016/S0186-1042(13)71202-6)

Z. Kouser, M. Singhal and A. M. Joshi. (23-25 de Diciembre de 2016). FPGA implementation of advanced Encryption Standard algorithm. Jaipur, India: 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE). Obtenido de <http://ieeexplore.ieee.org/document/7939594/>

ANEXOS

Anexo A: Código fuente JAVA

Código Servidor

```
package servidor;
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.ServerSocket;
import java.net.Socket;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import sun.misc.IOUtils;
import java.awt.Image;
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileWriter;
import java.io.PrintWriter;
import javax.imageio.ImageIO;
import org.jcodec.api.awt.AWTSequenceEncoder;

/**
 *
 * @author Alex
 */
public class ServerThread implements Runnable {

    public final int SOCKET_PORT = 8888;
    public final int FILE_SIZE = 58087;
    JLabel labenCapture = null;
    int bytesRead;
    int current = 0;
    ServerSocket servsock = null;
    Socket sock = null;

    boolean stop = false;
```

```

public ServerThread(JLabel labenCapture) {
    this.labenCapture = labenCapture;
}

@Override
public void run() {
    try {

        try {
            long tiempo1, tiempo2, tiempo3, tiempo4, tiempo5;
            FileWriter fw = null;
            fw = new FileWriter("LogServidorJava.txt");
            PrintWriter guardar = new PrintWriter(fw);
            String linea;
            servsock = new ServerSocket(SOCKET_PORT);
            linea = ( "Inicio Recibe Conexion,Tiempo recepcion
Imagen,Tiempo Mostrar Imagen\n");
            guardar.write(linea);
            guardar.flush();
            while (true) {
                System.out.println("Waiting...");
                current++;
                try {

                    sock = servsock.accept();
                    System.out.println("Accepted connection : " +
sock);

                    tiempo1 = new
java.util.Date().getTime();//Tiempo recepción de imagen

                    byte[] mybytearray = new byte[FILE_SIZE];
                    InputStream is = sock.getInputStream();
                    byte[] bytes = IOUtils.readFully(is, -1,
false);

                    tiempo2 = new java.util.Date().getTime();
//Tiempo preparar imagen
                    mostrarImagenDesktop(bytes);
                    tiempo3 = new java.util.Date().getTime();
//Tiempo mostrar magen
                    crearVideo(bytes);

```

```

        linea = (tiempo1 + "," + tiempo2 + "," +
tiempo3 + "\n");

        guardar.write(linea);
        guardar.flush();
    } catch (Exception ex) {
        guardar.close();
        fw.close();
        fw = null;
        System.out.println("Error al capturar");
    } finally {
        if (sock != null) {
            sock.close();
        }
    }
} finally {
    if (servsock != null) {
        servsock.close();
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e.getMessage(),
"Error al iniciar el servidor socket",
    JOptionPane.ERROR_MESSAGE);
}

}

public void mostrarImagenDesktop(byte[] imageData) {
    Thread thread = new Thread() {
        public void run() {
            ImageIcon icon = new ImageIcon(imageData);
            int w = icon.getIconWidth() + icon.getIconWidth();
            int h = icon.getIconHeight() + icon.getIconHeight();
            labenCapture.setIcon(scaleImage(icon, w, h));
        }
    };
    thread.start();
}
}

```

```

public ImageIcon scaleImage(ImageIcon icon, int w, int h) {

    Image img = icon.getImage();
    Image newimg = img.getScaledInstance(w, h,
java.awt.Image.SCALE_SMOOTH);
    return new ImageIcon(newimg);
}

public void crearVideo(byte[] imageData) {
    Thread thread = new Thread() {
        public void run() {
            try {

                if (Utils.recording) {
                    InputStream in = new
ByteArrayInputStream(imageData);
                    BufferedImage bi = ImageIO.read(in);
                    Utils.enc.encodeImage(bi);
                    Utils.enc.encodeImage(bi);
                    System.out.println("se a\u00f1adio imagen" +
current + " a la grabacion");
                }

                } catch (Exception ex) {
                    System.out.println("Error al a\u00f1adir la imgagen ->
" + current + " a la grabacion ->" + ex.getMessage());
                }

            }
        };
        thread.start();
    }
}

```

C\u00f3digo Cliente:

```

package cliente;

import com.github.sarxos.webcam.Webcam;
import com.github.sarxos.webcam.ds.v4l4j.V4l4jDriver;
import java.awt.Dimension;

```

```

import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;
import java.util.Scanner;
import javax.imageio.ImageIO;

/**
 *
 * @author Alex
 */
public class Cliente {

    static {
        try {
            Webcam.setDriver(new V4l4jDriver());
        } catch (Exception e) {
            System.out.println("error ->" + e.getMessage());
        }
    }

    public static void main(String[] args) throws IOException {
        int bytesRead;
        int current = 0;

        OutputStream os = null;

        Socket sock = null;
        try {

            System.out.println("Ingrese la ip del servidor Socket
ejemplo: 192.168.43.105 ");
            Scanner s = new Scanner(System.in);
            String host = s.next();

```

```

        System.out.println("Puerto del servidor socket Ejemplo
8888 ");

        Scanner s2 = new Scanner(System.in);
        int puerto = s2.nextInt();
        System.out.println("Connecting...");

        Webcam webcam = Webcam.getDefault();
        webcam.setViewSize(new Dimension(160, 120)); //raspberrry
        webcam.open();
        while (true) {
            try {
                current++;
                sock = new Socket(host, puerto);
                ByteArrayOutputStream baos = new
ByteArrayOutputStream();
                ImageIO.write(webcam.getImage(), "jpg", baos);
                byte[] mybytearray = baos.toByteArray();
                System.out.println("Sending " + current + "(" +
mybytearray.length + " bytes)");
                os = sock.getOutputStream();
                os.write(mybytearray, 0, mybytearray.length);
                os.flush();
                System.out.println("Done.");
                if (os != null) {
                    os.close();
                }
                if (sock != null) {
                    sock.close();
                }
                Thread.sleep(30);
            } catch (Exception e) {
                System.out.println("Error al capturar");
            }

        }

    } finally {
    }
}
}

```

Anexo B: Código fuente Python

Código Servidor

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
#     server.py
#
#     Copyright 2014 Recursos Python - www.recursospython.com
#
#

from socket import socket, error
def main():
    s = socket()

    # Escuchar peticiones en el puerto 6030.
    s.bind(("localhost", 6030))
    s.listen(0)

    conn, addr = s.accept()
    f = open("recibido2.jpg", "wb")

    while True:
        try:
            # Recibir datos del cliente.
            input_data = conn.recv(1024)
        except error:
            print("Error de lectura.")
            break
        else:
            if input_data:
                # Compatibilidad con Python 3.
                if isinstance(input_data, bytes):
                    end = input_data[0] == 1
                else:
                    end = input_data == chr(1)
                if not end:
                    # Almacenar datos.
                    f.write(input_data)
```

```
        else:
            break
        print("El archivo se ha recibido correctamente.")
    f.close()
if __name__ == "__main__":
    main()
```

Código Cliente

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
#     client.py
#
#     Copyright 2014 Recursos Python - www.recurso python.com
#
#
from socket import socket
def main():
    s = socket()
    server_address = ('localhost', 6030)
    #s.bind(server_address)
    s.connect(server_address)
    print("Conectado ..")
    while True:
        f =
open("D:\development/python/cameraRaspberry/socket/envios/prueba2.jpg
", "rb")
        content = f.read(1024)

        while content:
            # Enviar contenido.
            s.send(content)
            content = f.read(1024)

        break

# Se utiliza el caracter de código 1 para indicar
# al cliente que ya se ha enviado todo el contenido.
try:
    s.send(chr(1))
except TypeError:
    # Compatibilidad con Python 3.
    s.send(bytes(chr(1), "utf-8"))

# Cerrar conexión y archivo.
s.close()
f.close()
```

```
print("El archivo ha sido enviado correctamente.")
```

```
if __name__ == "__main__":  
    main()
```

