

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Modelado de un Canal de Comunicación de datos de una Aplicación de
Teleconsulta para la Ciudad de Guayaquil

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:

Xavier Aizaga Villón

Karen Alarcón Ballesteros

GUAYAQUIL - ECUADOR

Año: 2020

DEDICATORIA

Xavier Aizaga

A Dios por darme la vida, a mi familia por ser esa red de soporte incondicional.

Karen Alarcón

El presente proyecto lo dedico a mis padres Sonia y Ángel, a mis abuelos Lidia y Hugo por haberme forjado como la persona que soy. Me formaron con reglas y libertades pero al final de cuentas me brindaron su apoyo y me motivaron a lograr mis metas. Gracias familia. A mis profesores Washington y José , personas de gran sabiduría que me guiaron a este punto. A mi compañero Xavier por motivarme y ayudarme durante toda esta experiencia.

AGRADECIMIENTOS

Xavier Aizaga

Mi agradecimiento a ESPOL por el conocimiento impartido estos años a través de grandes profesores. A mis compañeros y ayudantes por ser esa guía a lo largo de mi carrera.

Karen Alarcón

Mi más sincero agradecimiento a mi familia la cual me motivó a seguir a pesar de todo lo que se ha puesto en el camino para lograr esta meta. Quiero aprovechar y agradecer a mis amigos que fueron un soporte importante durante toda mi travesía por la universidad, en especial a alguien con quien he compartido muchos años el cual nunca soltó mi mano y me incentivó a continuar, Carlos Luis Ponguillo Burbano.

DECLARACIÓN EXPRESA

”Los derechos de titularidad y explotación, me(nos) corresponde conforme al reglamento de propiedad intelectual de la institución; (nombre de los participantes) y doy(damos) mi(nuestro) consentimiento para que la ESPOOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Xavier Aizaga

Karen Alarcón

EVALUADORES

José Eduardo Córdova García
PROFESOR DE LA MATERIA

Washington Velásquez Vargas
PROFESOR TUTOR

RESUMEN

En la ciudad de Guayaquil a raíz de la pandemia Covid-19 el sistema de salud se saturó debido a la cantidad de personas afectadas por este virus. Esta investigación se desarrolla en base a un análisis previo donde se verificó que las aplicaciones de teleconsulta eran poco utilizadas debido al desconocimiento de la población sobre la telemedicina en la ciudad. El presente trabajo tiene como propósito plantear una solución basada en un modelo de datos para la gestión de un sistema de teleconsulta utilizando la arquitectura de la plataforma FIWARE. Se diseñó una aplicación web de teleconsulta con framework ReactJS y librerías de Ant Design. Adicional, se usaron componentes como FIWARE Orion para orquestar el ciclo de vida de datos usando comunicación de tipo publicador/suscriptor y FIWARE Draco para el flujo de datos y notificaciones. En las pruebas de estrés y rendimiento realizadas con la herramienta Apache Jmeter se observó que la aplicación funciona en ambiente local y nube. En base a estos resultados la propuesta se puede implementar a gran escala por su modelo distribuido, su óptimo consumo de recursos y su procesamiento para gestionar eventos en tiempo real. Esta arquitectura inteligente que ofrece FIWARE es de fácil integración con otras tecnologías por su diseño.

Este documento se divide en cuatro capítulos: Capítulo I; introducción y análisis de la problemática del uso de los sistemas de teleconsulta y la aplicación de la telemedicina en el Ecuador, Capítulo II; se detalla la solución propuesta, su diseño y tecnologías para su implementación; Capítulo III; se analiza los resultados de las pruebas, en cuanto a métricas y funcionamiento. Finalmente, en el capítulo IV se manifiestan las conclusiones con respecto a los resultados obtenidos en los capítulos previos y posibles mejoras o soluciones a futuro con el uso de la tecnología moderna.

Palabras Clave: API, código abierto, Seguridad de la información, Sistemas distribuidos, Teleconsulta.

ABSTRACT

In the city of Guayaquil, as a result of the Covid-19 pandemic, the health system was saturated due to the number of people affected by the virus. This research is developed based on a previous analysis where it was verified that teleconsultation applications were little used due to the lack of knowledge of the population about telemedicine in the city. The purpose of this work is a solution based on a data model for the management of medical data in a teleconsultation system using the FIWARE platform architecture. A teleconsultation web application was designed with ReactJS framework and AntDesign libraries, and additional components such as FIWARE Orion were used to orchestrate the data lifecycle using the communication channel and FIWARE Draco for data flow and notifications. In the stress and performance tests performed to the communication channel with the Apache Jmeter tool, it was observed that the application works in local and cloud environment, but better response times were obtained in the cloud environment. In based on these results, the proposal can be implemented on a large scale due to its distributed model, its optimal resource distributed model, its optimal resource consumption and its processing to manage events in real time. This intelligent architecture offered by FIWARE is easy to integrate with other technologies due to its design.

This document is divided into four chapters: Chapter I; introduction and analysis of the problem of the use of teleconsultation systems and the application of telemedicine in Ecuador, Chapter II; the proposed solution, its design and technologies for its implementation are detailed; Chapter III; The results of the tests are analyzed, in terms of amometric and performance. Chapter IV; the conclusions regarding the results obtained in the previous chapters and possible future improvements or solutions are expressed with the use of modern technology.

Keywords: Teleconsulting, distributed systems, information security, API, open source.

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	iii
ABREVIATURAS	ix
SIMBOLOGÍA	xi
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xiv
ÍNDICE DE ALGORITMOS	xv
1 INTRODUCCIÓN	1
1.1 DESCRIPCIÓN DEL PROBLEMA	2
1.2 JUSTIFICACIÓN DEL PROBLEMA	2
1.3 OBJETIVOS	3
1.3.1 OBJETIVO GENERAL	3
1.3.2 OBJETIVOS ESPECÍFICOS	3
1.4 MARCO TEÓRICO	3
2 Aproximación Metodológica e Investigativa	5
2.1 ¿POR QUÉ LA TELE-CONSULTA?	5
2.2 TIPO Y DISEÑO DE INVESTIGACIÓN	6
2.2.1 CRITERIOS DE INCLUSIÓN	6
2.2.2 CRITERIOS DE EXCLUSIÓN	7
2.3 MODELO DE AMBIENTE DE ESTUDIO	7
2.3.1 ARQUITECTURA GENERAL	7
2.3.2 ACTORES	7

2.3.3	POLÍTICAS Y SERVICIOS	8
2.3.4	MÉTRICAS Y VARIABLES	9
2.3.4.1	PRINCIPALES	9
2.3.4.2	SECUNDARIAS	10
2.4	PLATAFORMA DE INTERACCIÓN USUARIO	10
2.4.1	DISEÑO DE LA APLICACIÓN	11
2.4.2	SERVICIO DE VIDEOLLAMADAS	11
2.4.3	FUNCIONALIDADES	12
2.4.4	CASOS DE USO FLUJO DE COMUNICACIÓN	13
2.5	INFRAESTRUCTURA TECNOLÓGICA	14
2.5.1	FIWARE-ORION (CONTEXT BROKER)	15
2.5.2	FWARE-DRACO (APACHE NIFI)	15
2.5.3	ALMACENAMIENTO DE DATOS	16
2.6	CANALIZACIÓN DE DATOS	17
2.6.1	ENTIDADES EN FIWARE-ORION	18
2.6.2	SUBSCRIPCIONES EN FIWARE-ORION	19
2.6.3	EVENTOS Y NOTIFICACIONES	20
2.7	ALTERNATIVAS TECNOLÓGICAS	21
3	ANÁLISIS Y RESULTADOS	23
3.1	FUNCIONAMIENTO	23
3.2	ESCENARIOS Y AMBIENTES DE PRUEBAS	27
3.3	RESULTADOS EN TÉRMINOS DE MÉTRICAS	29
3.3.1	PARÁMETROS DEL SISTEMA	29
3.3.2	RECURSOS DEL SISTEMA	31
3.3.2.1	AMBIENTE EN LA NUBE	32
3.3.2.2	AMBIENTE LOCAL	33
3.4	RESOLUCIÓN DEL PROBLEMA	34
3.5	COSTOS Y ESCALABILIDAD	35
4	CONCLUSIONES Y RECOMENDACIONES	37
4.1	Conclusiones	37
4.2	Recomendaciones	38

4.3 Líneas Futuras	39
BIBLIOGRAFÍA	41
APÉNDICES	43
Apéndice A: Diagrama para Integración de elementos telemáticos	47
Apéndice B: Repositorio y Encuesta	49

ABREVIATURAS

IESS	Instituto Ecuatoriano de Seguridad Social
API	Application Programming Interface
TIC	Tecnologías de la información y la comunicación
FTP	File Transfer Protocol
IPSec	Internet Protocol Security
VPN	Virtual Private Network
FHIR	Fast Healthcare Interoperability Resources
OCB	Object context broker
GE	Generic Enables
JSON	Javascript Object Notation
HTTP	Hypertext Transfer Protocol

SIMBOLOGÍA

S Segundos

Mb Megabyte

Gb Gigabyte

Tb Terabyte

ÍNDICE DE FIGURAS

2.1	Arquitectura General de la Aplicación de Teleconsulta	8
2.2	Actores Involucrados	8
2.3	Diagrama de flujo de la comunicación	13
2.4	Actualización de medicinas en stock	13
2.5	Agendar cita	14
2.6	Modificación del historial clínico	14
2.7	Funcionamiento de FIWARE Orion	15
2.8	Flujo de datos Draco Apache Nifi	16
2.9	Almacenamiento de datos en MySql	17
2.10	Actualización de medicina	17
2.11	Datos a través de FIWARE Draco	18
2.12	Datos en la base de datos MySql	18
2.13	Flujo de datos de la entidad	18
2.14	Notificación de creación del perfil	21
2.15	Notificación de actualización en el stock de medicinas a vía Twitter . .	21
2.16	Notificación para agendar citas	21
3.1	Funcionamiento de frontend	24
3.2	Funcionalidades de cada usuario dentro de la aplicación	24
3.3	Funcionamiento de backend	24
3.4	Servicios en Docker	25
3.5	Servicio GE FIWARE Orion	25
3.6	Servicio GE FIWARE Draco sin errores	25
3.7	Notificación de creación de usuario	26
3.8	Notificación de creación de cita	26
3.9	Notificación de la actualización del stock	26
3.10	Videoconferencia de prueba entre un paciente y un doctor	27

3.11	Gráfico tiempo de respuesta vs solicitud	30
3.12	Gráfico respuesta HTTP por segundo	30
3.13	Gráfica de usuarios vs. tiempo de respuesta	31
3.14	CPU utilizado por escenario de prueba (%CPU Vs. Tiempo)	32
3.15	Memoria RAM utilizado por escenario de prueba (GB vs. Tiempo)	32
3.16	Espacio de disco duro ambiente nube (KB vs Tiempo)	33
3.17	CPU y memoria RAM ambiente local	34
3.18	Disco duro ambiente local	34
3.19	Diagrama de Gantt	35
1	Integración de signos vitales con Sygnus	47
2	Encuesta de aceptación	49

ÍNDICE DE TABLAS

3.1	Escenario de prueba frontend	27
3.2	Requisitos de Hardware y software local	28
3.3	Requisitos de Hardware y software nube	28
3.4	Tabla resultado de pruebas escenario 2	29
3.5	Tabla resultado de pruebas escenario 2	29
3.6	Tabla de actividades del proyecto	36
3.7	Tabla de costos del proyecto	36

ÍNDICE DE ALGORITMOS

1	Creación de entidades (Stock de medicina)	19
2	Creación de suscripciones (Stock de medicina)	20

CAPÍTULO 1

1. INTRODUCCIÓN

En Ecuador se aplican normas de bioseguridad donde se prioriza el distanciamiento social para reducir la cantidad de contagios, cuidar la integridad física y mental de la población¹. Los servicios y entidades de salud se arriesgan a un colapso causado por una sobrecarga de consultas médicas que se deben atender por medios virtuales. Las Tecnologías de la información y Comunicación (TICs) tienen un impacto importante en el desarrollo de las sociedades latinoamericanas, especialmente en aspectos relacionados con la salud pública [1]. Las TICs ofrecen arquitecturas que permiten a aplicaciones, plataformas web o sistemas inteligentes mejorar la calidad de vida del ser humano [2], con el paso del tiempo estas herramientas incrementan su participación en el ámbito de la salud, e.g., el uso de tecnologías de Blockchain para el manejo de datos en el área de la salud [3] y la monitorización de signos vitales del paciente en tiempo real [4].

La telemedicina se refiere a las interacciones que ocurren entre un médico y un paciente con el fin de proporcionar asesoramiento a través de medios electrónicos [5], también es complementaria a la consulta médica, llegan a ser cruciales para obtener el diagnóstico médico a distancia y mantener cierta vigilancia del paciente, cuando son escasos los médicos especialistas [6]. Las ventajas de la aplicación de la telemedicina son: la accesibilidad; el paciente puede interactuar con su médico desde cualquier lugar, la eficiencia, la calidad de la atención médica y la equidad de servicios a las poblaciones rurales y urbano marginales [7]. Por tal motivo, es imprescindible que las instituciones y/o entidades de salud pública, tengan a su disposición un sistema adecuado que permita a los pacientes comunicarse con su médico y revisar su historial clínico en tiempo real.

¹<https://www.mgc.es/blog/ventajas-de-la-teleconsulta-y-la-videoconsulta-medica/>

1.1 DESCRIPCIÓN DEL PROBLEMA

La tasa de mortalidad de acuerdo a las cifras del Registro Civil del Ecuador² hasta Octubre del 2020 muestran un exceso de 37.070 decesos, en comparación del promedio realizado en los años 2018 y 2019. Las estadísticas indican que están aumentando los casos de contagio, generando una alta demanda en el servicio de consultas para medicina general y médicos especialistas. El Instituto Ecuatoriano de Seguridad Social (IESS) cuenta con un sistema de teleconsultas por teléfono, que es muy deficiente³ para la atención y el análisis de la situación del paciente. Esto ocasiona una incorrecta evaluación por parte del médico. La mala gestión administrativa en las bodegas del IESS ha causado escasez⁴ de los fármacos privando a los pacientes de medicina para su tratamiento. En las aplicaciones de teleconsulta se debe tener presente los siguientes criterios: seguridad de los datos, la concurrencia, tolerancia a fallos y la confiabilidad. Además requieren que los datos médicos, la confidencialidad y el anonimato de los datos del paciente sean prioridad.

