



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE INGENIERÍA DE SISTEMAS

**CONSTRUCCIÓN DE UN SISTEMA MÓVIL BASADO EN AGENTES PARA EL
DEPARTAMENTO DE RADIOLOGÍA, QUE INTEGRE SISTEMAS HIS Y PACS**

Trabajo de titulación previo a la obtención del

Título de Ingeniero de Sistemas

AUTOR: EDWIN MAURICIO HUERTAS TOAPAXI

TUTOR: WASHINGTON ARSENIO RAMÍREZ MONTALVAN

Quito – Ecuador

2022

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Edwin Mauricio Huertas Toapaxi, con documento de identificación No 1718938853, manifestó que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 06 de abril del 2022

Atentamente,

A handwritten signature in blue ink, appearing to read 'Edwin Huertas Toapaxi', written in a cursive style.

Edwin Mauricio Huertas Toapaxi

1718938853

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Edwin Mauricio Huertas Toapaxi, con documento de identificación No 1718938853, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy el autor del Proyecto Técnico: “Construcción de un sistema móvil basado en agentes para el departamento de radiología, que integre sistemas HIS y PACS”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 6 abril del 2022

Atentamente,



Edwin Mauricio Huertas Toapaxi

1718938853

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Washington Arsenio Ramírez Montalvan con documento de identidad No 1710804681, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: CONSTRUCCIÓN DE UN SISTEMA MÓVIL BASADO EN AGENTES PARA EL DEPARTAMENTO DE RADIOLOGÍA, QUE INTEGRE SISTEMAS HIS Y PACS, realizado por Edwin Mauricio Huertas Toapaxi, con documento de identificación No 1718938853, obteniendo como resultado final el trabajo de titulación bajo la opción de proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 6 de abril del 2022



Ing. Washington Arsenio Ramírez Montalvan Phd.

1710804681

DEDICATORIA

Dedico el presente trabajo a Dios. A mi madre Blanca Toapaxi y a mi abuelito Miguel Ángel Toapaxi, que con su infinito amor me apoyaron y motivaron a culminar una meta propuesta hacía mucho tiempo atrás en el lugar que estén, se que siempre están a mi lado. Mi tía Gloria, mi prima Betty las cuales con su apoyo me enseñaron a perseguir mis objetivos y no rendirme. A mi esposa Irene que es la persona más importante de mi vida que con su amor y comprensión me permitió alcanzar este sueño. Y a toda su familia que me han brindado palabras de aliento y que siempre se han preocupado por mí. Gracias a todos por su apoyo, ya que son y se han convertido en el CNS motor de mi vida.

INDICE GENERAL

INDICE GENERAL	6
INDICES DE TABLAS	9
INDICES DE FIGURAS	9
RESUMEN	11
ABSTRACT	12
CAPÍTULO 1	1
INTRODUCCIÓN	1
1.1. ANTECEDENTES	1
1.2. PROBLEMA	3
1.3. JUSTIFICACIÓN	5
1.4. OBJETIVOS	8
1.4.1. <i>Objetivo General</i>	8
1.4.2. <i>Objetivos específicos</i>	8
1.5. MARCO METODOLÓGICO	8
CAPÍTULO 2	10
FUNDAMENTO TEÓRICO	10
2.1. AGENTES	10
2.2. SISTEMAS MULTIAGENTES	12
2.3. JAVA AGENT DEVELOPMENT FRAMEWORK (JADE)	13
2.4. JADE LEAP	14

2.5.	FLUTTER.....	15
2.5.1.	<i>Funcionalidades</i>	16
2.6.	HIS.....	16
2.7.	RIS & PACS	18
2.8.	HL7.....	19
2.9.	DICOM.....	20
CAPÍTULO 3		21
	METODOLOGÍA	21
3.1.	PRERREQUISITOS EN HARDWARE	22
3.2.	ONTOLOGÍA DE AGENTES	22
3.3.	INTERCAMBIO DE INFORMACIÓN.....	26
3.4.	SOLICITUDES CON LA PÁGINA	34
3.5.	SOLICITUD CON EL CELULAR	40
CAPÍTULO 4		43
	RESULTADOS.....	43
4.1.	HERRAMIENTA JMETER.....	43
4.2.	CARGA Y ESTRÉS EN EL USO DEL CPU Y MEMORIA DEL COMPUTADOR	43
4.2.1.	<i>Prueba 1: Uso del CPU</i>	43
4.2.2.	<i>Prueba 2: Uso de la memoria</i>	46
4.3.	MÉTRICAS DE TIEMPOS DE RESPUESTAS.....	48
4.4.	MÉTRICAS DE CALIDAD DEL PROYECTO.....	50
4.4.1.	<i>Código duplicado</i>	50
4.4.2.	<i>Código muerto</i>	50

4.4.3. <i>Estándares de codificación</i>	50
4.4.4. <i>Bugs</i>	51
CONCLUSIONES Y RECOMENDACIONES	54
5.1. CONCLUSIONES.....	54
5.2. RECOMENDACIONES	55
REFERENCIAS BIBLIOGRÁFICAS	57

INDICES DE TABLAS

Tabla 1	17
Tabla 2	51

INDICES DE FIGURAS

Figura 1	11
Figura 2	15
Figura 3	19
Figura 4	24
Figura 5	25
Figura 6	26
Figura 7	32
Figura 8	33
Figura 9	34
Figura 10	35
Figura 11	36
Figura 12	41
Figura 13	44
Figura 14	45
Figura 15	45
Figura 16	46
Figura 17	47
Figura 18	47

Figura 19	48
Figura 20	49
Figura 21	49
Figura 22	52
Figura 23	53
Figura 24	53

RESUMEN

El presente proyecto de investigación que tiene como título la construcción de un sistema móvil basado en agentes para el departamento de radiología que integre los sistemas Health Information System (HIS, por sus siglas en inglés) y Picture Archiving and Communication System (PACS, por sus siglas en inglés) que fue previamente desarrollada por (Ortiz Jácome & Noroña García, 2019). Con el objetivo principal de permitir la comunicación entre estos sistemas utilizando multiagentes, con herramientas de código abierto para la optimización de procesos de almacenamiento, distribución y gestión de la información de los pacientes por parte del personal médico hospitalario. Los Agentes son tareas que se ejecutan cada cierto tiempo y realizan varias acciones, de forma asíncrona sin consumir muchos recursos del computador. La solución se construyó en tres partes, una aplicación para que sea el contenedor principal de los agentes; la segunda parte es la implementación de un sistema web para que nos permita crear, modificar los pedidos de los estudios, y poder visualizar los resultados que se hicieron al paciente; como tercera parte se implementó una aplicación móvil con la finalidad de que el médico puedan ver desde el celular los resultados que se hizo al paciente. Finalmente, el entregable es un sistema que permita ayudar a los médicos para poder realizar un análisis más oportuno de los resultados del departamento de radiología realizados a los pacientes.

ABSTRACT

The present research project whose title is the construction of a mobile system based on agents for the radiology department that integrates the Health Information System (HIS) and the Picture Archiving and Communication System (PACS, for its acronym in English). acronym in English) that was previously developed by (Ortiz Jácome & Noroña García, 2019). With the main objective of allowing communication between these systems using multi-agents, with open-source tools for the optimization of storage processes, distribution and management of patient information by hospital medical staff. Agents are tasks that are executed from time to time and perform various actions, asynchronously without consuming many computer resources. The solution was built in three parts, an application to be the main container for the agents; the second part is the implementation of a web system that allows us to create, modify the orders of the studies, and be able to visualize the results that were made to the patient; As a third party, a mobile application was implemented so that the doctor can see the results that were done to the patient from the cell phone. Finally, the deliverable is a system that helps physicians to perform a timelier analysis of the results of the radiology department performed on patients.

CAPÍTULO 1

Introducción

1.1. Antecedentes

El artículo de opinión de (Dibarboure, Sgarbi, & Febles, 2021) señala a la Inteligencia Artificial (IA), el Biomodelado 3D (BM) como las nuevas tecnologías que se han consolidado dentro de la medicina y la Imagenología. Estas incidirán en la Imagenología en algunos escenarios que son: la cuantificación y estratificación de riesgo, la complementación y la optimización de métricas operativas. Este artículo representa un excelente aporte para esta investigación, ya que nos ofrece la visión de la importancia que tienen las nuevas tecnologías en el área de la salud. A demás se enfatiza en que se debe enfocar la ciencia y la tecnología para trabajar en la solución de los problemas de salud de la sociedad.

Por otro lado, se encuentra el estudio de (Llontop, 2019) su objetivo se baso en el análisis del programa de archivos de salud que estaba relacionado con los tecnólogos del departamento de radiología. Como principal conclusión que es esencial el conocimiento y dominio de la web en la Telerradiología con la metodología formativa en el sistema PACS dirigido a los tecnólogos médicos del servicio de emergencia. A demás se concluye que sus principales beneficios es dar información segura, eficiente y pertinente en el tiempo.

Posteriormente, se analizó el estudio de (Dibarboure, Sgarbi, & Febles, 2021) donde se plantea la importancia de una solicitud de estudio de imagen bien realizada, así como del informe radiológico y la importancia del PACS, referente a los sistemas de archivo y comunicación de imágenes, facilitando la interpretación de imágenes en tiempos eficaces y ofrece apoyo a “subespecialidades” en Imagenología; mejora las prácticas educativas de los

médicos imagenólogos; promueve la eficacia, la calidad y sirve para desarrollar y perfeccionar la Telemedicina.

El trabajo investigativo de (Rodríguez, 2014) que uno de sus objetivos era el de definir nuevas sugerencias para la gestión de los datos del paciente, a través de los métodos de estandarización, con el fin de depurar las historias clínicas, así como la interrelación entre datos estadísticos y clínicos, para evitar la información duplicada. Empleando como metodología el Rational Unified Process (RUP, por sus siglas en inglés) donde se utiliza el Modelo Integrado de Capacidad y Madurez (CMMI, por sus siglas en inglés) con el fin de evaluar y mejorar los procesos.

Los resultados que indica (Rodríguez, 2014) es que al utilizar estándares como Health Level Seven International (HL7, por sus siglas en inglés) dirigidos por los perfiles de Integrating the Healthcare Enterprise (IHE, por sus siglas en inglés), indica que se puede disminuir la duplicidad de la información, en la tercera versión del módulo de Gestión de Pacientes del sistema alas Radiology Information System (RIS, por sus siglas en inglés) indica que para un mejor funcionamiento es necesario la existencia de un PACS para su integración, el sistema es funcional en cualquier sistemas operativos.

Como una conclusión (Rodríguez, 2014) que se plantea son las funcionalidades que existen en el módulo son de gran ayuda para los procesos para cubrir las necesidades de los hospitales implementando la arquitectura de diseño Model View Controller (MVC, por sus siglas en inglés), con el fin de garantizar que cada clase cumpla solo con la responsabilidad asignada, cabe indicar que existen nuevas inquietudes que aparecieron para una posterior implementación.

También, se consultó la investigación de (Mazzoncini, Marques, Elías, & Santos, 2011) la cual indica que la recuperación de imágenes de manera ágil y rápida en la base de datos facilita comparar información relevante relacionada con el examen. Todo esto resulta esencial para nuestro trabajo investigativo, por todos los beneficios que aporta, además de la información teórica que permite recopilar para la profundización del tema.

1.2. Problema

(Llontop, 2019) indica que el tecnólogo médico en radiología tiene entre sus funciones más importantes, la asistencial que desempeña a diario en las áreas de radiodiagnóstico. Donde uno de sus objetivos es mejorar la salud del paciente así mismo su rehabilitación. Sin embargo, también debe realizar otras tareas, las cuales son complejas, debido a la actualización constante de la ciencia y la tecnología. Por tanto, esta situación les exige capacitarse constantemente.

Con la actualización del sistema de salud, unido a la tecnología, se emplea de manera, cada vez más común los exámenes radiológicos que emplean sistemas bivalentes de película, sin embargo, estos resultan muy costosos en comparación con los sistemas sin películas, conocidos como PACS. Siendo capaz de reducir los costos y sustituir los exámenes radiológicos físicos. Por lo tanto, optimiza la atención del paciente en emergencia. (Dibarboure L. , 2016) con el avance de la tomografía computarizada, con multi detector y de resonancia magnética, ha existido un incremento exponencial de las imágenes que se obtienen para los estudios de salud; lo que ha traído como consecuencia que no sea viable documentar en películas radiográficas cada estudio. En un momento determinado, se seleccionaron imágenes y estas se entregaban a los pacientes, pero esto aumentaba significativamente los costos.

De esta manera, señala (Dibarboure L. , 2016) que los PACS permiten archivar las imágenes, desde donde los médicos imagenólogos comunican los estudios con estaciones de trabajo que tienen herramientas de visualización, reconstrucción de imágenes y post procesamiento. Por otro lado, poseen la ventaja de que se pueden enviar las imágenes a diferentes puntos de los hospitales o instituciones, ya sean consultorios, salas internación, block quirúrgico y, así mismo a distancia, posibilitando la telerradiología. De esta forma, se incorporan las imágenes a la historia clínica electrónica de los pacientes.

De igual modo, el lenguaje en común que emplean estas imágenes se denomina Digital Imaging and Communication in Medicine (DICOM, por sus siglas en inglés), por tanto, no existen inconvenientes para su lectura, ya que todas las imágenes de tomografía computada y resonancia magnética son digitales por lo que se pasan directamente al PACS independientemente de la marca comercial de los equipos. En la actualidad, si un Servicio de imagenología aspira o pretende a tener buena calidad debe informar todos los estudios que en él se realicen, en forma oportuna y con un contenido adecuado brindando el menor riesgo posible a sus pacientes y al personal con los mejores resultados.

Según (Health Level Seven International, 2019) los PACS permiten almacenar las imágenes, desde donde los radiólogos almacenan los estudios radiológicos con los equipos de trabajo utilizando herramientas de visualización, recuperación de imágenes y post procesamiento. Este medio, permite la trasmisión de las imágenes a diferentes áreas en los hospitales o instituciones que lo requieran e incorporan los estudios a las historias clínicas electrónicas de los pacientes.

Por otro lado, se debe mencionar el aspecto que posibilita la comunicación y conectividad entre las herramientas que integran la atención hospitalaria. Para (Martínez,

Vega, & Ferras, 2014) siendo RIS capaz de combinar los departamentos de imágenes y la información del paciente, para gestionar desde el momento que ingresan al centro hospitalario.

En la actualidad, la comunicación entre los módulos de la historia clínica HIS y de los estudios radiológicos PACS al momento carece de un medio que permitan enlazar ambas bases de datos y una interfaz que permita una adecuada administración, para la distribución de las imágenes, presentación de resultados y procesos administrativos integrando en un sistema HIS/PACS. Es por esto, que resulta esencial crear un módulo y una aplicación móvil que permita la comunicación entre los sistemas de HIS y PACS con sistemas multiagentes, usando herramientas de código abierto.

1.3. Justificación

La mayor importancia de este trabajo radica en la radiología y la información que es posible obtener mediante esta, son de gran ayuda para el diagnóstico y tratamiento de los pacientes. La mayor importancia de este trabajo radica en la radiología y la información que es posible obtener mediante esta, son de gran ayuda para el diagnóstico y tratamiento de los pacientes. A demás funciona como un método inigualable para mejorar la calidad del trabajo del tecnólogo.

De igual manera, en la cotidianeidad y la práctica médica, de los PACS agrandan valor a las instituciones hospitalarias. Entre las principales ventajas que este ofrece se encuentra es la disminución de tiempos y el almacenamiento físico que justifican la utilización de sistemas. La principal motivación para realizar este trabajo consiste en perfeccionar y hacer más eficiente el trabajo del personal médico de Radiología, sobre todo

en las funciones asistenciales, donde se requiere una actualización constante y una mejoría en la calidad de las imágenes radiológicas.

Por otro lado, los principales beneficiarios de este trabajo serían los profesionales de la radiología que podrían trabajar con herramientas más modernas y, por tanto, eficientes. Además, también se beneficiarían los pacientes pues con esto pueden recibir una atención médica de calidad, rapidez y eficiencia. También se benefician las instituciones hospitalarias, en general, pues los costos disminuirían y la calidad de su centro estaría respaldada por excelentes resultados de diagnóstico y prevención.