Anexo C: Mediciones de los indicadores

Métricas Internas

Métricas Internas Java

Lenguaje	Captura Imagen (ms)	Transmisión Imagen (ms)	Preparación Imagen (ms)	Visualización de la Imagen (ms)	TOTAL	Tiempo de Respuesta
Java	137	3	63	1	204	66
Java	139	2	64	1	206	66
Java	151	2	67	1	221	69
Java	232	3	65	0	300	68
Java	264	2	85	1	352	87
	136	3	73	0	212	76
Java	138	2	66	0	206	68
Java	155	2	65	0	222	67
Java	138	2	65	0	205	67
Java	145	2	65	0	212	67
Java	244	3	66	0	313	69
Java	244	2	79	0	325	81
Java	153	2	71	1	227	73
Java	138	2	67	0	207	69
Java	151	2	66	0	219	68
Java	138	2	65	0	205	67
Java	139	2	65	0	206	67
Java	298	3	72	1	374	75
Java	219	3	65	0	287	68
Java	137	2	75	0	214	77
Java	138	3	65	0	206	68
Java	137	2	65	0	204	67
Java	154	2	64	0	220	66
Java	138	3	66	0	207	69
Java	277	2	65	0	344	67
Java	202	2	71	0	275	73
Java	137	2	78	0	217	80
Java	153	2	64	0	219	66
Java	140	2	68	0	210	70
Java	137	2	65	0	204	67
Java	137	2	64	0	203	66
Java	270	3	66	0	339	69

Java	198	2	65	0	265	67
Java	137	2	78	0	217	80
Java	137	3	65	0	205	68
Java	137	2	66	0	205	68
Java	142	2	66	0	210	68
Java	175	3	69	1	248	72
Java	272	12	65	0	349	77
Java	167	2	63	0	232	65
Java	138	2	86	0	226	88
Java	152	2	62	0	216	64
Java	137	3	64	1	205	67
Java	138	2	64	0	204	66
Java	154	2	72	0	228	74
Java	275	3	71	0	349	74
Java	206	2	65	0	273	67
Java	137	3	85	0	225	88
Java	141	2	65	0	208	67
Java	137	2	64	1	204	66
Java	139	2	65	0	206	67
Java	188	13	64	0	265	77
Java	278	3	65	0	346	68
Java	145	3	64	0	212	67
Java	139	2	213	0	354	215
Java	142	3	64	0	209	67
Java	160	2	65	1	228	67
Java	138	2	66	0	206	68
Java	205	2	65	0	272	67
Java	280	3	65	1	349	68
Java	152	3	66	0	221	69
Java	141	3	79	0	223	82
Java	138	2	79	0	219	81
Java	138	3	66	0	207	69
Java	139	2	64	0	205	66
Java	251	14	65	0	330	79
Java	270	2	66	0	338	68
Java	140	2	64	0	206	66
Java	138	2	78	0	218	80
Java	137	2	67	0	206	69
Java	152	3	64	0	219	67
Java	140	2	64	0	206	66
Java	249	2	64	1	316	66

Java	252	3	64	0	319		67
Java	148	4	65	0	217		69
Java	161	3	66	0	230		69
Java	138	2	85	0	225		87
Java	139	3	64	1	207		67
Java	138	3	66	0	207		69
Java	298	22	64	0	384		86
Java	206	2	64	0	272		66
Java	137	3	65	1	206		68
Java	138	2	73	0	213		75
Java	139	3	77	1	220		80
Java	169	3	66	0	238		69
Java	139	2	66	0	207		68
Java	275	2	69	1	347		71
Java	188	2	65	0	255		67
Java	139	3	66	0	208		69
Java	155	2	65	0	222		67
Java	139	2	77	0	218		79
Java	138	2	65	0	205		67
Java	137	3	67	0	207		70
Java	281	22	64	0	367		86
Java	196	2	64	1	263		66
Java	139	2	64	1	206		66
Java	138	3	70	0	211		73
Java	139	2	79	0	220		81
Java	139	2	65	0	206		67
Java	165	2	65	0	232		67
Java	279	2	65	0	346		67
Java	173	3	70	0	246		73
Java	138	2	64	1	205		66
Java	165	2	65	1	233		67
Java	138	3	88	0	229		91
Java	138	2	65	0	205		67
Java	165	2	64	0	231		66
Java	283	3	64	0	350		67
Java	172	2	64	0	238		66
Java	138	2	64	0	204		66
Java	137	3	71	0	211		74
Java	136	2	77	0	215		79
Java	137	2	64	0	203		66
Java	219	2	64	0	285		66

Java	271	3	64	1	339		67
Java	142	2	64	1	209		66
Java	138	3	69	1	211		72
Java	137	2	74	1	214		76
Java	152	2	78	0	232		80
Java	139	2	64	0	205		66
Java	170	13	65	0	248		78
Java	284	2	66	0	352		68
Java	168	3	64	1	236		67
Java	140	2	64	1	207		66
Java	139	2	64	0	205		66
Java	138	2	78	0	218		80
Java	139	2	66	0	207		68
Java	204	3	63	0	270		66
Java	278	2	65	0	345		67
Java	142	3	64	0	209		67
Java	139	2	64	0	205		66
Java	156	3	64	0	223		67
Java	154	3	77	0	234		80
Java	143	2	64	0	209		66
Java	205	2	64	0	271		66
Java	279	2	65	0	346		67
Java	138	3	69	0	210		72
Java	157	2	64	0	223		66
Java	138	2	64	0	204		66
Java	146	2	64	0	212		66
Java	137	4	87	0	228		91
Java	256	2	64	0	322		66
Java	269	2	64	0	335		66
Java	138	2	64	1	205		66
Java	137	2	65	0	204		67
Java	138	2	65	0	205		67
Java	154	2	65	0	221		67
Java	137	3	79	0	219		82
Java	243	12	65	0	320		77
Java	257	3	66	0	326		69
Java	138	3	65	0	206		68
Java	157	2	66	1	226		68
Java	143	2	72	0	217		74
Java	137	2	65	0	204		67
Java	138	2	79	0	219		81

Java	300	2	66	0	368		68
Java	243	2	71	0	316		73
Java	137	3	64	0	204		67
Java	139	2	65	0	206		67
Java	144	2	65	0	211		67
Java	152	2	73	0	227		75
Java	138	2	77	0	217		79
Java	268	11	63	0	342		74
Java	234	2	72	2	310		74
Java	137	2	65	0	204		67
Java	153	2	64	0	219		66
Java	138	3	65	0	206		68
Java	137	3	66	0	206		69
Java	138	3	78	0	219		81
Java	277	15	65	0	357		80
Java	245	2	64	0	311		66
Java	138	2	65	0	205		67
Java	139	2	74	0	215		76
Java	138	2	63	1	204		65
Java	138	2	65	0	205		67
Java	157	2	77	0	236		79
Java	279	2	73	1	355		75
Java	974	2	63	0	1039		65
Java	156	2	65	0	223		67
Java	304	12	74	0	390		86
Java	160	2	64	0	226		66
Java	139	2	65	0	206		67
Java	139	3	65	0	207		68
Java	137	3	78	0	218		81
Java	154	3	68	0	225		71
Java	167	2	65	1	235		67
Java	289	2	64	0	355		66
Java	161	2	65	0	228		67
Java	138	2	74	0	214		76
Java	155	2	65	0	222		67
Java	140	2	78	0	220		80
Java	139	2	64	0	205		66
Java	164	15	63	1	243		78
Java	274	2	71	0	347		73
Java	170	2	64	0	236		66
Java	139	2	63	1	205		65

Java	153	2	67	0	222		69
Java	142	2	78	1	223		80
Java	153	2	69	0	224		71
Java	211	2	73	0	286		75
Java	276	3	67	0	346		70
Java	137	2	65	0	204		67
Java	138	3	65	0	206		68
Java	154	2	65	0	221		67
Java	138	2	77	1	218		79
Java	138	2	66	0	206		68
Java	220	13	65	0	298		78
Java	284	3	66	0	353		69
Java	156	2	65	0	223		67
Java	284	2	64	0	350		66
Java	192	3	65	0	260		68
Java	137	3	84	0	224		87
Java	139	2	64	0	205		66
Java	152	2	67	0	221		69
Java	138	2	72	0	212		74
Java	148	3	64	1	216		67
Java	271	13	66	0	350		79
Java	192	2	70	0	264		72
Java	149	2	85	0	236		87
Java	138	3	64	0	205		67
Java	146	2	73	1	222		75
Java	142	3	65	0	210		68
Java	170	2	64	0	236		66
Java	303	22	65	0	390		87
Java	337	3	63	0	403		66
Java	137	2	87	0	226		89
Java	138	3	65	0	206		68
Java	154	2	63	1	220		65
Java	138	2	63	0	203		65
Java	270	2	64	0	336		66
Java	241	2	64	0	307		66
Java	138	2	64	0	204		66
Java	156	2	77	0	235		79
Java	138	2	64	0	204		66
Java	138	2	65	0	205		67
Java	138	2	65	0	205		67
Java	271	2	67	0	340		69