1.2 JUSTIFICACIÓN DEL PROBLEMA

Los sistemas de teleconsultas mejoran la atención, aseguran la información del historial clínico de los pacientes, evitan el colapso de los servicios de salud pública y privada, reducen el contagio de enfermedades virales, optimizan la economía del hogar evitando gastos de movilización innecesarios, permiten llegar a zonas de difícil acceso geográfico generando igualdad a la población, e.g., el historial clínico puede ser compartido entre profesionales, acceso a la información desde cualquier parte del mundo y propone un servicio de notificaciones de medicinas disponibles en farmacia, minimizando la asistencia al hospital o centro de salud [8]. La teleconsulta ayuda a países como Ecuador a mejorar sus servicios de salud, ya que su cobertura es global, descentraliza la asistencia directa en los hospitales públicos y maneja pacientes de manera individual y de forma confidencial [5].

²<https://www.eluniverso.com/noticias/7785484/mapa-coronavirus-covid-ecuador>

³<https://www.pressreader.com/ecuador/el-diario-ecuador/20200527/281612422610358>

⁴<http://www.pichinchacomunicaciones.com.ec/existe-escasez-de-medicinas-en-hospitales-del-iess-y-del-msp/>

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Modelar un canal de comunicación de datos utilizando tecnologías de código libre y de la plataforma FIWARE para la distribución confiable de datos médicos en un sistema de teleconsulta.

1.3.2 OBJETIVOS ESPECÍFICOS

- Diseñar una aplicación Web reactiva utilizando el framework ANTD Design que permita la comunicación con una infraestructura basada en contexto.
- Implementar una infraestructura tecnológica mediante la plataforma FIWARE para la distribución confiable de datos de ámbito médico.
- Analizar el modelo de comunicación utilizando métricas de rendimiento y técnicas de stress para la verificación de la calidad del sistema .

1.4 MARCO TEÓRICO

Una red de telemedicina estándar se compone de tres elementos: servidor web; registro de las transacciones del usuario y proveer control de acceso, un servidor de transferencia de archivos (FTP) y un servidor de base de datos. Las limitantes de esta arquitectura cliente-servidor son: la escalabilidad vertical al momento de falta de recursos, memoria o disco duro, latencia en el envío de peticiones hacia un mismo servidor y tolerancia a fallos con interrupción del servicio [9].

Por ejemplo, un sistema que se desarrolló en la telemedicina extrajo información mediante un sensor de pulso que detecta el ritmo cardíaco e identifican la localización del paciente, siendo estos almacenados, interpretados y presentados mediante una aplicación web. Este sistema se desarrolló en un servidor web diseñado en Django con framework basado en python usando una base de datos relacional (MySql) [10].

Los micro-servicios representan patrones de arquitectura de software para el desarrollo de aplicaciones de servicios que se ejecutan de forma independiente,

trabajando con mecanismos de comunicación ligeros como HTTP en diversos escenarios, e.g., en el ámbito médico, mejoran la precisión del diagnóstico diferencial estableciendo predicciones basadas en sistemas expertos u otros tipos de inteligencia artificial [11], ó en [12] se implementó una técnica de procesamiento en la niebla para la monitorización remota de datos biométricos (presión arterial y electrocardiogramas), y ambientales en un entorno doméstico diseñado para la población anciana proporcionando un sistema seguro y flexible. Asimismo, en [13] se aplicó una arquitectura basada en micro-servicios para la implementación de un esquema de detección y diagnóstico de fallas a dispositivos de rehabilitación de rodilla, con el fin de facilitar un protocolo de desarrollo médico para la recuperación de la movilidad del paciente en caso de vulnerabilidad geográfica, demostrando que una arquitectura distribuida aumenta la eficiencia del desarrollo, debido a que el diseño de los componentes, implementación y la validación se realizan de forma independiente.

Considerando el análisis de las ciencias aplicadas en el área de la telemedicina, Ecuador necesita una plataforma web de teleconsultas con tecnologías que facilite y satisfaga las necesidades humanas, se propone un sistema open source que utilice módulos como: FIWARE-Draco basado en Apache Nifi; un sistema fácil de usar, potente y confiable para el procesamiento y la distribución de datos, otro módulo es FIWARE-Orion Context Broker; administra todo el ciclo de vida de la información de contexto y su disponibilidad, incluidas actualizaciones, consultas, registros y suscripciones⁵. Además de integrar la tecnología Docker con la creación de contenedores ligeros y portables que puedan ejecutarse en cualquier máquina⁶. La información se protege mediante el protocolo PEP-Proxy⁷ en los puntos finales expuestos por habilitadores genéricos de FIWARE o servicios REST, y se controla el acceso a las aplicaciones backend. En el frontend, se utiliza ANTD Design basado en Reactjs que permite crear aplicaciones mucho más flexibles y escalables de forma inmediata.

⁵<https://www.FIWARE.org/developers/catalogue/>

⁶<https://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>

⁷<https://FIWARE-pep-proxy.readthedocs.io/en/latest/>

CAPÍTULO 2

2. Aproximación Metodológica e Investigativa

Las aplicaciones de teleconsulta se vieron solicitadas en la Ciudad de Guayaquil el impacto que causó fue a tal punto que motivaron la realización de una investigación previa que dió paso a esta propuesta, mediante la metodología design thinking se identificó la necesidad que atravesó tanto el sistema de salud público como privado por medio de investigaciones, encuestas y entrevistas a los usuarios afectados por el colapso de servicios durante la pandemia Covid-19, que surgió a partir de Febrero del 2020 [14].

En base a esta necesidad se aplicó la técnica lluvia de ideas (brainstorming), entre ellas la mejor opción escogida fué el modelado de un canal de comunicación para las aplicaciones de teleconsultas. A continuación realizando el prototipo de la solución seleccionada se diseñó una aplicación de teleconsulta con las librerías de Ant Design simulando el ambiente de interacción con los usuarios finales, en la cual se realiza un canal de comunicación para enviar los datos recolectados hacia la infraestructura diseñada con la tecnología FIWARE.

Se identificaron posibles clientes potenciales que no cuentan con una plataforma interactiva y amigable, que permita visualizar y gestionar la información del paciente, en otros casos, clientes que tienen una aplicación pero necesita una infraestructura que integre la información con otros dispositivos proporcionando coherencia y persistencia de sus datos. La solución se evaluó mediante técnicas de stress y métricas de rendimiento verificando la calidad de la aplicación.

2.1 ¿POR QUÉ LA TELE-CONSULTA?

Las teleconsultas se impulsaron de forma exponencial a raíz de la pandemia que se propagó en el país, brindando apoyo a las consultas médicas cuidando la integridad

física y respetando las normas de bioseguridad establecidas, agregando valor sobre la experiencia y el conocimiento de las tecnologías implicadas, disminuyendo costos y resolviendo problemas asistenciales [15].

Las teleconsultas son una herramienta esencial para usar durante una pandemia debido a la seguridad y efectividad que proporcionan al momento de evaluar casos sospechosos, guiar el diagnóstico y tratado de pacientes, minimizando el riesgo de transmisión de la enfermedad. Además aligera la sobrecarga en los centros de salud reduciendo las visitas innecesarias a los servicios de emergencias [16].

2.2 TIPO Y DISEÑO DE INVESTIGACIÓN

La investigación que se realiza tiene un enfoque aplicativo y cuantitativo. El enfoque aplicativo se debe al diseño e implementación de una plataforma web con infraestructura tecnológica que se orienta a la seguridad y al flujo de los datos brindando compatibilidad con diferentes tecnologías. El enfoque cuantitativo se aplica mediante métricas de rendimiento y técnicas de stress al sistema valorando su calidad.

2.2.1 CRITERIOS DE INCLUSIÓN

Las aplicaciones de teleconsulta se destina a pacientes y doctores que tengan dispositivos inteligentes conectados a una red con acceso a internet. La educación en escuelas, colegios, universidades y organismos estatales juega un papel importante al momento de fomentar el uso de este tipo de tecnologías en la población [17]. Los factores a considerar son: la compatibilidad con aplicaciones de terceros que hacen posible publicar la información de un Context Broker mediante una página web pública donde cualquier usuario puede acceder de manera fácil, la integración con las diferentes tecnologías que el cliente tenga en funcionamiento mediante typeORM y el ajuste de la escalabilidad dependiendo de su infraestructura¹.

¹<https://skyone.solutions/es/beneficios-de-la-escalabilidad>

2.2.2 CRITERIOS DE EXCLUSIÓN

Las limitantes del proyecto son: la integración solo con una aplicación web, la obtención de datos en tiempo real mediante wearables u otros dispositivos que capten información de signos vitales. Además, la administración de estos sistemas requieren conocimientos avanzados sobre tecnologías, programación y redes de datos. Es necesario tomar en cuenta que algunos procedimientos necesitan realizarse estrictamente de forma presencial, e.g., intervenciones quirúrgicas. Personas limitadas por la disponibilidad de internet en la región o que no posean la capacidad de costear un dispositivo móvil [18].

2.3 MODELO DE AMBIENTE DE ESTUDIO

2.3.1 ARQUITECTURA GENERAL

El servicio de teleconsulta en la ciudad de Guayaquil se proyecta como solución a la comunicación segura de datos, los pacientes necesitan que su información personal y confidencial esté correcta. Se proporcionó una interfaz amigable al usuario para el registro de los datos y una infraestructura tecnológica para su gestión y almacenamiento correspondiente. Se puede ejecutar este sistema tanto en su máquina local como en la nube, utilizando software que no ocupa demasiados recursos en su computador o teléfono celular, desde el navegador pueden acceder a la plataforma, iniciar sesión y realizar la actividad que desean.

En la figura 2.1 se muestra la arquitectura general de esta solución donde se visualizan las tecnologías a usar de FIWARE, los puertos activos para el flujo de los datos de la aplicación de teleconsulta y su interacción entre la aplicación web, los GE de FIWARE y las bases de datos.

2.3.2 ACTORES

En [14], se realizaron encuestas sobre las teleconsultas en la Ciudad de Guayaquil, las cuales ayudaron a definir los siguientes actores para esta aplicación:

- **PACIENTE:** Es el usuario que precisa recibir algún servicio de salud a través de un médico o un agente intermediario (administrador) con la opción de agendar su cita

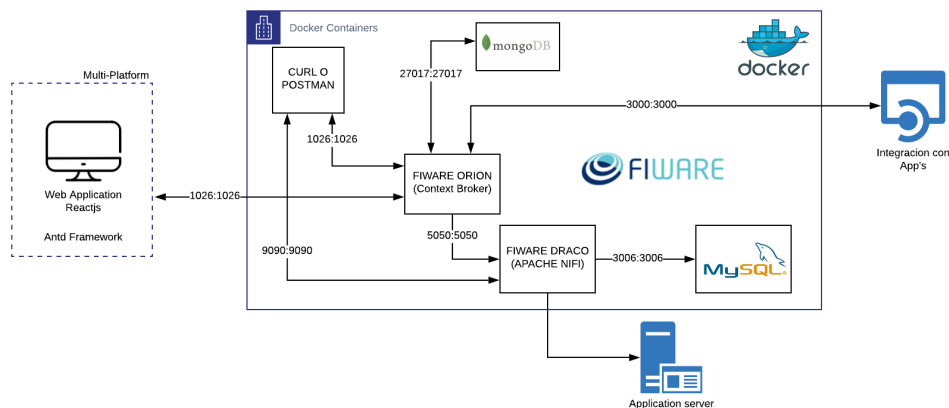


Figura 2.1: **Arquitectura General de la Aplicación de Teleconsulta**

mediante un dispositivo portátil desde la comodidad de su hogar.

- **DOCTOR:** Este usuario manipula el historial clínico y autoriza la entrega de medicamento al paciente según su propio diagnóstico, sin necesidad de exponerse a alguna situación de riesgo.
- **ADMINISTRADOR:** Este usuario con privilegios se encarga de que la información, tanto de pacientes como de doctores se mantenga en orden, actualizando el stock de medicinas con el fin de notificar al paciente mediante correo electrónico el retiro de las mismas, previa autorización del médico y maneja la frecuencia de horarios de atención del médico.



Figura 2.2: **Actores Involucrados**

2.3.3 POLÍTICAS Y SERVICIOS

El canal de comunicación junto con su plataforma y su infraestructura se esfuerza por proteger a los usuarios frente a cualquier modificación, divulgación o destrucción de los

datos que se recolecte, almacene, maneje y use en el acceso a la plataforma. Se utiliza la información para mejorar el contenido, usabilidad y experiencia de los usuarios dentro de la aplicación y sincronización de la misma². A continuación se detallan ciertas políticas que se deben tener en cuenta en la implementación:

- En cuanto a la recolección de datos se compromete a asegurar la confidencialidad/privacidad de la información mediante los servicios en línea ofrecidos.
- El cliente es responsable de la seguridad de su contraseña y nombre de usuario, a menos que compruebe ser víctima de un ataque externo o falsificación.
- La información que es enviada por el canal de comunicación se almacena en la base de datos, la misma que está encriptada por contraseña y un código de seguridad adicional para cada usuario.
- La actualización de la información se notifica vía correo electrónico al usuario, así como la creación de su cuenta en la plataforma.³
- La aplicación puede recopilar cierta información automáticamente, incluyendo, pero no limitado al tipo de dispositivo móvil que utilice, los dispositivos móviles de identificación única del dispositivo, la dirección IP de su dispositivo móvil, el sistema operativo móvil, el tipo de navegadores de Internet que utiliza, la información sobre el modo de usar la aplicación y sus cookies.

2.3.4 MÉTRICAS Y VARIABLES

Las métricas que se evaluaron para la verificación de la calidad de este canal de comunicación fueron:

2.3.4.1 PRINCIPALES

- **RENDIMIENTO:** Se tomó el tiempo promedio que requirió enviar la información desde la aplicación web a la infraestructura, con una sobrecarga de actualizaciones de stock mediante la aplicación. Debido a la dependencia de la conexión que tenga el usuario influye en la realización de alguna actividad.