Para (Martínez, Vega, & Ferras, 2014) HIS se orienta en la integración de tecnologías para los profesionales siendo capaz de disminuir recursos con la optimización de las necesidades de los hospitales. Señala (Chalen, 2017) que el sistema de información almacene el registro de los datos del paciente, la especialidad y los equipos médicos, así como su historia clínica y los resultados, tanto para radiólogos, y las imágenes en el formato DICOM que es el estándar digital que permite la interacción con imágenes de radiología.

PACS no solo es un sistema de distribución de imágenes, sino que en conjunto con RIS permite ofrecer la accesibilidad a imágenes radiológicas digitales dentro de una red hospitalaria. Además, los médicos tratantes y especialistas pueden visualizar los estudios desde cualquier estación de trabajo o medio tecnológico, a tan solo segundos de que el examen es realizado permitiendo de esta manera la reducción de costos y tiempos.

Nuestro trabajo se fundamenta en el desarrollo de un módulo para el departamento de radiología el cual permitirá la incorporación de HIS y PACS que fue previamente desarrollado por (Ortiz & Noroña, 2019), este resulta necesario pues carece de multiagentes.

Los agentes resultan imprescindibles como tecnología, ya que indican (Aguilar, Bolívar, Hidrobo, & Cerrada, 2013) que poseen características como la reactividad, es decir son capaces de mantener una interacción constante con su entorno. Por tanto, esto facilita las acciones rápidas, que son cruciales para el sistema en tiempo real.

Los agentes toman las acciones necesarias para lograr las metas del sistema. También señalan (Aguilar, Bolívar, Hidrobo, & Cerrada, 2013) que una de sus propiedades más importantes es la autonomía, pues esta es capaz de eliminar la intervención humana o de otros sistemas externos. De esta manera, el sistema puede tener un comportamiento propio, y reaccionar a los estímulos externos basándose en sus estados internos.

De esta manera, el empleo de un sistema multiagentes es esencial, ya que tienen mucha interrelación sobre todo con la Inteligencia Artificial. Según (Aguilar, Bolívar, Hidrobo, & Cerrada, 2013) un sistema multiagentes se refiere al conjunto de agentes autónomos que tratan de solucionar un problema dado, además son capaces de interactuar consigo mismo, empleando protocolos y expresiones de comunicación. Sus principales beneficios, y por lo cual es factible que se utilicen dentro de esta investigación, son en primer lugar capaces de solventar un problema por su autonomía.

La utilización de multiagentes tiene el propósito de mejorar y agilizar la comunicación entre la historia clínica del paciente y los estudios radiográficos realizados, donde su objetivo es la reducción de tiempo y dinero, así como la centralización de imágenes médicas evitando la duplicidad de estudios, su pérdida; esto ayuda a un oportuno y pronto tratamiento del paciente.

1.4. Objetivos

1.4.1. Objetivo General

- Desarrollar un módulo y una aplicación móvil que permita la comunicación entre los sistemas de HIS y PACS con sistemas multiagentes, usando herramientas de código abierto.

1.4.2. Objetivos específicos

- Analizar la comunicación para el desarrollo de sistemas multiagentes
- Desarrollar el módulo de radiología y una aplicación móvil para el departamento médico con sistemas multiagentes
- Integrar el módulo el cual optimizará la comunicación entre HIS y PACS
- Implementar métricas de rendimiento para la evaluación del sistema web y móvil con agentes

1.5. Marco Metodológico

Se van a realizar dos desarrollos que son un aplicativo web y móvil para el departamento de radiología, en base a este requerimiento la metodología que mejor se adapta al proyecto es RUP, donde su pilar fundamental son los componentes que son pequeños aplicativos que permiten realizar un mejor seguimiento al desarrollo con sus respectivas pruebas dando como resultado un componente fiable y modular. Las herramientas que se va a utilizar para el manejo de datos entre agentes es el framework JADE, el cual nos brinda un soporte para la comunicación entre agentes y poder intercambiar datos entre las bases que tienen la información de los pacientes HIS y los resultados de los exámenes radiológicos PACS. Para la comunicación con dispositivo móvil se utilizará Lightweight Extensible Agent

Platform (LEAP, por sus siglas en inglés), este complemento de JADE ayuda a la comunicación con la plataforma web.

Con la ayuda de agentes nos permite obtener la información de los pacientes que se encuentran en el módulo HIS y poderlos enlazar con las imágenes que se encuentran en el módulo PACS, los agentes son útiles ya que cuando exista un nuevo pedido de un examen radiológico desde el módulo de HIS o desde el aplicativo móvil, este se pueda visualizar desde el módulo de radiología. En el ambiente web se utiliza OPENMRS, es un software que permite el diseño de un sistema de registro médico personalizados y modular, el sistema se basa en una estructura de datos conceptual por lo que se puede personalizar en los diferentes usos. La visualización de imágenes radiológicas es con DCM4CHE es una implementación que funciona con java para la visualización y recuperación de las imágenes DICOM.

En este proyecto se tomará en cuenta las métricas asociadas a las características de calidad de funcionalidad midiendo el rendimiento, disponibilidad, concurrencia y la batería del dispositivo móvil. Para lo cual se utilizará la herramienta Visual VM la cual proporciona una interfaz visual que permite mostrar la información detallada sobre las aplicaciones Java que se están ejecutando en una maquina Máquina Virtual Java (Hurka, 2019).

CAPÍTULO 2

Fundamento teórico

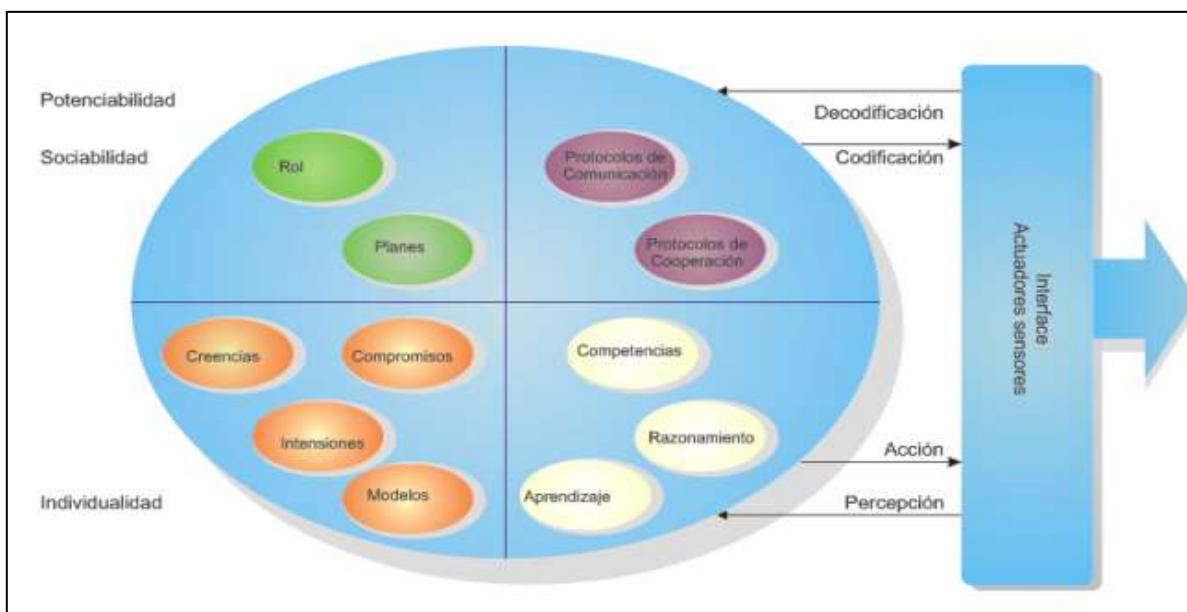
2.1. Agentes

Para (Bravo, Aguilar, & Rivas, 2004) una agente consiste en un “sistema computacional que está situado en un ambiente, y que es capaz de tomar acciones autónomas en ese ambiente con el fin de cumplir sus objetivos de diseño” (p. 23). Entre sus principales propiedades se puede mencionar: la independencia, movilidad, lógica, reactividad, atención y proactividad. Señalan (Bravo, Aguilar, & Rivas, 2004) que están dotados de herramientas que les permiten el razonamiento de las situaciones de manera inteligente y así como la capacidad de evolucionar por la experiencia, estos se denominan agentes inteligentes.

Según (Jennings & Wooldridge, 1995) “Los agentes constituyen el próximo avance más significativo en el desarrollo de sistemas y pueden ser considerados como la nueva revolución en el software” (p.17) La arquitectura de una agente se caracteriza por aspectos como la potencialidad, sociabilidad, individualidad, también se suceden procesos de codificación, decodificación, así como de acción y percepción. A continuación, se presenta en la Figura 1 la arquitectura de un agente:

Figura 1

Arquitectura de un agente



Fuente: (Aguilar, Bolivar, Hidrobo, & Cerrada, 2013)

Entre otras definiciones se puede mencionar la de (Shoham, 1993) donde indica que comúnmente, esta se refiere a “una entidad que funciona continua y autónomamente, en un entorno en el cual otros procesos ocurren y existen otros agentes” (p.54). Por su parte, (Russell & Norvig, 2004) “Un agente es una entidad que percibe su entorno y actúa bajo estas percepciones” (p.15)

Con todo esto, se puede determinar que un agente satisface funcionalidades sumamente complejas, de una manera sencilla, independiente, con un razonamiento inteligente y eficaz. A demás que, según todos los conceptos analizados, no existe uno definitivo, sino que depende de múltiples factores y circunstancias. Su concepto va a estar influido por sus actividades que le posibilitan cumplir sus funciones; también por su información o conocimiento, es decir las normas que debe seguir para realizar sus tareas; y

finalmente, por su comunicación, definida como la manera en que interactúa con el medio y con los demás agentes.

2.2. Sistemas Multiagentes

Por otro lado, se encuentran los System Management Agent (SMA, por sus siglas en inglés) que se definen por (Alvares & Sichman, 1997) la incorporación de los agentes en un ambiente social, los cuales trabajan de manera colaborativa para conseguir fines individuales y colectivos; que conforman el medio del SMA. El correcto trabajo de un SMA se fundamenta en el uso de protocolos para la comunicación y coordinación que legitimen la apropiada repartición de actividades, información y compromisos de cada componente del sistema.

Señala (FIB, ECSDI, 2019) que un sistema multiagentes está conformado por diversos agentes inteligentes y autónomos, su principal característica es que ofrece soluciones a problemas con mayor rapidez conservando una comunicación mínima con flexibilidad y cooperación. En este caso, (Russell & Norvig, 2004) los sistemas multiagentes, obligan a que los agentes sean heterogéneos, por lo que deben tener autonomía y capacidad de razonamiento lógico, así como características tales como: la planificación, el aprendizaje, interacción, capacidad de negociación, coordinación y competición con otros para poder cumplir sus tareas.

De esta forma, resumen (Aguilar, Bolivar, Hidrobo, & Cerrada, 2013) que los sistemas multiagentes hacen referencia a los conflictos de boceto de agentes autónomos, de esta forma se busca responder tres cuestiones claves: ¿Por qué y cómo cooperan los agentes? ¿Cómo los agentes pueden reconocer y resolver conflictos? ¿Cómo los agentes pueden

negociar? Entonces sus principales características son que priorizan la coordinación de acciones o negociación entre los agentes; son asíncronos y son factibles para situaciones de emergencia.

2.3. Java Agent Development Framework (JADE)

(Marchetti & García, 2013) indican que JADE es una plataforma de software que suministra funcionalidades básicas de un middleware, son independientes de la aplicación y facilitan el desarrollo de aplicaciones distribuidas. Uno de los principales beneficios de JADE es que efectúa la abstracción sobre un lenguaje orientado a objetos, llamado JAVA, que provee un API simple y amigable. También mencionan (Marchetti & García, 2013) que la comunicación asíncrona se fundamenta en mensajes como forma básica de interacción entre agentes en JADE; por lo que el agente que desee comunicarse envía un mensaje a un destino determinado. De esta forma, no hay dependencia temporal entre el emisor y los receptores, por lo que es posible que un receptor no está disponible cuando el remitente exprese el mensaje.

Para (Bellifemine, Caire, & Greenwood, 2007) este sistema es peer to peer: quiere decir que cada agente se identifica con un nombre único global, denominado como AgentIdentifier o (AID, por sus siglas en inglés). Siendo así, un agente puede iniciar la comunicación con cualquier otro agente en cualquier momento que lo desee y puede ser igualmente el objeto de una comunicación entrante en cualquier momento. Según (Hurtado, y otros, 2018) JADE-LEAP, es una extensión de JADE que puede funcionar no solo en computadoras y servidores, sino también en dispositivos ligeros como celulares y tablets autorizados para JAVA. Con la finalidad de lograrlo se debe conformar según las configuraciones de Java Micro Edition y Android Dalvik Java Virtual Machine.

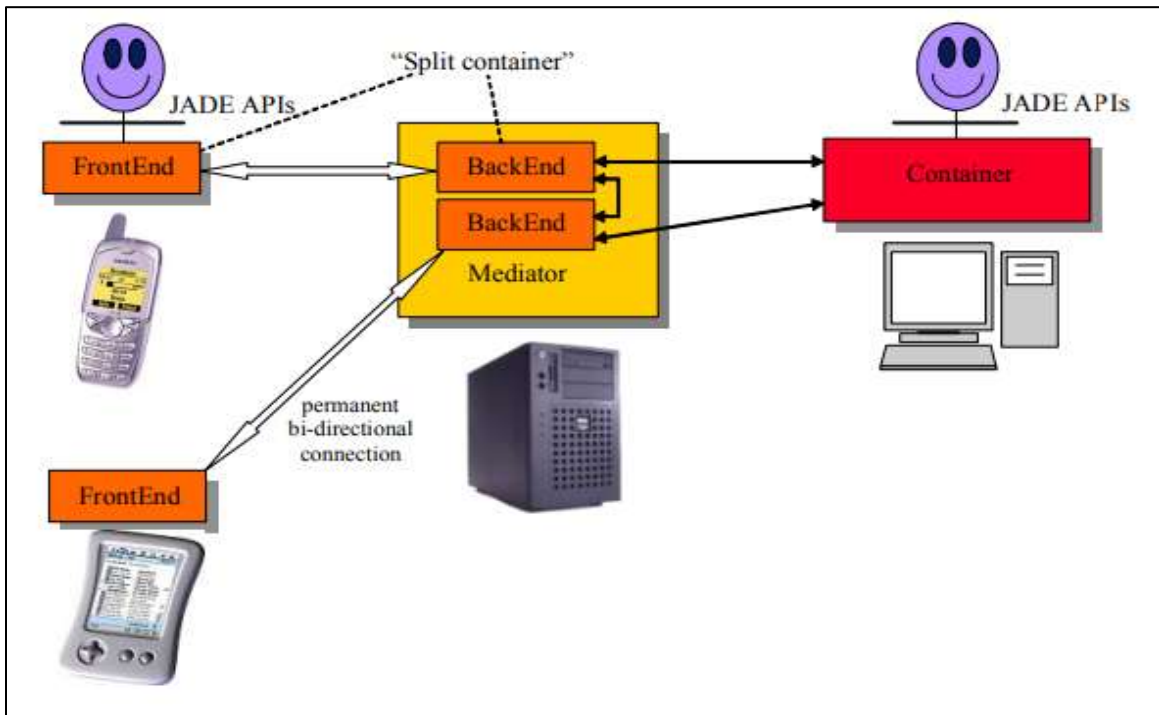
Por otro lado, (Ughetti, Trucco, & Gotta, 2008) señalan que LEAP hace posible la optimización de las herramientas de comunicación en el caso de los dispositivos móviles con recursos limitados y conectados a través de redes inalámbricas. Para poder activar este módulo, el contenedor JADE está fragmentado, como se representa en la Figura 2. Por un extremo funciona el terminal móvil y posee un back-end que se ejecuta en la red cableada. Por otra parte, se tiene un componente denominado mediador, que tiene la capacidad de instanciar y conservar los back-ends. Esto es porque afronta de mejor manera los escenarios de alta carga de trabajo, por lo que es posible dispersar varios mediadores, cada uno de ellos gestionando un conjunto de back-ends. Cada front-end se vincula con su back-end correspondiente mediante una conexión bidireccional permanente.

2.4. JADE LEAP

LEAP, este complemento de JADE ayuda a la comunicación con la plataforma web, existen versiones ad hoc de JADE diseñadas para implementar agentes JADE de forma transparente en diferentes entornos orientados a Java, como dispositivos. En la Figura 2, se presenta el esquema de ejecución de JADE-LEAP.