Java	253	3	69	0	325		72
Java	138	2	73	0	213		75
Java	143	2	77	0	222		79
Java	136	3	62	0	201		65
Java	137	3	63	1	204		66
Java	158	2	65	0	225		67
Java	284	2	65	0	351		67
Java	206	2	64	0	272		66
Java	137	2	64	0	203		66
Java	151	2	64	0	217		66
Java	145	2	81	0	228		83
Java	140	2	64	0	206		66
Java	141	2	65	0	208		67
Java	281	2	64	0	347		66
Java	217	2	66	0	285		68
Java	138	2	64	1	205		66
Java	138	2	65	0	205		67
Java	138	1	78	0	217		79
Java	135	2	65	0	202		67
Java	155	2	64	0	221		66
Java	281	2	69	1	353		71
Java	199	2	64	1	266		66
Java	137	2	64	0	203		66
Java	137	3	65	0	205		68
Java	159	2	78	0	239		80
Java	138	2	61	1	202		63
Java	138	15	64	0	217		79
Java	266	13	64	0	343		77
Java	183	2	64	0	249		66
Java	167	2	64	0	233		66
Java	138	2	64	0	204		66
Java	143	2	81	1	227		83
Java	137	2	66	0	205		68
Java	191	2	64	0	257		66
Java	286	2	64	0	352		66
Java	151	2	63	1	217		65
Java	134	2	63	1	200		65
Java	138	2	64	0	204		66
Java	154	3	79	0	236		82
Java	146	2	64	1	213		66
Java	177	2	72	1	252		74

Java	273	12	64	0	349		76
Java	162	2	64	0	228		66
Java	154	3	64	0	221		67
Java	138	2	64	0	204		66
Java	136	3	83	0	222		86
Java	137	2	64	1	204		66
Java	162	12	65	0	239		77
Java	532	2	65	0	599		67
Java	137	2	65	0	204		67
Java	137	2	63	0	202		65
Java	137	3	66	0	206		69
Java	137	2	77	0	216		79
Java	209	8	64	0	281		72
Java	270	2	64	0	336		66
Java	145	2	64	0	211		66
Java	136	2	63	0	201		65
Java	147	2	63	0	212		65
Java	162	2	62	0	226		64
Java	138	2	64	0	204		66
Java	205	13	85	1	304		98
Java	276	2	65	0	343		67
Java	140	2	64	0	206		66
Java	149	2	63	0	214		65
Java	137	2	64	0	203		66
Java	138	2	63	0	203		65
Java	137	2	63	1	203		65
Java	228	2	77	0	307		79
Java	275	2	65	0	342		67
Java	144	2	64	0	210		66
Java	138	2	64	1	205		66
Java	138	2	65	0	205		67
Java	154	2	64	0	220		66
Java	137	2	65	0	204		67
Java	225	13	88	0	326		101
Java	272	2	64	1	339		66
Java	144	2	65	0	211		67
Java	152	2	65	0	219		67
Java	136	2	64	0	202		66
Java	143	3	66	0	212		69
Java	148	3	65	0	216		68
Java	245	2	94	1	342		96

Java	278	1	66	0	345		67
Java	136	2	64	0	202		66
Java	138	2	66	0	206		68
Java	138	2	65	0	205		67
Java	146	2	64	0	212		66
Java	153	2	65	0	220		67
Java	269	13	78	1	361		91
Java	240	2	64	0	306		66
Java	137	3	63	0	203		66
Java	138	2	67	0	207		69
Java	154	2	65	0	221		67
Java	139	2	64	0	205		66
Java	136	2	64	0	202		66
Java	281	2	78	0	361		80
Java	217	2	64	1	284		66
Java	150	2	64	1	217		66
Java	137	2	64	0	203		66
Java	136	2	72	0	210		74
Java	136	2	64	0	202		66
Java	138	2	69	0	209		71
Java	290	10	82	0	382		92
Java	225	2	64	0	291		66
Java	144	2	64	0	210		66
Java	138	3	65	0	206		68
Java	138	2	66	0	206		68
Java	153	2	65	0	220		67
Java	143	3	65	0	211		68
Java	269	15	213	0	497		228
Java	219	2	67	0	288		69
Java	137	2	64	0	203		66
Java	154	2	64	1	221		66
Java	138	2	64	0	204		66
Java	139	3	64	0	206		67
Java	138	3	65	0	206		68
Java	273	2	81	1	357		83
Java	243	3	72	0	318		75
Java	136	2	64	0	202		66
Java	136	3	63	0	202		66
Java	137	2	64	0	203		66
Java	138	2	62	1	203		64
Java	153	3	59	0	215		62

Java	138	2	40	0	180		42
Java	138	2	54	0	194		56
Java	137	2	193	0	332		195
Java	137	2	87	0	226		89
Java	152	2	98	0	252		100
Java	177	2	75	0	254		77
Java	138	2	85	0	225		87
Java	137	2	87	0	226		89
Java	138	2	94	0	234		96
Java	153	2	73	1	229		75
Java	137	2	82	0	221		84
Java	144	3	92	0	239		95
Java	138	2	84	1	225		86
Java	138	2	1070	0	1210		1072
Java	153	2	71	0	226		73
Java							
PROMEDIO	174,493	2,987	71,579	0,171	249,229		74,565
DESVIACIÓN	68,102	3,014	53,585	0,384	86,060		53,693

Realizado por: Paucar Rosman, 2018

Métricas Internas Python

Lenguaje	Captura Imagen (ms)	Transmisión Imagen (ms)	Preparación Imagen (ms)	Visualización de la Imagen (ms)	TOTAL	Tiempo de Respuesta
Python	15	73	190	2	280	263
Python	14	75	193	2	284	268
Python	15	67	185	2	269	252
Python	13	63	179	2	257	242
Python	16	71	190	2	279	261
Python	15	63	184	2	264	247
Python	16	72	191	2	281	263
Python	13	76	192	2	283	268
Python	16	71	190	2	279	261
Python	14	78	195	2	289	273
Python	15	81	200	2	298	281
Python	14	75	195	2	286	270
Python	15	77	195	2	289	272
Python	14	77	195	2	288	272
Python	16	70	190	2	278	260
Python	13	77	197	1	288	274
Python	16	75	198	2	291	273
Python	13	78	198	2	291	276
Python	15	77	193	2	287	270
Python	14	73	189	2	278	262
Python	16	81	199	2	298	280
Python	15	83	201	2	301	284
Python	15	74	192	2	283	266
Python	14	73	192	2	281	265
Python	15	82	199	2	298	281
Python	14	72	189	2	277	261
Python	15	72	189	2	278	261
Python	14	68	185	2	269	253
Python	15	83	201	2	301	284
Python	14	79	195	2	290	274
Python	15	85	205	2	307	290
Python	15	74	192	2	283	266
Python	15	81	199	2	297	280
Python	14	73	191	2	280	264
Python	14	78	195	2	289	273
Python	14	71	188	1	274	259
Python	14	80	198	1	293	278

Python	14	74	190	2	280		264
Python	16	75	194	2	287		269
Python	13	78	195	2	288		273
Python	16	68	186	2	272		254
Python	14	70	191	2	277		261
Python	15	76	194	2	287		270
Python	14	70	187	2	273		257
Python	16	78	196	2	292		274
Python	16	66	185	2	269		251
Python	16	78	196	2	292		274
Python	15	62	179	2	258		241
Python	14	72	189	1	276		261
Python	14	83	200	2	299		283
Python	15	82	200	2	299		282
Python	15	79	196	2	292		275
Python	15	78	195	2	290		273
Python	14	80	196	2	292		276
Python	15	74	191	2	282		265
Python	13	73	189	2	277		262
Python	14	78	195	2	289		273
Python	13	84	200	2	299		284
Python	16	80	198	2	296		278
Python	14	84	200	2	300		284
Python	15	84	202	2	303		286
Python	13	79	194	2	288		273
Python	15	71	188	2	276		259
Python	14	78	195	2	289		273
Python	15	79	197	2	293		276
Python	14	65	181	2	262		246
Python	16	71	191	2	280		262
Python	14	63	184	2	263		247
Python	14	82	201	2	299		283
Python	14	83	200	2	299		283
Python	16	85	202	2	305		287
Python	15	68	185	2	270		253
Python	14	70	187	1	272		257
Python	15	84	205	2	306		289
Python	15	82	201	2	300		283
Python	13	78	194	2	287		272
Python	14	80	197	2	293		277
Python	14	84	200	2	300		284