²<https://www.gobiernoelectronico.gob.ec/politica-datos-personales/>

³<http://www.refanet.com/apps/politicas.html>

- **PAQUETES POR SEGUNDO:** Esta métrica proporcionó información vital del enlace de comunicación de la aplicación con la infraestructura, para una comprensión más puntual sobre las características que ofrece el modelo de comunicación.
- **PERDIDA DE PAQUETES Y ERRORES:** Las piezas que integra un paquete del envío de datos del frontend, se utilizaron para medir cuan fiable es el envío de la información entre los componentes del modelo de comunicación.
- **DISPONIBILIDAD:** Se midió la proporción del tiempo de uso de la aplicación sin fallas variando la cantidad de usuarios que interactuaban en la aplicación cierta cantidad de veces y cierta cantidad de tiempo.

2.3.4.2 SECUNDARIAS

- **RECURSOS:** La monitorización de los recursos del sistema asegura que la estabilidad del servidor de aplicaciones se mantuvo durante la teleconsulta con el paciente permitiendo ya sea al administrador o al doctor realizar la actividad pertinente.
- **SATISFACCIÓN DE USUARIO:** Esta métrica cuantifica el porcentaje de usuarios doctores y administrativos que desean mejorar o cambiar su infraestructura tecnológica mediante encuestas o entrevistas realizadas a ellos mismos, indagando sus necesidades y sus opiniones en cuanto al uso de las tecnologías para brindar una mejor atención al paciente.

2.4 PLATAFORMA DE INTERACCIÓN USUARIO

La pandemia de COVID-19 destacó la importancia y la utilidad de las aplicaciones de teleconsultas, para proporcionar una manera de poner en contacto a pacientes y profesionales de la salud cuando no es posible una consulta presencial⁴. Brindando una plataforma fácil de manipular para aquellos usuarios que en ocasiones se sienten en desventaja cuando se presenta algo muy complejo, y a la empresa le toma recursos y tiempo pues requieren capacitaciones, para aliviar un poco esta carga se diseñó una plataforma reactiva para el modelo de comunicación con funcionalidades básicas,

⁴<https://telemedico.com.ec/telemedicina-en-la-pandemia-sirve/>

diseñadas con librerías Ant Design conectando los datos ingresados de los diferentes usuarios, para luego enviarse a la infraestructura tecnológica mediante formato JSON⁵.

2.4.1 DISEÑO DE LA APLICACIÓN

ReactJS⁶ permite que la aplicación de teleconsulta pueda actualizar y renderizar los componentes adecuados cuando la información del paciente cambie. La librería de Ant design ofrece un diseño de vistas simples que proporciona compatibilidad con los navegadores modernos haciendo ligero su funcionamiento, también optimiza el uso de recursos del computador o teléfono celular, minimiza redundancias, ahorra costos de producción y facilita la creación de aplicaciones.

En [19], se implementó una aplicación de Planificación de Recursos Empresariales (ERP) basada en ReactJS para mejorar los problemas de la interfaz de usuario de las soluciones ERP basadas en Access, o en [20], ReactJS se utilizó por E-Force para el desarrollo de interfaces web, permitiendo combinar y emplear el patrón de diseño de Modelo de Vista de Controlador (MVC) y en [21], ReactJS ayudó a gestionar la información y solucionó los problemas de automatización de los procesos operativos.

Finalmente, en la implementación de esta solución se utilizó el gestor de dependencias de Javascript, yarn⁷ porque optimiza los recursos computacionales y descarga solo los paquetes necesarios llamados en el archivo package.json, almacenados en la carpeta node modules. Esta descarga se realiza una sola vez a diferencia del gestor de paquetes npm que descarga repetidas veces. Su instalación solo requiere ejecutar el comando **npm install -g yarn**.

2.4.2 SERVICIO DE VIDEOLLAMADAS

Se implementó como servicio adicional dentro de la interfaz web la realización de videollamadas entre doctores y pacientes mediante la herramienta OpenVidu la cual es gratuita que incluía funcionalidades como compartimento de pantalla, audio, video y chat. Se incorporó este servicio haciendo uso de librerías compatibles con ReactJS. OpenVidu Library React es uno de los tutoriales más simples y rápidos para agregar

⁵<https://ant.design/docs/react/introduce>

⁶<https://es.reactjs.org/>

⁷<https://openwebinars.net/blog/que-es-yarn/>

capacidades de videoconferencia a su aplicación web existente. Este tutorial usa la biblioteca `openvidu-react` ⁸.

2.4.3 FUNCIONALIDADES

Las funcionalidades están divididas en tres roles: pacientes, doctores y administradores de la aplicación, a cada uno se le asignó características únicas para mantener la información coherente y concisa.

La autenticación se empleó en el inicio de la sesión utilizando `typescript` para la gestión de datos de la plataforma web, además incluye un tipo de Mapeo relacional de objetos (`typeORM`) para la integración con cualquier base de datos del cliente o cuando se deseen migrar sus datos⁹. Se añadió una capa de seguridad con código único para cada usuario llamado `salt`, que permite asegurar y encriptar la base de datos. Los datos manejados son enviados por la aplicación web a la plataforma `FIWARE`.

Se describen algunas de las principales funcionalidades que se consideraron para el modelado de comunicación:

- **Historial clínico:** Esta característica incluye los datos del paciente y una actualización de su progreso y evolución en su salud, es editado de forma periódica por el doctor.
- **Stock de medicina:** Es utilizada por el administrador de la aplicación que actualiza el stock y envía notificaciones al paciente sobre la entrega de su medicamento.
- **Frecuencia:** En esta actividad el usuario que administra la aplicación registra los horarios de atención de los doctores, según su área de especialidad.
- **Agendar cita:** Esta funcionalidad es designada al usuario paciente, refleja el médico que lo atenderá junto con la fecha y la hora de la cita.
- **Ingresar doctores y pacientes:** Estas opciones sirven recolectar datos dentro de la aplicación, es manejada por el administrador de la aplicación manteniendo un orden.
- **Mis citas:** en esta función se visualiza las citas creadas y se ingresa a la sesión de videollamada para hacer la consulta.

⁸<https://docs.openvidu.io/en/2.16.0/tutorials/openvidu-library-react/>

⁹<https://typeorm.io/>

2.4.4 CASOS DE USO FLUJO DE COMUNICACIÓN

El ingreso a la aplicación permite identificar el tipo de usuario y desplegar el menú designado para cada necesidad. En la figura 2.3 se observa que el usuario doctor manipula el historial clínico, crea también una receta para el paciente y aparece en su perfil la videoconferencia que crea el paciente; el usuario paciente solicita su cita médica con agendar cita la cual genera una videoconferencia con el doctor que ha seleccionado y el usuario administrador manipula el stock de medicinas con actualización de medicina. Todos son usuarios y hacen uso del menú de forma particular.

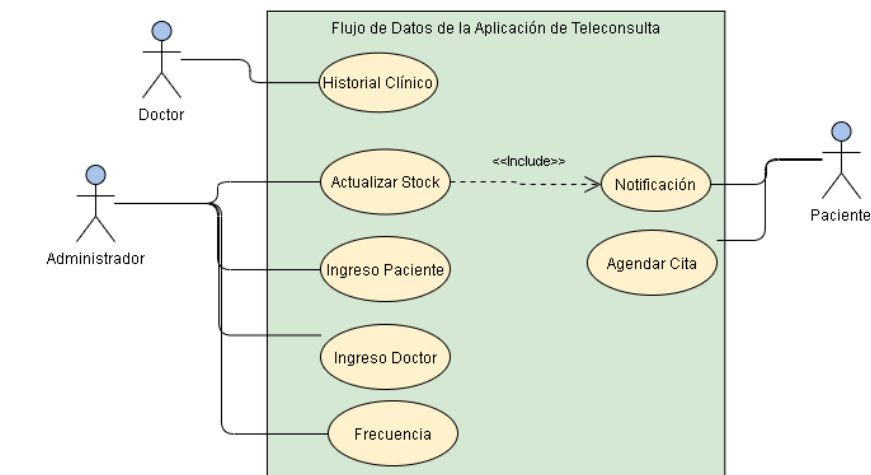


Figura 2.3: Diagrama de flujo de la comunicación

En las figuras 2.4, 2.5, 2.6 se detalla el flujo de comunicación, es decir una interacción entre los actores con sus actividades y la dependencia entre ellas al momento de interactuar en la plataforma web.

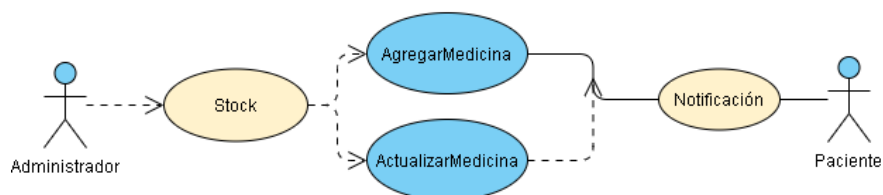


Figura 2.4: Actualización de medicinas en stock

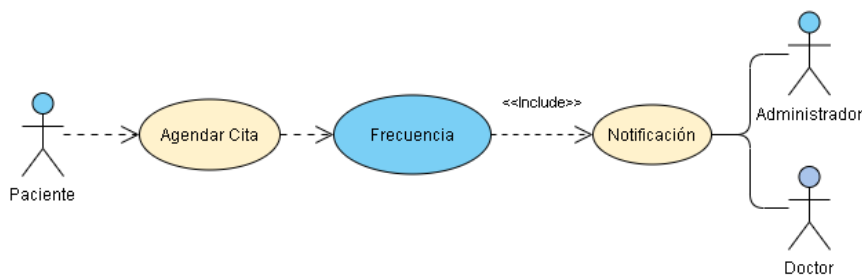


Figura 2.5: **Agendar cita**

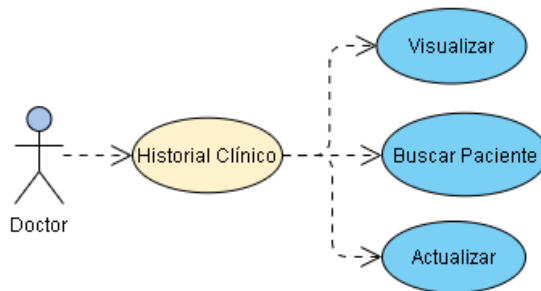


Figura 2.6: **Modificación del historial clínico**

2.5 INFRAESTRUCTURA TECNOLÓGICA

La implementación se centra en un modelo de comunicación escalable, distribuida, concurrente, tolerante a fallos y confiable; permite la integración con diferentes API's, aplicaciones móviles, aplicaciones web y sensores de toma de datos en tiempo real. La plataforma FIWARE proporcionó un diseño de backend mediante un sistema de pruebas controlado en un host local que junto a los habilitadores genéricos o componentes de FIWARE (GE) integran aplicaciones de interfaces de programación y permiten el funcionamiento de la aplicación de teleconsulta.

La plataforma FIWARE obtiene la información de contexto de sistemas ya existentes, interacciones de usuario o valores leídos por sensores, es por esto que surge la necesidad de una solución que permita integrar los servicios actuales y futuros de manera no intrusiva. También establecen y promueven directrices modelando datos, definiendo entidades que junto a sus atributos de forma sean susceptibles a ser reutilizados. Se deben tener en cuenta los modelos de datos ya existentes, evitar el uso de metadatos y utilizar solo partes del modelo de datos específicos para reducir el intercambio de información [22].

2.5.1 FIWARE-ORION (CONTEXT BROKER)

El componente principal de FIWARE es el GE FIWARE Orion¹⁰, este orquestador proporciona la administración del ciclo de vida del historial clínico y medicinas en stock mediante su object context broker (OCB) que realiza consultas y actualizaciones de datos de forma descentralizada y escalable mediante la API RESTful NGSIv2 por medio del formato JSON. Se usaron los métodos para obtener las entidades (v2/entities/) y suscripciones (v2/subscriptions) detalladas en la sección 2.6.1 y 2.6.2 respectivamente. FIWARE Orion usa el canal de comunicación publicador/subscriptor por el puerto 1026, e.g., en [22] se usó FIWARE Orion para obtener las coordenadas en tiempo real a fin de desarrollar un sistema de detección de desplazamientos para turismo inteligente. En la figura 2.7 se muestra la interacción de como los productores generan las suscripciones y como los consumidores la solicitan a través del puerto por defecto.

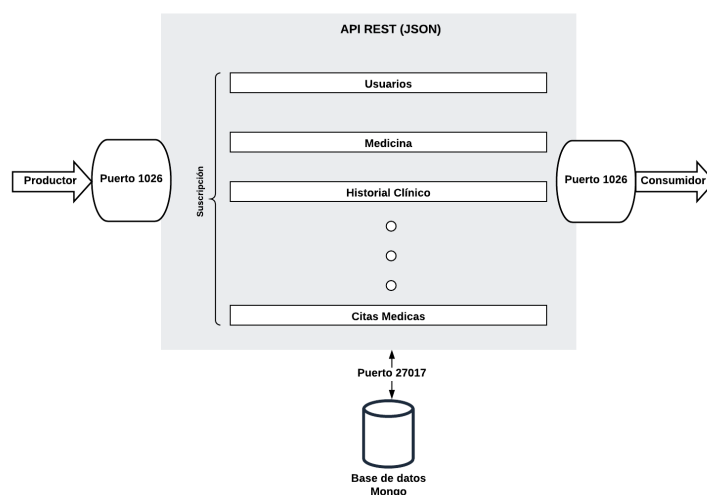


Figura 2.7: Funcionamiento de FIWARE Orion

2.5.2 FIWARE-DRACO (APACHE NIFI)

El GE FIWARE Draco¹¹ permitió procesar y distribuir la información de contexto proveniente de FIWARE Orion por el puerto 5050 con Apache Nifi, administrado mediante el socket `http://localhost:8080/nifi/`, el cual gestiona los datos usando programación de flujo. En la infraestructura se utilizó la plantilla ORIONtoMYSQL como se muestra en la

¹⁰https://FIWARE-training.readthedocs.io/es_MX/latest/ecosistemaFIWARE/ocb/

¹¹<https://github.com/ging/FIWARE-draco>

figura 2.8, los procesadores ListenHTTP y NGSIToMySQL permiten enviar, recibir, enrutar y procesar el flujo de archivos (flowfile) que son datos compuestos por atributos clave - valor entre sistemas y almacenarlo en la base de datos mySQL a través del puerto 3306, e.g., en [23] se usó FIWARE Draco para el diseño, programación y despliegue de un procesador para persistencia de datos de contexto en una base de datos distribuida.