Figura 2

El entorno de ejecución de JADE-LEAP



2.5. Flutter

Es un framework que permite la creación de aplicaciones móviles y de escritorio permitiendo la integración en un solo código. El framework es de código abierto de elaborado por Google para elaborar aplicaciones multiplataforma compiladas de forma nativa tanto para Android como para iPhone a partir de una única base de código. Para el desarrollo de aplicaciones móviles el lenguaje en el que se debe programar es Dart. Se compila en código de máquina ARM o Intel, así como en JavaScript, para un rendimiento rápido en cualquier dispositivo.

2.5.1. Funcionalidades

Calidad nativa: entre las principales ventajas de Flutter es obtener la calidad en el resultado de las aplicaciones que son específicas para los sistemas.

Experiencia de usuario: Flutter incluye Material Design de Google y Cupertino de Apple, lo que permite que las aplicaciones se visualicen como si fueran elaboradas por las propias compañías dando una experiencia agradable al usuario.

Tiempo de carga: se experimenta tiempos de carga menores que un segundo en los sistemas operativos de Android o iOS en aplicaciones móviles, lo que garantiza el no abandono de la aplicación.

Desarrollo ágil y rápido: el Framework integra hot-reload lo que permite visualizar los cambios en tiempo real desde cualquier simulador o dispositivo conectado (Quality devs, 2019).

2.6. HIS

(Rodríguez, 2014), es un mecanismo que se encarga de la recopilación, el procesamiento, la transmisión, recuperación y acopio de datos médicos dentro de las unidades de salud. Por otro lado, señalan (Gómez & Yunda, 2013) que estos sistemas de información hospitalarios se conciben con el objetivo de realizar la gestión clínica, administrativa y financiera de cualquier centro o institución hospitalaria. Además, están orientados a solucionar los requerimientos de almacenamiento, procesamiento e interpretación de información clínica y administrativa para que la institución pueda optimizarse y generar una mejor toma de decisiones. Como se indica en la Tabla 1, los HIS son la base de los sistemas de información de salud. También se debe señalar que algunos

servicios como imágenes y laboratorio radiológico cuentan con sistemas de información especializados. En la siguiente tabla se exponen cada uno de los dominios y sus procesos respectivos:

Tabla 1

Dominio de Negocios con un Enfoque de Arquitectura Empresarial

DOMINIO DE NEGOCIO	PROCESOS DE NEGOCIO
Servicios de salud	Registro de paciente Historia clínica Registros de nacimiento y mortandad. Reporte de enfermedades
Laboratorio	Reporte de resultados Clasificación de enfermedades Asociación de resultados a pacientes Registro de muestras
Farmacia	Inventario Registro de inventario Tratamientos
Recursos humanos de salud	Clasificación de trabajadores Reclutamiento, contratación de trabajadores Monitoreo de trabajadores Informe de contrataciones y formación
Soporte a la toma de decisiones	Protocolos de salud e investigación Presupuestos Análisis y presentación de datos Monitoreo de salud urgente
Financieros	Tarifa por servicios Nómina Facturación

Fuente: (Stansfield, Orobato, Lubinski, Uggowitz, & Mwanyika, 2008)

Para (Pérez, Arencibia, Jiménez, & Tellería, 2018) el Sistema HIS posee diversas funcionalidades como por ejemplo firma y validación digital que hace posible la integridad, temporalidad, autenticidad y autoría de los documentos clínicos electrónicos que se generan en las instituciones de salud que lo emplean.

2.7. RIS & PACS

Por otra parte, según (Chalen, 2017) los PACS, son sistemas que se relacionan e interactúan con diversas modalidades radiológicas en un centro de salud. Sobre todo, tiene el objetivo de remitir, recoger, almacenar, visualizar, reportar e imprimir las imágenes radiológicas para la sistematización de los procedimientos de diagnóstico. Además, permite el acceso a la información tanto dentro como fuera del centro hospitalario. Este sistema permite que la visualización sea posible desde diferentes tipos de dispositivos como visores especiales, computadores personales, smartphones y tablets.

Por otra parte, señalan (Gómez & Yunda, 2013) que, para poder soportar estos servicios, es necesaria la implementación de una red de comunicación de alta velocidad. De esta forma, resulta posible transferir imágenes de gran tamaño, además de sobrellevar el acceso concurrente, todo esto con el fin de atender peticiones desde las terminales de visualización, procesamiento y reportes de imágenes radiológicas.

En cambio, RIS es un sistema de información radiológica. Para (Vega, Ciudad, Duque, & Soler, 2020) se desarrolla para brindar soporte y gestionar el flujo de trabajo del departamento radiológico, por tanto, se puede presentar como un sistema independiente. Ambos sistemas trabajando en conjunto posibilitan y automatizan la gestión de las listas de trabajo para los servicios de imágenes, además gestiona datos del paciente, así como también se pueden realizar citaciones, contribuye a la organización del trabajo de los especialistas de radiología.

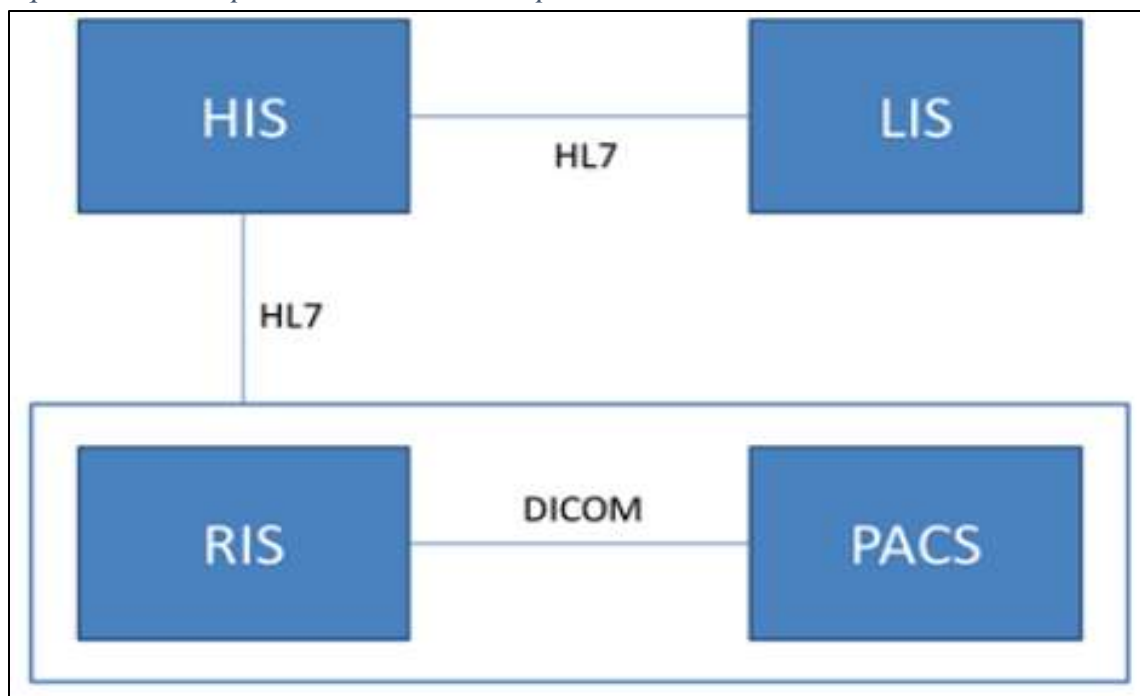
2.8. HL7

Señala que HL7, es una organización sin fines de lucro, acreditada por el Instituto Nacional Estadounidense de Estándares. Esta se dedica a promover un marco de trabajo y estándares que permitan intercambiar, integrar, colaborar y consultar información de salud dentro de formatos electrónicos. Para su desarrollo se basaron en un modelo orientado a objetos, denominado modelo de referencia (RIM), por lo que HL7, incorpora un conjunto de estándares basados sobre todo en mensajería, gestión, manejo de documentación, entre otros

Por otro lado, indica (Health Level Seven International, 2019) que HL7 concentra su aplicación en otorgar interoperabilidad para diversos sistemas. Su principal propósito no se refiere solo al entorno hospitalario, sino que este protocolo se toma como un indicador entre los diferentes sistemas implementados como se visualiza en la Figura 3.

Figura 3

Esquema de interoperabilidad a través de protocolo HL7



Para (Martínez, Vega, & Ferras, 2014) se debe tomar en cuenta que las funcionalidades de RIS no se orientan solo a la adquisición, análisis o acopio de los estudios, sino que también se concentra en la gestión a un nivel más alto. De esta manera, PACS es el encargado de almacenar y tramitar los estudios con la finalidad de alcanzar la comunicación entre sistemas, para lo cual se emplean estándares que posibilitan la interoperabilidad.

2.9. DICOM

Según (Clinic Cloud, 2021), DICOM es un estándar promovido por American College of Radiology y National Electrical Manufactures Association. Para esto se conforma un comité con el propósito de desarrollar un estándar que posibilite la trasmisión de imágenes radiológicas con información relacionada con el paciente. (Gómez & Yunda, 2013) indican que el formato de archivos DICOM se puede integrar con la mayoría de los sistemas PACS, debido a que en varias estaciones de trabajo están separadas del hardware de captura y visualización de imágenes. El estándar incluye un complejo protocolo de comunicación TCP/IP, gracias a esto es posible tener equipo conectados a través de internet con capacidades de trasmisión por este medio.

Por otro lado, se debe resaltar que para (Clinic Cloud, 2021) el beneficio más importante de las imágenes DICOM es la seguridad que ofrece en contraposición con sistemas similares. En este aspecto, se destaca que el fundamento de los archivos DICOM es precisamente, el protocolo que emplea para cifrar la transferencia de información. Este procedimiento contribuye a la protección y privacidad de las imágenes, al igual que de todos los archivos adjuntos referente a la salud del paciente.

CAPÍTULO 3

Metodología

En este tercer capítulo, se describen los requerimientos necesarios para la creación de un módulo y una aplicación móvil que permitan la comunicación entre los sistemas de HIS y PACS con sistemas multiagentes, usando herramientas de código abierto. Además, se detallará la comunicación para el desarrollo de sistemas multiagentes, también se busca optimizar la comunicación entre HIS y PACS y finalmente, se exponen las métricas de rendimiento para la evaluación del sistema web y móvil con agentes. La arquitectura de este proyecto se dividió en tres partes, las cuales serán expuestas posteriormente.

En primer lugar, se debe explicar que el módulo de radiología RIS será un aplicativo web y móvil con ayuda de agentes para la gestión de citas de los pacientes, el agendamiento de exámenes radiológicos, recordatorios por notificaciones y correo electrónico y la presentación de informes de resultados por Internet o correo electrónico.

La información del proyecto RIS se almacenará en la base de datos en MYSQL, la cual tiene la siguiente estructura que se encuentra en la Figura 5. Cabe mencionar que en la tabla con el nombre HISPACS es la encargada de almacenar la información de los identificadores de los pacientes de los sistemas de OPENMRS y DCM4CHEE, la forma en la que se va actualizar la información es por medio de agentes RIS que corren bajo el proyecto de AGENTESJADE.

3.1. Prerrequisitos en hardware

Para la instalación del sistema y la realización de las respectivas pruebas se instalaron los siguientes servidores con las siguientes características que se detallan a continuación.

Servidor OpenMRS

- Memoria de 3GB
- Procesador de cuarta generación con 1 núcleo
- Disco no sólido de 100 GB
- Dirección IP 192.168.1.103

Servidor DCM4CHEE

- Memoria de 4GB
- Procesador de cuarta generación con 1 núcleo
- Disco no sólido de 100 GB
- Dirección IP 192.168.1.102

Servidor SysRis

- Memoria de 8GB
- Procesador de cuarta generación con 1 núcleo
- Disco no sólido de 100 GB
- Dirección IP 192.168.1.99

3.2. Ontología de Agentes

OBTENER_DATOS_HIS. Cuando el AGENTE RIS va a actualizar la información de la tabla HISPACS este agente envía un mensaje al AGENTE HIS con esta ontología, con el fin de que el AGENTE HIS sepa que debe hacer cuando el reciba el mensaje.

OBTENER_DATOS_PACS. Cuando existe la necesidad de actualizar la tabla HISPACS el AGENTE RIS envía un mensaje al AGENTE PACS con esta ontología para que el AGENTE PACS sepa que acción debe realizar.

CREAR_PACIENTE_PACS. Con esta ontología es enviada al AGENTE HIS es para que este agente cree nuevos pacientes que han sido encontrados en la base de datos de PACS.

OBTENER_DATOS_TIPO_IDENTIFICACION. Cuando se quiere conocer los datos de un determinado paciente por el agente PETICION o PETICIONWS se consulta al agente RIS que este enviara la respectiva solicitud al agente HIS

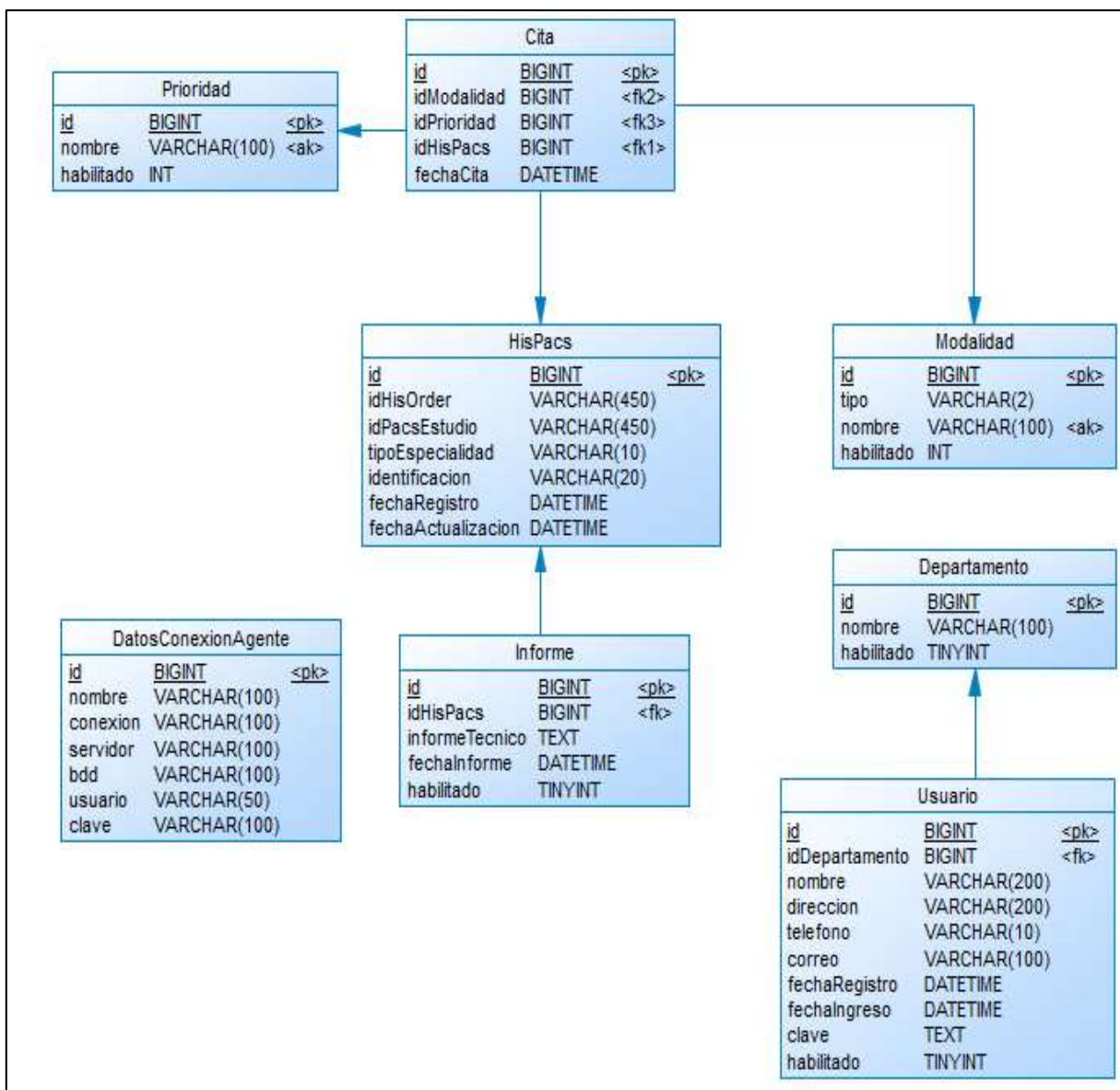
CONSULTAR_PACIENTE. Se consulta la información de los pacientes que están enlazados en la base de datos de RIS para luego obtener la información con el agente HIS.

