

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA ELECTRÓNICA

TRABAJO DE TITULACIÓN PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO
ELECTRÓNICO

PROYECTO TÉCNICO CON ENFOQUE GENERAL:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN Y
PROCESAMIENTO DE SEÑALES CON UNA ARQUITECTURA ORIENTADA
A SERVICIOS PARA LA DETECCIÓN DE FALLAS EN MÁQUINAS
ROTATIVAS MEDIANTE SEÑALES ACÚSTICAS”

AUTOR:

JULIO FERNANDO SIGUENCIA URGILES

TUTOR:

DIEGO CABRERA MENDIETA, Ph.D

CUENCA – ECUADOR

2018

CESIÓN DE DERECHOS DE AUTOR

Yo, Julio Fernando Siguencia Urgiles con documento de identificación N° 0302490354 manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy el autor del trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑALES CON UNA ARQUITECTURA ORIENTADA A SERVICIOS PARA LA DETECCIÓN DE FALLAS EN MÁQUINAS ROTATIVAS MEDIANTE SEÑALES ACÚSTICAS”**, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, julio del 2018



Julio Fernando Siguencia Urgiles

CI: 0302490354

CERTIFICACIÓN

Yo declaro que bajo mi **tutoría** fue desarrollado el trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑALES CON UNA ARQUITECTURA ORIENTADA A SERVICIOS PARA LA DETECCIÓN DE FALLAS EN MÁQUINAS ROTATIVAS MEDIANTE SEÑALES ACÚSTICAS”**, realizado por Julio Fernando Sigüencia Urgiles, obteniendo el **Proyecto Técnico con enfoque general**, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, julio del 2018

A handwritten signature in blue ink, enclosed in a circular scribble. The signature reads "Diego Cabrera Mendieta".

Diego Cabrera Mendieta, Ph.D

CI: 0104028303

DECLARATORIA DE RESPONSABILIDAD

Yo, Julio Fernando Siguencia Urgiles con número de cédula CI. 0302490354, autor del trabajo de titulación “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑALES CON UNA ARQUITECTURA ORIENTADA A SERVICIOS PARA LA DETECCIÓN DE FALLAS EN MÁQUINAS ROTATIVAS MEDIANTE SEÑALES ACÚSTICAS**” certifico que el total contenido del **Proyecto Técnico con enfoque general**, es de mi exclusiva responsabilidad y autoría

Cuenca, julio del 2018



Julio Fernando Siguencia Urgiles

CI: 0302490354

AGRADECIMIENTOS

Mi expresión sincera de agradecimiento a Dios, por haberme acompañado en este escalón de mi vida lleno de altos y bajos que me han enseñado a ser una persona fuerte y perseverante.

El más sincero agradecimiento a mis padres y hermanos por el apoyo incondicional, su cariño, esfuerzo y duros momentos pasados por verme feliz y cumplir mis metas, gracias por tanto, los amo.

Gracias a mis amigos, compañeros de clase y a todos quienes fueron parte de mi proceso de formación académica y humana.

Extiendo mi agradecimiento al Dr. Diego Cabrera Mendieta, y al grupo de Investigación y Desarrollo de Tecnologías Industriales GIDTEC, por compartir sus conocimientos, el apoyo y consejos brindados para el desarrollo de este proyecto.

Julio Fernando Siguencia Urgiles

DEDICATORIAS

Este trabajo está dedicado a mis padres Julio y Rosa que siempre soñaron verme formado como un profesional. A la Virgencita del Cisne en quien he confiado desde el primer día que llegue a la universidad y a una persona muy especial, mi abuelita, que está en el cielo gracias “*mamá*” por nunca haberme dejado solo.

Expreso mi dedicatoria a mis hermanos, Hector, Julio Alejandro, José, Jenny, Kevin y a todos mis sobrinos que son mi adoración y mi inspiración para seguir adelante.

Mi dedicatoria sincera a una persona especial que me enseñó a creer en mí y que formo parte del camino de vida que me concedió DIOS.

El puente entre la vida y la muerte son los recuerdos.

Julio Fernando Siguencia Urgiles

ÍNDICE GENERAL

AGRADECIMIENTOS.....	II
DEDICATORIAS.....	III
ÍNDICE GENERAL.....	IV
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	VIII
RESUMEN.....	IX
INTRODUCCIÓN.....	X
ANTECEDENTES DEL PROBLEMA DE ESTUDIO.....	XII
JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES).....	XV
OBJETIVOS.....	XVI
OBJETIVO GENERAL.....	XVI
OBJETIVOS ESPECÍFICO.....	XVI
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA O ESTADO DEL ARTE.....	1
1.1 Arquitectura Orientada a Servicios (SOA).....	2
1.2 Aplicaciones Móviles.....	3
1.3 Servidor apache.....	5
1.4 Protocolo ssh.....	6
1.5 Protocolos tcp/ip.....	7
1.6 Detección de fallas en máquinas rotativas.....	7
CAPÍTULO 2: METODOLOGÍA Y ARQUITECTURA DEL SISTEMA.....	15
2.1 Metodología para la detección de fallos.....	15
2.2 Arquitectura del sistema basado en SOA.....	19
CAPÍTULO 3: DESARROLLO DEL SISTEMA.....	23
3.1 Aplicación movil.....	23
3.2 ServiDor web.....	30
3.3 Servidor de cómputo.....	35
CAPÍTULO 4: FUNCIONAMIENTO Y RESULTADOS OBTENIDOS.....	39

4.1 Interfaz de FDAPP	39
4.2 Registro o inicio de sesión de usuario.....	40
4.3 Menú principal	41
4.4 Directorios en el servidor de cómputo	43
4.5 Resultados obtenidos.....	44
4.6 Análisis de resultados.....	53
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	57
REFERENCIAS BIBLIOGRÁFICAS	59

ÍNDICE DE FIGURAS

Figura 1 <i>Flujo de configuración de una conexión de Shell segura.</i>	6
Figura 2 <i>Autenticación ssh usando llaves públicas y privadas.</i>	7
Figura 3 <i>Conjunto de protocolos TCP/IP.</i>	7
Figura 4 <i>Proceso de adquisición de señales.</i>	9
Figura 5 <i>Calidad de sonido y bits</i>	10
Figura 6 <i>Adquisición de señales acústicas.</i>	16
Figura 7 <i>Representación de vectores de soporte en el plano x,y,z.</i>	17
Figura 8 <i>Flujograma del procesamiento de vectores de soporte.</i>	17
Figura 9 <i>Flujograma del procesamiento de la señal de audio con presunción de falla.</i>	18
Figura 10 <i>Arquitectura del sistema aplicación móvil para detección de fallas.</i>	19
Figura 11 <i>Servicio de presentación de la arquitectura SOA del sistema.</i>	20
Figura 12 <i>Árbol de directorios de almacenamiento en el servidor de cómputo.</i>	20
Figura 13 <i>Servicio de almacenamiento de la arquitectura SOA del sistema.</i>	21
Figura 14 <i>Servicio de registro e inicio de sesión de usuario.</i>	21
Figura 15 <i>Diagrama de bloques con el contenido de la app móvil.</i>	24
Figura 16 <i>Flujograma del proceso de inicio y registro de usuario.</i>	25
Figura 17 <i>Flujograma del proceso para crear el directorio raíz en el servidor de cómputo.</i>	25
Figura 18 <i>Flujograma del proceso registrar máquina.</i>	26
Figura 19 <i>Flujograma del proceso seleccionar máquina.</i>	27
Figura 20 <i>Configuración de base de datos en MySQL.</i>	31
Figura 21 <i>Flujograma con el proceso registrar máquina.</i>	32
Figura 22 <i>Flujograma del servicio de presentación.</i>	33
Figura 23 <i>Flujograma del servicio de directorios.</i>	33
Figura 24 <i>Flujograma del servicio de almacenamiento.</i>	34
Figura 25 <i>Flujograma del servicio de registro o inicio de sesión de usuario.</i>	35
Figura 26 <i>Flujograma del script de python procesar.</i>	36
Figura 27 <i>Flujograma del script de python process_inf.</i>	37
Figura 28 <i>Icono de la aplicación FDAPP.</i>	40
Figura 29 <i>Registro o inicio de sesión.</i>	40

Figura 30 Registro de usuario y validación de datos ingresados en caso de espacios en blanco, contraseñas incorrectas e usuario ya registrado.	41
Figura 31 Proceso del registro o selección de la máquina que desea analizar.	42
Figura 32 Proceso para obtener el estado de inferencia de la máquina.	43
Figura 33 Directorios en el servidor de cómputo.	43
Figura 34 Adquisición de señales acústicas en el banco de vibraciones de la Universidad Politécnica Salesiana.	44
Figura 35 Registro de usuarios en la base de datos del servidor web.	45
Figura 36 Almacenamiento de grabaciones de audio en el servidor de cómputo.	45
Figura 37 Resultados obtenidos de la primera muestra de audio capturada a velocidad baja en el banco de vibraciones.	46
Figura 38 Resultados obtenidos de la última muestra de audio capturada a velocidad baja en el banco de vibraciones.	47
Figura 39 Resultados obtenidos de la primera muestra de audio capturada a velocidad media en el banco de vibraciones.	47
Figura 40 Resultados obtenidos de la última muestra de audio capturada a velocidad media en el banco de vibraciones.	48
Figura 41 Resultados obtenidos de la primera muestra de audio capturada a velocidad alta en el banco de vibraciones.	48
Figura 42 Resultados obtenidos de la última muestra de audio capturada a velocidad alta en el banco de vibraciones.	49
Figura 43 Resultados de inferencia del banco de vibraciones a velocidad baja con fallo.	50
Figura 44 Resultados de inferencia del banco de vibraciones a velocidad baja en estado de funcionamiento normal.	50
Figura 45 Resultados de inferencia del banco de vibraciones a velocidad media con fallo.	51
Figura 46 Resultados de inferencia del banco de vibraciones a velocidad media con una señal de estado de funcionamiento normal.	51
Figura 47 Resultados de inferencia del banco de vibraciones a velocidad alta con fallo.	52
Figura 48 Resultados de inferencia con una señal de estado normal de funcionamiento del banco de vibraciones a velocidad alta	52

Figura 49 *Señal de audio de amplitudes en función del tiempo del banco de vibraciones a velocidad alta.* 53

ÍNDICE DE TABLAS

Tabla 1 Herramientas para el desarrollo de aplicaciones móviles. 4

Tabla 2 Resultados estadísticos de las señales de audio en estado normal de funcionamiento de la máquina. 54

Tabla 3 Valores estadísticos promedio de las señales de audio en estado normal de funcionamiento de la máquina. 54

RESUMEN

Las industrias hoy en día buscan mejorar su productividad, afianzados en nuevas tecnologías que mejoran tiempos y procesos en la línea de producción. El desarrollo tecnológico apunta a métodos de control automático y de respuesta inmediata. Sin embargo, la maquinaria que se encuentra en operación sigue expuesta a fallos debido a desgaste de sus partes, fenómenos de vibración, etc. La identificación temprana o predicción de fallos en el mejor de los casos, complementa el proceso de mejora de productividad. La recopilación de información es primordial para generar controles de calidad y además nuevas técnicas de mantenimiento, que reduzcan las paradas por fallos imprevistos.

En este proyecto de titulación se propone un sistema para la detección de fallos en máquinas rotativas mediante la adquisición y procesamiento de señales acústicas. Cuando la máquina se encuentra funcionando correctamente genera un sonido el cual puede ser utilizado como referencia para procesar y comparar características entre señales de estado normal y señales con posibles fallos. Si el resultado de la comparación varía considerablemente, da a conocer que alguna de las partes de la máquina se encuentra en mal estado y necesita servicios de mantenimiento.