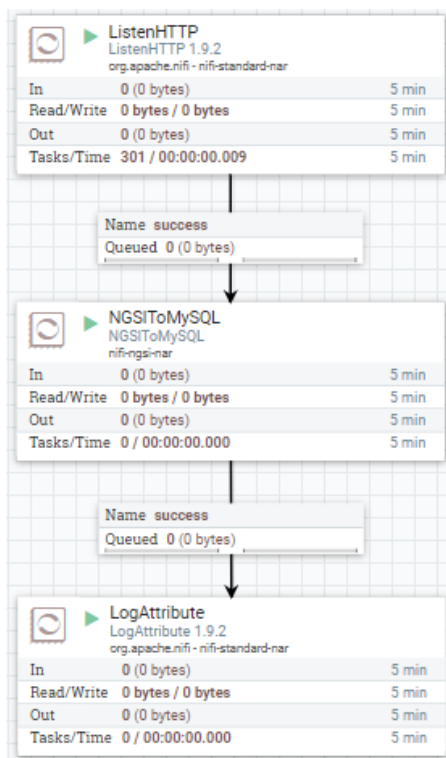


Figura 2.8: Flujo de datos Draco Apache Nifi

2.5.3 ALMACENAMIENTO DE DATOS

La infraestructura tecnológica está soportada por dos bases de datos para su persistencia; por defecto el GE FIWARE Orion integra al MongoDB¹², una base de datos no relacional escalable horizontalmente, que guarda los datos compatibles con formato JSON y tolerante a fallas, su comunicación se da a través del puerto 27017.

Por otra parte, se escogió la base de datos de FIWARE Draco integrada con MySQL, porque cuenta con varias capas de seguridad, es compatible y flexible con todas las versiones de windows y linux, consume pocos recursos y está diseñada para mantener los datos de contexto procedentes del NGSI de FIWARE Orion por el puerto 5050, los

¹²<https://university.mongodb.com/courses/catalog>

mismos que se transforman en eventos de NGSi. En tanto, como se muestra en la figura 2.9, toda la información actualizada se mapea al servidor de base de datos (medicina) MySQL¹³ a través del puerto 3006 en la tabla x002fmedicina.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| hospital |
| mysql |
| performance_schema |
| sys |
+-----+
1 row in set (0.00 sec)

mysql> show tables FROM hospital;
+-----+
| Tables_in_hospital |
+-----+
| x002fmedicina |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM hospital.x002fmedicina limit 10;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| recvTimeTs | recvTime | fiwareServicePath | entityId | entityType | attrName | attrType | attrValue | attrId |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 160724292802 | 12/06/2020 13:56:22 | medicina | urn:ngsi-ld:Medicina:37 | Medicina | stock | Integer | 100 | () |
| 160724292402 | 12/06/2020 13:56:22 | medicina | urn:ngsi-ld:Medicina:37 | Medicina | nombre | Text | DOLOREBIBION FORTE | () |
| 160724299436 | 12/06/2020 13:56:26 | medicina | urn:ngsi-ld:Medicina:38 | Medicina | stock | Integer | 200 | () |
| 160724299436 | 12/06/2020 13:56:26 | medicina | urn:ngsi-ld:Medicina:38 | Medicina | nombre | Text | LOGASTAN LOGSD | () |
| 160724306869 | 12/06/2020 13:57:45 | medicina | urn:ngsi-ld:Medicina:39 | Medicina | stock | Integer | 45 | () |
| 160724306869 | 12/06/2020 13:57:45 | medicina | urn:ngsi-ld:Medicina:39 | Medicina | nombre | Text | ALOPRINDOL 300MG | () |
| 160724307471 | 12/06/2020 13:57:57 | medicina | urn:ngsi-ld:Medicina:40 | Medicina | stock | Integer | 44 | () |
| 160724307471 | 12/06/2020 13:57:57 | medicina | urn:ngsi-ld:Medicina:40 | Medicina | nombre | Text | XEFEMIDOL | () |
| 160724308774 | 12/06/2020 13:58:07 | medicina | urn:ngsi-ld:Medicina:41 | Medicina | stock | Integer | 43 | () |
| 160724308774 | 12/06/2020 13:58:07 | medicina | urn:ngsi-ld:Medicina:41 | Medicina | nombre | Text | SALBUTAMOL | () |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Figura 2.9: Almacenamiento de datos en MySQL

2.6 CANALIZACIÓN DE DATOS

La canalización de datos empieza en el frontend, que implica realizar una actualización del stock de medicina con rol de administrador como se muestra en la figura 2.10.

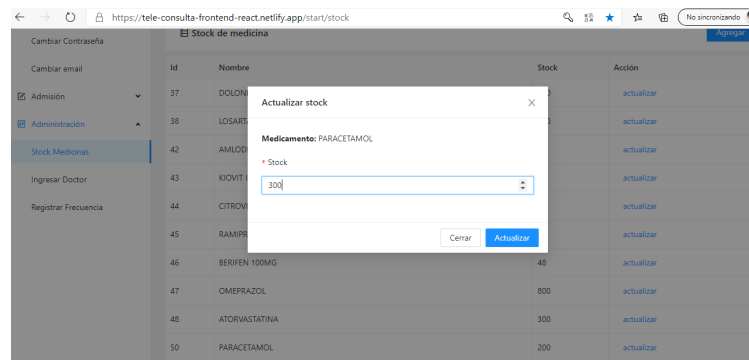


Figura 2.10: Actualización de medicina

Los datos se relacionan con la información de medicina que se envían mediante un POST a FIWARE Orion a través del puerto 1026, guardandose en una entidad como se observa en la figura 2.11. Si la actualización de datos es exitosa, se realiza una replicación a través de FIWARE Draco por el puerto 5050 a la url (v2/notify).

Los datos de medicina que provienen de FIWARE Orion son almacenados en la base de datos MySQL en el puerto 3006. Por ejemplo, en la figura 2.12 se realizó la actualización de la medicina Paracetamol.

¹³<https://dev.mysql.com/doc/>

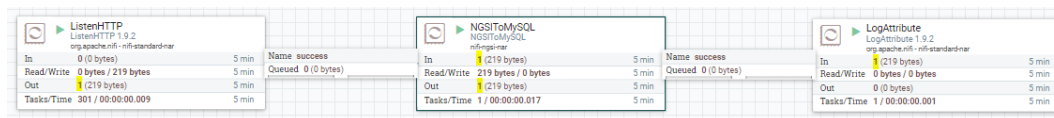


Figura 2.11: Datos a través de FIWARE Draco

1607440588780	12/08/2020 15:16:28	medicina	urn:ngsi-Id:Medicina:39	Medicina	nombre	Text	ALOPURINOL 300MG	[]
1607441703774	12/08/2020 15:35:03	medicina	urn:ngsi-Id:Medicina:40	Medicina	stock	Integer	175	[]
1607441703774	12/08/2020 15:35:03	medicina	urn:ngsi-Id:Medicina:40	Medicina	nombre	Text	ATENULOL	[]
1607444624056	12/08/2020 16:23:44	medicina	urn:ngsi-Id:Medicina:41	Medicina	stock	Integer	75	[]
1607444624056	12/08/2020 16:23:44	medicina	urn:ngsi-Id:Medicina:41	Medicina	nombre	Text	SALBUTAMOL	[]
1607447412540	12/08/2020 17:10:12	medicina	urn:ngsi-Id:Medicina:50	Medicina	stock	Integer	300	[]
1607447412540	12/08/2020 17:10:12	medicina	urn:ngsi-Id:Medicina:50	Medicina	nombre	Text	PARACETAMOL	[]

Figura 2.12: Datos en la base de datos MySql

2.6.1 ENTIDADES EN FIWARE-ORION

Una entidad es una representación de medicina. Las entidades en FIWARE Orion¹⁴ están representadas por una estructura denominada elementos de contexto (id, type), mientras que los atributos (name, type, value) pertenecen al contexto como se muestra en la figura 2.13.



Figura 2.13: Flujo de datos de la entidad

La entidad implementada de medicina cuenta con cuatro atributos:

- **Id:** Atributo único, su valor es el código de la medicina.
- **Type:** Tipo de entidad, su valor es el tipo de medicina.
- **Nombre:** Tipo texto, su valor es el nombre del medicamento.
- **Stock:** Tipo entero, su valor es la cantidad de medicina en stock.

El envío de peticiones requirió un cliente REST e incluyó la especificación del método HTTP (GET, POST, PUT, DELETE, OPTIONS, HEAD, TRACE, CONNECT). Así mismo se adicionó la URL <http://localhost:1026/v2/entities/>, se colocó una cabecera

¹⁴https://FIWARE-training.readthedocs.io/es_MX/latest/ecosistemaFIWARE/ocb/#orion-context-broker

(Content-Type: "application/json" o Accept: application/json) y finalmente se adjuntó el cuerpo con los atributos de medicina. En la creación de la entidad se usó el comando POST con la url como se observa a continuación:

Algoritmo 1 Creación de entidades (Stock de medicina)

```
POST http://localhost:1026/v2/entities
Header: Content-Type: "application/json"
{
  "id": "med001",
  "type": "Medicina",
  "nombre": {
    "type": "Text",
    "value": "Amoxicilina 60 mg"
  },
  "stock": {
    "type": "Text",
    "value": "100"
  }
}
```

Para consultar la entidad se realizó un GET a la url **http://localhost:1026/v2/entities**, mientras que el context broker respondió con la información de las medicinas en formato JSON. En cuanto a la actualización de la entidad, se realizó un PATCH a la url **http://localhost:1026/v2/entities/idmedicina/attrs/** con los atributos que se necesitaban cambiar. En tanto, para eliminar una medicina se aplicó el comando DELETE a la url **http://localhost:1026/v2/entities/idmedicina/**.

2.6.2 SUBSCRIPCIONES EN FIWARE-ORION

Las suscripciones¹⁵ permiten registrar la información del contexto solicitado. Cuando ocurre un cambio en el atributo requerido, FIWARE Orion envía una notificación sin necesidad de realizar una consulta a la suscripción, para el efecto, debe estar habilitado y configurado el servicio **http://localhost:1028/accumulate**.

Para crear la suscripción se usó el comando POST con la url terminada en **/v2/subscriptions**. Cuando el stock de medicinas, que es monitorizado por FIWARE Orion, cambia en su numero de fármacos emite una notificación, como se muestra en la siguiente configuración:

En la configuración de la suscripción los campos usados fueron:

¹⁵https://FIWARE-training.readthedocs.io/es_MX/latest/ecosistemaFIWARE/ocb/#interaccion-con-la-api-restful-ngsi-v

Algoritmo 2 Creación de suscripciones (Stock de medicina)

```
Content-Type: application/json
POST http://localhost:1026//v2/subscriptions
{
  "description": "Obtener informacion de stock medicina",
  "subject": {
    "entities": [{"id": ".*", "type": "Medicina"}],
    "condition": {"attrs": ["stock"]} },
  "notification": {
    "http": {
      "url": "http://localhost:3000/app/monitor"},
      "attrs": ["nombre", "stock"]},
    "expires": "2040-01-01T14:00:00.00Z",
    "throttling": 5
  }
}
```

- **Entities:** Entidades que hace referencia la suscripción.
- **Id:** Admite el id único de la entidad para la suscripción.
- **Type:** Admite el tipo de la entidad para la suscripción.
- **Condition:** Condición del criterio para envío de las notificaciones.
- **Notification:** Parámetros de los atributos solicitados a notificar.

La validación de la suscripción se realiza mediante un GET a la url <http://localhost:1026//v2/subscriptions> GE FIWARE Orion responde con un formato JSON.

2.6.3 EVENTOS Y NOTIFICACIONES

Los eventos se accionan a través del elemento conditions dentro de la suscripción permitiendo que la notificación se active.

Las notificaciones se gestionaron por medio de tres escenarios:

- **Creación de usuario:** El usuario solicita al administrador un perfil para la aplicación de teleconsulta, la que se notifica por correo electrónico. Su procedimiento se muestra en la figura 2.14.

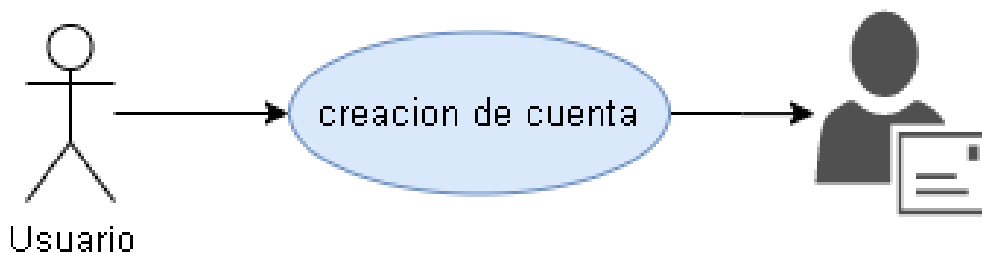


Figura 2.14: **Notificación de creación del perfil**

- **Actualización de stock:** Las actualizaciones del stock de medicinas realizadas por el administrador se notifican públicamente vía Twitter. En la figura 2.15 se muestra el procedimiento a través de FIWARE Draco usando un script.

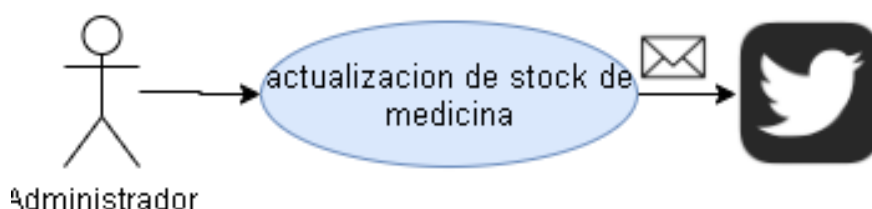


Figura 2.15: **Notificación de actualización en el stock de medicinas a vía Twitter**

- **Agendar cita:** Al momento que el usuario crea la cita con el médico especialista, se notifica por correo electrónico como se muestra en la figura 2.16.