Python	16	70	188	2	276		258
Python	14	73	188	2	277		261
Python	15	75	194	2	286		269
Python	14	77	193	2	286		270
Python	15	78	196	2	291		274
Python	14	83	200	2	299		283
Python	15	76	194	2	287		270
Python	14	78	195	2	289		273
Python	16	83	203	2	304		286
Python	14	76	193	2	285		269
Python	14	81	198	2	295		279
Python	13	79	197	2	291		276
Python	15	77	195	2	289		272
Python	13	77	193	2	285		270
Python	14	74	192	2	282		266
Python	15	76	193	2	286		269
Python	14	76	193	2	285		269
Python	14	66	183	2	265		249
Python	17	82	201	2	302		283
Python	14	76	192	2	284		268
Python	16	75	195	2	288		270
Python	16	73	192	2	283		265
Python	14	64	185	2	265		249
Python	14	72	189	2	277		261
Python	15	72	190	2	279		262
Python	14	79	197	2	292		276
Python	15	82	202	2	301		284
Python	13	84	200	2	299		284
Python	16	66	184	2	268		250
Python	14	84	201	2	301		285
Python	17	74	198	2	291		272
Python	15	75	193	2	285		268
Python	17	74	193	2	286		267
Python	13	74	190	2	279		264
Python	16	70	189	2	277		259
Python	14	67	184	1	266		251
Python	15	73	190	2	280		263
Python	14	74	190	2	280		264
Python	14	82	200	2	298		282
Python	14	77	194	2	287		271
Python	15	84	201	2	302		285

Python	13	71	187	2	273		258
Python	14	80	198	2	294		278
Python	14	76	192	2	284		268
Python	13	77	193	2	285		270
Python	15	79	195	2	291		274
Python	16	72	191	1	280		263
Python	13	75	191	2	281		266
Python	15	82	200	2	299		282
Python	13	79	194	2	288		273
Python	16	81	201	2	300		282
Python	14	77	194	2	287		271
Python	14	69	190	2	275		259
Python	14	71	186	2	273		257
Python	15	65	183	2	265		248
Python	14	83	199	2	298		282
Python	15	68	189	2	274		257
Python	14	77	195	2	288		272
Python	16	65	187	2	270		252
Python	15	80	199	2	296		279
Python	14	74	192	2	282		266
Python	14	72	189	2	277		261
Python	15	79	197	2	293		276
Python	13	85	201	2	301		286
Python	16	75	193	2	286		268
Python	14	74	190	2	280		264
Python	15	76	195	2	288		271
Python	15	72	189	2	278		261
Python	13	70	190	2	275		260
Python	14	82	200	2	298		282
Python	15	77	195	2	289		272
Python	14	66	181	2	263		247
Python	14	72	188	2	276		260
Python	13	73	197	2	285		270
Python	14	83	203	2	302		286
Python	13	73	191	2	279		264
Python	15	82	200	2	299		282
Python	13	73	189	2	277		262
Python	14	70	187	2	273		257
Python	13	81	199	2	295		280
Python	16	73	191	2	282		264
Python	13	83	199	2	297		282

Python	15	72	190	2	279		262
Python	15	81	199	2	297		280
Python	13	64	184	2	263		248
Python	14	74	191	1	280		265
Python	15	65	183	2	265		248
Python	14	78	198	2	292		276
Python	14	65	181	2	262		246
Python	14	66	181	2	263		247
Python	15	78	195	2	290		273
Python	14	74	191	2	281		265
Python	16	68	187	2	273		255
Python	14	69	185	2	270		254
Python	14	76	195	2	287		271
Python	14	78	196	2	290		274
Python	15	82	199	2	298		281
Python	14	78	195	1	288		273
Python	15	79	197	2	293		276
Python	14	73	189	2	278		262
Python	14	71	187	2	274		258
Python	14	85	203	2	304		288
Python	16	69	190	2	277		259
Python	13	67	183	2	265		250
Python	16	71	190	2	279		261
Python	14	71	188	2	275		259
Python	16	67	188	2	273		255
Python	15	75	196	2	288		271
Python	15	74	193	2	284		267
Python	15	64	181	2	262		245
Python	15	74	193	1	283		267
Python	14	82	201	2	299		283
Python	14	77	194	2	287		271
Python	13	82	198	2	295		280
Python	14	67	184	2	267		251
Python	14	72	189	2	277		261
Python	15	67	184	2	268		251
Python	13	73	189	2	277		262
Python	16	71	190	2	279		261
Python	15	77	194	2	288		271
Python	16	64	183	2	265		247
Python	13	68	183	2	266		251
Python	15	77	195	2	289		272

Python	14	69	187	1	271		256
Python	16	81	200	2	299		281
Python	14	73	189	2	278		262
Python	16	72	190	2	280		262
Python	15	78	195	2	290		273
Python	14	71	188	1	274		259
Python	14	68	184	2	268		252
Python	14	77	195	2	288		272
Python	15	81	202	2	300		283
Python	15	76	193	2	286		269
Python	14	83	199	2	298		282
Python	15	83	201	1	300		284
Python	15	82	199	2	298		281
Python	15	79	197	2	293		276
Python	13	78	196	2	289		274
Python	14	76	193	2	285		269
Python	14	80	196	2	292		276
Python	14	75	193	2	284		268
Python	15	67	184	2	268		251
Python	15	69	186	2	272		255
Python	14	76	194	2	286		270
Python	14	72	193	2	281		265
Python	14	70	187	2	273		257
Python	16	62	182	2	262		244
Python	14	85	202	2	303		287
Python	14	68	189	2	273		257
Python	14	61	177	2	254		238
Python	15	63	181	2	261		244
Python	15	72	188	3	278		260
Python	14	75	193	2	284		268
Python	14	76	193	2	285		269
Python	15	63	181	1	260		244
Python	13	76	193	2	284		269
Python	16	74	193	2	285		267
Python	13	78	194	2	287		272
Python	15	70	191	2	278		261
Python	14	73	191	2	280		264
Python	14	74	190	2	280		264
Python	14	65	182	1	262		247
Python	14	74	194	2	284		268
Python	15	67	184	2	268		251

Python	15	73	191	2	281		264
Python	15	77	196	2	290		273
Python	16	78	196	2	292		274
Python	15	72	191	1	279		263
Python	15	81	199	2	297		280
Python	14	69	186	2	271		255
Python	15	83	201	2	301		284
Python	14	73	191	2	280		264
Python	15	79	198	2	294		277
Python	15	81	199	2	297		280
Python	16	63	182	2	263		245
Python	14	77	196	2	289		273
Python	15	75	193	2	285		268
Python	14	85	202	2	303		287
Python	15	75	192	2	284		267
Python	14	81	197	2	294		278
Python	15	76	194	2	287		270
Python	14	80	196	2	292		276
Python	16	82	201	2	301		283
Python	15	80	197	2	294		277
Python	16	62	185	1	264		247
Python	14	73	191	2	280		264
Python	14	81	198	2	295		279
Python	14	75	192	2	283		267
Python	14	65	187	2	268		252
Python	13	71	187	2	273		258
Python	14	81	200	2	297		281
Python	14	83	202	2	301		285
Python	14	75	192	1	282		267
Python	14	81	199	2	296		280
Python	15	84	202	2	303		286
Python	14	78	193	2	287		271
Python	14	80	197	2	293		277
Python	13	78	193	2	286		271
Python	16	72	191	2	281		263
Python	14	69	185	2	270		254
Python	14	78	195	2	289		273
Python	13	75	190	2	280		265
Python	15	68	186	2	271		254
Python	14	68	186	2	270		254
Python	16	74	192	2	284		266