CREAR_PACIENTE. Envía toda la información del paciente para que el agente HIS lo cree en la base de datos del servidor OPENMRS.

OBTENER_DATOS_PACIENTE_ESTUDIO. Obtiene la información de los estudios realizado al paciente que se encuentran almacenados en el servidor de DM4CHEE.

Figura 4

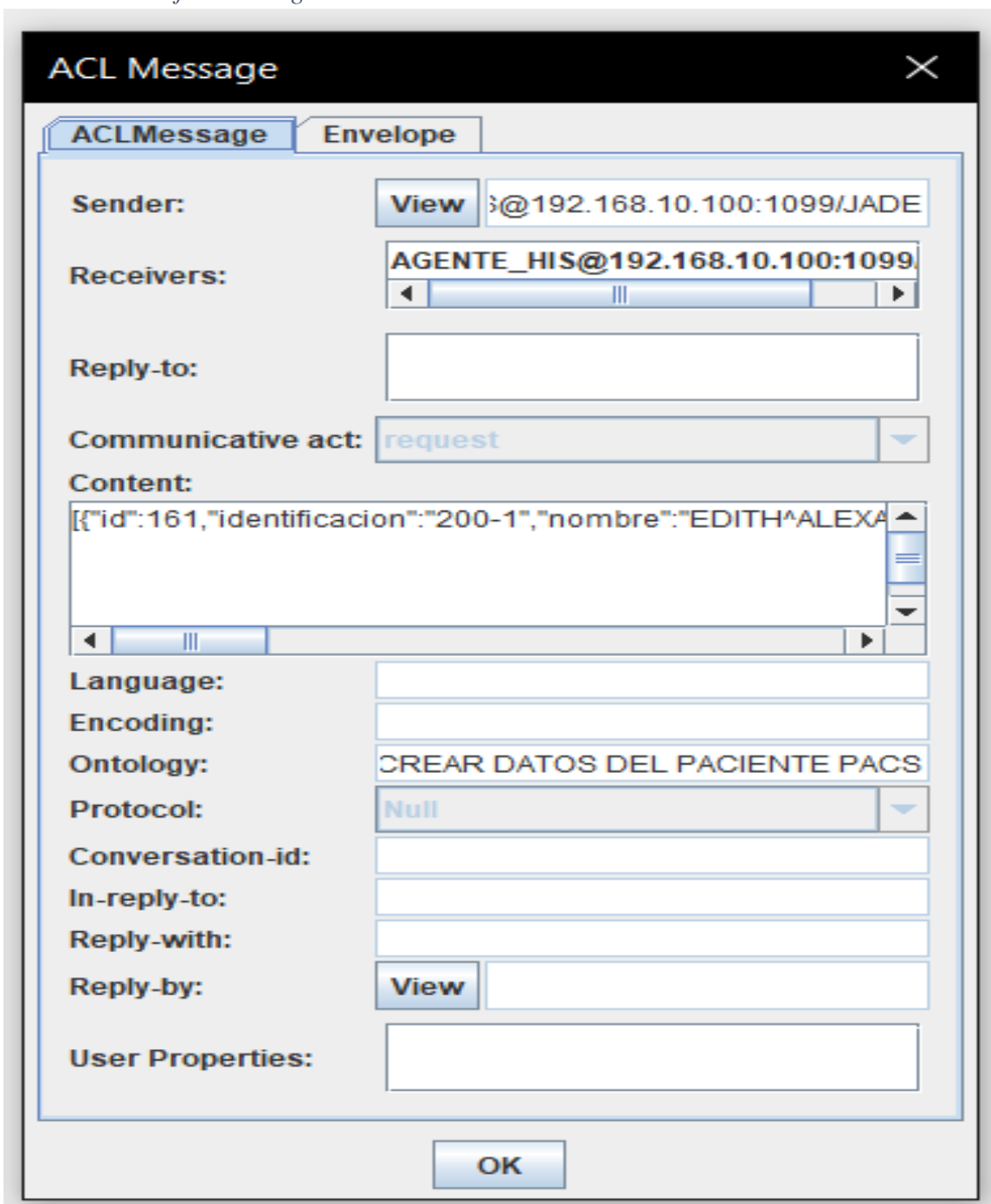
Modelo de la base de datos



La forma en la que se comunican los agentes es utilizando mensajes ACL Message con ontologías, la estructura que se utilizó para enviar mensajes un agente hacia otro es la estructura tipo JSON como se muestra en la Figura 5, para posteriormente deserializarlo y ejecutar la instrucción necesaria.

Figura 5

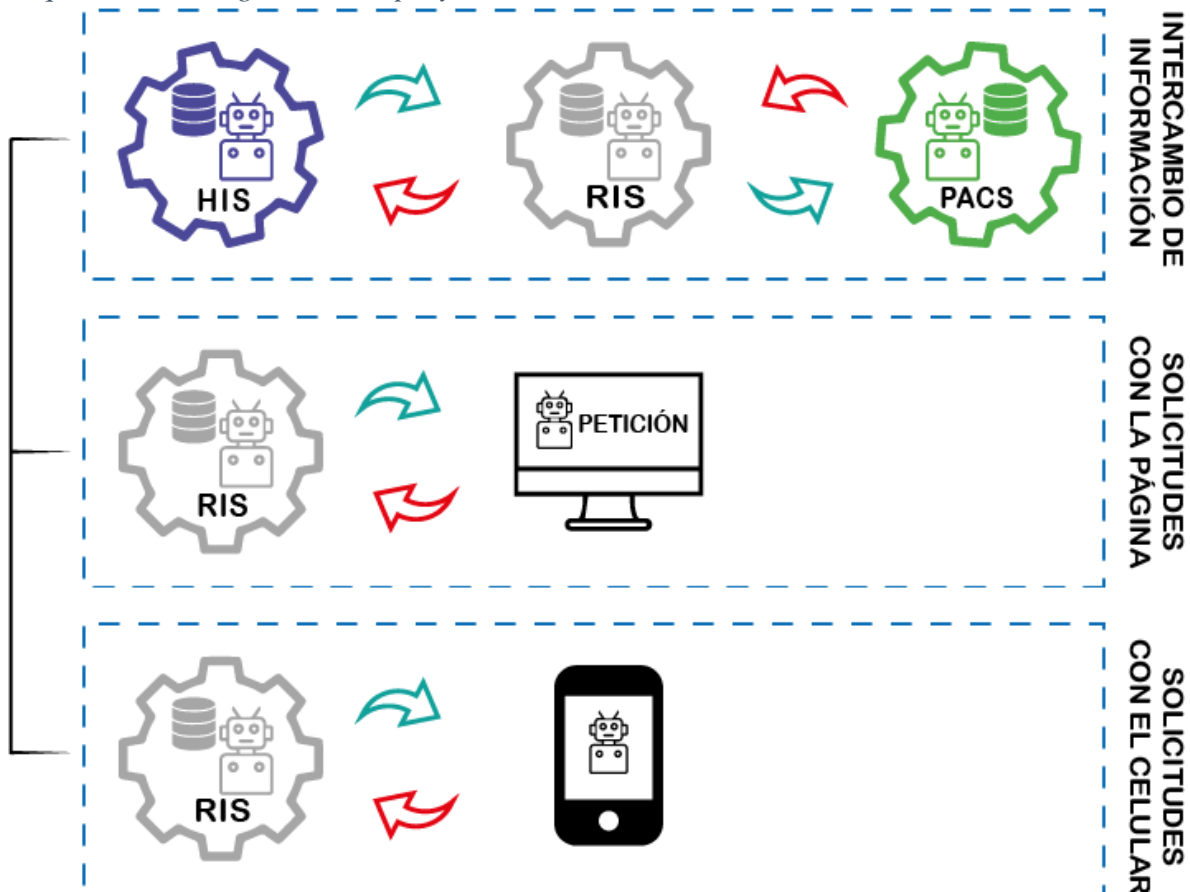
Envió de mensajes entre agentes



La arquitectura de los agentes con su funcionamiento involucrados en el sistema, así como la relación que se establece entre sus partes y sus distintas funciones, se presenta en la Figura 6.

Figura 6

Arquitectura de Agentes en el proyecto



3.3. Intercambio de información

Para el intercambio de la información de los pacientes en los servidores de DCM4CHEE y OPENMRS fue necesario la creación de tres agentes que se puso los siguientes nombres HIS, RIS y PACS. El agente RIS cada cierto tiempo pregunta si existen nuevos pacientes al agente HIS que no estén todavía relacionados y de la misma forma

preguntara al agente PACS y si existen nuevos estudios que no estén relacionados, esto permitirá mantener actualizada la información que se encuentra en la base de datos de RIS.

A continuación, se describen los algoritmos que se utilizaron para la creación e implementación de los agentes en esta primera parte:

Algoritmo 1. Creación de la instancia del contenedor principal

Output: booleano

1. Runtime runtime \leftarrow Runtime.instance();
 2. ProfileImpl profileImplMain \leftarrow new ProfileImpl(true);
 3. profileImplMain.setParameter(ProfileImpl.CONTAINER_NAME, HOST);
 4. profileImplMain.setParameter(ProfileImpl.GUI, "false");
 5. AgentContainer agentContainer \leftarrow runtime.createMainContainer(profileImplMain);
-

El Algoritmo 1, es primordial para que funcionen los agentes debido a que de esta forma se crea el contenedor principal donde se acoplan o unen el resto de contenedores con sus respectivos agentes. Este método realiza un retorno de un booleano para saber si se puede crear el siguiente contenedor o mostrar un mensaje de error indicando que no se puede crear, la descripción del algoritmo es el siguiente

- ✓ En la línea 1 se crea una instancia en tiempo de ejecución esto ayuda que la CPU y memoria no se sobrecarguen.
- ✓ En la línea 2 se indica que la instancia va a contener un contenedor principal.
- ✓ En la línea 3 se establece el nombre del contenedor y cual seria el host donde se va a crear.
- ✓ En la línea 4 se indica que no se va a utilizar un entorno gráfico para la visualización de los agentes.

- ✓ En la línea 5 se asigna el contenedor principal.

Cabe indicar las razones por las que no se podría crear el contenedor principal son por que ya existe un contenedor principal o el puerto esta ocupado.

Algoritmo 2. Inicialización de Agentes

1. AgentController agenteRIS \leftarrow agentContainer.createNewAgent(Agente.AGENTE_RIS, RisAgente.class.getName(), null);
 2. agenteRIS.start();
 3. AgentController agenteHIS \leftarrow agentContainer.createNewAgent(Agente.AGENTE_HIS, HisAgente.class.getName(), null);
 4. agenteHIS.start();
 5. AgentController agentePACs \leftarrow agentContainer.createNewAgent(Agente.AGENTE_PACS, PacsAgente.class.getName(), null);
 6. agentePACs.start();
-

En el algoritmo 2 se inicializan los agentes en el contenedor creado anteriormente, la descripción dl algoritmo es la siguiente:

- ✓ En la línea 1, 3 y 5 se crea los agentes en los contenedores con su respectivo nombre.
- ✓ En la línea 2, 4 y 6 se inicializa el agente.

Algoritmo 3. Envio mensajes de respuesta entre agentes

1. ACLMessage reply \leftarrow msg.createReply();
 2. reply.setPerformative(typeMessage);
 3. reply.setContent(contentData);
 4. myAgent.send(reply);
-

La forma de comunicación entre agentes es utilizando mensajes ACL en los cuales el contenido puede ser un entero, booleano o un string, la descripción del funcionamiento del Algoritmo 3 para la recepción de mensajes es:

- ✓ En la línea 1 se instancia el mensaje de quien se va a recibir la información.
- ✓ En la línea 2 se establece la prioridad del tipo de mensaje que se va a enviar puede ser informativo, error etc.
- ✓ En la línea 3 se establece el mensaje que se va a enviar al agente.
- ✓ En la línea 4 se realiza el respectivo envió.

Algoritmo 4. Envío mensajes entre agentes

1. `ACLMessage request ← new ACLMessage(ACLMessage.REQUEST);`
 2. `request.addReceiver(new AID(agenteRecibe, AID.ISLOCALNAME));`
 3. `request.setOntology(ontology);`
 4. `request.setContent(contentData);`
 5. `myAgent.send(request);`
-

En el algoritmo 4 nos permite el envió de mensaje entre agentes la descripción de del algoritmo es:

- ✓ En la línea 1 se instancia el mensaje a quien se va a enviar la información.
- ✓ En la línea 2 se define el receptor del mensaje y su ubicación
- ✓ En la línea 3 se establece la ontología para que el agente cuando reciba el mensaje el sepa que debe hacer y que mensaje deberá enviar.
- ✓ En la línea 4 se envía la data para que reciba el agente.
- ✓ En la línea 5 se realiza el respectivo envió.

En los agentes HIS y PACS su comportamiento es similar su única variación es la ontología que se ejecuta en cada agente. Esta clase tiene una herencia del comportamiento preestablecido por jade en este caso en ambas clases se estableció, CiclycBehaviour donde su principal función es estar pendiente de la recepción de mensaje para ejecutar una determinada acción a continuación se define el Algoritmo 5

Algoritmo 5. Comportamiento agentes HIS y PACS

```
1. HISComportamiento(Agent agent) {
2.   super(agent);

3.   @Override
4.   public void action() {

5.     ACLMessage mensaje ← myAgent.receive();
6.     if (mensaje != null) {
7.       String contenido;

8.       switch (mensaje.getOntology()) {
9.         break;
```

-
- ✓ La definición del Algoritmo 5 es la siguiente:
 - ✓ En la línea 1, 2 se instancia el agente respectivo.
 - ✓ En la línea 3 se realiza una sobreescritura al método.
 - ✓ Desde la pagina 5,6,7 y 8 se espera el respectivo mensaje, donde que se va hacer lo determina la ontología enviada

Algoritmo 6. Comportamiento agentes RIS

```
1. addBehaviour(new RISComportamiento(this, 3000));

2. addBehaviour(new CyclicBehaviour() {
3.   @Override
4.   public void action() {
5.     ACLMessage message ← myAgent.receive();
6.     if (message != null && message.getContent() != null)
7.       Conexion con la base para actualizar los datos
8.     else
9.       Cancela
10.  End if
11. }
12. });
```

En el Algoritmo 6 se define el comportamiento del agente RIS, cabe indicar que solo este agente tiene definido dos comportamientos que son

CyclicBehaviour. Ejecuta una tarea determinada cada tiempo.

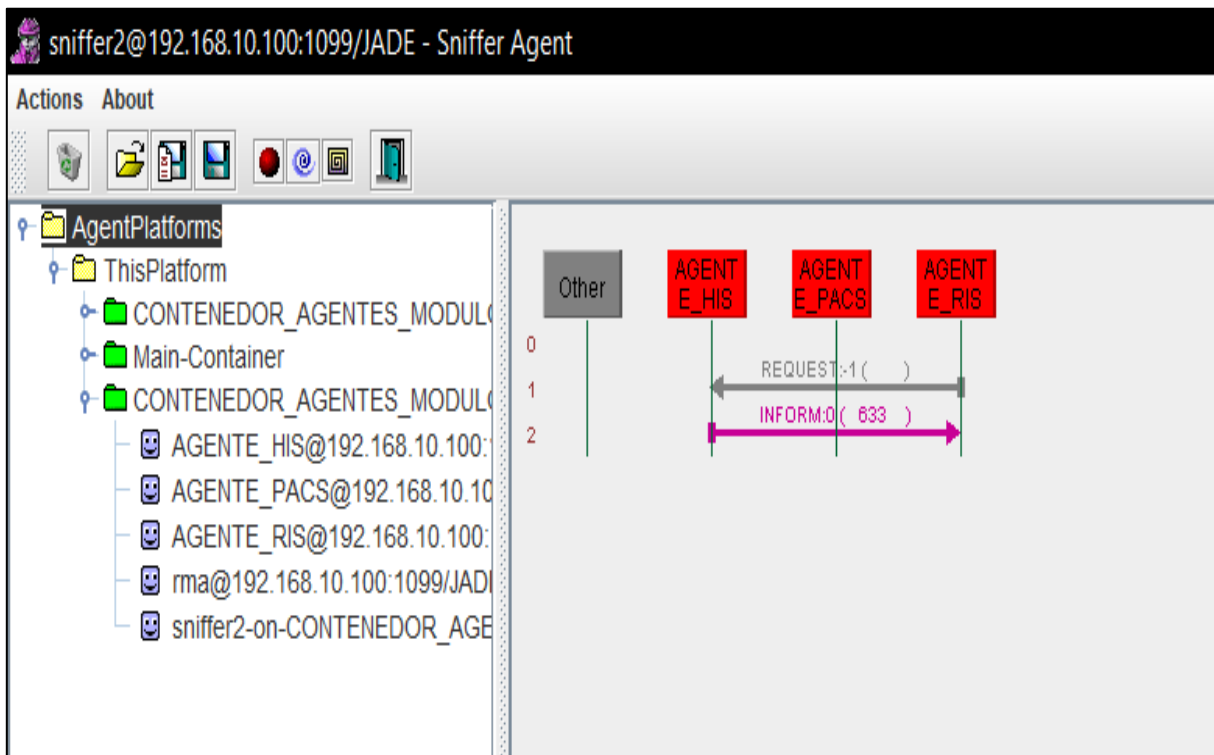
TickerBehaviour. Ejecuta una tarea indefinidamente cada tiempo.