Figura 2.16: **Notificación para agendar citas**

2.7 ALTERNATIVAS TECNOLÓGICAS

La banca digital ha alejado a las personas de los bancos y las tiendas se han alejado de los centros comerciales con el comercio electrónico, se estima que para la siguiente década la teleconsulta cambie la atención en los hospitales y clínicas por atención en casa con monitoreo mediante dispositivos móviles. Estos casos pueden ser posibles gracias a dos grandes variables:

- costos más bajos.
- aumento de la comodidad.

Se realizó un ensayo a pacientes con enfermedades agudas analizando la comparación entre la atención hospitalaria versus la atención en casa con llamadas de audio, video con médicos y monitoreo remoto. Como resultado se obtuvo menos reingresos, pruebas, consultas y costos en la atención domiciliaria. Aunque estos hallazgos apuntan al potencial de la teleconsulta requieren replicación y validación adicional por parte de los médicos [24].

Los avances tecnológicos tienen el potencial de mejorar la atención médica y de hacerla accesible a más personas¹⁶ mediante las TICs. La nueva frontera es brindar atención especializada a una escala global. Se han creado una gran diversidad de aplicaciones para ayudar a los consumidores a organizar su información médica en un lugar seguro. Estas herramientas digitales pueden ofrecer en un futuro lo siguiente:

- Revisar los signos vitales en tiempo real.
- Calcular y hacer un seguimiento de la ingesta calórica.
- Programar recordatorios para tomar los medicamentos.
- Registrar la actividad física, como el recuento de pasos diarios.
- Combinarse con la Robótica, permitiría intervenir quirúrgicamente o asistir a un cirujano con un equipo de especialistas, ambos en distintos lugares¹⁷.

Haciendo énfasis en la regulación de la prescripción médica se impulsa a que los farmacéuticos comunitarios amplíen la educación del paciente, mediante farmacias por internet las cuales brindan información de calidad y orientación adecuada [25].

¹⁶<https://www.mayoclinic.org/es-es/healthy-lifestyle/consumer-health/in-depth/telehealth/art-20044878>

¹⁷<http://scielo.sld.cu/scielo.php>

CAPÍTULO 3

3. ANÁLISIS Y RESULTADOS

En este capítulo se detalla el funcionamiento, escenarios, ambientes y resultados en términos de métricas, además se presenta la selección de diversos escenarios y ambientes de pruebas, sobre los que se evaluó el canal de comunicación entre una aplicación de teleconsulta y la infraestructura tecnológica basada en FIWARE, con la finalidad de sustentar la efectividad que aportan las tecnologías nuevas a los sistemas que se encuentran implementados hoy en día en el ámbito de la salud.

3.1 FUNCIONAMIENTO

Para la validación del funcionamiento del sistema de teleconsulta se verificó que tanto los servicios de la aplicación e infraestructura estuvieran inicializados correctamente.

Para iniciar la aplicación web se usó *Visual Studio Code* donde se ejecutaron los siguientes comandos:

- yarn: descarga y actualiza los paquetes que se necesitan o se han modificado y se registran en la carpeta node models.
- yarn start: inicia el proyecto desde la terminal de *visual studio code* y apertura una ventana en el navegador.

En la figura 3.1 se observa el funcionamiento de la aplicación desde la consola de *Visual Studio Code* cuando se ejecutan los comandos mencionados anteriormente, esta es la forma en que se inicializa un ambiente local para la realización de pruebas además se muestra en que dirección y puerto está corriendo la aplicación.

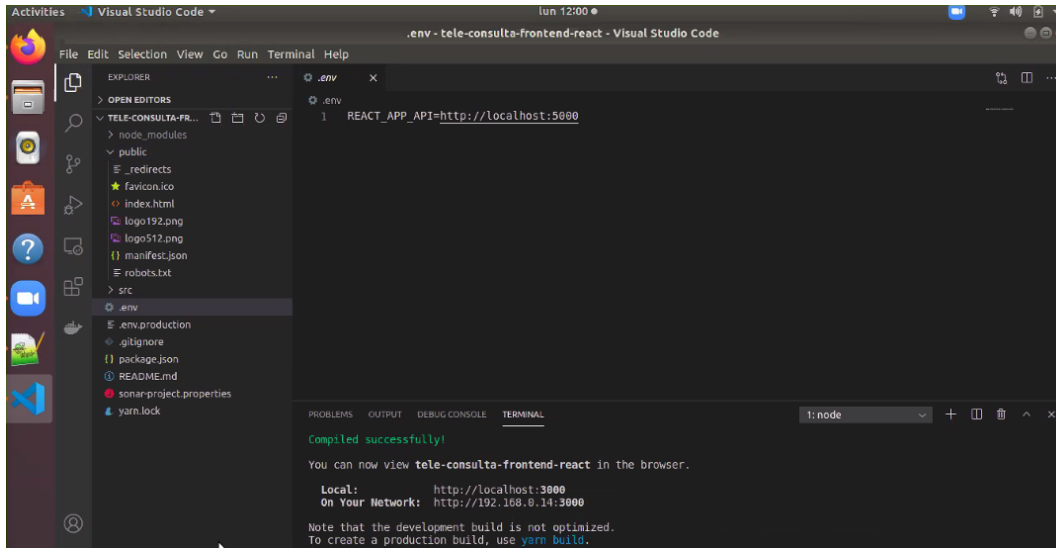


Figura 3.1: Funcionamiento de frontend

En la figura 3.2 se muestra un detalle de las funcionalidades dependiendo si el usuario es un administrador, un doctor o un paciente.



Figura 3.2: Funcionalidades de cada usuario dentro de la aplicación

En la figura 3.3 se observa el funcionamiento del backend de la aplicación de teleconsulta, mostrando el levantamiento de la base de datos y los servicios.

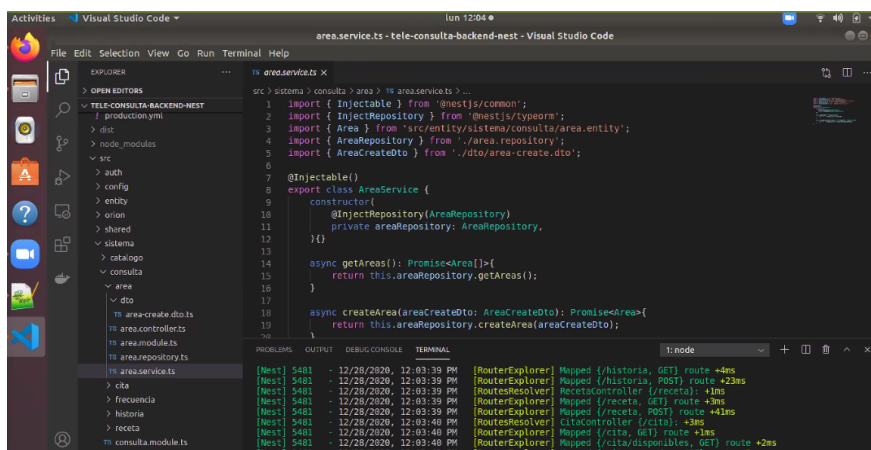


Figura 3.3: Funcionamiento de backend

Para el funcionamiento de la infraestructura se escribió el comando `sudo docker ps` y se verificó que las imágenes de FIWARE dentro de docker esten ejecutándose, con un estado saludable como se observa en la figura 3.4.

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
965e00e4535a	fiware/tutoriala-context-provider	fiware-context-provider	"/nodejs/bin/node ./..."	13 days ago	Up 2 days (healthy)	0.0.0.0:3000-3001->3000-3001/tcp
8d2b6d9c284	fiware/orion:2.4.2	fiware-orion	"/usr/bin/contextBro..."	13 days ago	Up 2 days (healthy)	0.0.0.0:1026->1026/tcp
5efb72732a7	fiware/iotagent-ul:1.14.0	fiware-iot-agent-ul	"docker/entrypoint.s..."	2 weeks ago	Up 2 days (healthy)	0.0.0.0:4041->4041/tcp, 0.0.0.0:7896->7896/tcp
034eb701c931	mongo:3.6	mongo	"docker-entrypoint.s..."	2 weeks ago	Up 2 days (healthy)	0.0.0.0:27017->27017/tcp
9fd9139cc79	qmg/fiware-draco:1.3.1	fiware-draco	"./scripts/start.sh"	2 weeks ago	Up 2 days (healthy)	8000/tcp, 8080/tcp, 0.0.0.0:9090->9090/tcp, 8080/tcp
07a08e6e12f	mysql:5.7	mysql	"docker-entrypoint.s..."	2 weeks ago	Up 2 days	0.0.0.0:3306->3306/tcp, 33060/tcp

Figura 3.4: Servicios en Docker

Se validó que el servicio del GE FIWARE Orion este arriba accediendo a la ip `http://localhost:1026/version` como se observa en la figura 3.5.

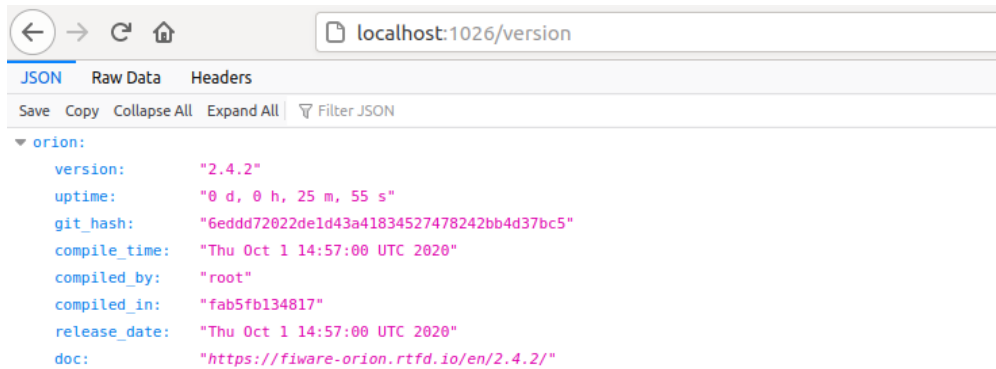


Figura 3.5: Servicio GE FIWARE Orion

El servicio del GE FIWARE Draco se comprobó accediendo a la dirección ip `http://localhost:9090/nifi`, en la figura 3.6 se muestra que los procesadores HTTPListen, NGSIToMySQL, y InvokeHTTP, no presentaron errores o advertencias.

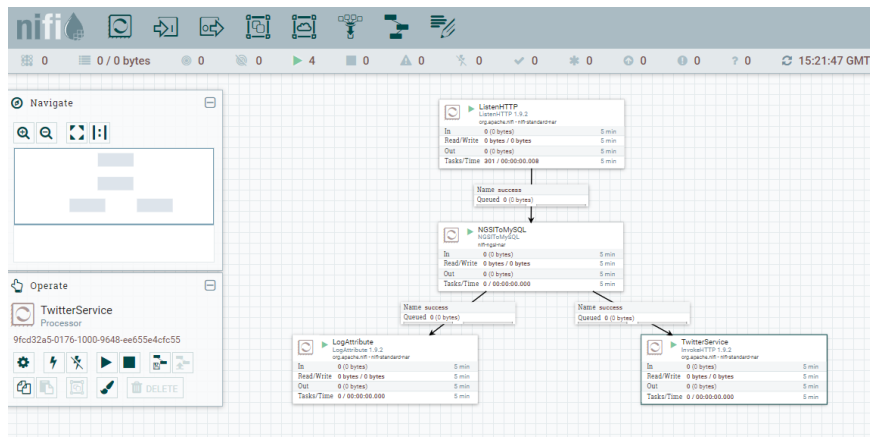


Figura 3.6: Servicio GE FIWARE Draco sin errores

Se corroboró que las notificaciones de creación de usuario, de cita y la actualización de stock de medicina estuvieran habilitadas como se muestran en las figuras 3.7, 3.8 y 3.9.

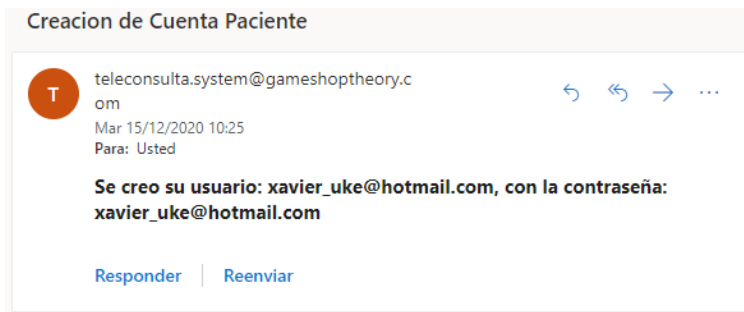


Figura 3.7: Notificación de creación de usuario

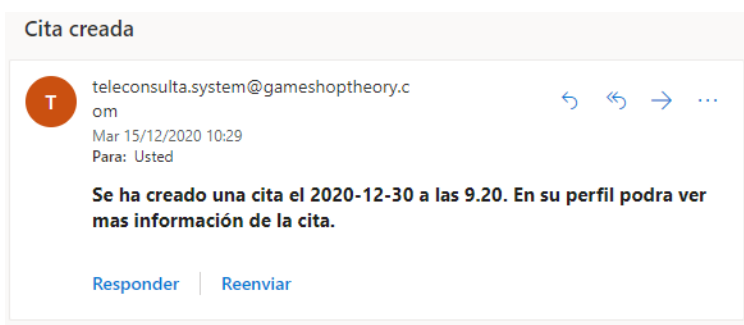


Figura 3.8: Notificación de creación de cita



Figura 3.9: Notificación de la actualización del stock

En la figura 3.10 se observa una prueba de la herramienta de videoconferencias que fueron agregadas a la aplicación de teleconsultas como un servicio, el cual se activa al realizar la agendación de una cita por parte del paciente.