Python	14	71	187	2	274		258
Python	15	71	189	1	276		260
Python	14	75	192	2	283		267
Python	14	82	199	2	297		281
Python	13	76	191	2	282		267
Python	15	75	195	2	287		270
Python	14	76	192	2	284		268
Python	14	69	187	2	272		256
Python	13	71	185	2	271		256
Python	15	81	200	2	298		281
Python	14	82	201	2	299		283
Python	14	70	190	2	276		260
Python	14	75	192	2	283		267
Python	15	87	206	2	310		293
Python	14	68	189	2	273		257
Python	14	79	197	2	292		276
Python	13	75	192	2	282		267
Python	16	81	198	2	297		279
Python	13	74	189	2	278		263
Python	14	78	200	1	293		278
Python	14	75	192	2	283		267
Python	14	68	188	2	272		256
Python	14	69	191	2	276		260
Python	16	70	189	2	277		259
Python	15	71	189	2	277		260
Python	16	62	179	2	259		241
Python	15	80	198	2	295		278
Python	14	72	188	2	276		260
Python	14	72	189	2	277		261
Python	16	67	189	2	274		256
Python	14	79	196	2	291		275
Python	15	76	194	2	287		270
Python	14	64	183	2	263		247
Python	13	83	201	2	299		284
Python	15	82	200	2	299		282
Python	14	83	201	2	300		284
Python	14	79	195	2	290		274
Python	16	90	209	2	317		299
Python	15	79	198	2	294		277
Python	13	63	180	2	258		243
Python	14	60	182	2	258		242

Python	16	68	190	2	276		258
Python	13	70	186	2	271		256
Python	14	74	190	2	280		264
Python	15	75	192	3	285		267
Python	15	75	194	2	286		269
Python	13	77	192	1	283		269
Python	16	81	200	2	299		281
Python	14	82	201	2	299		283
Python	15	67	185	2	269		252
Python	14	74	190	2	280		264
Python	16	69	188	2	275		257
Python	15	78	195	1	289		273
Python	16	72	191	2	281		263
Python	14	77	193	2	286		270
Python	14	81	198	2	295		279
Python	13	73	190	2	278		263
Python	16	73	191	2	282		264
Python	13	68	185	2	268		253
Python	14	84	202	2	302		286
Python	15	80	200	2	297		280
Python	14	78	200	2	294		278
Python	14	69	189	2	274		258
Python	15	84	201	2	302		285
Python	14	66	183	2	265		249
Python	13	81	196	2	292		277
Python	13	84	200	2	299		284
Python	16	75	193	2	286		268
Python	13	71	186	2	272		257
Python	14	66	186	1	267		252
Python	13	80	196	1	290		276
Python	13	78	195	2	288		273
Python	14	70	187	2	273		257
Python	14	68	185	2	269		253
Python	14	82	199	2	297		281
Python	15	76	193	2	286		269
Python	14	64	182	2	262		246
Python	16	62	180	2	260		242
Python	15	63	181	1	260		244
Python	14	72	188	2	276		260
Python	15	62	180	2	259		242
Python	14	82	199	2	297		281

Python	14	72	189	2	277		261
Python	16	71	189	2	278		260
Python	15	75	192	2	284		267
Python	15	73	191	2	281		264
Python	14	72	189	2	277		261
Python	15	85	202	2	304		287
Python	13	79	196	2	290		275
Python	14	76	193	2	285		269
Python	13	74	191	2	280		265
Python	15	79	196	2	292		275
Python							
Python	14,459	74,829	192,605	1,939	283,832		267,435
DESVIACIÓN	0,929	5,900	5,896	0,262	11,731		11,682

Realizado por: Paucar Rosman, 2018

Métricas Externas

Métricas Externas Java

Lenguaje	CPU (%)	RAM (%)	Ancho de Banda (Kb)
Java	71	9,7	298
Java	70,8	9,7	416
Java	70,6	9,7	584
Java	70,7	9,7	601
Java	70,5	9,7	602
Java	70,6	9,7	604
Java	70,4	9,7	170
Java	70,6	9,7	454
Java	70,4	9,8	585
Java	70,5	10,1	523
Java	70,4	10,2	558
Java	70,5	10,3	570
Java	70,4	10,4	311
Java	70,5	10,4	570
Java	70,3	10,4	570
Java	70,4	10,4	566
Java	70,3	10,4	572
Java	70,4	10,4	568
Java	70,5	10,4	571
Java	70,4	10,4	566
Java	70,5	10,4	288
Java	70,3	10,4	369
Java	70,4	10,4	428
Java	70,3	10,4	383
Java	70,4	10,4	408
Java	70,3	10,4	444
Java	70,4	10,4	251
Java	70,3	10,4	403
Java	70,1	10,4	434
Java	70,2	10,4	447
Java	70,1	10,4	440
Java	70,2	10,4	239
Java	70,1	10,4	403
Java	70,2	10,4	428
Java	70,3	10,4	299
Java	70,2	10,4	401
Java	70,3	10,4	436

Java	70,1	10,4	436
Java	70,2	10,4	436
Java	70,1	10,4	400
Java	70,2	10,4	406
Java	70,1	10,4	456
Java	70,2	10,4	574
Java	70	10,4	587
Java	70,1	10,4	527
Java	70	10,5	600
Java	70,1	10,5	500
Java	70	10,5	577
Java	70,1	10,5	548
Java	70	10,5	547
Java	70,1	10,5	549
Java	70	10,5	548
Java	70,1	10,5	529
Java	69,9	10,5	564
Java	70	10,5	574
Java	69,9	10,5	366
Java	70	10,5	567
Java	69,9	10,5	611
Java	70	10,5	565
Java	69,9	10,5	564
Java	70	10,5	587
Java	69,9	10,5	571
Java	70	10,5	591
Java	69,8	10,5	497
Java	69,9	10,5	557
Java	69,8	10,5	561
Java	69,9	10,5	574
Java	69,8	10,5	574
Java	69,9	10,6	574
Java	69,8	10,6	574
Java	69,9	10,6	574
Java	69,8	10,6	523
Java	69,9	10,6	579
Java	69,8	10,6	578
Java	69,9	10,6	578
Java	69,8	10,6	578
Java	69,8	10,6	579
Java	69,7	10,6	579

Java	69,8	10,6	580
Java	69,7	10,6	578
Java	69,8	10,6	579
Java	69,9	10,6	579
Java	69,8	10,6	579
Java	69,9	10,6	578
Java	69,8	10,6	579
Java	69,9	10,6	579
Java	69,8	10,6	578
Java	69,9	10,6	579
Java	69,8	10,6	614
Java	69,8	10,6	574
Java	69,8	10,6	122
Java	69,8	10,6	122
Java	69,7	10,6	122
Java	69,8	10,6	121
Java	69,7	10,6	122
Java	69,8	10,6	122
Java	69,7	10,6	122
Java	69,8	10,6	121
Java	69,7	10,6	122
Java	69,8	10,6	122
Java	69,7	10,6	122
Java	69,7	10,6	122
Java	69,7	10,6	113
Java	69,6	10,6	122
Java	69,7	10,6	122
Java	69,6	10,6	122
Java	69,7	10,6	122
Java	69,6	10,6	130
Java	69,7	10,6	121
Java	69,6	10,6	121
Java	69,7	10,6	122
Java	69,6	10,6	121
Java	69,7	10,6	122
Java	69,6	10,6	122
Java	69,7	10,6	121
Java	69,6	10,6	122
Java	69,6	10,7	131
Java	69,5	10,8	122

Java	69,6	10,8	122
Java	69,6	10,8	122
Java	69,5	10,8	121
Java	69,5	10,8	121
Java	69,4	10,8	122
Java	69,5	10,8	122
Java	69,4	10,8	121
Java	69,5	10,8	121
Java	69,4	10,8	122
Java	69,5	10,8	123
Java	69,4	10,8	121
Java	69,5	10,8	122
Java	69,4	10,8	113
Java	69,4	10,8	121
Java	69,4	10,8	121
Java	69,4	10,8	121
Java	69,4	10,8	122
Java	69,4	10,8	121
Java	69,3	10,8	122
Java	69,4	10,8	121
Java	69,3	10,8	122
Java	69,4	10,8	121
Java	69,3	10,8	122
Java	69,4	10,8	122
Java	69,3	10,8	121
Java	69,4	10,9	122
Java	69,3	10,9	121
Java	69,4	11	122
Java	69,3	11	122
Java	69,4	11	121
Java	69,3	11	122
Java	69,3	11	122
Java	69,3	11	122
Java	69,3	11	122
Java	69,3	11	122
Java	69,3	11	122
Java	69,3	11	122
Java	69,3	11	122
Java	69,2	11	113
Java	69,3	11	123
Java	69,2	11	122
Java	69,3	11	122
Java	69,2	11	113