El sistema propuesto se compone de 3 partes principales como son: dispositivos móviles como medios de adquisición de señales acústicas, servicios web para comunicación a través del internet y servidor de cómputo para el procesamiento de la información. Su arquitectura está orientada a servicios con el fin de volver a todos sus componentes interoperables y con la posibilidad de agregar servicios que se puedan desarrollar a futuro sin alterar la arquitectura inicial. La aplicación móvil mediante el micrófono del dispositivo permite capturar señales acústicas, las cuales pueden ser almacenadas y procesadas en un servidor remoto. Además, cuenta con servicios de registro o inicio de sesión de usuarios y un sistema de directorios para almacenar la información de forma organizada y específica para cada cliente del sistema de detección de fallos. Los resultados obtenidos con una señal de fallo muestran valores estadísticos con cierta variación con respecto a los valores estadísticos de una señal en estado normal. El factor de similitud de histogramas resulta más cercano a cero cuando se analiza una señal con estado de fallo.

INTRODUCCIÓN

El desarrollo tecnológico ha beneficiado también al sector industrial para mejorar procesos, usar técnicas de trabajo en colaboración hombre máquina y reducir procesos totalmente mecánicos por procesos semi-automáticos y en algunos casos totalmente automáticos. De igual forma para los procesos de mantenimiento existen sofisticados elementos de medición y control que permiten identificar con mayor facilidad errores en el funcionamiento de las máquinas. Por lo general las técnicas de mantenimiento se aplican periódicamente y en caso de que ocurra un fallo inesperado en la máquina, sin embargo, se enfoca solamente a corregir el error. La nueva era de la revolución industrial se basa en la recolección y almacenamiento de los datos para procesarlos y predecir fallos, además encontrar los factores que producen errores comunes.

En varias investigaciones se proponen enfoques basados en modelos para el diagnóstico de fallas. Por ejemplo *X. Ren* en [1], propone una arquitectura orientada a servicios basada en tecnologías Grid para integración de diferentes esquemas de diagnóstico de fallas en un marco común. Utiliza una metodología basada en FDI (*Fault Detection Isolation*) que se divide en dos pasos: generación residual y evaluación residual. La generación residual es un proceso donde la entrada y la salida de un sistema se supervisan y manipulan para generar una señal denominada residual. El residuo es normalmente cercano a cero cuando no hay falla presente, y diferente de cero cuando ocurre una falla. La evaluación residual es el análisis del residuo para examinar la probabilidad de fallas. La decisión es tomada en base al conocimiento sobre el proceso y los síntomas.

Otra metodología para el diagnóstico de fallos se presenta en [2], donde proponen una aplicación móvil para realizar monitoreo basado en la condición de máquinas industriales. La aplicación utiliza grabaciones acústicas y técnicas de minería de datos para el reconocimiento de patrones acústicos, es decir reconocer estados defectuosos de un estado saludable de las máquinas. El proceso de reconocimiento se logra en 5 pasos que son: adquisición de datos, procesamiento de datos, extracción de características, selección de características y finalmente clasificación.

Es así como en el presente trabajo se propone un sistema para detección de fallos de máquinas rotativas basado en una arquitectura orientada a servicios. El cliente tiene la posibilidad de consumir los servicios que presenta la arquitectura tales como: registro o inicio de sesión de usuarios, almacenamiento de la información, procesamiento de la información e inferencia del estado de la máquina. El sistema está compuesto por una aplicación móvil que puede ser instalada en dispositivos con un sistema operativo Android, servicios web para conexiones a través de internet y un servidor con poder computacional para el procesamiento de la información.

La memoria técnica está organizada de la siguiente manera. En el capítulo 1 se hace un breve análisis del estado del arte, donde se abordan temas sobre arquitecturas orientadas a servicios para la detección de fallas en máquinas industriales, tipos de aplicaciones móviles, servicios web, y herramientas matemáticas para la detección de fallas. En el capítulo 2 se propone la metodología y arquitectura para el desarrollo del sistema. En el capítulo 3 se habla específicamente sobre el desarrollo del sistema, se especifica los algoritmos utilizados y el software para el desarrollo de la aplicación móvil, los servicios web y los algoritmos para el procesamiento de la información en el servidor de cómputo. En el capítulo 4 se presenta el funcionamiento del sistema y los resultados obtenidos. Finalmente, en el capítulo 5 se da a conocer las conclusiones obtenidas del proyecto y recomendaciones para futuros trabajos.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

Hoy en día el desarrollo de software industrial apunta a modelos o arquitecturas que permitan optimizar los recursos computacionales disponibles, para cumplir con las expectativas de la industria a menor costo y de manera inmediata. En consecuencia, nace una nueva tendencia llamada Arquitectura Orientada a Servicios (SOA por sus siglas en inglés), que se ha convertido en el paradigma predilecto para implementar sistemas que permiten acoplar esquemas estandarizados para compartir información dentro del sistema y generar resultados óptimos y a menor costo. En [3] se menciona el siguiente concepto “Una arquitectura orientada a servicios (SOA) es un conjunto de principios arquitectónicos para la construcción de sistemas autónomos pero interoperables”.

Los principios de orientación a servicios son la clave fundamental para conseguir ciertas propiedades deseables en la arquitectura [4], tales como:

- Contrato de servicio estandarizado, requiere que se tengan en cuenta consideraciones específicas al diseñar la interfaz técnica pública de un servicio y evaluar la naturaleza y la cantidad del contenido que se publicará como parte del contrato oficial de un servicio.
- Acoplamiento, este principio aboga por la creación de un tipo específico de relación dentro y fuera de los límites del servicio, con un énfasis constante en la reducción de las dependencias entre el contrato de servicio, su implementación y los consumidores de servicios.
- Abstracción, este principio enfatiza la necesidad de ocultar de los detalles subyacentes de un servicio como sea posible.
- Autonomía, Apoya la medida en que otros principios de diseño pueden ser efectivamente realizados en entornos de producción del mundo real mediante el fomento de características de diseño que aumentan la fiabilidad de un servicio y la previsibilidad del comportamiento.
- Descubrimiento, para que los servicios se posicionen como activos de TI, necesitan ser fácilmente identificados y comprendidos cuando se presentan oportunidades de reutilización.

Existen enfoques SOA aplicados a diferentes áreas. Por ejemplo en el área de negocios brinda capacidad de agilidad empresarial es decir que las arquitecturas TI implementadas soporten cambios dinámicos en respuesta a exigencias de los clientes, innovaciones tecnológicas y sobre todo panorama competitivo [5]. En el ámbito industrial permite la distribución inteligente de dispositivos con mecanismos de diagnóstico eficiente para mejorar el tiempo de actividad del equipo, brindando una pronta reacción antes del colapso o recuperándose sin inconvenientes del fallo [6]. Además, cabe mencionar que SOA proporciona sistemas de fabricación reconfigurables (RMS) [7] y sistemas de producción evolutivos (EPS) [8]. En el mismo sentido, en la investigación faculta la posibilidad de integración y re-uso de modelos de diferentes disciplinas para simular y modelar sistemas más complejos reduciendo la heterogeneidad de los nombres de las variables y facilitando el uso de interfaces de comunicación [9]; todos estos enfoques referidos hacia un mismo objetivo, la interoperabilidad de sus componentes.

Haciendo énfasis sobre este paradigma aplicado para el desarrollo de diferentes aplicaciones industriales se pueden ver aplicaciones como las mostradas en [10] donde se utiliza SOA para el diseño de un software que permita el diagnóstico de fallos mediante el uso de técnicas machine learning en máquinas rotativas. Otra aplicación la podemos encontrar en [1] donde se propone un servicio Grid o servicio web, basado en un conjunto de procesos distribuidos apoyados en internet, utilizando estándares como XML, SOAP, WSDL, UDDI. Demostrándose en las evidencias anteriores la integración de SOA para procesos robustos de supervisión para aplicaciones de e-Mantenimiento, paralelismo en el desarrollo de software e implementación de nuevos servicios.

Por otro lado, el avance tecnológico ha logrado mejorar las características hardware y software de los dispositivos móviles permitiendo ofrecer servicios de alto nivel y rendimiento. Sin embargo, demandan un elevado uso de recursos del dispositivo, ancho de banda y servidores de internet. A continuación, se da a conocer varios ejemplos de aplicaciones móviles:

- Reconocimiento facial, en [11] se ha desarrollado una aplicación móvil para el entretenimiento del usuario. Mediante el uso de la cámara permite el reconocimiento de rostro de los jugadores donde el objetivo es disparar cuando se

reconozca o se apunte el rostro del jugador. Para ello la aplicación de juego envía fotogramas de cámara que contienen las caras de los jugadores al servidor donde está contenido el módulo de reconocimiento facial.

- Aplicaciones de realidad aumentada (MAR), estas aplicaciones demandan un alto nivel computacional del dispositivo y un ancho de banda considerable para el tráfico de datos, en [12] se profundiza sobre aplicaciones basadas en MAR para dispositivos móviles y equipos portátiles donde se definen directrices para futuros protocolos de transporte en tiempo real de datos y multimedia.
- Procesamiento de sonido, en [13] proporcionan una visión general del uso de estas aplicaciones que han sido desarrolladas en base a las características de los dispositivos móviles e infraestructura de red.

Con el objetivo de reducir costos y ofrecer un servicio de calidad, se han desarrollado aplicaciones móviles basadas en SOA. Por ejemplo, en [14] encontramos una aplicación para el cuidado de la salud basadas en SOA. Esto les permite un monitoreo en tiempo real de los pacientes mediante uso de dispositivos móviles, mejor flexibilidad y extensibilidad a menor costo a través de módulos de software reutilizables y autónomos, además permite el desarrollo de aplicaciones variadas mediante el uso de estándares reduciendo la complejidad de comunicación entre ellas. Se habla también en [15] de una plataforma web abierta orientada a servicios desarrollados para sistemas distribuidos, dispositivos móviles y servicios de TI médicos. El componente central de la plataforma web es un bus de servicio que permite la integración y organización de servicios basados en la nube. Todos los servicios inicialmente se conectan al bus de servicio independientemente si son clientes que solicitan un servicio o proveedores que proporcionan el mismo.

JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

Para el monitoreo del estado de la máquina es necesario implementar un sistema de adquisición de datos e instrumentación que involucra en algunos casos readecuación de la infraestructura del lugar de trabajo de la máquina. Esto genera un alto costo debido a la implementación de sensores, tarjetas de adquisición, sistemas SCADA y personal capacitado para la calibración y ajuste de dichos instrumentos. Como un ejemplo de este tipo de sistemas, en la Universidad Politécnica Salesiana sede Cuenca dentro del grupo de investigación GIDTEC se cuenta con un sistema de detección de fallos y software de diagnóstico, que permite proponer técnicas de mantenimiento basados en la condición. Para ello el sistema cuenta con; acelerómetros, sensores acústicos, sensores de corriente, sensores de voltaje, etc. Además de tarjetas de adquisición, equipos de almacenamiento y procesamiento de datos.

En este trabajo técnico se propone una solución para reducir costos en la etapa de adquisición y almacenamiento de datos por parte de los usuarios, mediante la implementación de un sistema que hará uso de una aplicación móvil que será la encargada de adquirir señales de tipo acústicas utilizando el micrófono que tiene incorporado el dispositivo. Estas señales serán enviadas a un servidor de cómputo que se encargará del procesamiento y extracción de características de la señal de audio.

La arquitectura del sistema será diseñada usando el paradigma SOA. La finalidad es aprovechar todos los recursos disponibles, permitir la interoperabilidad de sus componentes, dinamizar todo el sistema, y admitir que se acople hacia las nuevas tendencias tecnológicas que se puedan desarrollar en un futuro no muy lejano. La aplicación móvil permitirá que el usuario pueda visualizar datos estadísticos del sonido que genera la máquina, cuando entra en funcionamiento. Además, proporcionara información gráfica mediante histogramas, del comportamiento de la señal de audio. Todo el proceso será versátil se integrará una interfaz amigable con el usuario, es así que el técnico podrá utilizar la aplicación sin la necesidad de tener mayores conocimientos sobre instrumentación electrónica.