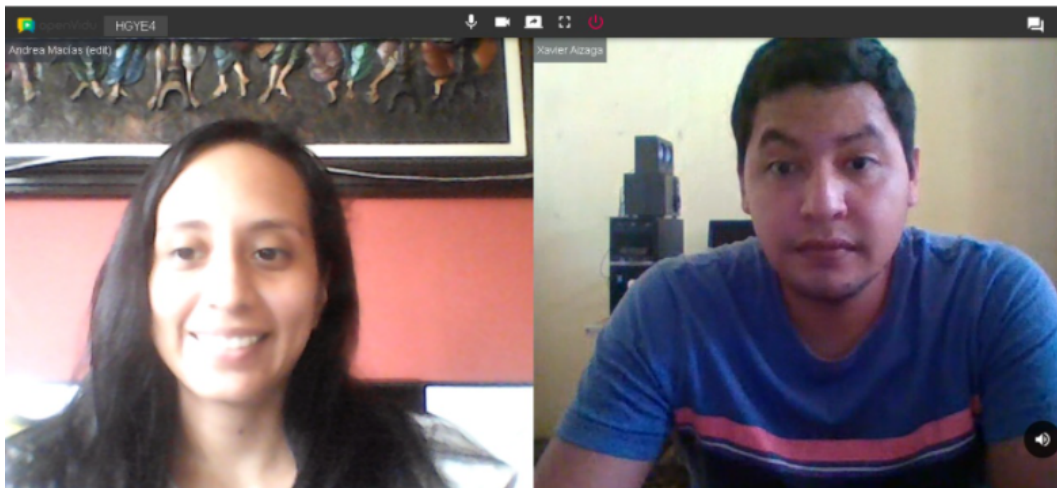


Figura 3.10: Videoconferencia de prueba entre un paciente y un doctor

3.2 ESCENARIOS Y AMBIENTES DE PRUEBAS

Los escenarios son ambientes controlados que permiten garantizar y verificar que la solución implementada y cumplir con el estándar adecuado para su uso en producción, mediante la herramienta de evaluación *Apache Jmeter*. Se eligió esta herramienta por ser open source y porque permite hacer pruebas de rendimiento y estrés con interfaz gráfica. Se midieron los indicadores clave de desempeño (KPI) para la verificación de la eficacia del sistema en base a las necesidades del usuario final.

- **Escenario 1:** Se evaluó el uso del sistema con 500 peticiones por minuto, entre citas agendadas por el paciente (25) multiplicado por las consultas atendidas por doctor (20) en sesenta segundos.

Tabla 3.1: Escenario de prueba frontend

Numero de citas	25
Numero de doctores	20
Tiempo	60 segundos

- **Escenario 2:** Se generó 1250 actualizaciones en el stock de medicinas, verificando la funcionalidad de las entidades y suscripciones hechas para el canal de comunicación entre la aplicación y la infraestructura.

Los ambientes que se consideraron para evaluar esta propuesta cuentan con las siguientes características de hardware y software:

- **Ambiente Local:** Este ambiente se usó como servidor donde se instalaron todos los servicios y requerimientos para el correcto funcionamiento del frontend y backend, como se muestra en la tabla 3.2:

Tabla 3.2: **Requisitos de Hardware y software local**

Hardware	Software	
	FrontEnd	BackEnd
Ubuntu 18.04.5 LTS	NodeJS LTS x64	FIWARE Draco 1.3.1
AMD A9-9425 radeon r5	Npm 6.14.9	FIWARE Orion 2.4.2
Ram 12 Gb	Yarn 1.22.10	Docker 19.03.13
SSD 100 Gb	And Design Librerías	MySql 5.7
		Mongo 3.6

- **Ambiente en la Nube:** En este ambiente se usaron dos servidores con las mismas características, el primero para almacenar el frontend mientras que el segundo para almacenar el backend ambos con similares requisitos de hardware como se muestra en la tabla 3.3:

Tabla 3.3: **Requisitos de Hardware y software nube**

Hardware	Software	
	FrontEnd	BackEnd
Debian GNU/Linux 10	NodeJS LTS x64	FIWARE Draco 1.3.1
Linux 4.19.0-12-cloud-amd64	Npm 6.14.9	FIWARE Orion 2.4.2
Ram 8 Gb	Yarn 1.22.10	Docker 19.03.13
SSD 30 Gb	And Design Librerías	MySql 5.7
		Mongo 3.6

3.3 RESULTADOS EN TÉRMINOS DE MÉTRICAS

3.3.1 PARÁMETROS DEL SISTEMA

En toda implementación se necesita demostrar su efectividad y validar si cumple los requerimientos funcionales. Para ello se simuló una carga de usuarios que se conectaban a la aplicación ejecutandose de forma independiente, *Apache JMeter* permite utilizar un script para realizar una pruebas con diferentes niveles de carga.¹

Las simulaciones dieron como resultados los siguientes parámetros descritos en las siguientes tablas:

En la tabla 3.4 muestra los resultados obtenidos de manera general para el escenario 1, se realizó pruebas a nivel del frontend sobrecargandolo con peticiones GET para analizar su comportamiento y tiempos de respuesta para el ambiente local y nube.

Tabla 3.4: Tabla resultado de pruebas escenario 2

Solicitudes	Ejecuciones	Tiempos de respuesta (ms)							Rendimiento	Red (KB / seg)	
Etiqueta	#Muestras	Promedio	Min	Max	Mediana	90° PCT	95° PCT	99° PCT	Transacciones / s	Recibido	Expedido
Total	2000	513,45	2	2083	610.50	1002,00	1069,00	1296,94	32,86	80,26	5,98
Local_Petición HTTP	500	6,71	2	64	5,00	12,00	15,00	27,99	8,35	16,92	0,99
Nube_Petición HTTP	500	1024.11	919	2083	981,00	1135.70	1228.80	1962.71	8.21	31,81	2,50
Nube_Petición HTTP-0	500	342,95	306	1308	325,00	388,00	431.80	543,87	8.30	3,51	1,26
Nube_Petición HTTP-1	500	680.05	608	1754	647,00	761,00	825,90	1564.26	8.28	28,56	1,26

En la tabla 3.5 muestra los resultados obtenidos del escenario 2, se saturó de solicitudes POST poniendo a prueba la respuesta del backend y su infraestructura tanto para el ambiente local y nube.

Tabla 3.5: Tabla resultado de pruebas escenario 2

Solicitud	Etiqueta	Tiempo de respuesta (ms)							Throughput	Red (KB/sec)	
Detalle	#Muestras	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transacciones/s	Recibido	Enviado
Total	2500	3428.60	174	8671	652.50	7807.90	8013.00	8330.00	39.61	15.55	15.22
Local_HTTP Request	1250	6625.97	320	8671	7272.50	8013.00	8197.00	8437.80	19.81	7.79	7.64
Nube_HTTP Request	1250	231.22	174	2899	197.00	230.00	454.95	633.47	20.75	8.13	7.94

- **Prueba de Rendimiento:** indica el tiempo de respuesta de nuestro servidor de teleconsulta contra las solicitudes de los usuarios y junto a la información de los recursos de hardware evaluar que tan rápido es nuestro servidor web.

¹<https://www.adictosaltrabajo.com/2010/01/14/jmeter-usuarios/>

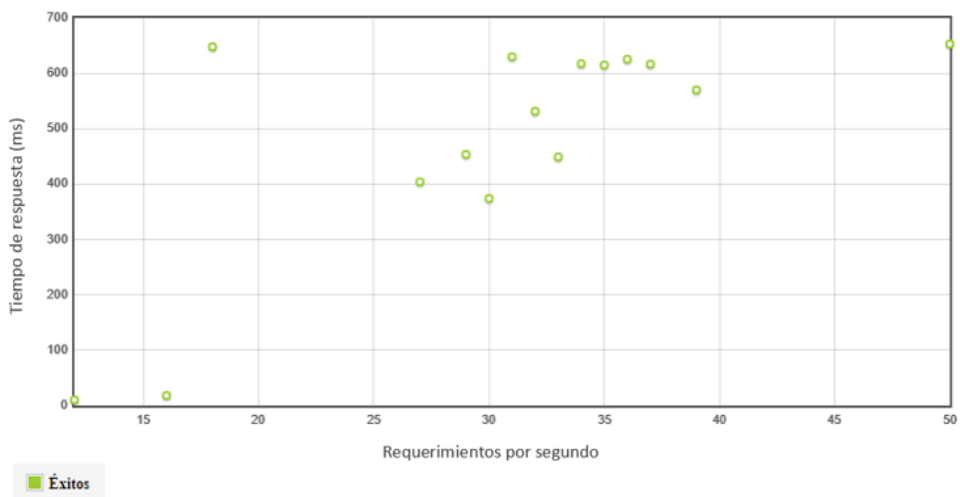


Figura 3.11: **Gráfico tiempo de respuesta vs solicitud**

Como muestra la figura 3.11, las solicitudes realizadas por los usuarios fueron exitosas y estuvieron en un rango de 27 a 39 solicitudes por segundo, con un tiempo de respuesta del servidor web de 374 a 630 milisegundos. Esto nos muestra que el ambiente de nube tiene un tiempo de respuesta de menor igual a 200 ms, mientras que el ambiente local posee tiempos de respuesta mayores a 200 ms y menores a 500 ms.

Los códigos de respuesta indican el estado del servidor de teleconsulta a cada segundo durante la prueba de rendimiento.

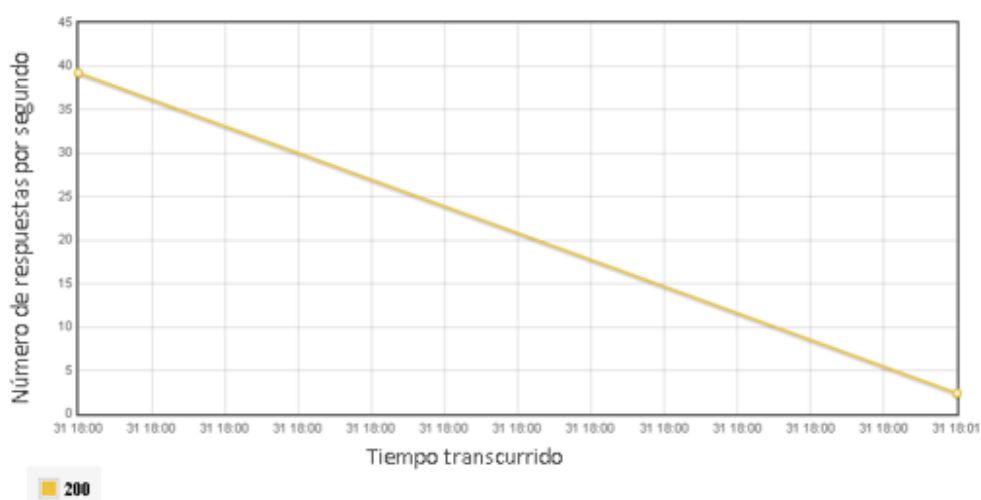


Figura 3.12: **Gráfico respuesta HTTP por segundo**

En la figura 3.12 se muestra que el código de respuesta es 200 e indica que la petición del usuario fue recibida, aceptada y procesada con éxito, este resultado se

mantuvo tanto para el ambiente local y nube.

- **Prueba de Stress:** Esta métrica muestra como se comportó la aplicación durante el aumento de usuarios en un tiempo determinado, para este caso el tiempo fué de 1 minuto.

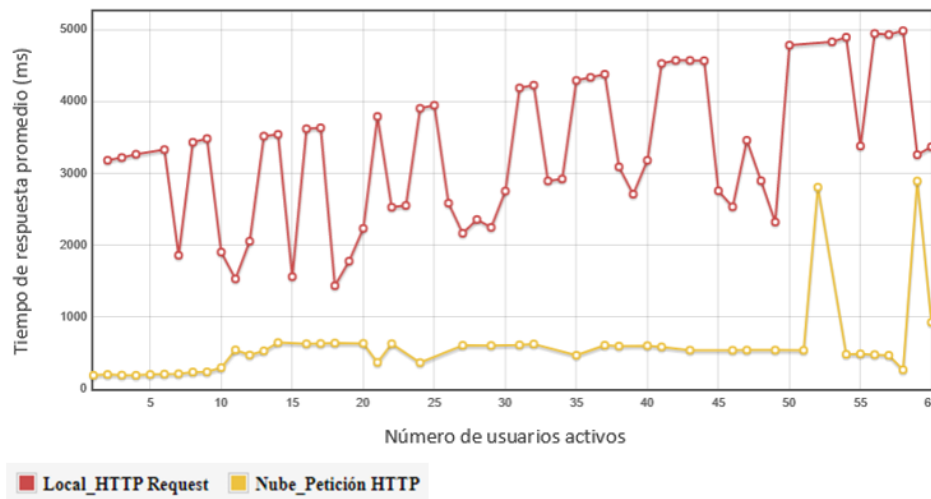


Figura 3.13: **Gráfica de usuarios vs. tiempo de respuesta**

La figura 3.13 proporciona la información del número de peticiones HTTP enviadas al servidor. Se observa que con 60 usuarios para el ambiente en nube atiende a un tiempo de 927.60 ms, mientras que el ambiente local su tiempo de respuesta es 3370 ms las peticiones. Indicándonos que el ambiente de nube es más tolerante a una carga masiva de usuarios.

3.3.2 RECURSOS DEL SISTEMA

A continuación se realiza un análisis de los recursos del sistema cuando la aplicación se encuentra operando en los diferentes escenarios y ambientes descritos anteriormente. Para evaluar así su funcionamiento.

- CPU: Permite medir el desempeño de la aplicación y su uso el cual no debe superar el 70 por ciento, debido a que afectaría su capacidad de atención del servidor web por usuario.
- Memoria Ram: proporciona el porcentaje de uso de memoria, si el consumo es más del 70 por ciento, indica que el servidor esta utilizando una alta cantidad de recursos, lo que afectaría los tiempos de respuesta o peticiones realizadas.

- Disco duro proporciona información de cuanto espacio en disco esta disponible al momento de almacenar el stock de medicina en las bases de datos.