- ✓ En la línea 1 se instancia el comportamiento Tickerbehaviour cada 3000 milisegundos
- ✓ En la línea 5 se asigna el mensaje de recepción al agente para luego poder enviar la respuesta.
- ✓ En la línea 7 se realiza la respectiva conexión para ejecutar una determinada acción que fue enviada por el agente.
- ✓ En la línea 9 si existe algún ya termina algoritmo

En la Figura 7 se puede visualizar la comunicación entre el AGENTE RIS, que solicita la información de un paciente al AGENTE HIS.

Figura 7

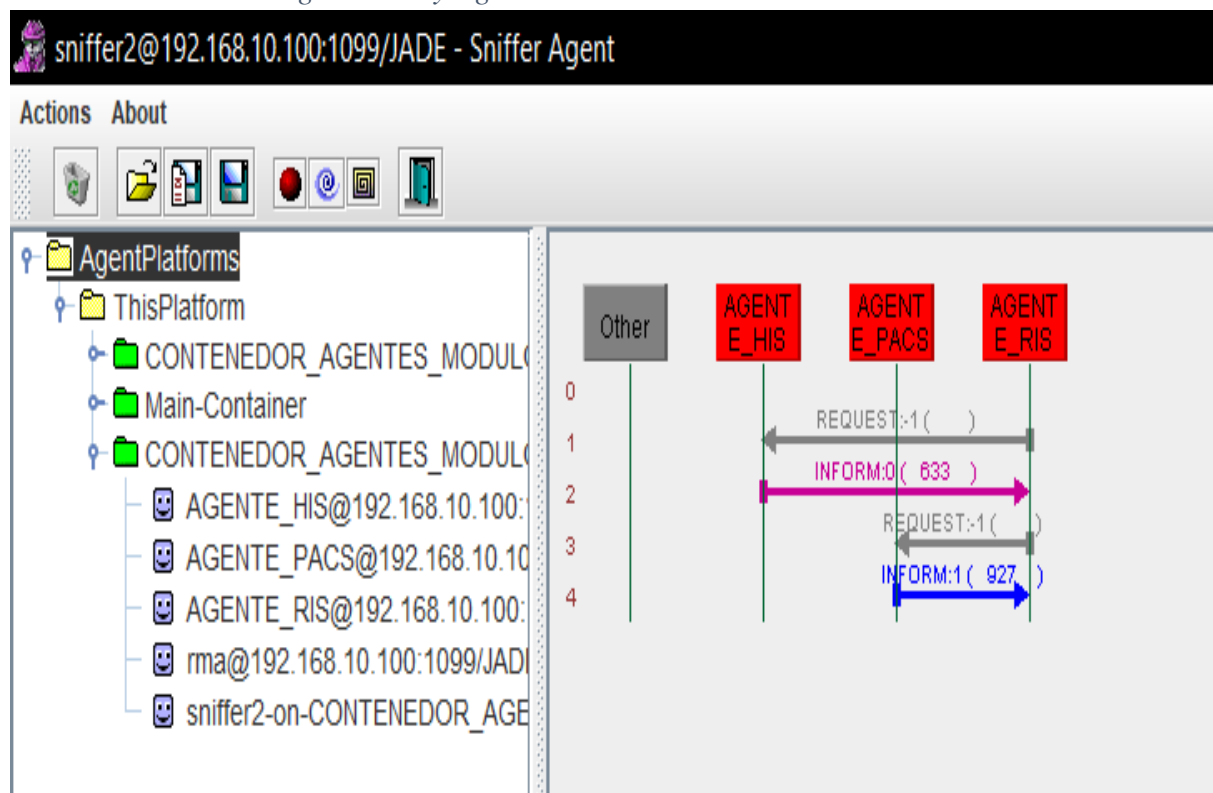
Comunicación entre Agente HIS y Agente RIS



En la Figura 8 se puede visualizar la comunicación entre el AGENTE RIS, que solicita la información de un estudio del paciente al AGENTE PACS.

Figura 8

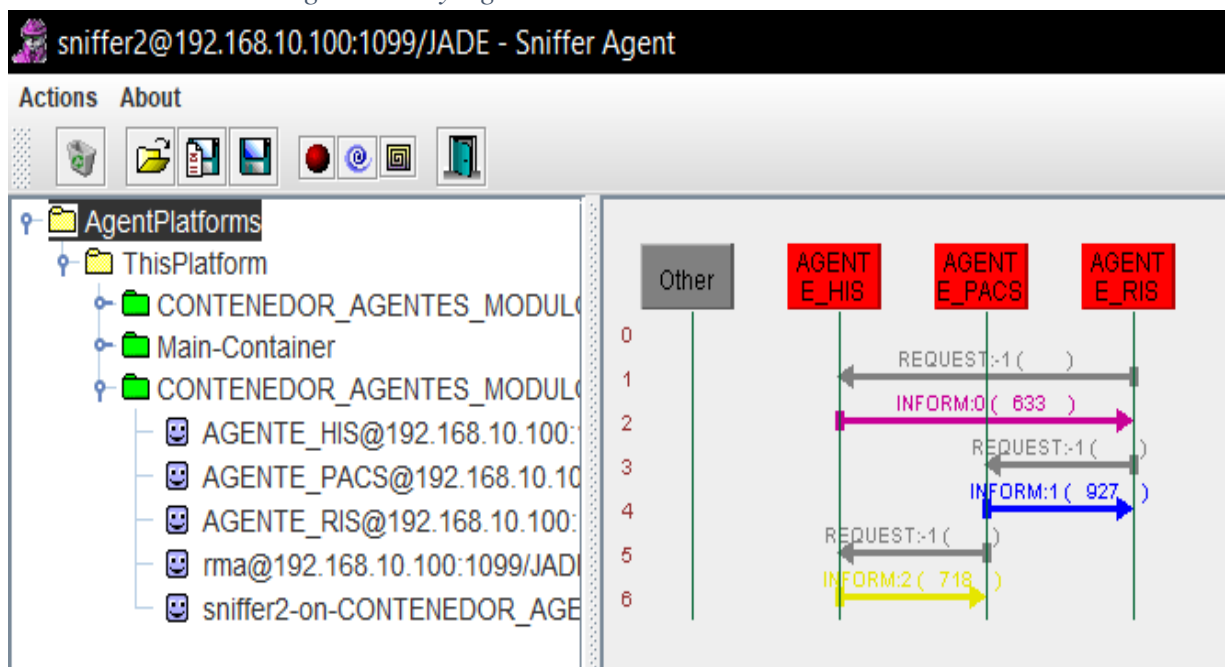
Comunicación entre Agente HIS y Agente RIS



En la Figura 9 se muestra cuenta el AGENTE PACS se comunica con el AGENTE HIS para registrar la información de un paciente si fuera necesaria.

Figura 9

Comunicación entre Agente HIS y Agente PACS

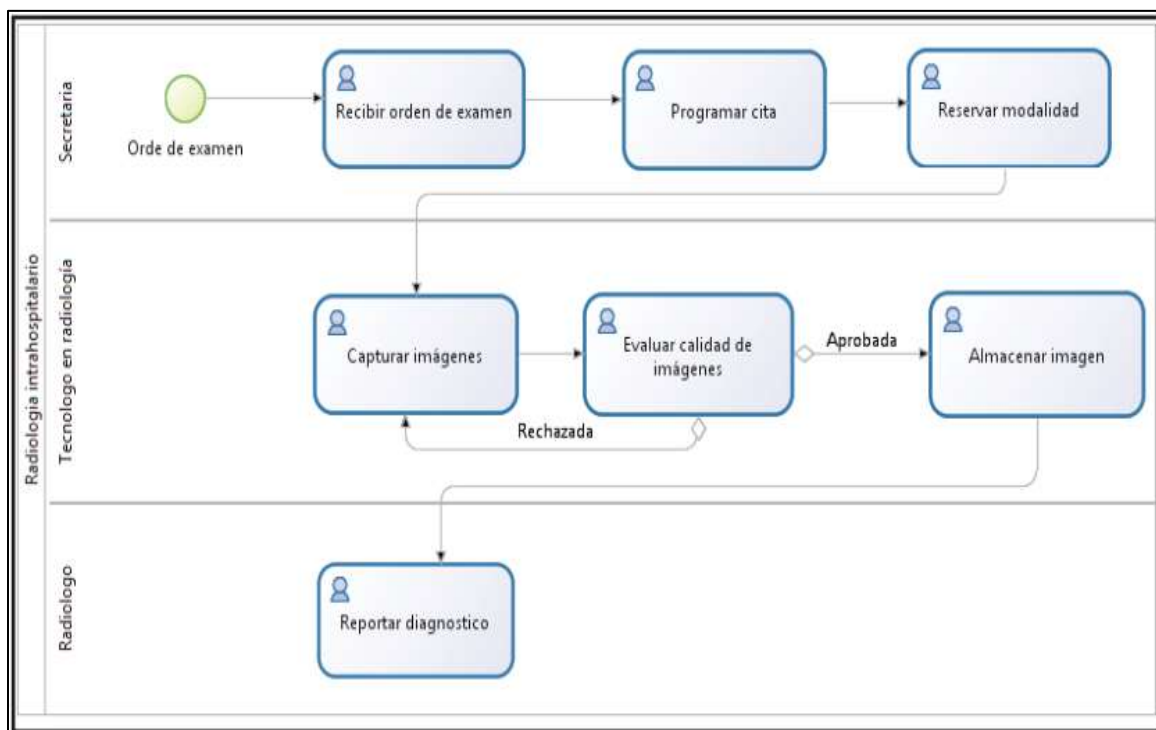


3.4. Solicitudes con la página

El flujo de trabajo dentro del departamento de Radiología sigue un proceso lineal que inicia en secretaria, donde se realiza la orden para el examen, a partir de aquí tiene lugar la programación de la cita y el tipo de modalidad. A partir de esto, el proceso pasa a ser atendido por el técnico radiólogo quien es el responsable de realizar la toma de imágenes, evaluar su calidad y una vez aprobada pasa a su almacenamiento, para finalmente enviar el reporte definitivo al radiólogo en la Figura 10.

Figura 10

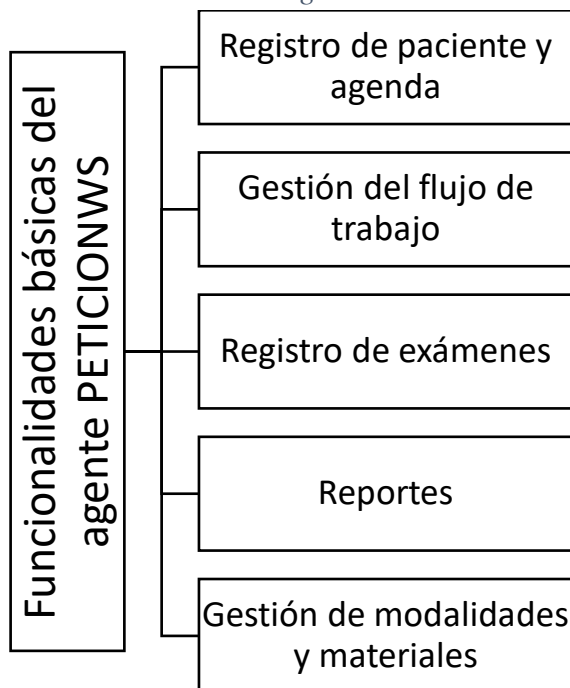
Flujo de trabajo del departamento de radiología



Para el intercambio de información con el agente RIS se creó un cuarto agente al que se le llamó PETICIONWS este es el encargado de atender las solicitudes que se realiza por la página para transferir los registros del paciente junto a las imágenes de los estudios que le han sido realizados en el Departamento de Radiología (como, por ejemplo: Ultrasonidos rayos x, resonancias magnéticas, tomografías entre otras modalidades). Una de las funciones básicas del agente de PETICIONWS se basan en la recopilación de datos de los pacientes, el registro de diferentes datos relacionados con los exámenes radiológicos y finalmente los reportes médicos y todo esto se representa en la Figura 11:

Figura 11

Funcionalidades del agente de PETICION



Con el agente de petición tiene dos clases que son: *PeticionComportamientoEnvio* y *PeticionComportamientoRespuesta*. A la clase *PeticionComportamientoEnvio* es para enviar el respectivo mensaje al agente, con su ontología y sepa el agente receptor que tipo de acción debe realizar. La clase *PeticionComportamientoRespuesta* es para enviar la respuesta al frontend después de realizar la respectiva consulta. En el Algoritmo 9 se define como se realizan las peticiones desde el agente de petición hacia cualquier agente.

Algoritmo 9. Petición de agentes WS

```
1. String datosConsulta = new ObjectMapper().writeValueAsString(objeto);
2. Object[] argumentos = new Object[]{agenteRecibe, ontologia, datosConsulta, dataAdicional};
3. agentePetición = creacionAgente(argumentos);
4. int intentos = 0;
5. do {
6. if (intentos++ > 5) throw new ExceptionApp("Error en la comunicación con el agente: " +
   ontologia);
7. Thread.sleep(1000);
8. respuesta = ConexionBDDRIS.getRespuestaAgente(agentePetición.getName(), agenteRecibe,
   ontologia);
9. } while (respuesta == null);
```

La definición del algoritmo es la siguiente:

- ✓ En la línea 1 se realiza el objeto en el formato JSON para ser enviado por medio de un mensaje.
- ✓ En la línea 2 se construye los argumentos para enviar al agente.
- ✓ En la línea 3 se crea el agente de petición.
- ✓ Desde la línea 6 hasta la línea 9 se realiza un ciclo para que de esta forma el agente de petición este esperando la respuesta que el de el agente RIS, realiza tiempo de espera de un segundo para volver a consultar si la respuesta ya esta almacenada en la base de datos si la respuesta en negativa continuara el ciclo hasta un tiempo mayor que 5 intentos, si se acaba el tiempo y no se tiene respuesta se envía una respuesta al usuario para que vuelva intentar procesar la transacción.

Para la creación del agente de petición debe realizarse cada vez que se ejecuta una petición desde el frontend este método se describe desde el Algoritmo 10

Algoritmo 10. Creación del agente de petición WS

1. `DateTimeFormatter formater ← DateTimeFormatter.ofPattern("yyyyMMdd_HHmssSSS");`
 2. `agentePetición ←`
`AdministraContenedor.getAgentContainer().createNewAgent(nombreAgente,`
`PeticiónAgente.class.getName(), argumentos);`
 3. `agentePetición.start();`
-

La definición del algoritmo es la siguiente:

- ✓ En la línea 1 se establece el nombre que va a tener el agente de petición donde se enlace con un formato de fecha tipo time stamp con eso se garantiza que el nombre del agente no se va a repetir.
- ✓ En la línea 2 se crea el agente en el contenedor ya creado.
- ✓ En la línea 3 se da por cancelada la operación.

Para la creación de ordenes en el servidor de DCM4CHEE se debía enviar mensajes HL7 para poder registrar los datos del paciente, en el Algoritmo 11 se define como se crea un mensaje para posterior enviarse al servidor.