Java	69,3	11	122
Java	69,4	11	122
Java	69,3	11	123
Java	69,2	11	123
Java	69	11	123
Java	69	11	123
Java	68,9	11	123
Java	69	11	124
Java	68,9	11	123
Java	69	11	114
Java	68,9	11	124
Java	69	11	124
Java	68,9	11	124
Java	69	11	124
Java	68,9	11	124
Java	69	11	124
Java	68,9	11	124
Java	69	11	125
Java	68,9	11	123
Java	68,9	11	115
Java	68,9	11	124
Java	68,9	11	124
Java	69	11	124
Java	68,9	11	124
Java	69	11	121
Java	68,9	11	124
Java	69	11	123
Java	68,9	11	123
Java	69	11	123
Java	68,9	11	123
Java	69	11	122
Java	68,9	11	122
Java	69	11	122
Java	68,9	11	122
Java	69	11	123
Java	69	11	113
Java	68,9	11	123
Java	68,9	11	123
Java	68,9	11	123
Java	68,9	11	122
Java	68,9	11	256
Java	68,9	11	551

Java	68,8	11	562
Java	68,9	11	572
Java	68,8	11	573
Java	68,9	11	511
Java	68,8	11	573
Java	68,9	11	519
Java	68,8	11	566
Java	68,9	11	590
Java	68,8	11	574
Java	68,9	11	357
Java	68,8	11	561
Java	68,9	11	589
Java	68,8	11	381
Java	68,9	11	594
Java	68,8	11	591
Java	68,9	11	576
Java	68,8	11	587
Java	68,9	11	579
Java	68,8	11	580
Java	68,9	11	570
Java	68,8	11	306
Java	68,9	11	569
Java	68,8	11	563
Java	68,9	11	562
Java	68,8	11	563
Java	68,9	11	565
Java	68,8	11	205
Java	68,9	11	612
Java	68,8	11	560
Java	68,8	11	492
Java	68,8	11	586
Java	68,8	11	381
Java	68,8	11	598
Java	68,8	11	595
Java	68,8	11	579
Java	68,8	11	540
Java	68,8	11	469
Java	68,8	11	556
Java	68,8	11	579
Java	68,8	11	588
Java	68,8	11	604

Java	68,8	11	595
Java	68,7	11	580
Java	68,8	11	612
Java	68,7	11	268
Java	68,8	11	325
Java	68,7	11	567
Java	68,8	11	600
Java	68,7	11	558
Java	68,8	11	463
Java	68,7	11	534
Java	68,8	11	528
Java	68,7	11	540
Java	68,8	11	556
Java	68,7	11	602
Java	68,8	11	526
Java	68,7	11	595
Java	68,8	11	613
Java	68,7	11	608
Java	68,8	11	583
Java	68,7	11	580
Java	68,8	11	556
Java	68,7	11	548
Java	68,8	11	570
Java	68,7	11	562
Java	68,8	11	605
Java	68,7	11	601
Java	68,8	11	630
Java	68,7	11	540
Java	68,7	11	602
Java	68,7	11	531
Java	68,7	11	621
Java	68,8	11	616
Java	68,7	11	554
Java	68,8	11	544
Java	68,7	11	570
Java	68,8	11	246
Java	68,7	11	585
Java	68,8	11	587
Java	68,7	11	573
Java	68,7	11	576
Java	68,6	11	501

Java	68,6	11	582
Java	68,6	11	557
Java	68,5	11	578
Java	68,6	11	559
Java	68,5	11	581
Java	68,6	11	551
Java	68,5	11	607
Java	68,6	11	605
Java	68,5	11	574
Java	68,6	11	545
Java	68,5	11	614
Java	68,6	11	539
Java	68,5	11	598
Java	68,6	11	549
Java	68,5	11	562
Java	68,6	11	533
Java	68,5	11	529
Java	68,6	11	599
Java	68,5	11	605
Java	68,6	11	534
Java	68,5	11	570
Java	68,6	11	537
Java	68,5	11	575
Java	68,6	11	635
Java	68,5	11	563
Java	68,6	11	573
Java	68,5	11	563
Java	68,6	11	593
Java	68,5	11	575
Java	68,6	11	593
Java	68,5	11	532
Java	68,6	11	465
Java	68,5	11	576
Java	68,6	11	592
Java	68,5	11	626
Java	68,6	11	544
Java	68,5	11	546
Java	68,6	11	551
Java	68,5	11	571
Java	68,6	11	578
Java	68,5	11	584

Java	68,6	11	591
Java	68,6	11	522
Java	68,6	11	507
Java	68,6	11	615
Java	68,6	11	558
Java	68,6	11	313
Java	68,6	11	622
Java	68,6	11	540
Java	68,6	11	554
Java	68,6	11	586
Java	68,6	11	595
Java	68,6	11	574
Java	68,5	11	572
Java	68,6	11	576
Java	68,5	11	573
Java	68,6	11	572
Java	68,5	11	578
Java	68,6	11	326
Java	68,5	11	530
Java	68,6	11	559
Java	68,5	11	619
Java	68,6	11	557
Java	68,5	11	556
Java	68,6	11	561
Java	68,5	11	541
Java	68,5	11	519
Java	68,5	11	607
Java	68,5	11	582
Java	68,5	11	563
Java	68,4	11	517
Java	68,4	11	531
Java	68,4	11	580
Java	68,4	11	568
Java	68,4	11	541
Java	68,4	11	610
Java	68,4	11	547
Java	68,4	11	583
Java	68,4	11	527
Java	68,5	11	568
Java	68,4	11	572
Java	68,4	11	572

Java	68,4	11,1	567
Java	68,5	11,1	583
Java	68,4	11,1	570
Java	68,5	11,1	530
Java	68,4	11,1	574
Java	68,5	11,1	561
Java	68,4	11,1	586
Java	68,4	11,1	555
Java	68,4	11,1	614
Java	68,4	11,1	616
PROMEDIO	69,198	10,814	417,301
DESVIACIÓN	0,638	0,287	202,276

Realizado por: Paucar Rosman, 2018

Métricas Externas Python

Lenguaje	CPU (%)	RAM (%)	Ancho de Banda (Kb)
Python	35,6	8,1	7080
Python	35,6	8,2	7090
Python	35,6	8,2	10600
Python	35,6	8,2	7090
Python	35,6	8,1	7090
Python	35,6	8,2	7090
Python	35,6	8,1	10600
Python	35,6	8,2	7090
Python	35,6	8,2	7100
Python	35,6	8,1	9420
Python	35,6	8,2	7090
Python	35,6	8,1	10600
Python	35,6	8,2	7090
Python	35,6	8,2	7090
Python	35,6	8,1	10600
Python	35,6	8,2	10600
Python	35,6	8,1	7090
Python	35,6	8,2	7100
Python	35,6	8,2	6320
Python	35,6	8,1	7090
Python	35,6	8,2	10600
Python	35,6	8,2	7080
Python	35,6	8,1	10600
Python	35,7	8,2	7090
Python	35,6	8,1	10600
Python	35,7	8,1	7100
Python	35,7	8,2	10600
Python	35,7	8,2	7080
Python	35,7	8,1	7100
Python	35,7	8,2	7090
Python	35,7	8,1	7080
Python	35,7	8,1	6340
Python	35,7	8,2	6290
Python	35,7	8,1	10600
Python	35,7	8,1	10600
Python	35,7	8,2	7090
Python	35,7	8,1	7090
Python	35,7	8,2	7090

Python	35,7	8,2	7070
Python	35,7	8,1	10600
Python	35,7	8,2	7090
Python	35,8	8,2	10600
Python	35,8	8,1	7090
Python	35,7	8,1	7090
Python	35,8	8,2	7090
Python	35,8	8,2	6260
Python	38,1	8,1	10600
Python	38,2	8	7090
Python	38,1	8	7090
Python	38,2	8,1	10700
Python	38	8	7090
Python	38,2	8	7090
Python	38	8,1	10600
Python	38,2	8	7090
Python	38	8	7090
Python	38,2	8,1	7080
Python	38,1	8	10600
Python	38,2	8,1	7100
Python	38	8,1	7080
Python	38,2	8	6340
Python	38,1	8,1	10600
Python	38,2	8	7100
Python	38	8,1	7540
Python	38,1	8,1	7090
Python	38	8	10600
Python	38,1	8,1	7100
Python	38	8	10600
Python	38,1	8,1	7090
Python	38	8,1	7080
Python	37,9	8	7100
Python	38	8,1	7090
Python	37,9	8	7080
Python	38	8	10600
Python	37,9	8,1	7090
Python	38	8	6570
Python	37,9	8	10600
Python	38	8,1	7110
Python	37,9	8	7090
Python	38	8,1	7090