3.3.2.1 AMBIENTE EN LA NUBE

En la figura 3.14 se observa que el porcentaje usado en el CPU por el escenario 1 es de 16.15%, mientras que en el escenario 2 el porcentaje de uso es de 26.09%, esto demuestra que el servidor en la nube puede soportar mas carga de usuarios.

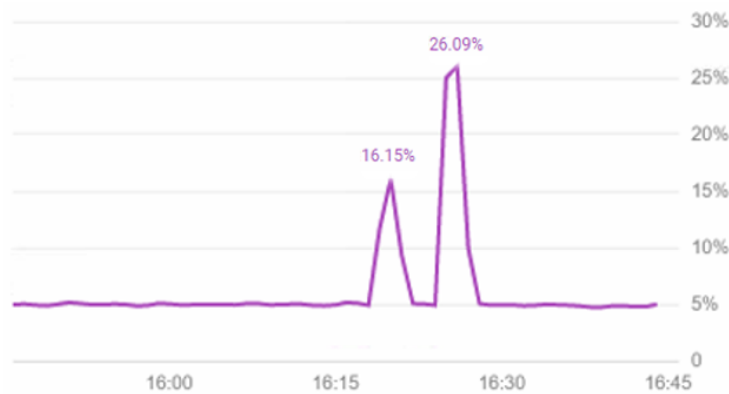


Figura 3.14: **CPU utilizado por escenario de prueba (%CPU Vs. Tiempo)**

En la figura 3.15 se observa que de 30GB disponibles solo se uso 7.255GB, esto muestra que el porcentaje de uso de memoria RAM promedio para los 2 escenarios fue de 24,18%, esto nos permite mas atención de peticiones.



Figura 3.15: **Memoria RAM utilizado por escenario de prueba (GB vs. Tiempo)**

En la figura 3.16 se observa cuanto espacio fue utilizado durante la ejecución de los escenarios 1 y 2, representados por los círculos azules y amarillos respectivamente.

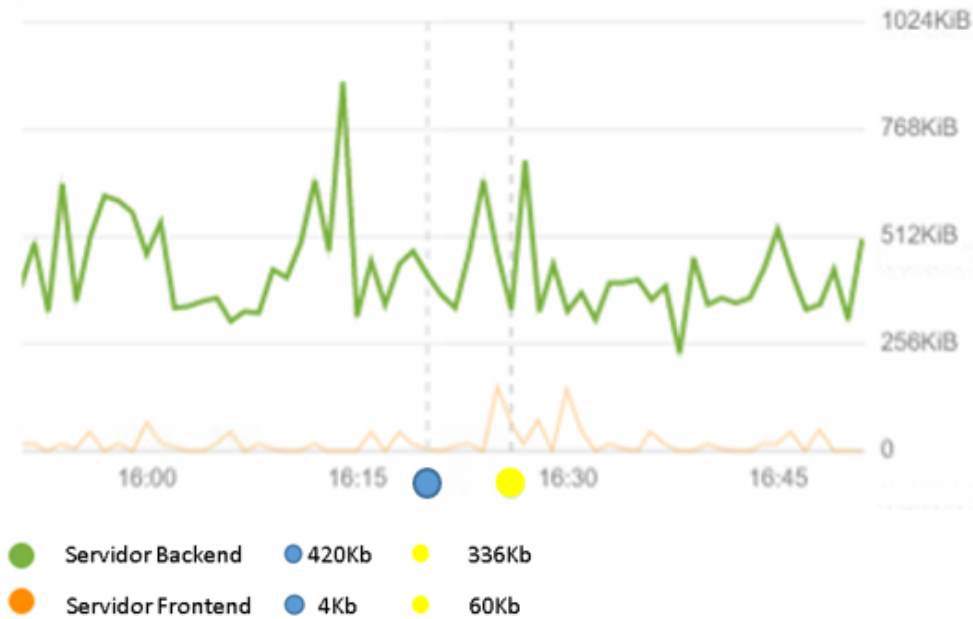


Figura 3.16: **Espacio de disco duro ambiente nube (KB vs Tiempo)**

3.3.2.2 AMBIENTE LOCAL

En general las aplicaciones necesitan ser instaladas ya sea en una computadora de escritorio o en un teléfono celular, esta aplicación puede tranquilamente ser accedida desde un navegador sin si quiera instalar ningún software. Debido a la demostración realizada para comparar su funcionamiento en los diferentes ambientes se necesitó enfocar todos los recursos en un solo servidor como se observa en las figuras 3.17, 3.18 la cantidad de recursos utilizados son diferentes para cada escenario que se definieron para medir la efectividad de la solución en un ambiente local.

- CPU y memoria RAM: El porcentaje de uso en las pruebas muestra que el CPU tuvo un 46% de consumo en sus dos hilos, mientras que la memoria RAM tuvo un consumo de 34,10%, lo cual indica su buen desempeño en ambientes con bajos recursos.

```

1 [|||||] 43.4% Tasks: 190, 912 thr: 1 running
2 [|||||] 46.7% Load average: 0.78 0.39 0.44
Mem [|||||] 3.82G/11.2G Uptime: 00:34:42
Swp [|||||] 0K/5.59G

PID USER PR  NI  VIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
5205 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:55.30 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5210 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:03.20 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5208 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:02.96 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5209 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:03.23 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5207 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:03.05 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5216 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:00.58 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5213 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:00.58 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5214 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:00.58 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5215 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:00.57 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5206 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:00.00 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front
5211 xavier 20  0  11.3G 445M 36000 S  0.0  3.9  0:00.00 /usr/bin/node /home/xavier/Documents/tele-consulta-frontend/tele-consulta-front

```

Figura 3.17: CPU y memoria RAM ambiente local

- Disco Duro: Después de la implementación y pruebas en los 2 escenarios, el tamaño de 100 GB de disco duro quedó lleno en un 80% en el ambiente local, esto demuestra el bajo consumo de recursos debido a que el frontend y el backend se encuentran implementados en el mismo servidor local.

```

xavier@linux001: ~
File Edit View Search Terminal Help
tmpfs          1171860      48  1171812    1% /run/user/1000
xavier@linux001:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            5,6G     0  5,6G   0% /dev
tmpfs           1,2G   2,2M  1,2G   1% /run
/dev/sda6       19G    14G  3,7G  80% /
tmpfs           5,6G   50M   5,6G   1% /dev/shm
tmpfs           5,0M   4,0K  5,0M   1% /run/lock
tmpfs           5,6G     0  5,6G   0% /sys/fs/cgroup
/dev/loop2      384K   384K     0 100% /snap/gnome-characters/570
/dev/loop0      98M    98M     0 100% /snap/core/10577
/dev/loop3      98M    98M     0 100% /snap/core/10444
/dev/loop4      1,0M   1,0M     0 100% /snap/gnome-logs/61
/dev/loop5      3,8M   3,8M     0 100% /snap/gnome-system-monitor/100
/dev/loop7      150M   150M     0 100% /snap/gnome-3-28-1804/67
/dev/loop6      163M   163M     0 100% /snap/gnome-3-28-1804/145
/dev/loop1      4,2M   4,2M     0 100% /snap/gnome-calculator/406
/dev/loop9      43M    43M     0 100% /snap/gtk-common-themes/1313
/dev/loop8      65M    65M     0 100% /snap/gtk-common-themes/1514
/dev/loop10     56M    56M     0 100% /snap/core18/1944
/dev/loop11     216M   216M     0 100% /snap/wine-platform-5-stable/13
/dev/loop13     1,0M   1,0M     0 100% /snap/gnome-logs/100
/dev/loop14     4,3M   4,3M     0 100% /snap/notepad-plus-plus/252
/dev/loop12     94M    94M     0 100% /snap/insomnia/118

```

Figura 3.18: Disco duro ambiente local

3.4 RESOLUCIÓN DEL PROBLEMA

Esta propuesta se planteó con el fin de que diera una ayuda a los problemas reflejados a raíz de la pandemia que atravesó el país, mejorando los sistemas de atención médica ya establecidos, diseñando un canal de comunicación como parte de la implementación de una aplicación web, fácil de manipular con funcionalidades básicas como la creación

de usuarios, citas, videoconferencias, ingreso de pacientes, creación de recetas, actualización de stock con notificaciones publicas para mantener informado al usuario del abastecimiento de las medicinas en los hospitales de Guayaquil.

La realización de una encuesta permitió medir que tan viable y aceptada es la solución propuesta para las empresas privadas. La encuesta indica que de 11 empresas privadas el 100 por ciento quiere tener una aplicación de teleconsulta que tenga un sistema de datos que permita integrar todas sus aplicaciones del hospital.

3.5 COSTOS Y ESCALABILIDAD

Para estimar los costos de la aplicación de teleconsulta se tomaron en cuenta criterios de un proyecto de consultoría tales como tiempo de duración del proyecto, infraestructura, costos del equipo de trabajo y derivados, estos costos realizados solo contemplan la implementación en sitio de la aplicación para un hospital [26].

En la figura 3.19 se muestra un diagrama de gantt asociado a la planificación del trabajo, se observa la estimación de un tiempo para su realización.

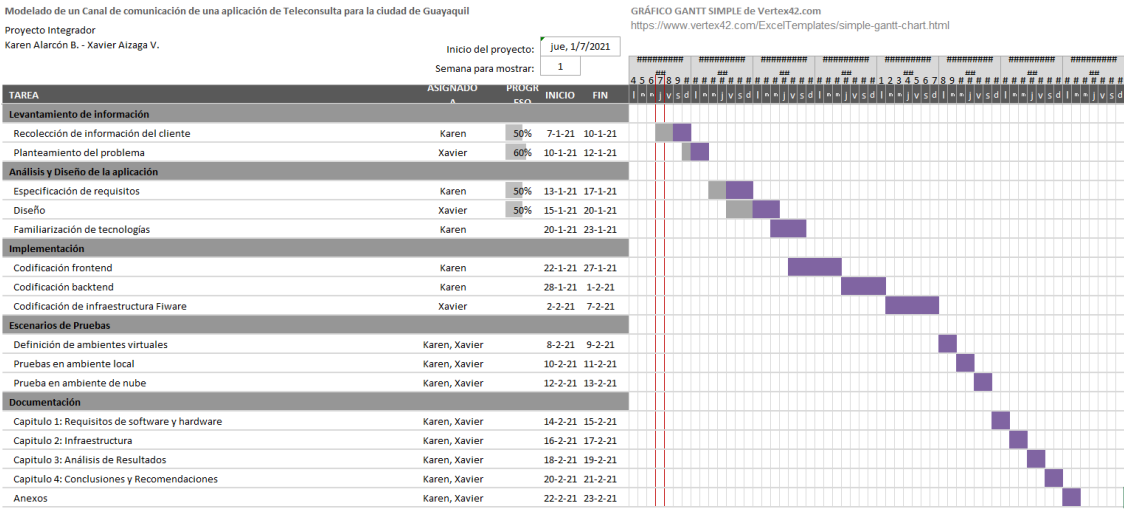


Figura 3.19: Diagrama de Gantt

En la tabla 3.6 se resumen las actividades para la realización del proyecto.

Para tener un valor aproximado del proyecto se detalla en la tabla 3.7 el salario del empleado por cargos.

Tabla 3.6: **Tabla de actividades del proyecto**

Duracion del Proyecto	
Actividades	Tiempo
Levantamiento de informacion	15 dias
Analisis y diseño de la aplicacion	27 dias
Implementacion	60 dias
Pruebas en ambientes virtuales	15 dias
Pruebas en ambiente de produccion	15 dias
Documentacion	15 dias

Tabla 3.7: **Tabla de costos del proyecto**

Salarios Equipo de trabajo			
Cargo	Sueldo total	Sueldo mensual	Costo (hora/empleador)
Jefe de proyectos	\$ 5400.00	\$ 1800.00	\$ 11.25
Consultor/Analista	\$ 3000.00	\$ 1000.00	\$ 6.25
Programador Senior	\$ 4500.00	\$ 1500.00	\$ 9.38
Programador Junior	\$ 2700.00	\$ 900.00	\$ 5.63

Realizando la suma del sueldo total por todo el proyecto, la cifra asciende a **\$15600.00** dolares americanos sin incluir iva.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- FIWARE ofrece una arquitectura de fácil integración con otras tecnologías debido a su diseño, el cual da un alcance más significativo a esta solución.
- El modelado de un canal de comunicación utilizó herramientas open source; ReactJS, Ant Design y FIWARE un formato JSON para enviar, recibir, actualizar y eliminar la información.
- Los costos se reducen debido a que es tecnología open source y no se compra un software externo.
- El canal de comunicación propuesto tiene diferentes beneficios debido a que incluye conceptos tales como concurrencia, escalabilidad, centralización, comunicación de tipo publicación/subscripción, resiliencia y alto rendimiento de las aplicaciones.
- El funcionamiento de este canal de comunicación propuesto fue exitoso en base a los resultados obtenidos mediante las pruebas de stress y rendimiento mostrados en la sección 3.3, todos los datos fueron recibidos, aceptados y procesados en los dos escenarios expuestos, adicional de las pruebas realizadas se puede observar que para el ambiente local las solicitudes enviadas al servidor web se atendieron pero con mucha latencia a diferencia del ambiente en nube que gestionó todas las solicitudes en menor tiempo debido a que sus servicios se encontraban distribuidos y no en un mismo servidor.
- Por su poco consumo de recursos tanto de memoria y disco, por su facilidad de implementación en un ambiente local o nube, se ajusta a pequeñas, medianas y

grandes empresas.

- Gracias al análisis de los resultados se puede decir que la propuesta es aceptada en un 90.9 por ciento por las empresas para mejorar la administración de su información, considerando servicios en la nube como los que ofrece la plataforma FIWARE.
- Las funcionalidades implementadas en esta solución fueron exitosas a través de una buena gestión y validación de las mismas.