OBJETIVOS

OBJETIVO GENERAL

- Diseñar e implementar un sistema de adquisición y procesamiento de señales con una arquitectura orientada a servicios para detección de fallas de máquinas rotativas mediante señales acústicas.

OBJETIVOS ESPECÍFICO

- Estudiar y analizar las características que ofrece SOA en el desarrollo de aplicaciones para la adquisición de señales.
- Diseñar la arquitectura acorde a SOA para la adquisición de señales con el dispositivo móvil, comunicación con el servidor de cómputo y procesamiento de señales para la detección de fallos.
- Implementar la aplicación móvil para Android con una interfaz amigable para los usuarios, con la capacidad de monitorear el estado de la máquina en tiempo real.
- Implementar en el servidor de cómputo el software de procesamiento de señales para la detección de fallos en maquinaria rotativa.
- Realizar pruebas experimentales de la plataforma en el banco simulador de fallos del grupo GIDTEC.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA O ESTADO DEL ARTE

El desarrollo de aplicaciones de software está basado en enfoques particulares que facilitan la solución de problemas encontrados durante el desarrollo de las mismas. En [16] se menciona 3 enfoques fundamentales los cuales son: programación estructurada, programación modular y la programación orientada a objetos. Estos paradigmas han evolucionado con el fin de construir sistemas más grandes y complejos que puedan cumplir con las exigencias de los usuarios como: manejo seguro de datos, comunicación entre sistemas, seguridad cibernética y rapidez en respuesta.

En base a los enfoques de programación se establece el software distribuido. En [17] se define software distribuido como *“un sistema cuyos componentes están ubicados en diferentes máquinas-servidores que se comunican entre sí mediante protocolos de comunicación”*. Es decir, un sistema que ofrece un servicio está compuesto por varios subsistemas que cumplen con un proceso en cada uno de los servidores y responden según lo solicitado por el usuario.

SOA comparte principios de los sistemas distribuidos, pero de una manera más organizada y estructurada para alcanzar la interoperabilidad de sus componentes. Esta arquitectura permite la integración entre aplicaciones, define las bases para que

diversos servicios independientes puedan cooperar entre si y da lugar a procesos mucho más grandes y complejos.

1.1 ARQUITECTURA ORIENTADA A SERVICIOS (SOA)

El término servicio dentro del desarrollo de sistemas computacionales se define como “*un programa que interactúa con otros mediante intercambio de mensajes*” [16]. Este servicio debe ser construido de forma consistente siendo estable y ofreciendo disponibilidad cuando el sistema necesite que este servicio entre en operación.

SOA se caracteriza por soportar servicios débilmente acoplados para facilitar la flexibilidad en las negociaciones de los elementos de la arquitectura de una manera interoperable e independiente de la tecnología. La arquitectura orientada a servicios permite que programas ejecutables, puedan ser solicitados por otros programas que actúan como consumidores de servicios. Los servicios pueden ser programas de aplicación, nuevos o heredados que se ejecutan como cajas negras (abstracción¹)[18].

Esto se traduce en eficiencia frente al desarrollo de nuevas tecnologías. Los desarrolladores pueden reemplazar servicios con diferentes estructuras y obtener el mismo resultado. Basados en la abstracción, cada servicio o componente se vuelve independiente de los demás a diferencia de aplicaciones monolíticas² como ocurría en el pasado. Dentro de este marco es importante recalcar una de las características principales de SOA, el hecho de que no está ligada a un tipo de tecnología concreta, tan solo basta que permita desarrollar servicios interoperables y manejen protocolos de comunicación bien definidos. Esto facilita a que los clientes puedan consumir los servicios, desde cualquier dispositivo, usando cualquier sistema operativo y en cualquier lenguaje.

La arquitectura SOA de los sistemas o aplicaciones dependerá de los servicios que se pretenda ofrecer y los componentes que van a conformar dicho sistema. SOA estará compuesto exclusivamente de servicios que puedan ser dinámicamente

¹ **Abstracción:** Oculta su lógica interna. No es necesario conocer la parte interna del programa, basta con saber la entrada que requiere, la salida que produce y cómo se invoca para su ejecución[18].

² **Aplicaciones Monolíticas:** Encapsulaba la interfaz del usuario, la lógica de los negocios y las operaciones de almacenamiento de datos en un solo componente[16].

localizados y entrelazados según las funcionalidades que puedan brindar. Las características básicas que deben cumplir los servicios [19] son:

- Las descripciones necesitan estar bien definidas para que sean independientes de la implementación del servicio.
- Los servicios deben ser independientes entre ellos.

En este sentido SOA, es una arquitectura que se rige por un conjunto de normas, niveles de abstracción, procedimientos y políticas que definen su estilo. La idea de usar una arquitectura basada en servicios es incrementar el rendimiento de las empresas que integren en sus procesos este tipo de arquitectura. El acoplamiento hacia nuevas estrategias y tecnologías debido a los desafíos de mercado que cambian rápidamente, se vuelve primordial ya que representan ahorro en el proceso de integración y menor tiempo de inproductividad. En el campo industrial se aplican estos nuevos enfoques en arquitecturas distribuidas basados en dispositivos inteligentes. La finalidad, adaptar capacidades de diagnóstico en las infraestructuras de TI(Tecnologías de la Información), de los sistemas de producción futuros[6].

1.2 APLICACIONES MÓVILES.

Hoy en día los dispositivos inteligentes se caracterizan por un alto nivel de rendimiento y gran capacidad de almacenamiento, dejando atrás el concepto de teléfono móvil que se utilizaba solamente para establecer una comunicación oral o escrita. En base a estas capacidades tecnológicas se desarrollan aplicaciones que son de utilidad para los usuarios, no solamente con fines de entretenimiento. Existen aplicaciones que son utilizadas como medios de estudio, por ejemplo bibliotecas virtuales[20], en el campo laboral como instrumentos de adquisición de datos [21], manuales de trabajo digitales para los operarios, etc.

El desarrollo de aplicaciones dependerá del sistema operativo que se encuentre instalado en el dispositivo. Actualmente las empresas líderes en software móvil son: Apple[®], Google Android[™], Windows Phone[®], y Blackberry[®]. Cada una de estas empresas maneja su plataforma de desarrollo y lenguaje de programación. El acceso a las apps móviles proporcionadas por sus propietarios se lo realiza a través de tiendas virtuales como Apple App Store[®], Android Play Store[™], Windows Store[®] y BlackBerry App World[®]. Algunas de las aplicaciones disponibles son gratuitas mientras que otras tienen un costo.

La aplicación móvil es un programa informático diseñado para ejecutarse en dispositivos móviles inteligentes [22]. Existen diferentes formas de desarrollar una aplicación y dependerán de las características y limitaciones que disponga el dispositivo, además de los destinatarios y funcionalidad de la aplicación. En base a estos criterios hoy en día se dispone de enfoques para el desarrollo de aplicaciones móviles. Cada enfoque contiene beneficios y limitaciones inherentes acorde a las necesidades del usuario. Entre los enfoques más importantes tenemos: desarrollo de aplicaciones nativas, desarrollo de aplicaciones web y desarrollo de aplicaciones híbridas. A continuación, se describe brevemente cada una de ellas[22]:

➤ **Aplicaciones Nativas:**

Son programas diseñados para ejecutarse en los dispositivos móviles. Por lo general estos archivos se descargan de las tiendas de aplicaciones de internet y se almacenan en el dispositivo. Tras su instalación la aplicación nativa se conecta directamente con el sistema operativo móvil, aprovechando de esta manera las funcionalidades del dispositivo. El programa se debe escribir usando las herramientas provistas por el distribuidor del sistema operativo SDK (*Software Development Kit*). Una vez escrito el código se debe compilar y crear el ejecutable en formato binario y empaquetarlo junto al resto de recursos para su distribución. En la Tabla 1 se detalla los lenguajes, herramientas, formatos y tiendas más populares de aplicaciones móviles de los diferentes sistemas operativos que hoy en día están en el mercado.

Tabla 1 Herramientas para el desarrollo de aplicaciones móviles.

Fuente: [23]

	Apple iOS	Android	Blacberry	Windows Phone
Lenguajes	C,C++	Java,C,C++	Java	C#,VB.NET,etc.
Herramientas	Xcode	Android SDK	BB Java Eclipse Plug-in	Visual Studio, Windows Phone
Formato	.app	.apk	.cod	.xap
Tiendas	Apple App Store	Google Play	Blacberry App World	Windows Phone Marketplace

➤ **Aplicaciones móviles basadas en la Web.**

En este enfoque no se utiliza un SDK, puesto que las aplicaciones web no necesitan instalarse. Pueden ser visualizadas usando el navegador del teléfono como si se tratara de un sitio web normal. Los nuevos dispositivos móviles poseen navegadores robustos que ofrecen soporte a funcionalidades como: HTML5, Cascading Style Sheets(CSS3) y JavaScript. Esto permite programar de forma independiente al sistema operativo en la cual se ejecutará la aplicación móvil.

➤ **Aplicaciones Híbridas**

Este tipo de aplicación móvil combina desarrollo nativo con tecnología Web. La parte Web de la aplicación se ubica en un servidor y por otro lado un conjunto de archivos HTML, JavaScript, CSS y medios incorporados en el código de la aplicación podrán almacenarse localmente en el dispositivo. Cuando sea necesario se establece un acceso directo a APIs nativas.

1.3 SERVIDOR APACHE

Apache es posiblemente el servidor web más utilizado en el mundo, debido a sus diversas características y sobre todo por ser un servidor web HTTP de código abierto. Está disponible para plataformas como: Unix, Windows, Macintosh, y otras, que hacen uso del protocolo de comunicación HTTP[24]. El servidor apache presenta una arquitectura modular³ y consta de una sección core y diversos módulos que brindan un gran aporte a la funcionalidad básica para un servidor web. Entre estos módulos tenemos[25]:

- **mod_ssl** - Comunicaciones Seguras vía TLS.
- **mod_rewrite** – Transforma páginas dinámicas php en páginas estáticas html.
- **mod_php** - Páginas dinámicas en PHP.
- **mod_python** - Páginas dinámicas en Python.
- **mod_aspdotnet** - Páginas dinámicas en .NET de Microsoft.
- **mod_security** - Filtrado a nivel de aplicación, para seguridad.

Una de las características usadas en este desarrollo es la comunicación segura vía TLS. El protocolo de seguridad TLS por sus siglas en inglés “*Transpor Layer*

³ **Arquitectura Modular:** Admite ampliar funcionalidades y características al servidor mediante incorporación de módulos externos[50].

Security” y SSL “Secure Socket Layer” son mecanismos de seguridad para el envío de datos seguro. Estos protocolos son definidos como criptográficos, debido a que operan por debajo de la capa de aplicación y proporcionan cifrado de extremo a extremo.

1.4 PROTOCOLO SSH

Secure Shell es un protocolo que permite administrar de forma segura el sistema y transferencia de archivos a través de redes de comunicación, utilizando cifrado para mantener segura la conexión entre un cliente y un servidor. Es decir se mantienen encriptados: las autenticaciones de usuario, comandos, resultados y transferencias de archivos, brindando seguridad de extremo a extremo en el envío y recepción de la información[26]. En la documentación de SSH Communications Security [27], mencionan el protocolo como un modelo cliente-servidor, donde la conexión la establece el cliente. En la Figura 1 se puede observar el flujo de comunicación de una conexión ssh.