Python	37,9	8,1	7090
Python	38	8	7090
Python	37,9	8,1	7090
Python	37,9	8	7090
Python	37,8	8	10600
Python	37,9	8,1	7090
Python	37,8	8	10600
Python	37,9	8,1	10600
Python	37,8	8	7090
Python	37,9	8,1	10600
Python	37,8	8,1	6330
Python	37,9	8	10600
Python	37,8	8,1	7090
Python	37,9	8	7110
Python	37,8	8	7080
Python	37,9	8,1	7100
Python	37,8	8	10600
Python	37,9	8	10200
Python	37,8	8,1	10500
Python	37,9	8	7090
Python	37,8	8,1	7080
Python	37,7	8	10600
Python	37,8	8,1	7100
Python	37,7	8,1	7090
Python	37,7	8	6320
Python	37,7	8,1	10600
Python	37,7	8	7090
Python	37,7	8,1	7070
Python	37,7	8,1	7090
Python	37,7	8	7090
Python	37,7	8,1	10600
Python	37,6	8,1	7090
Python	37,5	8,1	10600
Python	37,6	8	7090
Python	37,6	8,1	7080
Python	37,6	8,1	7100
Python	37,6	8	7110
Python	37,7	8,1	7090
Python	37,6	8	6070
Python	37,6	8,1	7080
Python	37,6	8,1	7090

Python	37,6	8	7090
Python	37,6	8,1	7090
Python	37,6	8,1	7090
Python	37,5	8	7090
Python	37,6	8,1	10600
Python	37,5	8	7080
Python	37,6	8,1	10600
Python	37,5	8,1	7090
Python	37,6	8	7080
Python	37,6	8,1	7080
Python	37,6	8,1	10600
Python	37,6	8	7100
Python	37,6	8,1	7100
Python	37,5	8,1	7090
Python	37,6	8	7090
Python	37,6	8,1	10600
Python	37,5	8	7090
Python	37,5	8	7080
Python	37,5	8,1	7100
Python	37,5	8	10600
Python	37,5	8,1	10600
Python	37,5	8,1	10600
Python	37,5	8	10600
Python	37,5	8,1	7090
Python	37,5	8	10600
Python	37,5	8	10600
Python	37,5	8,1	6590
Python	37,6	8	10600
Python	37,5	8,1	10700
Python	37,5	8,1	10600
Python	37,4	8,1	7090
Python	37,4	8,1	10600
Python	37,3	8	7090
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,2	8	7090
Python	37,2	8	10600
Python	37,3	8	7030
Python	37,2	8	10600
Python	37,3	8,1	7080

Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,2	8,1	10600
Python	37,1	8	7100
Python	37,2	8,1	10600
Python	37,1	8	7090
Python	37,2	8,1	10600
Python	37,1	8,1	7090
Python	37,2	8	10600
Python	37,2	8	10600
Python	37,2	8,1	6350
Python	37,2	8	10600
Python	37,2	8,1	10600
Python	37,2	8,1	7090
Python	37,2	8	10600
Python	37,2	8,1	10600
Python	37,3	8,1	7090
Python	37,2	8	7090
Python	37,3	8,1	10600
Python	37,2	8,1	10600
Python	37,3	8	7090
Python	37,2	8,1	7090
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8,1	7080
Python	37,2	8	7090
Python	37,3	8	10600
Python	37,2	8,1	10600
Python	37,3	8,1	7090
Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,3	8,1	10600
Python	37,3	8	7080
Python	37,3	8,1	7090
Python	37,3	8	5540
Python	37,3	8,1	7090
Python	37,3	8,1	7090
Python	37,2	8,1	7100
Python	37,3	8	10600
Python	37,3	8,1	7100
Python	37,3	8,1	10600

Python	37,3	8	7090
Python	37,2	8,1	10600
Python	37,3	8	10600
Python	37,2	8,1	10600
Python	37,3	8,1	7070
Python	37,2	8	10600
Python	37,3	8,1	7090
Python	37,3	8,1	10600
Python	37,3	8	5470
Python	37,3	8	7100
Python	37,3	8,1	10600
Python	37,3	8	10600
Python	37,3	8,1	7070
Python	37,3	8	7090
Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8	10600
Python	37,2	8,1	10600
Python	37,3	8	10600
Python	37,2	8,1	10600
Python	37,3	8	10600
Python	37,2	8,1	7090
Python	37,3	8,1	7090
Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,2	8	7070
Python	37,3	8,1	7090
Python	37,2	8,1	7090
Python	37,3	8	10600
Python	37,3	8,1	7090
Python	37,2	8,1	7090
Python	37,3	8	7090
Python	37,2	8,1	10600
Python	37,3	8	7090
Python	37,2	8	7080
Python	37,3	8,1	10600
Python	37,2	8	7090
Python	37,3	8,1	10700
Python	37,2	8,1	10600
Python	37,3	8	10600
Python	37,2	8,1	7100
Python	37,3	8,1	8960

Python	37,3	8	9620
Python	37,3	8,1	7090
Python	37,3	8,1	10600
Python	37,3	8	5970
Python	37,3	8,1	7100
Python	37,3	8	10600
Python	37,3	8,1	7080
Python	37,3	8,1	10600
Python	37,3	8	7080
Python	37,3	8,1	10600
Python	37,3	8	7100
Python	37,2	8	10600
Python	37,3	8,1	7090
Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8	7090
Python	37,2	8,1	10600
Python	37,3	8	3610
Python	37,3	8	10600
Python	37,3	8,1	10600
Python	37,3	8	10600
Python	37,3	8,1	7090
Python	37,3	8,1	7080
Python	37,3	8	10600
Python	37,3	8,1	7090
Python	37,3	8,1	7090
Python	37,3	8	7090
Python	37,3	8,1	7100
Python	37,3	8	10600
Python	37,3	8,1	7090
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8,1	6740
Python	37,2	8	10600
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8	10600
Python	37,3	8,1	7080
Python	37,3	8	10600
Python	37,3	8	10600

Python	37,3	8,1	10600
Python	37,3	8	7080
Python	37,3	8,1	10600
Python	37,3	8	10600
Python	37,3	8	7090
Python	37,2	8,1	7090
Python	37,3	8	6950
Python	37,2	8,1	6930
Python	37,3	8,1	10600
Python	37,2	8	7090
Python	37,3	8,1	7090
Python	37,3	8	10100
Python	37,3	8	10600
Python	37,3	8,1	10600
Python	37,3	8	10600
Python	37,3	8	7090
Python	37,2	8,1	10700
Python	37,3	8	7090
Python	37,2	8,1	10600
Python	37,3	8	7100
Python	37,2	8,1	10600
Python	37,3	8,1	10600
Python	37,2	8	10600
Python	37,3	8,1	6730
Python	37,3	8,1	10600
Python	37,3	8	6350
Python	37,3	8,1	8010
Python	37,3	8	7090
Python	37,3	8,1	10600
Python	37,3	8,1	10600
Python	37,3	8,1	7090
Python	37,3	8,1	7080
Python	37,2	8,1	10600
Python	37,2	8	10600
Python	37,1	8,1	10600
Python	37,2	8	7080
Python	37,1	8	10600
Python	37,2	8,1	10600
Python	37,1	8	10600
Python	37,2	8	7090
Python	37,1	8,1	6650

Python	37,2	8	10600
Python	37,1	8,1	7090
Python	37,2	8,1	7090
Python	37,2	8	10600
Python	37,2	8,1	10600
Python	37,2	8	7090
Python	37,2	8,1	7090
Python	37,2	8,1	10600
Python	37,2	8	7090
Python	37,2	8,1	10600
Python	37,2	8,1	7080
Python	37,2	8	10600
Python	37,2	8,1	10600
Python	37,2	8	10600
Python	37,2	8,1	6020
Python	37,2	8	6080
Python	37,2	8,1	7110
Python	37,2	8	7090
Python	37,2	8,1	7090
Python	37,2	8	10700
Python	37,2	8	10600
Python	37,2	8,1	7080
Python	37,2	8,1	7110
Python	37,2	8	10600
Python	37,2	8,1	10600
Python	37,2	8	8420
Python	37,2	8	7090
Python	37,2	8,1	10600
Python	37,2	8	10600
Python	37,2	8	10600
Python	37,2	8,1	6260
Python	37,2	8	10600
Python	37,2	8,1	8850
Python	37,2	8	10600
Python	37,2	8	7090
Python	37,2	8,1	10600
Python	37,2	8	10600
Python	37,2	8,1	7090
Python	37,2	8,1	7100
Python	37,2	8	10600
Python	37,2	8,1	10600

Python	37,2	8,1	7090
Python	37,2	8	7090
Python	37,2	8,1	10700
Python	37,2	8,1	10600
Python	37,2	8	3320
Python	37,2	8,1	7090
Python	37,2	8	4190
Python	37,2	8	10600
Python	37,2	8,1	7090
promedio	37,202	8,066	8579,520
desviación	0,639	0,060	1834,499

Realizado por: Paucar Rosman, 2018

Ancho de Banda

Captura con el comando IFTOP del Ancho de Banda para Java

The screenshot shows the MobaXterm application window. On the left, there is a file explorer for the /home/pi directory. The main terminal window displays the output of the iftop command, showing network traffic between raspberry1.mshome.net and DESKTOP-C5UT2CG.mshome.net. At the bottom of the terminal, a summary table provides cumulative and peak statistics for the session.