4.2 Recomendaciones

- Durante la implementación de esta solución se necesitaron diversos requerimientos para establecer los ambientes pertinentes de cada escenario de evaluación, detallados en las tablas 3.2 y 3.3.
- Seguir las guías proporcionadas por los mismos creadores de las tecnologías para familiarizarse con las herramientas de diseño como ReactJS, un framework basado en javascript y las tecnologías utilizadas para la distribución entre ellas, FIWARE y Docker, disponibles en internet.
- A lo largo de esta investigación se realizó un análisis de varios sistemas de teleconsultas y proyectos que utilizaban las mismas herramientas de diseño y codificación, las cuales ayudaron a implementar las funcionalidades de la aplicación.
- Debido a que esta solución se encuentra diseñada en base a un sistema de gestión hospitalaria, la arquitectura diseñada se integra al mismo sistema y los niveles de privilegios para cada usuario se designaron según un sistema de gestión real de un hospital.
- Aunque el sistema se puede implementar en un ambiente local se recomienda su implementación en nube por su diseño distribuido y por su rápida escalabilidad al momento de falta de recursos como disco, memoria.
- Realizar investigaciones futuras de la integración de dispositivos IoT a la plataforma de teleconsulta, lo cual permitirá hacer diagnósticos más precisos.

- Analizar el ajuste de la herramienta a otros escenarios médicos debido a que el enfoque fué a la interacción de un paciente en un hospital público.
- Se debe analizar costos con una implementación en la nube y el precio individual a los usuarios finales.

4.3 Líneas Futuras

- La implementación de la plataforma FIWARE deja una guía establecida con el fin de extender las funcionalidades en futuros aplicativos, donde se tomen en cuenta la compatibilidad NSGv2.
- La plataforma FIWARE tiene muchos módulos para realizar un aplicación más completa si se desea e.g. implementar un sistema de detección de signos vitales que sean almacenados dentro de la misma aplicación sin necesidad de adquirir algún servidor extra para este servicio, FIWARE orion permitiría la gestión eon de esos datos de modo que podrían inclusive generar un diagnóstico como plantilla y enviar al correo del paciente esa información.
- Por su característica de orquestador de GE FIWARE Orion nos permite integrar otras apps, webs, wearables e IoT mediante el puerto 3000 lo cual se puede profundizar en una segunda investigación, esto permitiría complementar la información de los pacientes tal como ubicación, ritmo cardíaco, presión o temperatura.
- Para priorizar la seguridad se propone introducir GE Keyrock, un gestor de identidad que permite añadir OAuth2 como mecanismo de autenticación y autorización. Esto se considera un factor imprescindible para la puesta en producción del sistema de forma distribuida y segura.

BIBLIOGRAFÍA

- [1] J. DeMaio, L. Schwartz, P. Cooley, and A. Tice, “The application of telemedicine technology to a directly observed therapy program for tuberculosis: a pilot project,” *Clinical infectious diseases*, vol. 33, no. 12, pp. 2082–2084, 2001.
- [2] E. Ayala and S. Gonzales Sánchez, “Tecnologías de la información y la comunicación,” 2015.
- [3] A. Dovale and J. Morales, “Blockchain y el sector salud,” *I+ S: Revista de la Sociedad Española de Informática y Salud*, no. 128, pp. 11–14, 2018.
- [4] M. Peñate Garrido, “Aplicación móvil y web para la monitorización de datos recogidos mediante la pulsera inteligente miband 2 usando ble y la plataforma fi-ware,” 2017.
- [5] I. Basagoitio, “Telemedicina: la revolución lenta,” *URL: <http://gm.upv.es/tele/telem-revlenta.pdf>*, 2003.
- [6] H. Nieto-Chaupis, “Are teleconsults enough efficient to tackle the progress of type-2 diabetes and depress in adult patients?,” in *2018 IEEE 1st Colombian Conference on Applications in Computational Intelligence (CoCACI)*, pp. 1–4, IEEE, 2018.
- [7] D. M. Bolaños, “Telemedicina en el ecuador: un mundo de desafíos y oportunidades,” *LA GRANJA. Revista de Ciencias de la Vida*, vol. 12, no. 2, pp. 32–35, 2010.
- [8] J. M. González Aguilar, “Estudio para la aplicación de la telemedicina en consultas y diagnósticos a distancia en el hospital teodoro maldonado carbo de guayaquil.,” 2015.
- [9] F. A. Romero Marín, “Diseño de una red interna de telemedicina para monitoreo de equipos de cardiología del hospital general babahoyo (iess).,” 2018.
- [10] R. B. Narváez, D. M. Villacís, T. M. Chalen, and W. Velásquez, “Heart rhythm monitoring system and iot device for people with heart problems,” in *2017*

International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–5, IEEE, 2017.

- [11] H. Calderón-Gómez, L. Mendoza-Pittí, M. Vargas-Lombardo, J. M. Gómez-Pulido, J. L. Castillo-Sequera, J. Sanz-Moreno, and G. Sención, “Telemonitoring system for infectious disease prediction in elderly people based on a novel microservice architecture,” *IEEE Access*, vol. 8, pp. 118340–118354, 2020.
- [12] D. Mendes, D. Jorge, G. Pires, R. Panda, R. António, P. Dias, and L. Oliveira, “Vitasenior-mt: A distributed and scalable cloud-based telehealth solution,” in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 767–772, IEEE, 2019.
- [13] J. M. Alvarez Q., J. A. Sanabria O., and J. I. Garcia M., “Microservices-based architecture for fault diagnosis in tele-rehabilitation equipment operated via internet,” in *2019 IEEE Latin American Test Symposium (LATS)*, pp. 1–6, 2019.
- [14] B. Perez and A. Soledispa, “Survey de telemedicina en ecuador: Aplicaciones de tele-consulta en servicios clínicos públicos y privados de la ciudad de guayaquil,”
- [15] C. H. Salvador, “Modelo de historia clínica electrónica para teleconsulta médica,” *An Electronic Health Record Model for Medical Teleconsultation*. *Doctoral Thesis*, 2004.
- [16] L. PAZ-BOLIVIA, “Seguro social universitario la paz : Plan de contingencia covid-19,” 2020.
- [17] A. Garcete and M. Torres, “El uso del teléfono celular y su significación en estudiantes universitarios,” *Revista ScientiAmericana*, vol. 4, no. 2, 2017.
- [18] M. H. Rosner and S. Khan, “Health economic implications of remote patient management,” in *Remote Patient Management in Peritoneal Dialysis*, vol. 197, pp. 133–142, Karger Publishers, 2019.
- [19] J. Koppala, “Erp solution with reactjs,” 2018.
- [20] P. B. Marin, J. A. C. Gómez, and O. C. Simón, “Desarrollo de un frontend en reactjs,” 2018.
- [21] J. M. Ceron Galindo *et al.*, “React js: la nueva tendencia en aplicaciones web, enfocadas en el control dinámico de datos,” 2019.

- [22] D. Peinado Sempere, “Aplicación de fiware en un destino turístico inteligente,” 2019.
- [23] G. Ortega and B. André, “Diseño, programación y despliegue de un procesador para persistencia de datos de contexto en una base de datos distribuida utilizando fiwaredraco y cassandradb,” 2019.
- [24] E. R. Dorsey and E. J. Topol, “Telemedicine 2020 and the next decade,” *The Lancet*, vol. 395, no. 10227, p. 859, 2020.
- [25] A. M. Luque Clavijo, G. Sepulveda Carrillo, and E. Cano Urrego, “Vigilancia tecnológica para el sena: propuesta de líneas de investigación aplicada en salud,” 2011.
- [26] N. Pemán Baena, “Desarrollo de un portal web para la gestión de información de investigación,” Master’s thesis, 2009.

APÉNDICES

Apéndice A: Diagrama para Integración de elementos telemáticos

Con el fin de alimentar y asegurar la información el historial clínico de los pacientes con sus signos vitales, diagnostico y recetas se puede hacer una integración con FIWARE Sygnus para hacer streaming de data y poder guardar los datos en la aplicación de teleconsulta.

La figura 1 muestra la integración de diferentes dispositivos telemáticos capaces de medir oxigenación de sangre, presión, niveles de estrés, temperatura, etc.

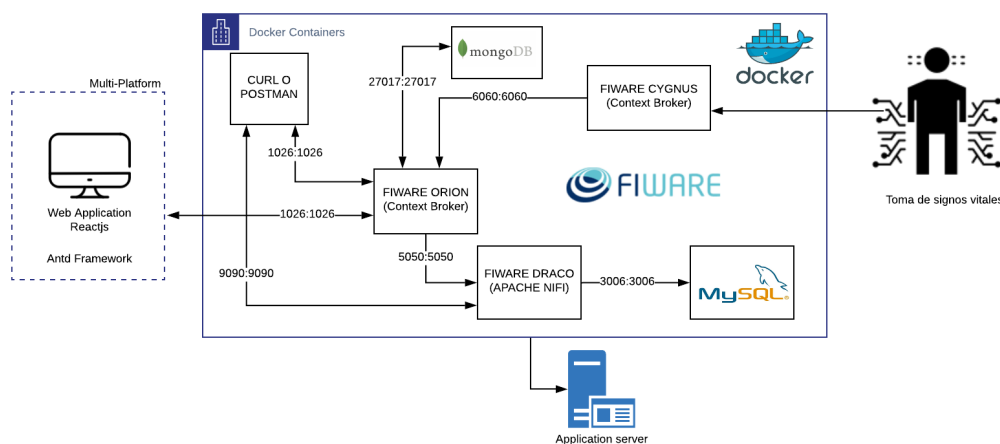


Figura 1: Integración de signos vitales con Sygnus

Se sugiere asegurar el backend con el FIWARE Pep-proxy para evitar vulnerabilidades en el flujo de los datos.

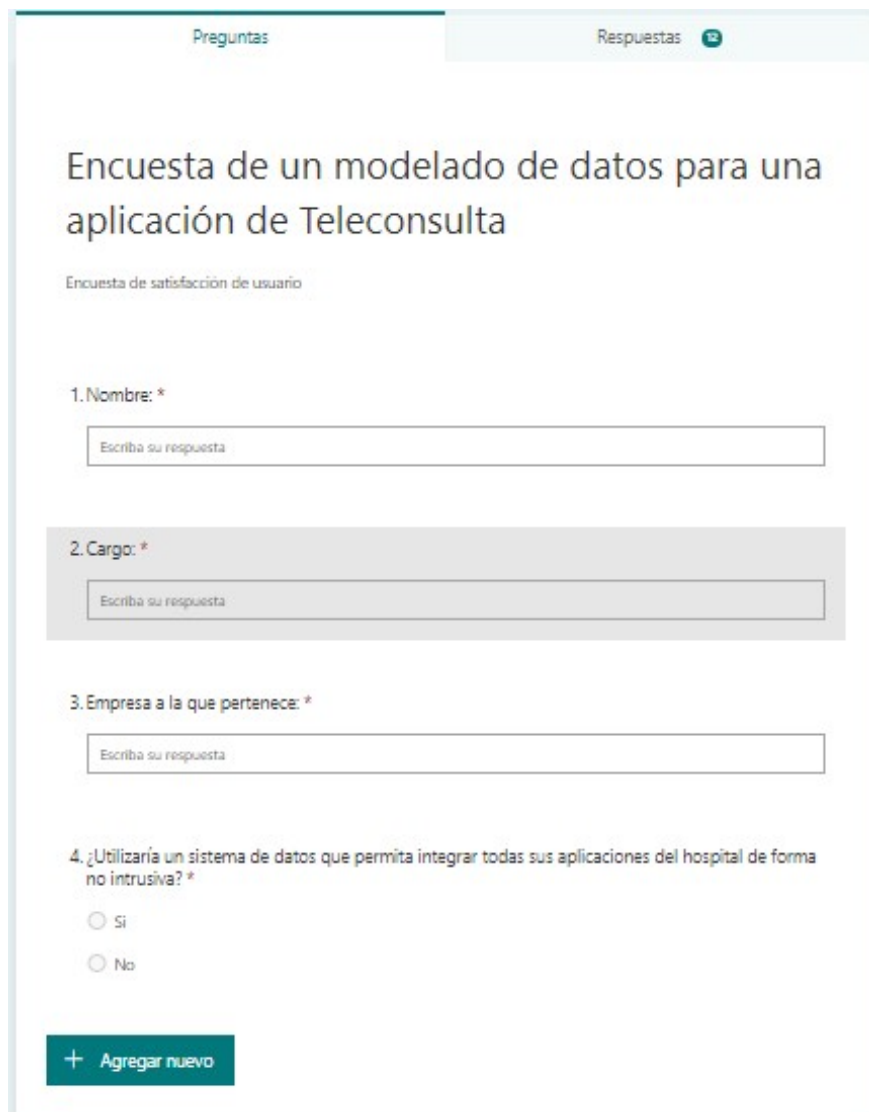
Mejoras a futuro de la aplicación:

- Utilizar un lector de signos vitales o toma de temperatura integrada a la aplicación con la finalidad de que el doctor pueda obtener información de su estado de salud en tiempo real.
- Crear una visualización de exámenes para que los doctores puedan acceder a esa

información sin necesidad de utilizar documentos físicos ni aplicaciones adicionales.

- Podría integrar farmacias a la base de datos para que cuando se crea una receta esta pueda ser despachada al paciente en la farmacia más cercana a su domicilio.

Apéndice B: Repositorio y Encuesta



The image shows a web-based survey form. At the top, there are two tabs: 'Preguntas' (Questions) and 'Respuestas' (Answers) with a count of 12. The main title is 'Encuesta de un modelado de datos para una aplicación de Teleconsulta'. Below the title, it says 'Encuesta de satisfacción de usuario'. The form contains four questions:

1. Nombre: *
Escriba su respuesta
2. Cargo: *
Escriba su respuesta
3. Empresa a la que pertenece: *
Escriba su respuesta
4. ¿Utilizaría un sistema de datos que permita integrar todas sus aplicaciones del hospital de forma no intrusiva? *
 Si
 No

At the bottom left, there is a button labeled '+ Agregar nuevo'.

Figura 2: **Encuesta de aceptación**

Para validar la aceptación de nuestra aplicación de teleconsulta se realizó una encuesta a usuarios como se muestra en la figura 2.

La programación realizada en el backend y frontend de la aplicación de teleconsulta se encuentra en los siguientes enlaces, para que las futuras generaciones puedan mejorar los procesos y toma de datos de esta aplicación.

- Gitlab backend: <https://gitlab.com/Karencita93/teleconsulta-backend>
- Gitlab frontend: <https://gitlab.com/Karencita93/teleconsulta-gye>