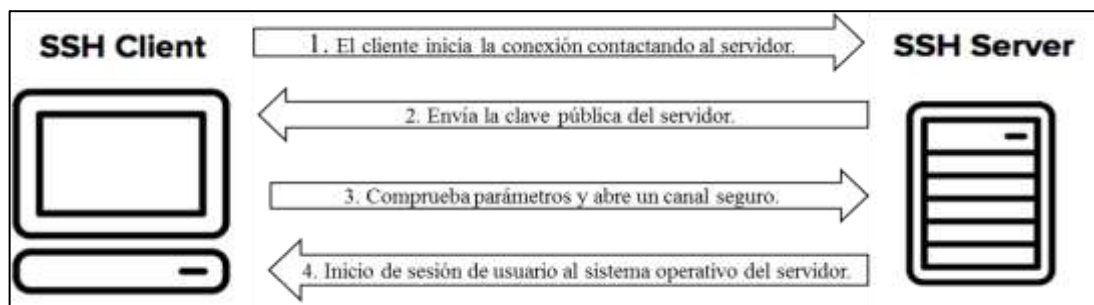


Figura 1 Flujo de configuración de una conexión de Shell segura.

Fuente: [26]

El protocolo de seguridad ssh se describe como un proceso de autenticación con llaves públicas y privadas ssh. Su funcionamiento se asemeja a la de un candado y su respectiva llave, es decir la clave pública es el candado y se encuentra alojada en el servidor mientras que la clave privada es la llave con la cual se abre el candado y es usada por el cliente como se puede observar en la Figura 2. Por lo general estas claves generadas se almacenan en un directorio del home del sistema operativo con el nombre .ssh, para el caso de la clave pública lleva el nombre id_rsa.pub y la llave privada id_rsa[28].



Figura 2 Autenticación ssh usando llaves públicas y privadas.

Fuente:[29]

1.5 PROTOCOLOS TCP/IP

El protocolo TCP/IP es un conjunto de normas y procedimientos que permiten a los dispositivos de red y programas de aplicación intercambiar información[30]. Es el protocolo de red con mayor uso en el mundo, debido a sus diversas funcionalidades y transmisión fiable de paquetes de datos sobre redes locales y a través del internet.

Este protocolo se puede interpretar en términos de capas como se observa en la Figura 3. El flujo empieza en la capa de aplicación, aquí se envían mensajes o datos a la capa de transporte UDP o TCP. Estos protocolos reciben los datos y los convierten en partes más pequeñas llamados paquetes, además añaden una dirección de destino y pasan al siguiente protocolo. En el protocolo IP o de internet el paquete se coloca en un datagrama, se añade una cabecera y cola del datagrama para enviar hacia el siguiente nivel o interfaz de red. Este nivel recibe los datagramas y los transmite como tramas a través de hardware de red específico o red física[31].

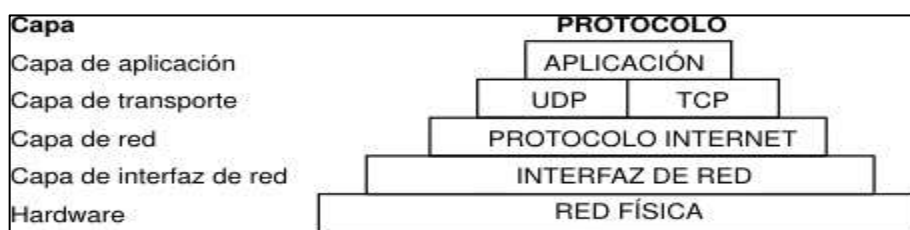


Figura 3 Conjunto de protocolos TCP/IP.

Fuente: [31]

1.6 DETECCIÓN DE FALLAS EN MÁQUINAS ROTATIVAS

La maquinaria en constante funcionamiento a lo largo del tiempo está expuesta a daños o disfuncionalidades, debido a condiciones de esfuerzo, rozamiento, picos de tensión y otros efectos que están presentes en el lugar de trabajo. Para reducir la cantidad de fallas y evitar paradas imprevistas las empresas

implementan estrategias de mantenimiento en base a sus propios procesos. Entre las estrategias de mantenimiento más utilizados tenemos: mantenimiento preventivo, mantenimiento correctivo, mantenimiento predictivo[32].

El mantenimiento correctivo se aplica cuando algún proceso o máquina presenta falencias en un estado de funcionamiento normal. En cambio, los preventivos son periódicos es decir se aplican cada cierto tiempo en el que los equipos o máquinas se vuelvan vulnerables a fallas. El tipo de mantenimiento predictivo o conocido también como mantenimiento basado en la condición (CBM), posibilita pronosticar el futuro de una falla que se pueda presentar en la máquina. La inferencia del estado de la máquina lo realiza mediante determinadas variables de operatividad como por ejemplo temperatura, vibración, consumo de energía, etc. La alteración de dichas variables dará un indicio de posibles problemas de las condiciones operativas normales.

La mayoría de las industrias optan por usar métodos de detección de fallas basadas en datos, que proporcionan información eficaz para futuros proyectos de mantenimiento o reestructuración de procesos. La ventaja de almacenar información permite crear sistemas de detección, clasificación y diagnóstico de fallas, mediante patrones que facilitan la identificación de dichas fallas. Como se mencionó, este método se caracteriza por disponer información para el procesamiento. La adquisición de datos y el proceso de extracción de características pueden ser cualitativas o cuantitativas[33].

- **Los métodos de extracción cualitativa:** Son sistemas expertos, métodos basados en lógica difusa, etc.
- **Los métodos de extracción cuantitativa:** Son sistemas de redes neuronales y técnicas estadísticas [39],[40].

Los métodos de extracción cuantitativa son la forma convencional de realizar análisis y diagnósticos de fallas en máquinas rotativas. Por lo general se utilizan metodologías como análisis de firmas de vibración, análisis de firmas de lubricantes, análisis de firmas de ruido y monitoreo de temperatura. El diagnóstico por vibración es el comúnmente utilizado debido a que se considera el espectro de cada máquina como normal. El momento que se presenta una falla el espectro cambia su forma. Por lo general se obtiene las señales de vibración de acelerómetros que se encuentran

ubicados en la máquina, en base a una respectiva configuración y posicionamiento de los sensores. En [34], por ejemplo se especifica la configuración y posición de acelerómetros de una caja de engranajes que consta de 3 ejes y 4 engranes.

Otra metodología para la detección de fallas es captar y procesar señales acústicas generadas por las máquinas a través del uso de micrófonos. Por ejemplo, en [35], proponen un sistema para detección de fallas de rodamientos de máquinas rotativas con el uso de señales acústicas grabadas por un teléfono móvil. Los resultados de las mediciones contenían información suficiente para distinguir casos de estado saludable y de falla de la máquina y además determinar el tipo de fallo específico. Un ejemplo similar se propone en [34], donde implementan un sistema para el diagnóstico de fallas en rodamientos de una máquina síncrona, utilizando micrófonos omnidireccionales para captar señales vibro-acústicas. Estas señales son procesadas dando como resultado la detección de una falla en la jaula del rodamiento de un generador síncrono en las coordenadas x , y .

Captar el sonido acústico resulta entonces un método confiable para la detección de fallas de máquinas rotativas. Sin embargo, para lograr un alto grado de efectividad de este método es necesario que el proceso de adquisición de señal sea el más apropiado. El proceso de adquisición de señales está conformado por sensores y/o transductores que captan la señal física y la convierte en una señal eléctrica, luego esta señal es acondicionada mediante filtros para eliminar ruido y por último la señal pasa por un convertidor analógico digital para que pueda ser procesada por un equipo informático. En la Figura 4 se puede observar el flujo para el proceso de adquisición de señales.

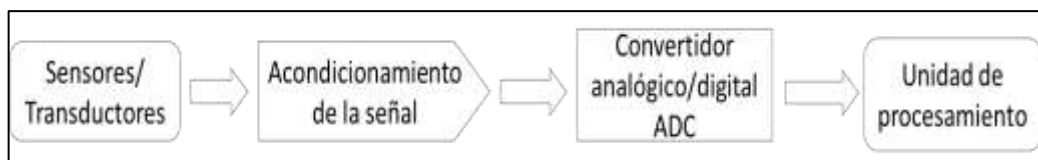


Figura 4 Proceso de adquisición de señales.

Fuente: Autor

➤ Sensores / Transductores

Son elementos que tienen la capacidad de adaptar una magnitud física de entrada, a una magnitud proporcional que pueda manejar un sistema. Por lo general la señal que manejan los sistemas para sus procesos es de tipo eléctrica. Un sensor es

un elemento que se encuentra en contacto directo con la magnitud evaluada, al variar esta magnitud las propiedades del sensor sufren cambios y reflejan un valor. En cambio, un transductor tiene la capacidad de convertir una señal física en una señal que el sistema pueda manejar[36].

➤ **Acondicionamiento de Señal**

El acondicionamiento de señal es un proceso por el cual la señal proveniente de un sensor o transductor pasa por una etapa de filtrado y adaptación a la entrada del convertidor analógico-digital. Sus características principales son adaptar el rango de la señal de salida del transductor al rango de la entrada del convertido ADC. Además, tiene la capacidad de acoplar las impedancias de salida del transductor con la entrada del convertidor analógico digital[36].

➤ **Conversor Analógico – Digital(ADC)**

El conversor ADC recibe una señal analógica continua en amplitudes (señal eléctrica) a su entrada, proveniente de sensores o transductores. El conversor efectúa un proceso de cuantificación y codificación para transformar la señal analógica adquirida a una señal digital o discreta en el tiempo. La señal digital es comprendida por los sistemas computacionales y por lo tanto permite realizar procesamiento de la información [46]. La cuantización convierte los valores continuos en discretos como se observa en la Figura 5. Mientras la tarjeta de adquisición de datos tenga mayor número de bits la resolución de la señal muestreada será mejor.

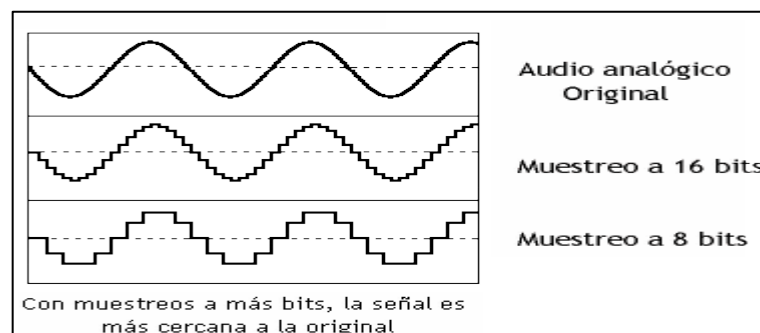


Figura 5 *Calidad de sonido y bits*
Fuente:[37]

En la actualidad encontramos este tipo de conversores en todos los dispositivos electrónicos que necesitan adquirir y procesar información. Por ejemplo, dispositivos móviles, computadoras, tarjetas de adquisición de datos (DAQ, Arduino,

RaspBerry, etc). Cabe mencionar que para asegurar la toma de muestras y conversión análoga-digital de forma correcta se debe considerar la tasa de muestreo. Para ello el Teorema de Nyquist establece que la frecuencia de muestreo f_s , deber ser mínimo el doble que el ancho de banda de la señal muestreada es decir $f_s > 2 \cdot f_m$. Esta consideración permitirá impedir el fenómeno aliasing, es decir interrumpir el paso de las frecuencias de alta ingresen a las frecuencias capturadas.

➤ **Unidad de procesamiento**

La unidad de procesamiento por lo general son sistemas con capacidad computacional, donde se aloja un software que se encarga de procesar los datos. El poder computacional dependerá de la capacidad de memoria que dispongan estas unidades. Para el análisis y manejo de datos hoy en día existe una variedad de software que realiza procesamiento de datos o maneja herramientas matemáticas para dicho proceso. Hay que tomar en cuenta que algunos de los softwares para procesamiento tienen licencia mientras que otros son libres.