Algoritmo 11. Creación de mensajes HL7

1. `String message = creacionMensaje(nombreConeccion, clienteVMR, diagnostico, prioridad, estudioUuid);`
 2. `Connection connection = context.newClient(servidor, puerto, false);`
 3. `Initiator initiator = connection.getInitiator();`
 4. `Parser parser = context.getPipeParser();`
 5. `Message msg = parser.parse(message);`
 6. `mensajeEnvio = parser.encode(msg);`
 7. `Message response = initiator.sendAndReceive(msg);`
 8. `respuesta = parser.encode(response);`
-

La definición del Algoritmo 11 se describe:

- ✓ En la línea 1 se llama a una función para la creación de mensajes HL7.
- ✓ En la línea 2 se establece la comunicación con la base de DCM4CHEE donde se va a registrar la información que se realizó por el paciente al momento de que crear una nueva cita desde el módulo de frontend.
- ✓ En la línea 3 se inicializa las conexiones
- ✓ En la línea 4 se realiza una instancia a bit para posteriormente se transfiera la información.
- ✓ En la línea 5 se castea el mensaje a Message para poder ser enviado.
- ✓ En la línea 6 se realiza el parseo del mensaje a message.
- ✓ En la línea 7 se realiza el respectivo envío.
- ✓ En la línea 8 se decodifica la respuesta.

A continuación, se detalla lo que hace el Algoritmo 12 que es una parametrización de los mensajes hacia el servidor DM4CHEE

Algoritmo 12. Creación de mensajes HL7

```
1. messageDCM = new MessageDCM();
2. messageDCM.initQuickstart("ORM", "O01", "P");
3. createMshSegment(currentDateTimeString, nombreConeccion);
4. createPidSegment(clienteVMR);
5. createPv1Segment();
6. createOrcSegment();
7. createObrSegment(diagnostico, prioridad, estudioUuid);
8. String msg = messageDCM.toString() + "\n" + "ZDS|" + estudioUuid;;
```

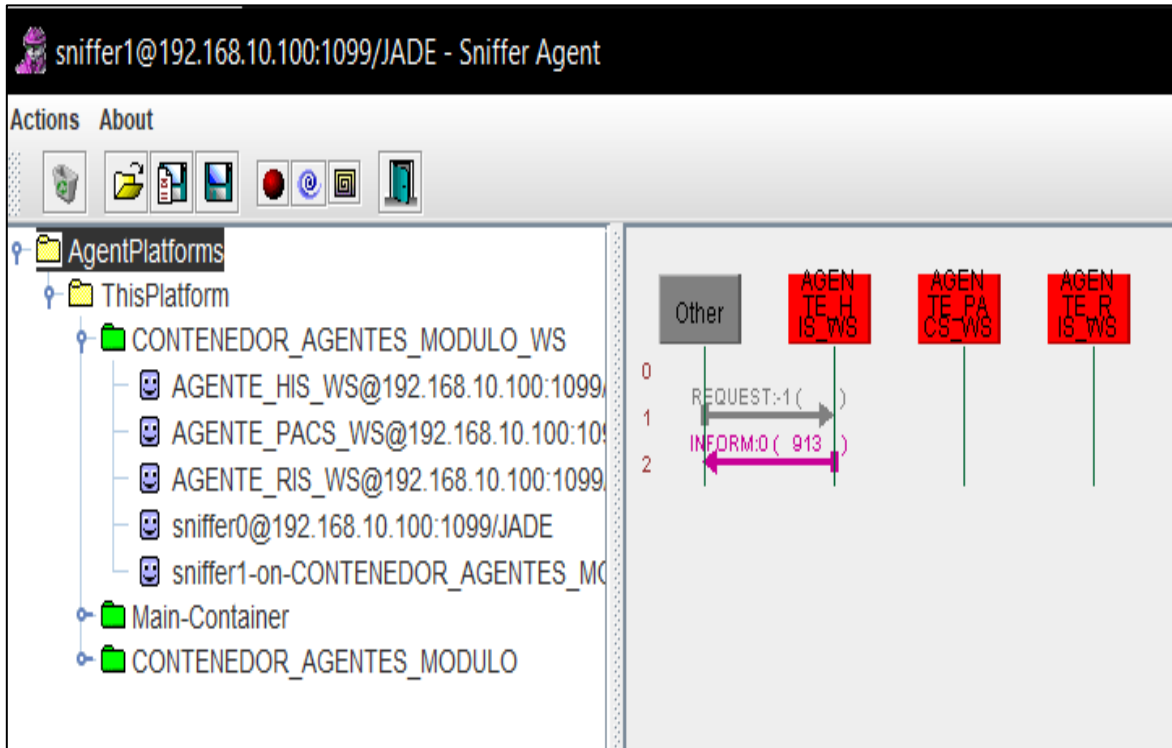
- ✓ En las líneas 2,3,4, 5,6 y 7 son parámetros por defecto para la respectiva configuración del servidor DM4CHEE.
- ✓ En la línea 8 se adjunta los mensajes con el uuid del estudio realizado al respectivo paciente.

3.5. Solicitud con el celular

Como tercera parte, para la comunicación con dispositivo móvil se realizó un complemento en Android Studio para utilizará Jade Leap y se lo utilice como plugin en el framework Flutter, después de realizar este complemento se crea el agente de PETICION para atender las solicitudes y con la ayuda de agentes se puede obtener información de los pacientes que se encuentran en el módulo HIS, además de establecer la relación con las imágenes que se encuentran en el módulo PACS. En este punto, los agentes son útiles para los momentos en que exista un nuevo pedido de un examen radiológico desde el módulo de SYSRIS o desde el aplicativo móvil.

En la Figura 12 se muestra cómo se comunica el agente de petición con el agente HIS para obtener los datos de un paciente

Figura 12
Comunicación con el agente de petición



En el Algoritmo 13 se define como se conecta el agente Móvil con el agente principal para el funcionamiento del agente se crea el contenedor secundario donde se acoplan o unen el contenedor con su respectivo agente.

Algoritmo 13. Creación de la instancia del contenedor movil

Output: AgentContainer

```
1. Runtime runtime ← Runtime.instance();
2. ProfileImpl profileImpl ← new ProfileImpl(true);
3. profileImpl.setParameter(ProfileImpl.CONTAINER_NAME, HOST);
4. profileImpl.setParameter(ProfileImpl.GUI, "false");
5. return ← runtime.createAgentContainer(profileImpl);
```

- ✓ En la línea 1 se crea una instancia en tiempo de ejecución esto ayuda que la CPU y memoria no se sobrecarguen.
- ✓ En la línea 2 se indica que la instancia va a contener un contenedor secundario.
- ✓ En la línea 3 se establece el nombre del contenedor y cual seria el host donde se va a crear.
- ✓ En la línea 4 se indica que no se va a utilizar un entorno gráfico para la visualización de los agentes.
- ✓ En la línea 5 se asigna el contenedor secundario.

CAPÍTULO 4

Resultados

4.1. Herramienta JMETER

Para medir los prerequisites de hardware se emplea JMeter, que es útil para el testeo de aplicaciones, es de código abierto desarrollada por Apache Software Foundation. Su objetivo es probar el rendimiento de las aplicaciones, además permite realizar pruebas unitarias y funcionales. A nivel empresarial se pueden realizar pruebas de carga y fue ejecutada por primera vez en 1998. Esta herramienta se basa en plugins lo que le permite ampliarse con facilidad (Dotcom-Monitor, 2020).

De las más populares de código abierto esta herramienta permite el análisis del rendimiento de un sitio web o aplicación. su principal ventaja es la de actualización ya que permite admitir nuevas tecnologías claro también posee desventajas que es su interfaz ya que es compleja debido a la gran cantidad que posee. Dado a esta complejidad la curva de aprendizaje es amplia por ende no le es tan fácil competir con otras herramientas que existen en el mercado.

4.2. Carga y estrés en el uso del CPU y Memoria del computador

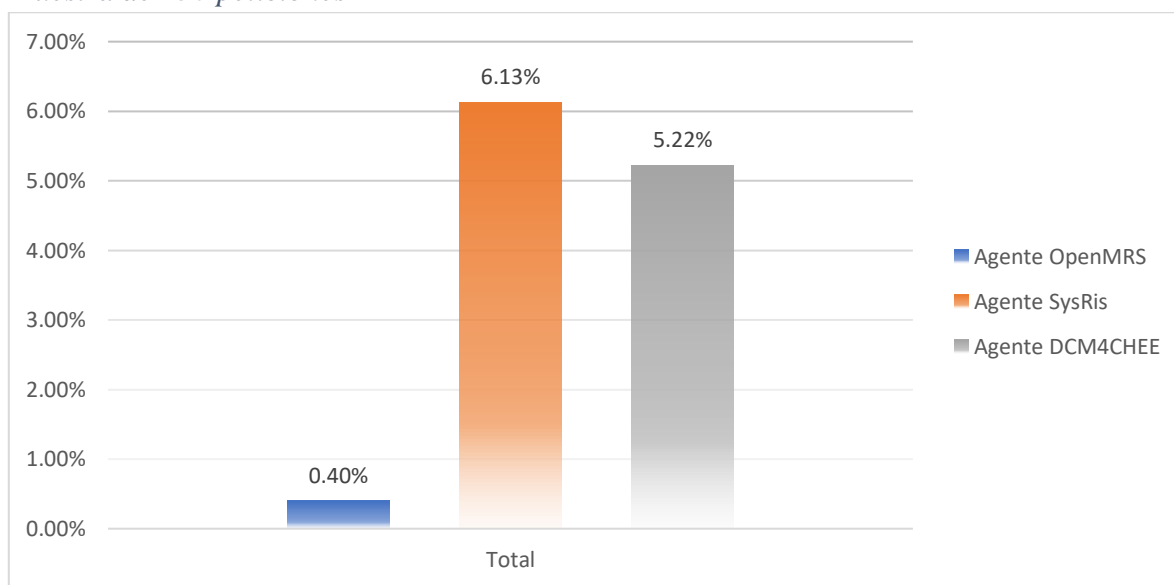
Se desea evaluar el número de peticiones que son soportadas por los siguientes agentes, para eso se realizaron tres escenarios para realizar las pruebas de performance: en primer lugar, se mide el uso del CPU, luego el uso de la memoria y finalmente, las métricas en tiempo de respuestas; con los agentes que se comunican a los servidores de OpenMRS, DCM4CHEE y SysRis.

4.2.1. Prueba 1: Uso del CPU

Como se puede observar en las figuras que se presentan a continuación, el uso del CPU, posee distintos valores para cada uno de los servidores que se emplean, en la Figura 13, se observa una muestra con 250 repeticiones:

Figura 13

Muestra de 250 peticiones



En el caso del Agente OpenMRS se observa un rendimiento de 0,40%; a este le sigue el Agente DCM4CHEE, con un índice del 5,22% y finalmente, se encuentra el Agente SysRis con un promedio de 6,13%, este resulta ser el más alto. En el resto de las Figuras (14 y 15), se exponen las muestras con 500 y 1000 repeticiones, respectivamente. En ambas figuras se observa que el rendimiento del CPU del agente SysRis es más alto y el agente OpenMRS y con el promedio más bajo.

Figura 14

Muestra de 500 peticiones

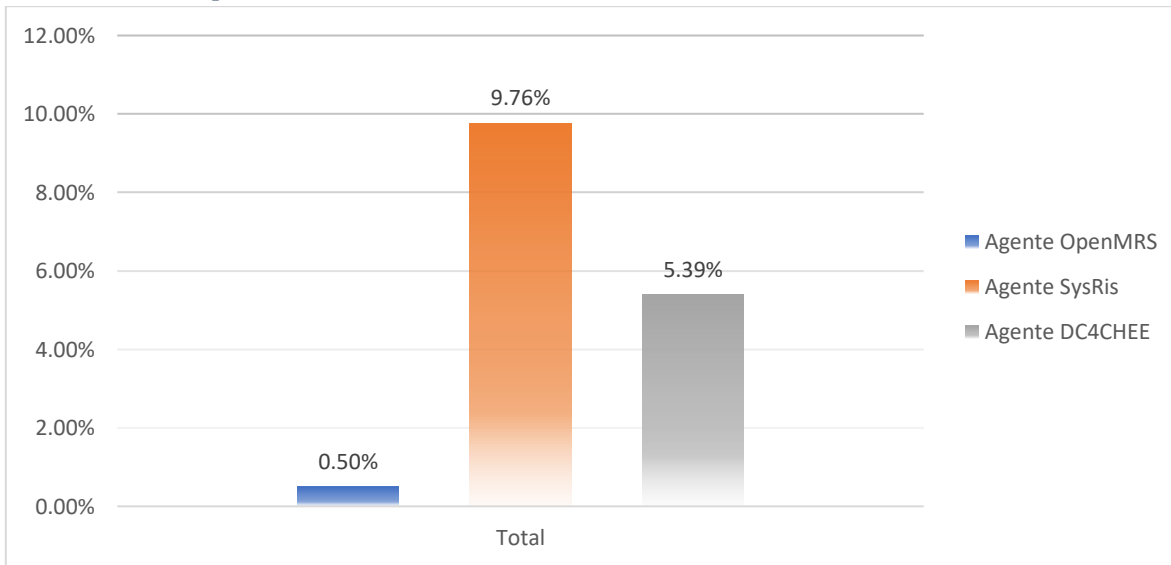
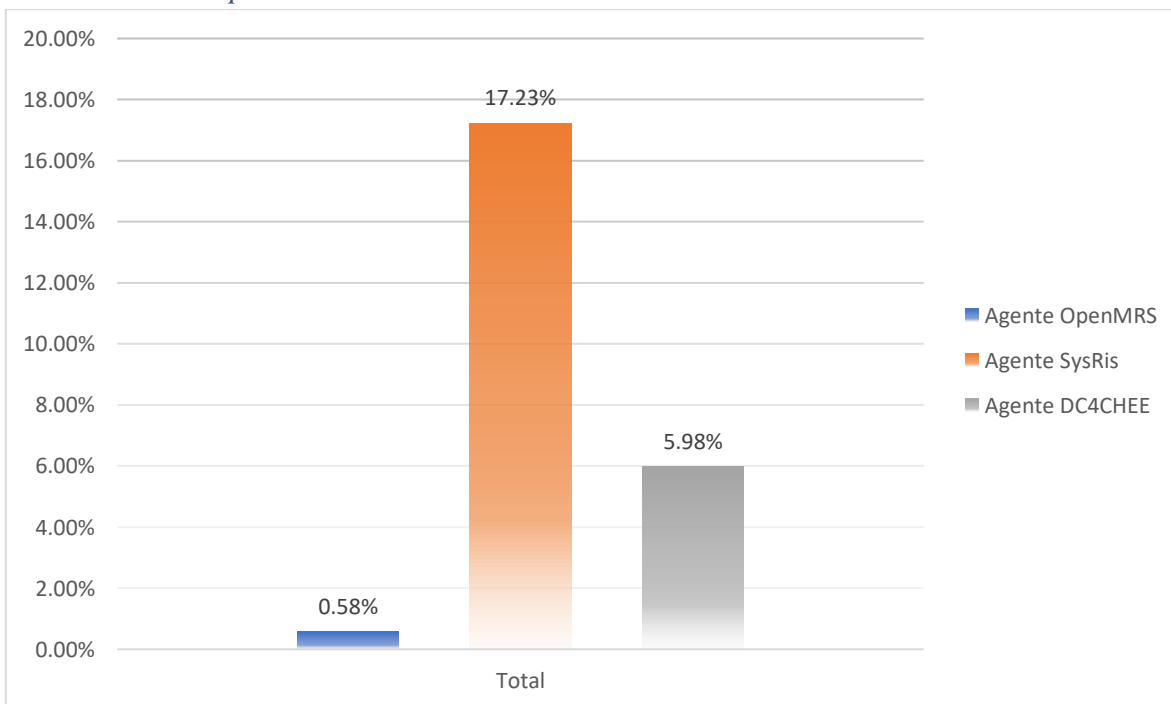


Figura 15

Muestra de 1000 peticiones



4.2.2. Prueba 2: Uso de la memoria

A continuación, se presentan las Figuras 16, 17 y 18 donde se comprueba el uso de la memoria con los tres agentes: OpenMRS, SysRis, DC4CHEE, al igual que en la prueba anterior se hacen mediciones con muestras de 250, 500 y 1000 repeticiones.