	1.91Mb	3.81Mb	5.72Mb	7.63Mb	9.54Mb
raspberrypi.mshome.net					
239.255.255.250	⇒	DESKTOP-C5UT2CG.mshome.net	335Kb	310Kb	325Kb
192.168.137.255	⇐	DESKTOP-C5UT2CG.mshome.net	9.59Kb	9.63Kb	10.2Kb
	⇒	DESKTOP-C5UT2CG.mshome.net	0b	0b	161b
	⇐	DESKTOP-C5UT2CG.mshome.net	0b	0b	0b
	⇒	DESKTOP-C5UT2CG.mshome.net	0b	0b	46b
	⇐	DESKTOP-C5UT2CG.mshome.net	0b	0b	46b

TX:	cum:	6.67MB	peak:	376Kb	rates:	335Kb	310Kb	325Kb
RX:		249KB		13.6Kb		9.59Kb	9.63Kb	10.4Kb
TOTAL:		6.91MB		388Kb		345Kb	320Kb	335Kb

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figura 31-2: Captura comando iftop ancho de banda Java

Realizado por: Paucar Rosman, 2018

Captura con el comando IFTOP del Ancho de Banda para Python

The screenshot shows a MobaXterm window with a terminal running the iftop command. The terminal output displays network traffic statistics for the local host (raspberrypi.mshome.net) and its connections to DESKTOP-C5UT2CG.mshome.net. The output is as follows:

	19.1Mb	38.1Mb	57.2Mb	76.3Mb	95.4Mb
raspberrypi.mshome.net =>	DESKTOP-C5UT2CG.mshome.net	7.19Mb	7.76Mb	8.39Mb	
192.168.137.255 <=>	DESKTOP-C5UT2CG.mshome.net	95.4Kb	105Kb	113Kb	
239.255.255.250 <=>	DESKTOP-C5UT2CG.mshome.net	0b	0b	0b	
		0b	184b	92b	
		0b	0b	0b	
		0b	0b	161b	
<hr/>					
TX:	cum: 324MB	peak: 10.8Mb	rates: 7.19Mb	7.76Mb	8.39Mb
RX:	4.27MB	150Kb	95.4Kb	105Kb	113Kb
TOTAL:	329MB	10.9Mb	7.28Mb	7.86Mb	8.50Mb

At the bottom of the terminal window, there is a notice: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>"

Figura 32-2: Captura comando iftop ancho de banda Python

Realizado por: Paucar Rosman, 2018

Anexo D: Desarrollo historias técnicas del sistema

Tabla 5-XV: Requisitos del Sistema

HISTORIA DE USUARIO	
ID: HT01	Nombre de la Historia: Requisitos del Sistema
Responsable: Team	Sprint: 01
Fecha Inicio: 01/03/2018	Fecha Fin: 30/03/2018
Descripción: Documento de Análisis de Requerimientos	
Pruebas de Aceptación: <ul style="list-style-type: none">• Reunión para establecer el alcance• Reunión para establecer los requisitos funcionales• Reunión para establecer los requisitos no funcionales	

Realizado por: Paucar Rosman, 2018

Tabla 5-XVI: Instalación Servidor

HISTORIA DE USUARIO	
ID: HT02	Nombre de la Historia: Instalación Servidor
Responsable: Team	Sprint: 01
Fecha Inicio: 01/03/2018	Fecha Fin: 30/03/2018
Descripción: Instalar el servidor para almacenamiento de videos	
Pruebas de Aceptación: <ul style="list-style-type: none">• Instalar Centos 7• Actualizar el sistema• Instalar la plataforma Java versión 8	

Realizado por: Paucar Rosman, 2018

Tabla 5-XVII: Diseño Global

HISTORIA DE USUARIO	
ID: HT03	Nombre de la Historia: Diseño Global
Responsable: Team	Sprint: 01
Fecha Inicio: 01/03/2018	Fecha Fin: 30/03/2018
Descripción: Diseño Global de la Solución	
Pruebas de Aceptación: <ul style="list-style-type: none">• Identificación de los actores• Identificación de casos de uso• Crear el documento del bosquejo general de la base de datos de la solución	

Realizado por: Paucar Rosman, 2018

Tabla 5-XVIII: Capa de Acceso a datos

HISTORIA DE USUARIO	
ID: HT04	Nombre de la Historia: Capa de Acceso a datos
Responsable: Team	Sprint: 01
Fecha Inicio: 01/03/2018	Fecha Fin: 30/03/2018
Descripción: Desarrollo de la capa de acceso a datos	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Desarrollo de la capa de acceso a datos • Conexión a la Base • Verificación 	

Realizado por: Paucar Rosman, 2018

Tabla 5-XIX: Subversión

HISTORIA DE USUARIO	
ID: HT05	Nombre de la Historia: Subversión
Responsable: Team	Sprint: 01
Fecha Inicio: 02/04/2018	Fecha Fin: 30/04/2018
Descripción: Instalación del Sistema de Control de Versiones (Subversión)	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Instalación de Apache • Instalación del Módulo Mod_subversión • Instalación de Subversión 	

Realizado por: Paucar Rosman, 2018

Tabla 5-XX: Diseño Conceptual

HISTORIA DE USUARIO	
ID: HT06	Nombre de la Historia: Diseño Conceptual
Responsable: Team	Sprint: 01
Fecha Inicio: 01/03/2018	Fecha Fin: 30/03/2018
Descripción: Diseño Global de la Solución	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Crear un documento con el diseño conceptual de la solución • Verificación y validación del documento 	

Realizado por: Paucar Rosman, 2018

Tabla 5-XXI: Documentación

HISTORIA DE USUARIO	
ID: HT07	Nombre de la Historia: Documentación
Responsable: Team	Sprint: 01
Fecha Inicio: 02/04/2018	Fecha Fin: 30/04/2018
Descripción: Documentación del sistema	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Manual técnico • Manual de usuario 	

Realizado por: Paucar Rosman, 2018

Tabla 5-XXII: Instalación Raspbian

HISTORIA DE USUARIO	
ID: HT08	Nombre de la Historia: Instalación Raspbian
Responsable: Team	Sprint: 01
Fecha Inicio: 02/04/2018	Fecha Fin: 30/04/2018
Descripción: Instalación del Sistema Operativo Raspbian	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Instalación del GRUB en la microSD de Raspberry • Instalación del Sistema Operativo Raspbian • Actualización del Sistema Operativo 	

Realizado por: Paucar Rosman, 2018

Tabla 5-XXII: Instalación plataformas

HISTORIA DE USUARIO	
ID: HT09	Nombre de la Historia: Instalación Plataformas
Responsable: Team	Sprint: 01
Fecha Inicio: 02/04/2018	Fecha Fin: 30/04/2018
Descripción: Instalación de las plataformas de desarrollo	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Instalación JDK Java en el servidor • Instalación Netbeans en el servidor • Instalación JDK Java en el cliente 	

Realizado por: Paucar Rosman, 2018

Tabla 5-XXIII: Configuración Raspberry

HISTORIA DE USUARIO	
ID: HT09	Nombre de la Historia: Configuración Raspberry
Responsable: Team	Sprint: 01
Fecha Inicio: 02/04/2018	Fecha Fin: 30/04/2018
Descripción: Configuración de dispositivos de red y webcam en la Placa Computadora (Raspberry)	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Instalación de drivers de tarjeta de red Wireless USB en Raspbian • Instalación de drivers de la cámara web en Raspbian 	

Realizado por: Paucar Rosman, 2018