El software maneja herramientas matemáticas y estadísticas para el tratamiento de los datos. Las medidas estadísticas son herramientas matemáticas que ayudan a obtener información sobre la situación, dispersión y otros patrones de comportamiento de datos de una muestra. La información resultante facilita la toma de decisiones, reduce la cantidad de datos en ciertos valores de interés y permite diferenciar y distinguir series de datos. Esta información faculta tener una idea preliminar de los datos antes de realizar un tratamiento más complejo de los mismos[38].

Las medidas estadísticas más importantes son:

- ✓ Medidas de posición central: Media, Mediana, Moda.
- ✓ Medidas de dispersión: Amplitud, Varianza, Desviación Estándar.
- ✓ Medidas de forma: Asimetría, Curtosis.

Los parámetros estadísticos por lo general son índices confiables para extraer información de la condición de las máquinas. La metodología que se aplique dependerá del tipo de datos que se adquiera. La mayor parte de investigaciones para la detección de fallas en máquinas utilizan señales de vibración y acústicas. La

información aportada tanto en el dominio del tiempo como en la frecuencia permiten determinar condiciones anormales.

En [39], demuestran que si se realiza un promedio de un conjunto de espectros de máquinas en buen estado y del mismo tipo generan una referencia confiable de funcionamiento. Existen casos en el que las máquinas son individualistas respecto a sus espectros en ese caso, también es posible obtener una referencia de cada máquina realizando el cálculo del promedio de varias mediciones captadas durante un periodo de tiempo con el fin de generar un espectro de referencia.

Por ejemplo, estas medidas estadísticas fueron aplicadas para implementar un sistema de identificación y clasificación de fallas de una caja de engrajes mediante redes neuronales convolucionales. La desviación estándar, la asimetría y la curtosis evaluadas en las señales de vibración se usaron para entrenar y probar la red neuronal convolucional implementada[40].

Otro caso particular en [34], proponen un diagnóstico de fallas en una caja de engrajes mediante redes neuronales multicapa. Las señales de vibración obtenidas en el dominio del tiempo generalmente cambian cuando se produce una falla. Entonces utilizaron valores estadísticos como el valor RMS para representar la amplitud de la vibración y la energía en el dominio del tiempo. Los valores de la desviación estándar, asimetría y curtosis para representar la distribución de la serie temporal de una señal de vibración en el dominio del tiempo. Estos valores se usaron como parámetros de entrada para la red neuronal multicapa.

A continuación, se analiza las medidas estadísticas más importantes y utilizadas en el diagnóstico de fallas:

✓ **Media**

Es una medida estadística de tendencia central y que devuelve como resultado un valor promedio de la muestra sin ser influida por valores extremos. De una muestra con valores x_1, x_2, \dots, x_n le media se denota de la siguiente manera[41]:

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (1)$$

✓ **Desviación Estándar**

La desviación estándar es una medida estadística de dispersión o variabilidad. Esta, permite determinar con mayor precisión los valores de una distribución de frecuencias con relación a la media. Se obtiene al calcular la raíz cuadrada de la varianza obteniendo como resultado unidades lineales, en tanto que la varianza muestral se mide en unidades cuadradas. La varianza de la muestra se denota[41]:

$$s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1} \quad (2)$$

La desviación estándar de la muestra, es la raíz cuadrada positiva de s^2 , es decir:

$$s = \sqrt{s^2} \quad (3)$$

✓ **Valor RMS**

El valor medio cuadrático o RMS por sus siglas en inglés *root mean square* es un valor estadístico que representa la magnitud de una cantidad de variables. Se puede obtener el valor rms de una serie de valores discretos o para funciones de variables continuas. Su denotación matemática es de la siguiente manera[42]:

$$Valor_{rms} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} \quad (4)$$

Donde $x_1^2 + x_2^2 + \dots + x_n^2$ representan cada dato de la muestra elevado al cuadrado y N representa la cantidad total de datos de la muestra.

✓ **Curtosis**

Es una medida estadística de forma, que cuantifica una distribución de probabilidad, si esta es esbelta o aplanada. Para entender el resultado o valor de curtosis se toma como referencia la distribución Normal de probabilidad, donde su curtosis es igual a 3. Si en una distribución su curtosis es mayor que la Normal quiere decir que su parte central es más puntiaguda y si el valor de curtosis es menor su forma será más plana y es más probable encontrar valores alejados de la media[43].

$$Curtosis = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^4}{Ns^4}} - 3 \quad (5)$$

La curtosis es el promedio a la cuarta potencia de la diferencia entre cada elemento del conjunto y la media, dividido entre la desviación típica elevado a la cuarta potencia y multiplicado por el número de datos de la muestra.

✓ **Factor de cresta**

El valor de factor de cresta nos indica cambios en el patrón de la señal debido a fuentes de vibración. Se define como la relación entre el valor pico de la señal y el valor RMS[44]:

$$FC = \frac{Valor\ Pico}{RMS} \quad (6)$$

CAPÍTULO 2: METODOLOGÍA Y ARQUITECTURA DEL SISTEMA

En esta sección se hablará sobre la metodología utilizada para implementar el sistema de detección de fallos en máquinas rotativas, mediante adquisición de señales acústicas. El diseño del sistema se basa en una arquitectura SOA con el propósito de volver a sus componentes interoperables y mantener el sistema abierto hacia nuevas tecnologías que en un futuro se podrían implementar. En la sección 2.1 se detalla el uso de tecnologías y algoritmos utilizados para el proceso de adquisición, almacenamiento y procesamiento de la señal acústica para el sistema de detección de fallos propuesto. Mientras que en la sección 2.2 se analiza detalladamente los componentes y servicios que componen el sistema.

2.1 METODOLOGIA PARA LA DETECCIÓN DE FALLOS

El uso de señales acústicas para detección de fallos es una metodología no invasiva y permite un diagnóstico eficaz de posibles fallos del estado de la máquina. Las máquinas rotativas producen un sonido específico de funcionamiento. Un síntoma de fallo puede alterar las señales acústicas que genera y que en ocasiones no puede ser detectado por los técnicos de mantenimiento. En esta sección se propone una metodología para identificar posibles fallos mediante adquisición de señales

sonoras, almacenamiento de señales y análisis estadístico para inferir el estado de la máquina.

Para el proceso se propone los siguientes pasos:

- Adquisición de datos
- Almacenamiento de las señales
- Procesamiento de las señales
- Inferencia del estado de la máquina

A continuación, se detalla cada uno de ellos.

➤ Adquisición de datos

Las señales acústicas pueden ser captadas por transductores de audio como por ejemplo micrófonos omnidireccionales o sensores de audio como se puede observar en la Figura 6. Mientras mejor sean las características del dispositivo de adquisición, mejor será la señal adquirida, debido a que permite percibir un mayor rango de frecuencias. Para esta fase de la metodología se propone el uso del micrófono que trae incorporado el dispositivo móvil. Gracias al avance tecnológico los smartPhones más utilizados disponen de un micrófono con altas prestaciones que posibilita captar las suficientes frecuencias de audio para el proceso de inferencia del estado de una máquina.

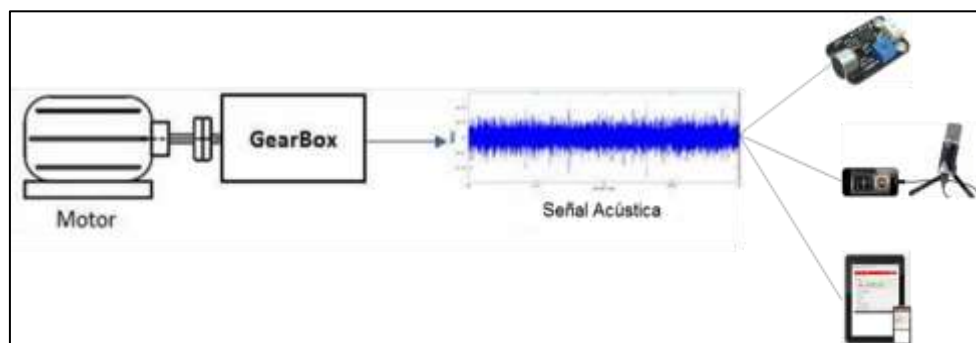


Figura 6 Adquisición de señales acústicas.

Fuente: Autor

➤ Almacenamiento de las señales

Cuando se obtiene un conjunto de señales de audio en condiciones normales de funcionamiento y se utilizan como base para establecer si una señal distinta pertenece o no pertenece a dicho conjunto, se denominan vectores de soporte. Cada señal almacenada en el sistema de detección de fallos de una máquina rotativa se

guarda como un vector de soporte. Una forma gráfica de representar dicho vector se observa en la Figura 7 donde cada punto representado en el plano cartesiano hace referencia a una señal.

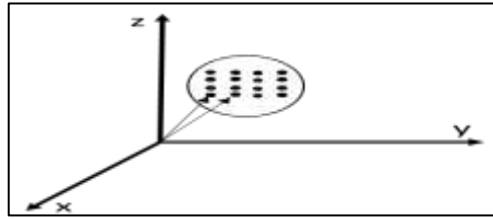


Figura 7 Representación de vectores de soporte en el plano x,y,z.

Fuente: Autor

➤ Procesamiento de las señales

Se procesan los vectores de soporte en condiciones normales del estado de la máquina, con el objetivo de generar un perfil de histograma común, que sirva como referencia de un estado de funcionamiento normal de la máquina. Este proceso se lleva a cabo concatenando cada señal almacenada y obteniendo un solo vector de soporte total. Además, se obtiene el valor de la media y desviación estándar del vector de soporte total, que indicará un valor promedio y el valor de dispersión de los datos que contiene dicho vector. En la Figura 8 se puede observar el diagrama de flujo del procesamiento de la señal, el cual requiere como entrada el directorio de la carpeta donde se encuentran almacenados los archivos de audio grabados en un estado de funcionamiento normal de la máquina.

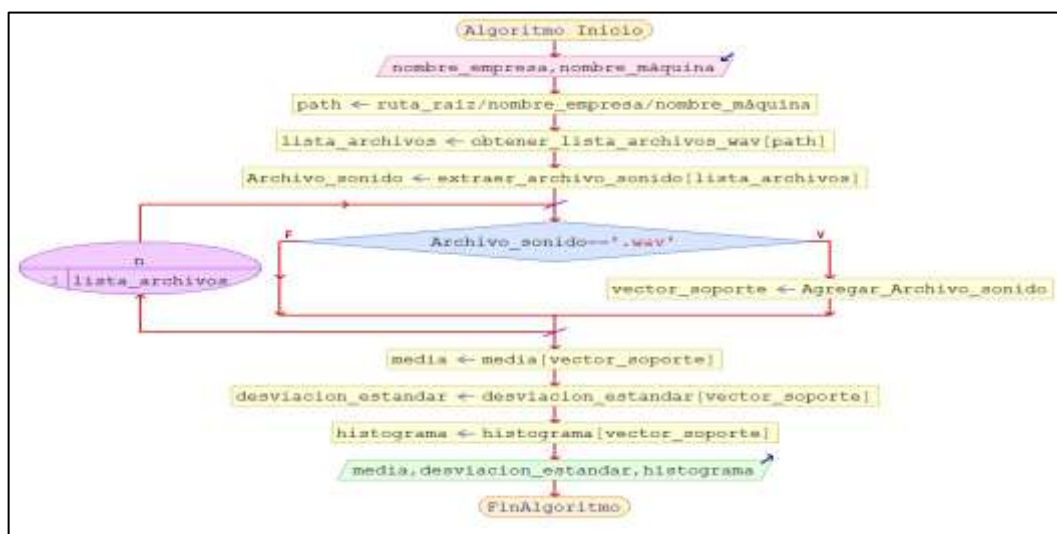


Figura 8 Flujograma del procesamiento de vectores de soporte.