Figura 16

Muestra de 250 peticiones

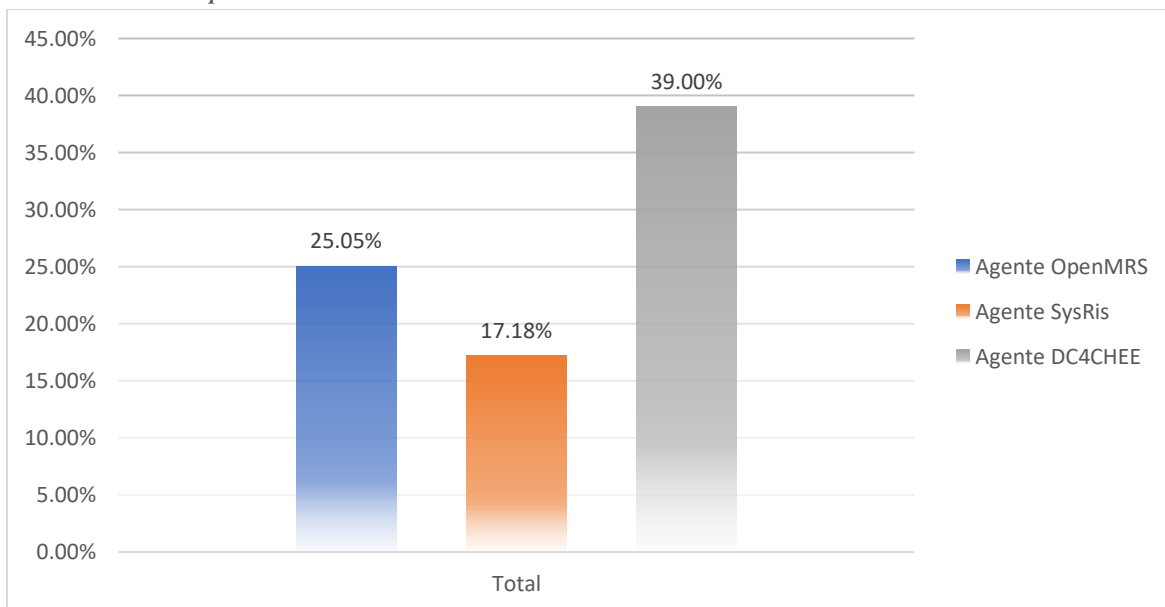


Figura 17

Muestra de 500 peticiones

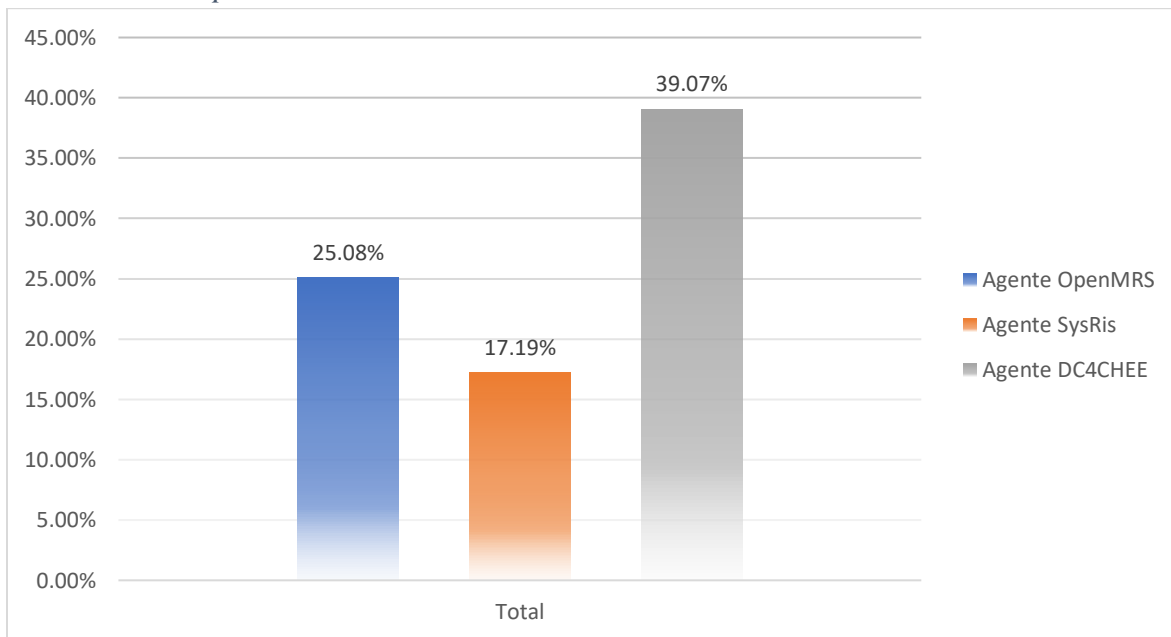
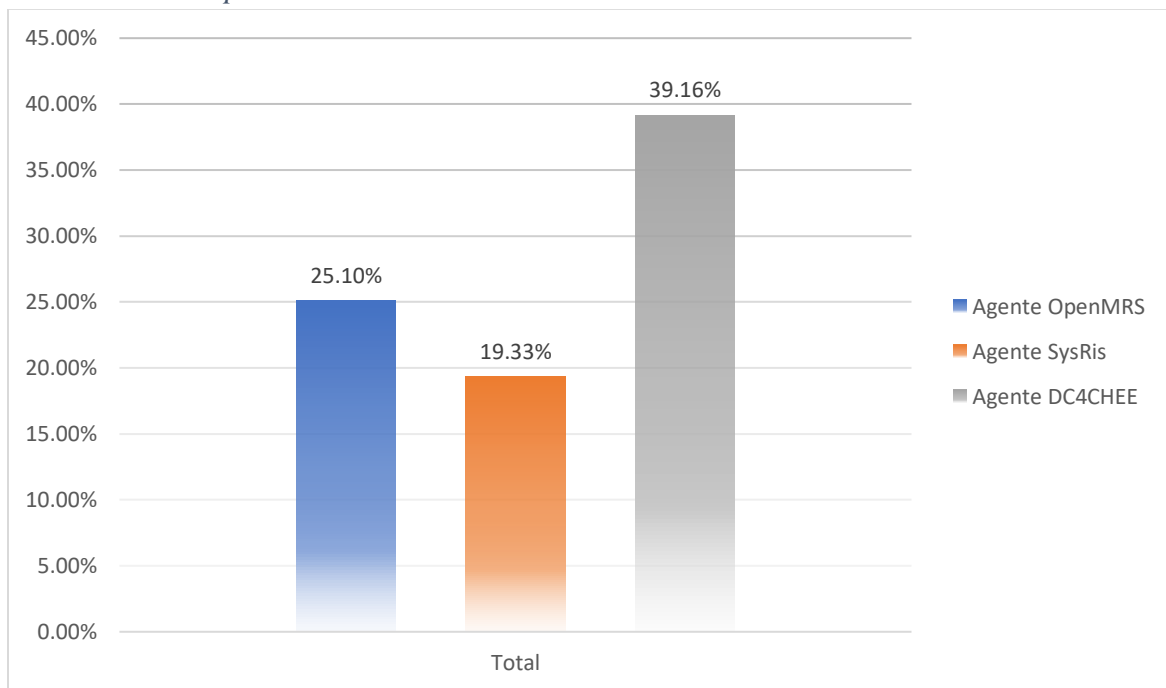


Figura 18

Muestra de 1000 peticiones



4.3. Métricas de tiempos de respuestas

Para esta prueba se detalla el tiempo de respuestas en milisegundos que se realizó con los escenarios de 250, 500 y 1000 solicitudes realizadas a los agentes OpenMRS y DC4CHEE, desde el agente RIS donde se puede observar en las Figuras 19, 20 y 21 respectivamente, cabe indicar que existe un mayor consumo en los tiempos por el Agente DCM4CHEE esto es debido a la transferencia de imágenes para ser mostradas al usuario final sea este desde el frontend de la página web o del teléfono móvil.

Figura 19

Muestra de 250 peticiones

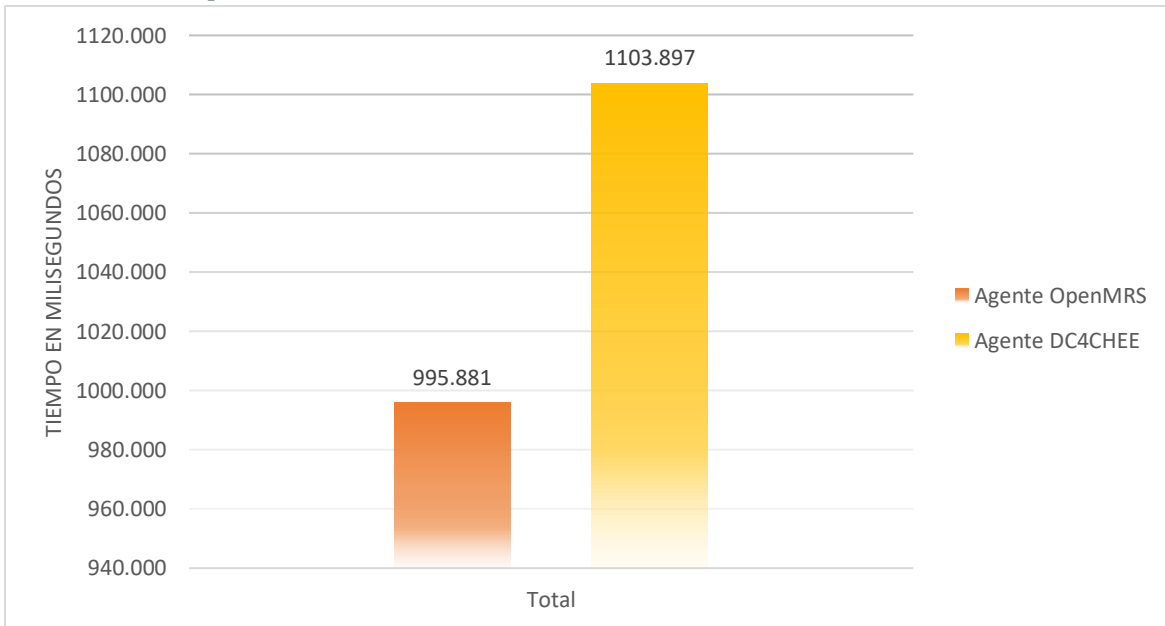


Figura 20

Muestra de 500 peticiones

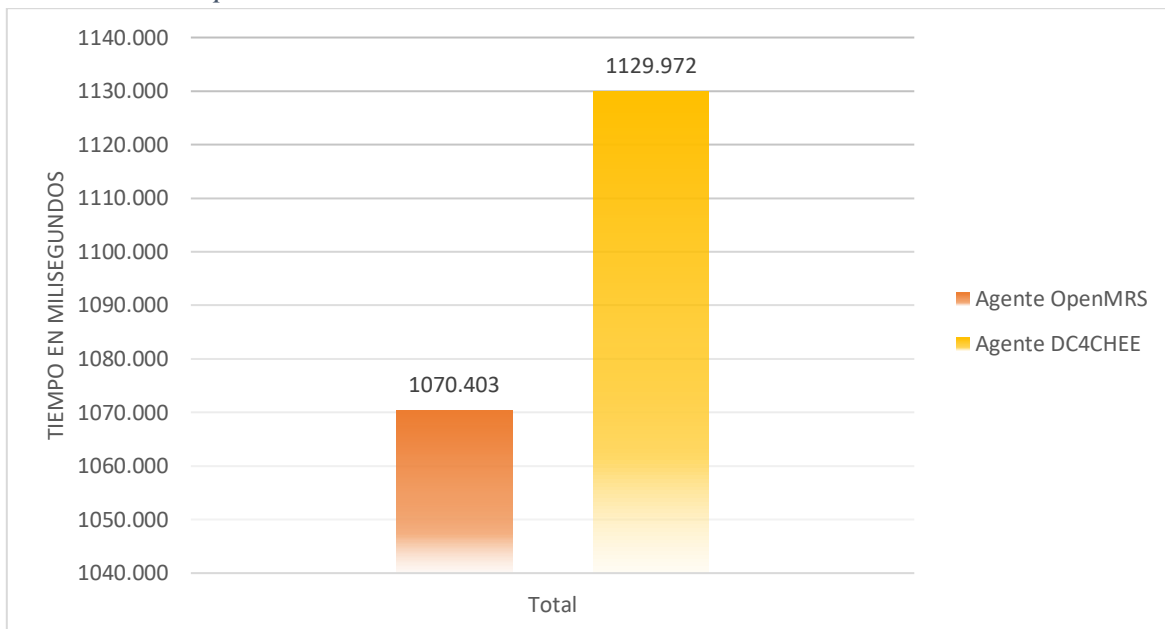
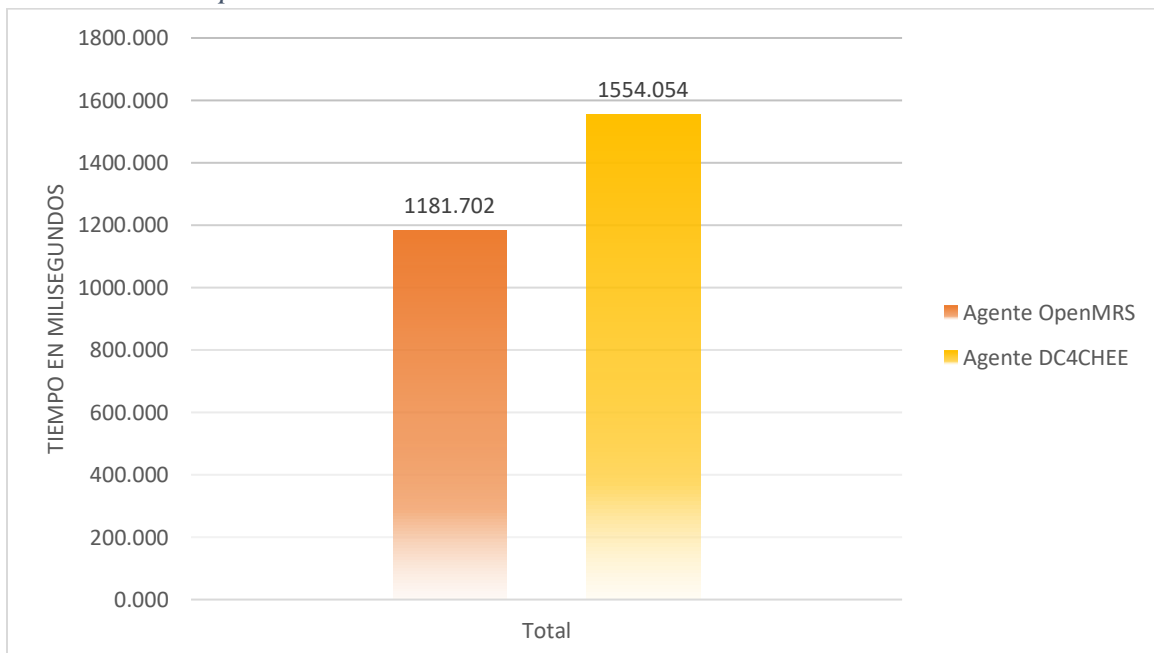


Figura 21

Muestra de 1000 peticiones



4.4. Métricas de calidad del proyecto

Calidad es un conjunto de condiciones establecidas en medidas, permitiendo determinar si un proyecto esta listo para ser lanzado a producción. El software de SONARQUEBE es una herramienta para la evaluación del código fuente, con datos estadísticos nos muestra su objetivo que es mejorar sobre diferentes métricas nuestro código permitiendo lanzar un software de calidad.

4.4.1. Código duplicado

Es una métrica relacionada con la duplicidad del código llámese a este la repetición de las mismas secuencias de código fuente, esto demuestra una programación pobre o perezosa lo que impide una de las buenas practicas que es la reutilización de código dando una gran desventaja que es el alto costo del mantenimiento.

4.4.2. Código muerto

Dado a la restructuración de código suele darse que parte del el no es utilizado o nunca lo fue, lo que se aconseja es eliminar el código no utilizado en nuestra aplicación para que este sea limpio y de fácil mantenibilidad. (Márquez, 2020).

4.4.3. Estándares de codificación

Son convenciones para escribir un código limpio, ordenado y que sea fácil de entender para cualquier persona, estas convenciones hacen referencia a tabulaciones, espacios, nombres de variables independientemente del lenguaje en el que se programe.