Fuente: Autor

➤ Inferencia del estado de la máquina

Esta etapa se divide en dos: la primera es analizar por separado la señal de audio con presunción de fallo y la segunda es realizar una comparación estadística y obtener la inferencia del estado de la máquina entre el histograma de la señal con presunción de falla y el histograma de las señales de audio almacenadas en estado de funcionamiento normal de la máquina. En la primera etapa se empieza por tomar una nueva señal de la máquina cuando se presume un fallo. Similar al proceso anterior se obtiene el histograma y los valores estadísticos como la media y desviación estándar de la señal adquirida con presunción de fallo. El diagrama de flujo del procesamiento de la señal adquirida se observa en la Figura 9, donde requiere como entrada la ruta del archivo.

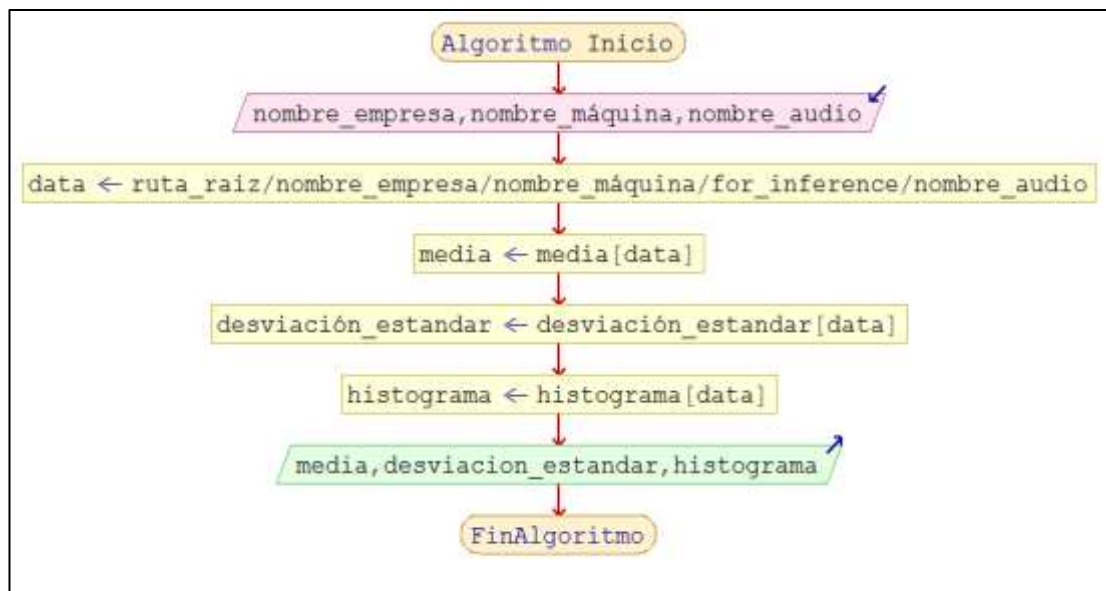


Figura 9 *Flujograma del procesamiento de la señal de audio con presunción de falla.*

Fuente: Autor

En la segunda etapa se realiza la comparación entre el histograma de los vectores de soporte y de la nueva señal capturada, con el objetivo de obtener un factor de similitud. Si la variación es considerable indicará un fallo en la máquina. Para la comparación entre histogramas se utiliza el algoritmo de intersección del histograma propuesto por Swain en [45]. Se interponen los histogramas del vector de soporte total y el histograma de la nueva señal capturada para obtener el área de intersección. El valor del área es normalizado entre 0 y 1. Cuando éste valor sea cercano a 1 indicará un normal funcionamiento de la máquina y si es cercano a 0 un fallo inminente en la máquina.

2.2 ARQUITECTURA DEL SISTEMA BASADO EN SOA

Para la consecución de la metodología expuesta anteriormente se plantea una arquitectura orientada a servicios, cuya arquitectura está formada por componentes y servicios que pueden ser consumidos por los clientes. Basado en una lógica de negocios y conceptos de abstracción vistos en la sección 1.1 cada servicio recibe una entrada y devuelve parámetros de salida de interés para el cliente. En la Figura 10 se expone la arquitectura propuesta para el sistema de detección de fallos de una máquina rotativa. Los componentes principales son: clientes, servidor web y servidor de cómputo o procesamiento que a continuación serán analizados a fondo.



Figura 10 Arquitectura del sistema aplicación móvil para detección de fallas.

Fuente: Autor

➤ Clientes

Los clientes son dispositivos móviles equipados con una aplicación diseñada como interface del usuario y medio de captura de señales de audio. Mediante la aplicación se puede observar los resultados del procesamiento de la señal y la inferencia sobre el estado de la máquina. La arquitectura basada en servicios permite que varios clientes puedan hacer uso del sistema al mismo tiempo.

➤ Servidor web

Este componente se caracteriza por alojar servicios que aportan al funcionamiento del sistema y además permite establecer un tipo de comunicación entre el dispositivo móvil y el servidor de cómputo. Los servicios alojados en este

componente son: Servicio de presentación, servicio de almacenamiento, servicio de directorios y servicio de registro e inicio de sesión de usuario.

- **Servicio de presentación:** Este servicio extrae la información del servidor de cómputo para luego enviarla hacia la aplicación móvil del usuario. El servicio recibe la petición del usuario por medio de la aplicación móvil. La petición realizada se procesa en el servidor de cómputo y se devuelve la información procesada hacia el servidor web y este a su vez responde con dicha información hacia la aplicación móvil para que el usuario pueda visualizarla. El flujo del proceso para un mejor entendimiento se observa en la Figura 11.

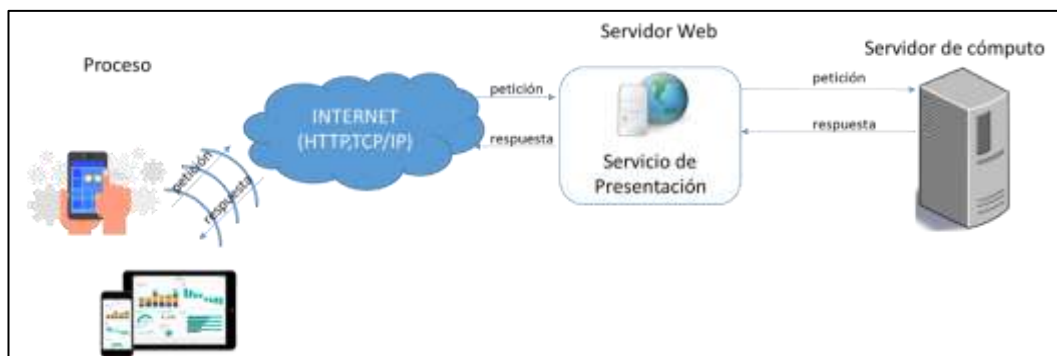


Figura 11 Servicio de presentación de la arquitectura SOA del sistema.
Fuente: Autor

- **Servicio de directorios:** Con el fin de almacenar los archivos de audio de manera organizada, el servicio genera directorios en base al nombre de la empresa y maquinaria que desea analizar el usuario. En la Figura 12 se observa el árbol de directorios dentro del servidor de cómputo. Este servicio requiere como entrada el nombre de la empresa y nombre de la máquina que se desea analizar. Con la información anterior, se crean subdirectorios donde se almacenará la información o archivos de audio grabados.



Figura 12 Árbol de directorios de almacenamiento en el servidor de cómputo.
Fuente: Autor

- **Servicio de almacenamiento:** Este servicio permite que los archivos de audio grabados en el dispositivo móvil se envíen hacia el servidor de cómputo. El servicio de almacenamiento necesita como entrada el directorio donde el cliente desea guardar la información y el archivo de audio. Esta información es comprimida en paquete de datos y es enviada hacia su destino usando el protocolo HTTP a través de redes de comunicación de datos. En la Figura 13 se observa el proceso de envío y almacenamiento del archivo de audio en el servidor de cómputo.



Figura 13 Servicio de almacenamiento de la arquitectura SOA del sistema.

Fuente: Autor

- **Servicio de registro e inicio de sesión de usuario:** Dispone de una base de datos mysql en el servidor web para el registro e inicio de sesión de usuario. Como se observa en la Figura 14, este servicio permite almacenar información con respecto al cliente como: sus datos personales y empresa al cual pertenece. Para el inicio de sesión el servicio requiere datos de nombre de usuario y contraseña. El servicio verifica si los datos se encuentran almacenados y responde con un verdadero o falso hacia la aplicación. En base a la respuesta generada se permitirá el acceso al resto de servicios del sistema.

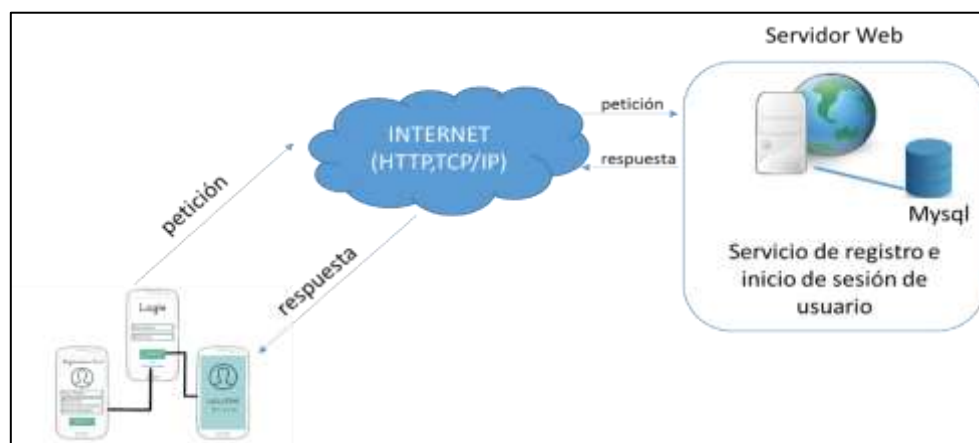


Figura 14 Servicio de registro e inicio de sesión de usuario.

Fuente: Autor

CAPÍTULO 3: DESARROLLO DEL SISTEMA

En este capítulo se proyecta el desarrollo técnico del sistema en función de la metodología y arquitectura propuestas anteriormente. Además, se hablará sobre el software que se utiliza en cada componente de la arquitectura propuesta, las configuraciones necesarias para la comunicación entre servidores y los algoritmos utilizados para los diferentes servicios del sistema de detección de fallos de una máquina rotativa.

3.1 APLICACIÓN MOVIL

El cliente dispone de una aplicación para consumir los servicios del sistema de detección de fallos de una máquina rotativa. La aplicación móvil fue desarrollada bajo el enfoque de programación nativa (véase la sección 1.2). La plataforma utilizada para el desarrollo de la aplicación móvil del sistema propuesto es *Android Studio™* (versión 2.3.3). El sistema operativo Android es el más utilizado por dispositivos móviles inteligentes que se encuentran en el mercado de hoy en día. En la Figura 15 se puede observar un diagrama con el contenido y el flujo de las opciones que el usuario dispone en la aplicación móvil.