4.4.4. Bugs

Es un error en el software que puede ser de distintos niveles bajo, alto o medio lo que hace es que este funcione de una manera incorrecta o haga que nuestro aplicativo falle en inesperadamente.

Para medir la calidad de la aplicación web nos vamos a referenciar a la Tabla 2 en la cual se detalla los parámetros con los que se va a evaluar la calidad del software.

Tabla 2

Medición de la calidad de la aplicación web

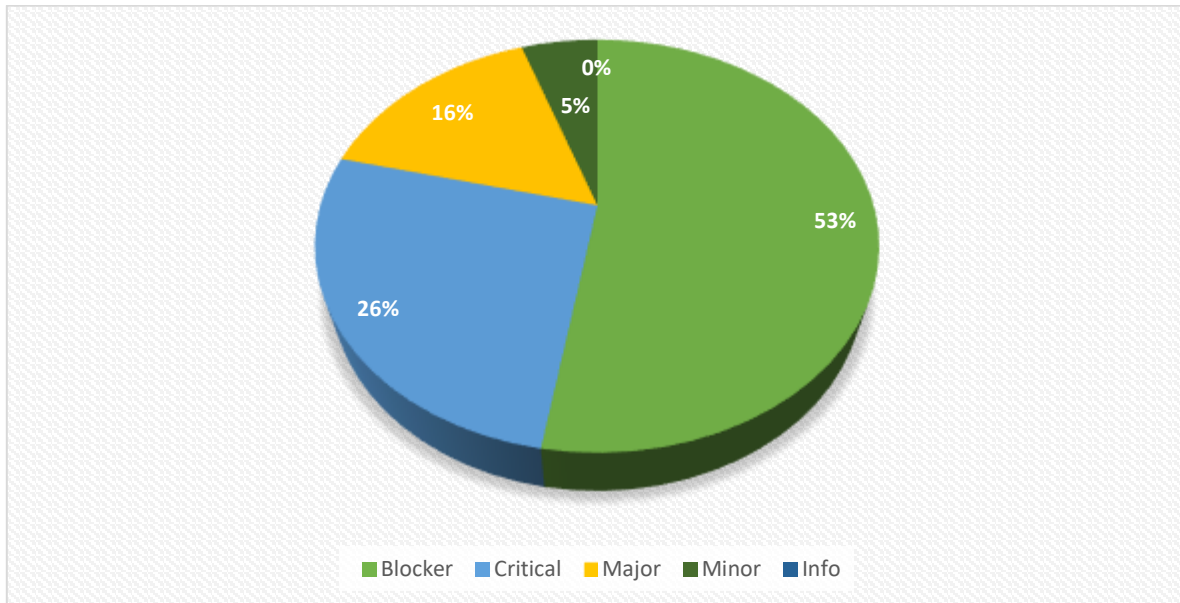
Evaluación de Calidad	Criterio	Reporte Obtenido	Totales
Blocker	10	15.00	150.00
Critical	5	28.00	140.00
Major	3	79.00	237.00
Minor	1	81.00	81.00
Info	0	0.00	0.00
		Totales	608.00
		Números de líneas de código	3579.00
		Calidad de Software	83.01

En la Figura 22 se muestra las estadísticas del análisis del código ejecutado por la herramienta de software SonarQube, donde se observa donde nos indica que se puede hacer mejoras en el código por sus vulnerabilidades en un 16%, código inseguro o repetido 5%,

código errado un 26%, Observamos un 53% con alta probabilidad de afectar el comportamiento de la aplicación en producción.

Figura 22

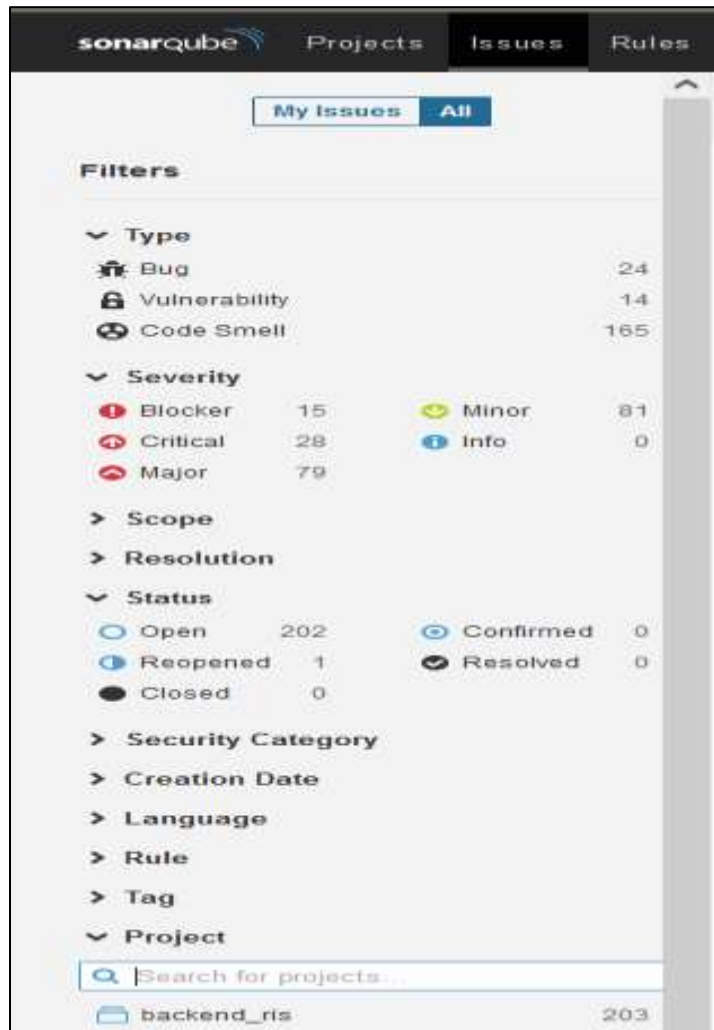
Calidad de la página web



En la Figura 23 se muestra el reporte que nos provee la herramienta SonarQube donde nos indica la severidad del código y nos muestra que la probabilidad de fallas graves en el es de 15 puntos, con poca probabilidad de afectar el comportamiento con 28 puntos, falla de calidad unos 79 puntos, errores ligeros unos 81 puntos eso puede ser debido a actualizaciones en plugins o el idioma con el que se desarrollo es el español, estos resultados son de un total de 3579 líneas analizadas.

Figura 23

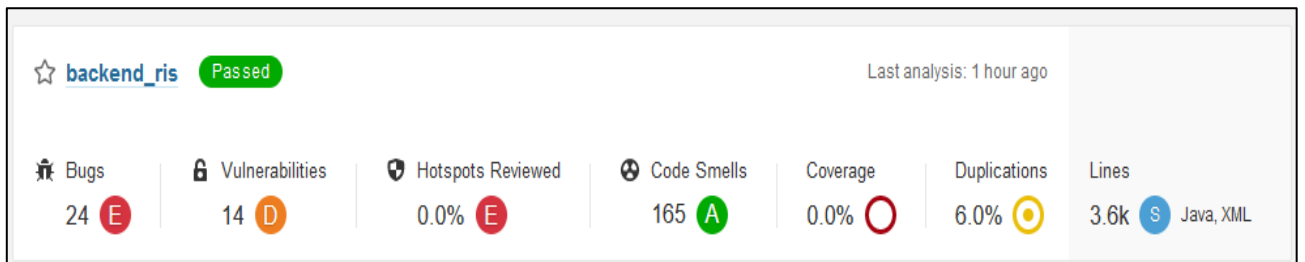
Reporte Software SonarQube detallado



En la Figura 24 nos indica que el código este aprobado, aunque se puede mejorar para minimizar los riesgos, corregir los bugs y sus respectivas vulnerabilidades, pero existe una buena calidad de software.

Figura 24

Reporte SonarQube



CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El presente trabajo realizó el desarrollo de un módulo y una aplicación móvil que tuvo como principal finalidad posibilitar la comunicación entre los sistemas de HIS y PACS con sistemas multiagentes, empleando herramientas de código abierto. Para esto se utilizó el lenguaje de programación de Java. Todo esto se realiza para facilitar los estudios médicos de pacientes que requieren de servicios de radiología. Este sistema es capaz de minimizar los costos y presentar toda la información en un mismo lugar, facilitando la comunicación entre médico, paciente e información visual.
- Se desarrolló el módulo de radiología y la aplicación móvil para el departamento médico con sistemas multiagentes, esto sin dudas posibilita la eficacia del sistema para que la información del paciente, sus imágenes de radiología y otras informaciones, se encuentren en el mismo lugar. Esto facilita el trabajo del médico y a la vez, de la satisfacción del cliente.
- Se logró la integración del módulo, que tuvo como objetivo facilitar la comunicación entre HIS y PACS. Los beneficios que esto trae consigo son diversos, con ello se logra un servicio de radiología digital, el acceso a las imágenes de manera factible y económica. En un futuro esto podría impulsar la telerradiología, donde los médicos revisen los casos de urgencia desde sus computadoras personales en cualquier momento del día.

- Las evaluaciones de las métricas de rendimiento del sistema web y móvil con agentes resultaron ser positivas, esto significa que el sistema es factible y eficaz. Además, se calificó la calidad de la página web a través del Reporte Software SonarQube detallado, midiendo indicadores como: Blocker, Critical, Major, Minor e Info. De esta manera, se puede concluir que las funciones del software satisfacen las necesidades del usuario. También se comprobó la eficiencia, donde se evaluó el nivel de rendimiento del software y el volumen de recursos utilizado.

5.2. Recomendaciones

- Se recomienda antes de la implementación del sistema PACS realizar estudios dirigidos a valorar el costo-beneficio, analizando los costos de mantenimiento anual. Aunque está claro que se aumentará la eficiencia del sistema de salud, esto muchas veces no es cuantificable.
- Otro tema que se recomienda estudiar es la posibilidad de “caídas del sistema”, esto representaría una gran dificultad pues un hospital no podría funcionar correctamente sin el acceso a la información de sus pacientes o las imágenes radiológicas.
- Dado que es un complemento del sistema se recomienda que este conectado a la red para que de esta manera se retroalimente de la información de los pacientes ya que si por alguna razón es desconectado y se lo vuelve a conectar después de un tiempo demoraría en actualizar los datos y puede ser que se colapse por la cantidad de data a actualizar.
- Resulta esencial una capacitación para los médicos y usuarios que manejan el sistema, pues deben tener un conocimiento mínimo de computación. Antes de

manejar se les sugiere la elaboración de una estandarización para que los datos siempre sean lo más homogéneos y entendibles para el resto del personal.

- Si se desea realizar modificaciones sobre alguna función en el software se sugiere que se lo haga bajo la metodología en el que los clientes están familiarizados para que de esta manera el impacto de algún cambio sea lo menor posible y no exista una resistencia.
- La importancia de definir la información a la cual el paciente pueda acceder y por otro lado la del medico, esto evitara una fuga de datos que podría traer consecuencias o problemas de confidencialidad sobre las historias de los pacientes.

REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, J., Bolivar, A., Hidrobo, F., & Cerrada, M. (2013). *Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial*. Mérida,: Universidad de Los Andes.
- Alvares, L., & Sichman, J. (1997). *Introdução aos sistemas multiagentes*. Brasilia: XVII Congresso da SBC-Anais JAI'97.
- Bellifemine, F., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE*. Chichester: Wiley.
- Bravo, C., Aguilar, J., & Rivas, F. (2004). Diseño de una arquitectura de automatización industrial basadas en sistemas multiagentes. *Revista Ciencia e Ingeniería*, 76.
- Chalen, J. (2017). *Análisis Y Diseño De Un Sistema PACS*.
- Clinic Cloud. (diciembre de 2021). *¿Qué es el formato DICOM? Las claves del estándar en imágenes médicas*. Obtenido de Clinic Cloud: <https://clinic-cloud.com/blog/formato-dicom-que-es-estandar-imagenes-medicas/>
- Dibarboure, L. (2016). Imagenología. Del Consultorio Radiológico al Departamento Clínico de Imagen. *Tendencias en Medicina*, 13-19.
- Dibarboure, L., Sgarbi, N., & Febles, G. (2021). Nuevas tecnologías en imagenología. Visión de referentes nacionales. *Revista de Imagenología*, 24(2), 05-08.
- Dotcom-Monitor*. (2020). Obtenido de <https://www.dotcom-monitor.com/wiki/es/knowledge-base/limitaciones-de-jmeter-en-las-pruebas-de-carga-spa/>

- FIB, ECS DI. (2019). *Sistemas Multiagentes*. Obtenido de Curso 2018/2019 : ECS DI - Curso 2018/2019 - FIB
- García, A., & Ortiz, A. (2019). *Análisis y diseño de un prototipo para un ecosistema de PACS virtuales*. Quito: Universidad Politécnica Salesiana.
- Gómez, L., & Yunda, L. (2013). *Informática Médica: Sistemas de Información y Estándares en Salud: Modelo de Aplicación*. Madrid: Paidós.
- Health Level Seven International. (2019). *Health Level Seven International*. Obtenido de <http://www.hl7.org/about/index.cfm?ref=nav>
- Hurka, T. (2019). *Visual VM*. Obtenido de <https://visualvm.github.io/graal.html>
- Hurtado, C., Ramirez, M., Alanis, A., Vazquez, S., Ramirez, B., & Manrique, E. (2018). Towards a multi-agent system for an informative healthcare mobile application. *KES international symposium on agent and multi-agent systems: technologies and applications*, 215-219.
- Jennings, N., & Wooldridge, M. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115–152.
- Llontop, F. (2019). *Metodología formativa en el sistema del programa archivo de comunicación de salud (PACS) dirigido a los tecnólogos médicos del servicio de emergencia, en radiología del Hospital Guillermo Almenara Irigoyen*. Lima: Universidad José Carlos Mariátegui.
- Marchetti, T., & García, A. (2013). *Plataformas para desarrollo de sistemas multiagente*. V Workshop de Investigadores en Ciencias de la Computación.

Márquez, R. (2020). *Paradigma*. Obtenido de <https://www.paradigmadigital.com/dev/evalua-la-calidad-de-tu-codigo-con-sonarqube/>

Martínez, Y., Vega, L., & Ferras, Y. (2014). Concepción del módulo de pacientes para el Sistema de Información Radiológica a las RIS. *Revista Cubana de Informática Médica*, 168-183.

Mazzoncini, P., Marques, S., Elías, J., & Santos, J. (2011). Implantación de un MINI-PACS en un Hospital Universitario. *Colegio Brasileño de Radiología y Diagnóstico por la Imagen*, 221-224.

Ortiz, A., & Noroña, A. (2019). *Análisis y diseño de un prototipo para un ecosistema de PACS virtuales*. Quito: Universidad Politécnica Salesiana.

Pérez, J., Arencibia, M., Jiménez, D., & Tellería, M. (2018). Hospital Clinical Information System: architecture and the map of beds. *Revista de Ciencias Médicas de Pinar del Río*, 22(1), 92-100.

Quality devs. (05 de 07 de 2019). Obtenido de <https://www.qualitydevs.com/2019/07/05/que-es-flutter/>

Rodríguez, Y. (2014). Concepción del módulo de pacientes para el Sistema de Información Radiológica a las RIS. *Revista Cubana de Informática Médica*, 169-183.

Russell, P., & Norvig, S. (2004). *Inteligencia Artificial: Un Enfoque Moderno*. Madrid: Prentice Hall Hispanoamericana.

Shoham, Y. (1993). Agent-oriented programming. *Artif. Intell.*, 60(1), 51–92.,.

- Stansfield, S., Orobato, N., Lubinski, D., Uggowitz, S., & Mwanika, H. (2008). *El caso de una arquitectura de sistema de información sanitaria nacional; un eslabón perdido para orientar el desarrollo y la implementación nacionales*. Bellagio : Making the eHealth Connection.
- Ughetti, M., Trucco, T., & Gotta, D. (2008). Development of agent-based, peer-to-peer mobile applications on ANDROID with JADE. *Segunda Conferencia Internacional sobre Computación, Sistemas, Servicios y Tecnologías Móviles Ubicuos* (págs. 287-294). Valencia: IEEE.
- Vega, L., Ciudad, F., Duque, E., & Soler, G. (2020). Radiology Information System XAVIA RIS. *Revista Cubana de Informática Médica*, 12(2).