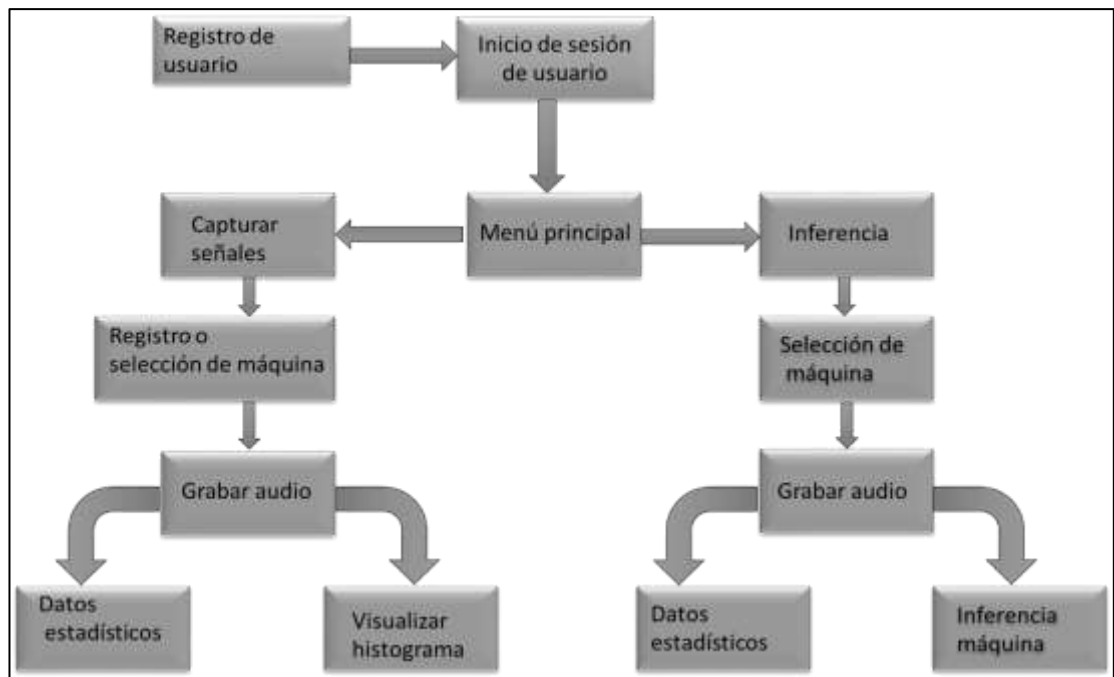


Figura 15 Diagrama de bloques con el contenido de la app móvil.

Fuente: Autor

Los componentes principales dentro de la aplicación móvil son:

- Registro o Inicio de sesión de usuario
- Menú Principal: Capturar señales, Inferencia
- Grabación de Audio

➤ Registro o Inicio de Sesión de Usuario

La aplicación móvil contiene una interfaz para que el usuario pueda ingresar sus datos personales tanto en la fase de registro como la de inicio de sesión. Toda la información es almacenada en un gestor de base de datos MYSQL disponible en el servidor web. Para el envío y recepción de datos entre el dispositivo móvil y el servidor web se utiliza el protocolo HTTP. El servidor estará a espera de conexiones por el puerto 80, para ello se dispone de una dirección IP pública proporcionada por el grupo de investigación GIDTEC. Este tipo de comunicación utiliza servicios Web Restfull con mensajería basada en JSON lo cual permite establecer un enlace entre el usuario del dispositivo y el servidor. En la aplicación móvil se utiliza la librería Volley desarrollada por Google que optimiza el envío de peticiones HTTP desde una aplicación móvil hacia los servidores remotos. En [46] se describe a Volley como librería encargada de simplificar procesos de peticiones a servicios web, evitando crear código repetitivo para manejar tareas asíncronas por cada petición. En la

Figura 16 se puede observar el diagrama de flujo del proceso de registro e inicio de sesión de usuarios, si el usuario no dispone de una cuenta deberá registrarse para tener acceso a los servicios que brinda la aplicación.

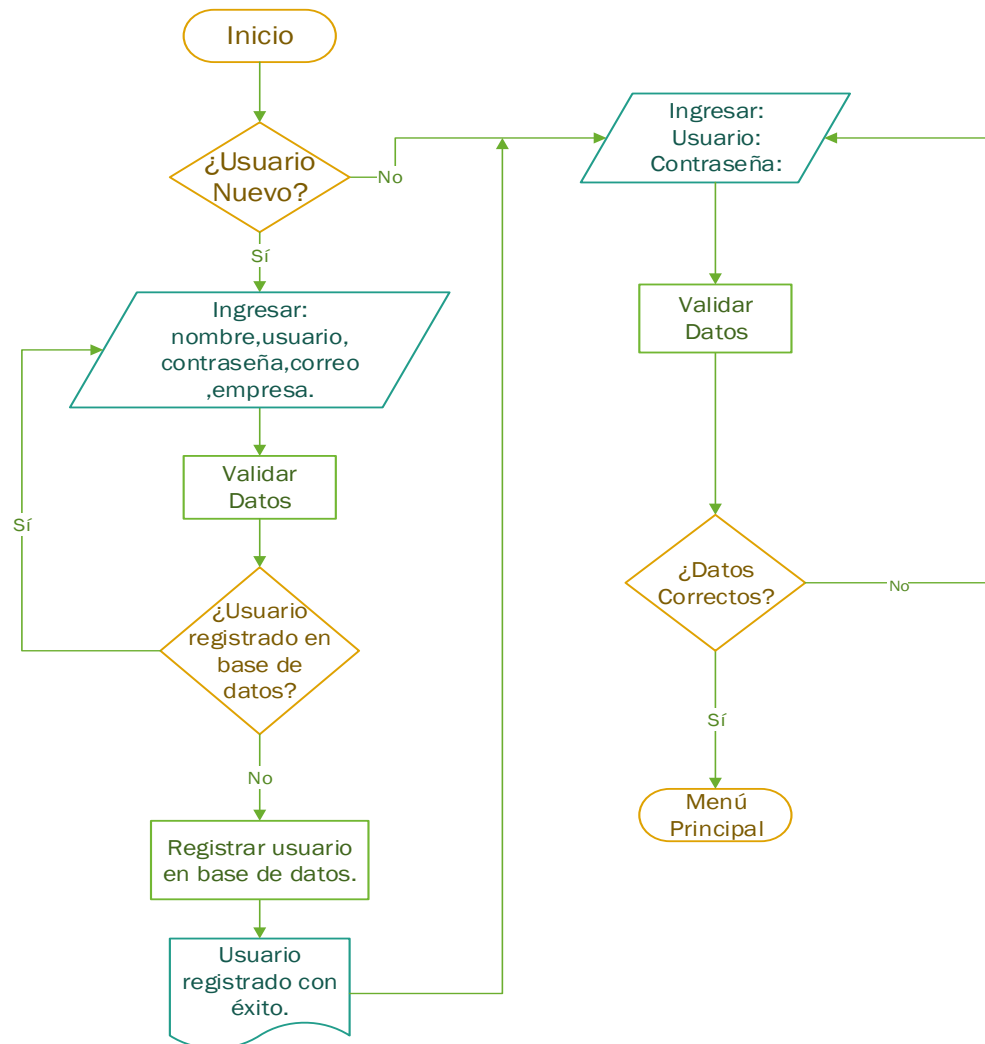


Figura 16 *Flujograma del proceso de inicio y registro de usuario.*

Fuente: Autor

Cabe mencionar que en el proceso de registro automáticamente se crea el directorio en el servidor de cómputo con el nombre de la empresa. En la Figura 17 se observa el flujograma del proceso.

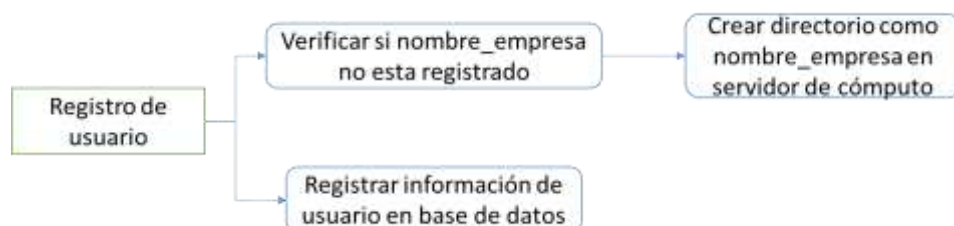


Figura 17 *Flujograma del proceso para crear el directorio raíz en el servidor de cómputo.*

Fuente: Autor

➤ Menú Principal: Capturar señales, Inferencia

La interfaz de la aplicación móvil para el menú principal consta de dos opciones que son: Capturar señales e Inferencia.

✓ Capturar señales

Esta opción le permite al usuario generar las firmas de audio de estado normal de la máquina. Este proceso sigue un flujo que empieza por el registro de la máquina que desea generar las firmas de audio. Una vez registrado se procede a grabar el audio y enviarlo hacia el servidor de cómputo para su respectivo almacenamiento en el directorio al que pertenece. Los directorios para cada máquina se generan cuando el usuario registra el nombre de la máquina. En el caso de que la máquina ya se encuentre registrada se desarrolló una sección para que el usuario pueda elegir con cuales de las máquinas antes registradas desea trabajar. En la Figura 18 se observa el flujograma del proceso registro de máquina.

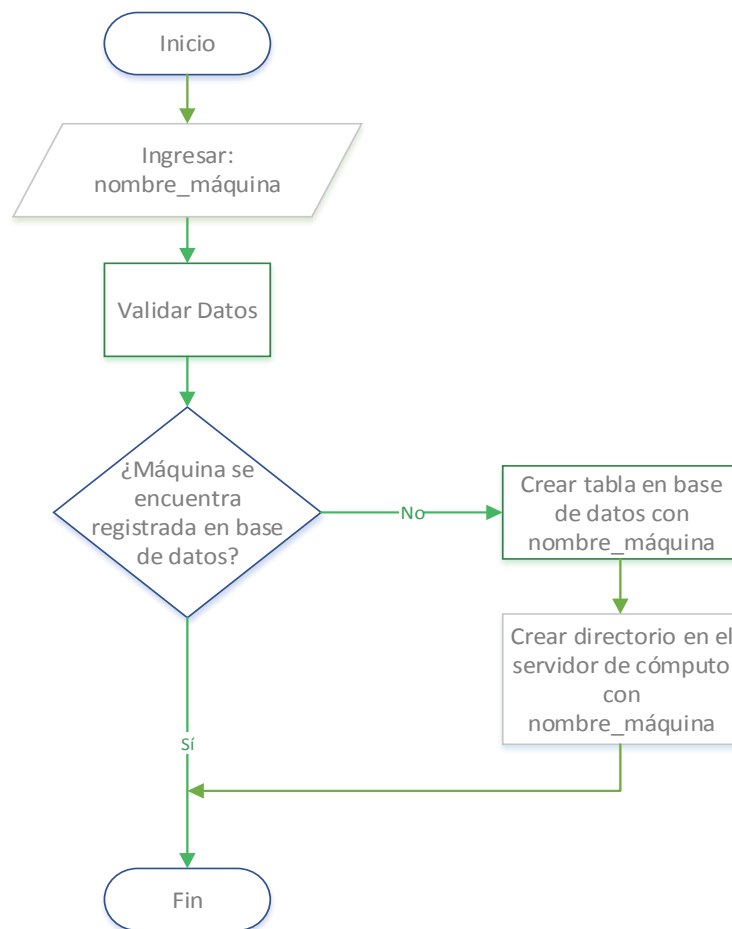


Figura 18 *Flujograma del proceso registrar máquina.*

Fuente: Autor

✓ Inferencia

La interfaz de la aplicación móvil para esta etapa consta de una sección para elegir una de las máquinas registradas en la sección anterior de cual se desea obtener el estado. El usuario podrá seleccionar la máquina del registro de almacenamiento para su respectivo análisis, entonces se crea un nuevo directorio dentro de la carpeta correspondiente a la máquina seleccionada con el nombre “for_inference” para almacenar los audios tomados en este proceso. El objetivo es no mezclar la información almacenada anteriormente y realizar una comparación estadística entre el audio tomado en ese instante y los audios almacenados con anterioridad. En la Figura 19 se representa el proceso selección de máquina mediante un diagrama de flujo.

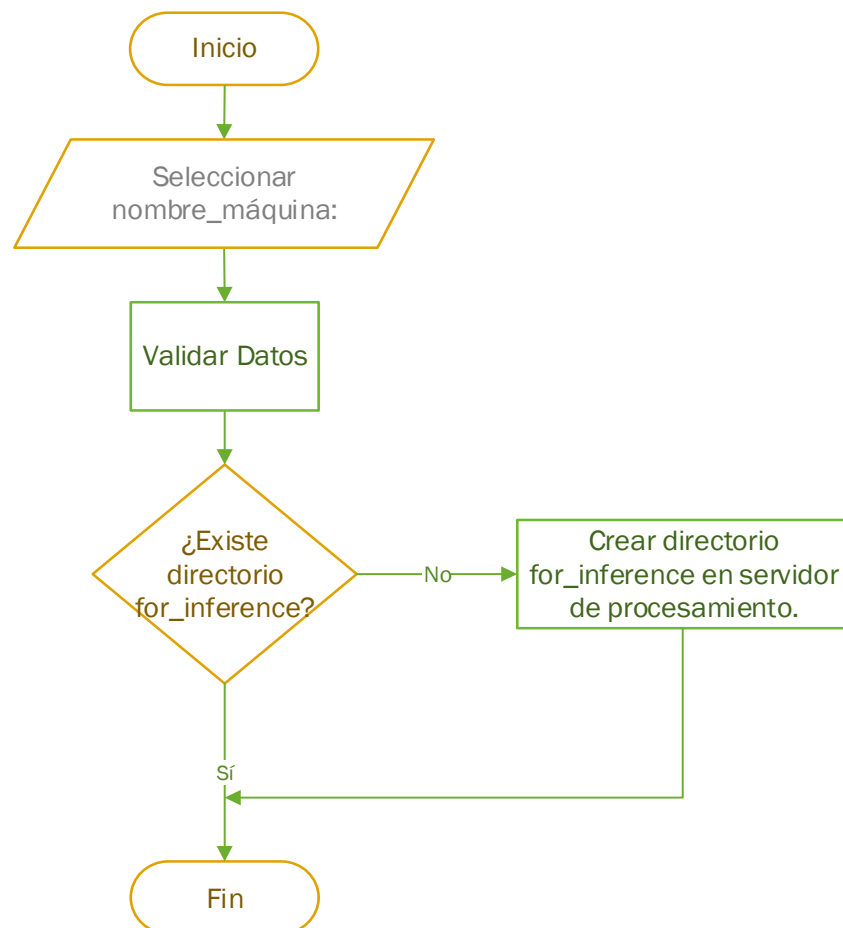


Figura 19 *Flujograma del proceso seleccionar máquina.*

Fuente: Autor

✓ Grabación de Audio

Para el procesamiento de audio es importante conocer las características del micrófono incorporado en el dispositivo móvil, por lo general están diseñados para un propósito esencial que es la conversación humana. En [47] se menciona que teléfonos y tabletas inteligentes en la actualidad pueden tener frecuencias de muestreo de hasta 48kHz y resolución de 24 bits. Sin embargo, dependerá también del chasis del dispositivo donde se aloje toda la electrónica necesaria para su funcionamiento. La señal de audio captada por el micrófono es analógica y para el procesamiento es necesario convertirla en una señal digital. La frecuencia de muestreo va a depender de las aplicaciones que se quieran dar con la señal muestreada y las características de la tarjeta de sonido del equipo electrónico.

Para la adquisición y reproducción de audio existe una frecuencia de muestreo estandarizada en 44100 Hz o 44.1kHz debido a que el rango de voz humana está entre los 20 Hz y 20.000 Hz. En base al teorema de Nyquist la frecuencia de muestreo 44.1kHz es el doble de la frecuencia máxima percibida por los seres humanos. Otra de las características es la profundidad de bits, pues representa la resolución de cada muestra de audio, en dispositivos móviles se dispone de 8,16 y 24 bits.

Android Studio trae incorporado en sus librerías la clase `AudioRecord` que se encarga de administrar los recursos de audio en aplicaciones Java para grabar contenido de audio desde el hardware de entrada del dispositivo. Esta clase presenta un método mutable para la lectura de datos entre ellos tenemos:

- `read(byte[], int, int)`,
- `read(short[], int, int)` y
- `read(ByteBuffer, int)`

La elección del método se basa en el formato de almacenamiento de los datos de audio conveniente para el usuario de la clase `AudioRecord` los cuales pueden ser `ENCODING_PCM_8BIT`, `ENCODING_PCM_16BIT` y `ENCODING_PCM_FLOAT`.

La clase `AudioRecord` está estructurada de la siguiente manera:

```
AudioRecord (int audioSource,  
  
             int sampleRateInHz,  
  
             int channelConfig,  
  
             int audioFormat,  
  
             int bufferSizeInBytes)
```

Donde:

- `int audioSource`: Es la fuente o entrada de audio.
- `int sampleRateInHz`: Es la frecuencia de muestreo expresada en Hz.
- `int channelConfig`: Describe la configuración de los canales de Audio.
- `int audioFormat`: El formato o profundidad de bits.
- `int bufferSizeInBytes`: Tamaño del buffer en Bytes donde se escriben los datos de audio durante la grabación.

Los datos correspondientes para grabar audio desde la aplicación móvil son los siguientes:

- ❖ Fuente de audio es el MIC incorporado en el dispositivo.
- ❖ Frecuencia de muestreo 44.1KHz.
- ❖ Configuración del canal de audio `CHANNEL_IN_MONO`.
- ❖ Formato de audio es de 16 bits `ENCODING_PCM_16BIT`.
- ❖ Tamaño del buffer es de 4096 Bytes.

El formato de audio digital donde se almacena la información es de tipo WAV debido que no comprime los datos y acepta archivos de audio mono y estéreo a diversas resoluciones y velocidades de muestreo. Además, .wav es compatible con varios codec y utiliza el formato no comprimido PCM (*Pulse Code Modulation*), brindando mayor calidad al archivo generado debido a que contiene toda la información que sale del convertidor ADC.

3.2 SERVIDOR WEB

Dentro de este componente se encuentran los servicios de registro e inicio de sesión de usuario y los scripts para la comunicación entre el dispositivo móvil y el servidor de cómputo. El servidor web esta implementado en apache sobre la plataforma Centos de **Linux**®. Para brindar seguridad en transmisión de datos entre servidores es necesario encriptar la vía de comunicación, y para ello se utilizó llaves de seguridad privada y pública en el protocolo de comunicación ssh; como se da a conocer en la sección 1.4 del marco metodológico. Este tipo de llaves son códigos de autenticación generados por el sistema operativo y que deben reposar en los directorios de ambos servidores.

Para permitir que el servidor de apache lea información emitida por servidores remotos es necesario utilizar una comunicación segura ssh y deshabilitar SELINUX (Modulo de seguridad para el kernel Linux) del sistema operativo, para ello se procedió a realizar la siguiente configuración:

1. Es necesario cambiar las configuraciones de inicio de sesión para el usuario apache, de forma que solicite autenticación mediante llaves ssh, en el directorio `vi /etc/passwd` se debe realizar el siguiente cambio:

```
De apache: x: 48: 48: Apache: /usr / share / httpd: / bin / nologin
```

```
A apache: x: 48: 48: Apache: /usr / share / httpd: / bin / bash
```

2. Crear una carpeta con el nombre `.ssh` en el directorio: `/usr / share / httpd/` y cambiar el propietario de este directorio al propietario apache. Esta carpeta servirá para alojar las llaves de autenticación.

```
mkdir /usr/share/httpd/.ssh
```

```
chown apache:apache /usr/share/httpd/.ssh
```

3. Iniciar sesión como usuario apache y generar las llaves privadas y públicas, luego copiar al servidor remoto con el que se desea comunicar, para ello ejecutar en la consola los siguientes comandos:

```
su - apache
```

```
cd .ssh
```

```
ssh-keygen -t rsa
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub usuario@IP_cómputo
```

4. Verificar claves de acceso desde el servidor web hasta el servidor remoto mediante el siguiente comando:
`ssh usuario@ IP_cómputo`
5. Y por último deshabilitar selinux para permitir la transmisión de datos entre servidores. Luego de ingresar el comando para desactivar selinux es necesario reiniciar el sistema operativo para que los cambios surjan efecto.
`echo 0 >/selinux/enforce`

➤ **Configuración de base de datos**

Para el servicio de registro o inicio de sesión de usuario, el servidor web dispone de una base de datos desarrollada en MySQL versión 5.6.40. El sistema de gestión de base de datos MySQL se basa en un lenguaje de consulta estructurado y admas se caracteriza por ser una base de datos de código abierto. La base de datos propuesta para el sistema de detección de fallas está compuesta por tablas que contienen información de los datos personales del usuario y tablas con información de las máquinas registradas para el proceso de inferencia de la máquina.

Para almacenar información personal del usuario se desarrolla una sola tabla compuesta por los siguientes campos: nombres; nombre de usuario, contraseña, dirección de correo electrónico y nombre de la empresa a la cual pertenece. En la Figura 20 se puede observar la configuración de los campos y características de cada uno de los datos de usuario.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(45)	YES		NULL	
username	varchar(45)	YES		NULL	
password	varchar(45)	YES		NULL	
email	varchar(45)	YES		NULL	
business	varchar(45)	YES		NULL	

Figura 20 Configuración de base de datos en MySQL.

Fuente: Autor

El resto de tablas que se generen llevaran el nombre de la empresa al que el usuario pertenece y dentro de las tablas generadas se almacenaran las máquinas que se registren para el análisis. Se genera un solo campo en cada tabla y pertenece al nombre de la máquina que el usuario genera. En la Figura 21 se observa el flujograma del proceso.

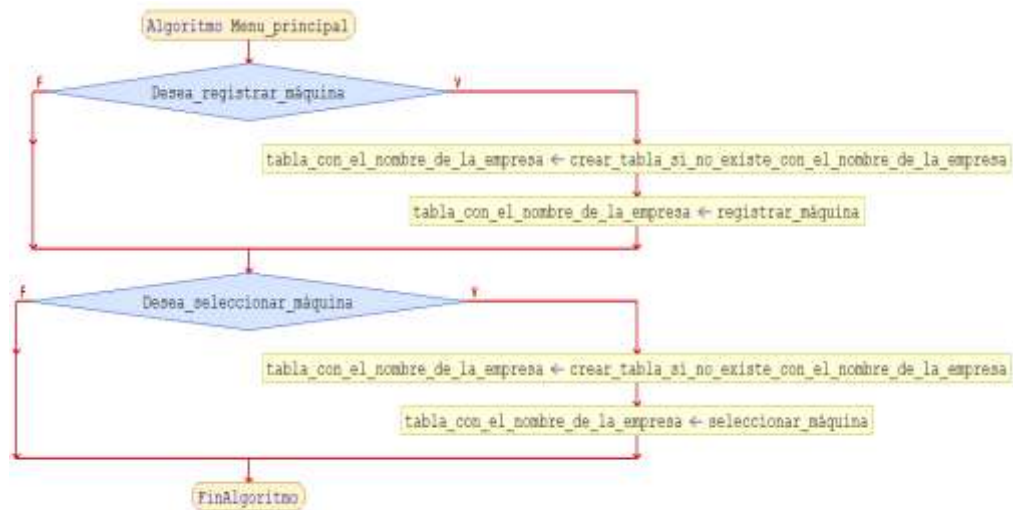


Figura 21 Flujograma con el proceso registrar máquina.

Fuente: Autor

➤ Desarrollo de servicios propuestos

Los servicios que se alojan en el servidor web permiten la comunicación entre la aplicación móvil y el servidor de cómputo. Además, registrar la información personal de los usuarios que accedan al sistema para detección de fallos propuesto. A continuación, se detallan los servicios propuestos en la arquitectura del sistema para el componente servidor web (véase en la sección 2.2):

1) Servicio de presentación

La información que solicita el usuario llega hacia el servidor web, aquí en base a la solicitud realizada se hace un requerimiento al servidor de cómputo mediante comunicación segura ssh2. Luego del procesamiento de la información por parte del servidor de cómputo, responde hacia el servidor web por el mismo medio ssh2. El servidor web mediante mensajería basado en JSON, envía la respuesta hacia la aplicación móvil para que pueda ser visualizada por el usuario. En la Figura 22 se observa el diagrama de flujo del proceso.

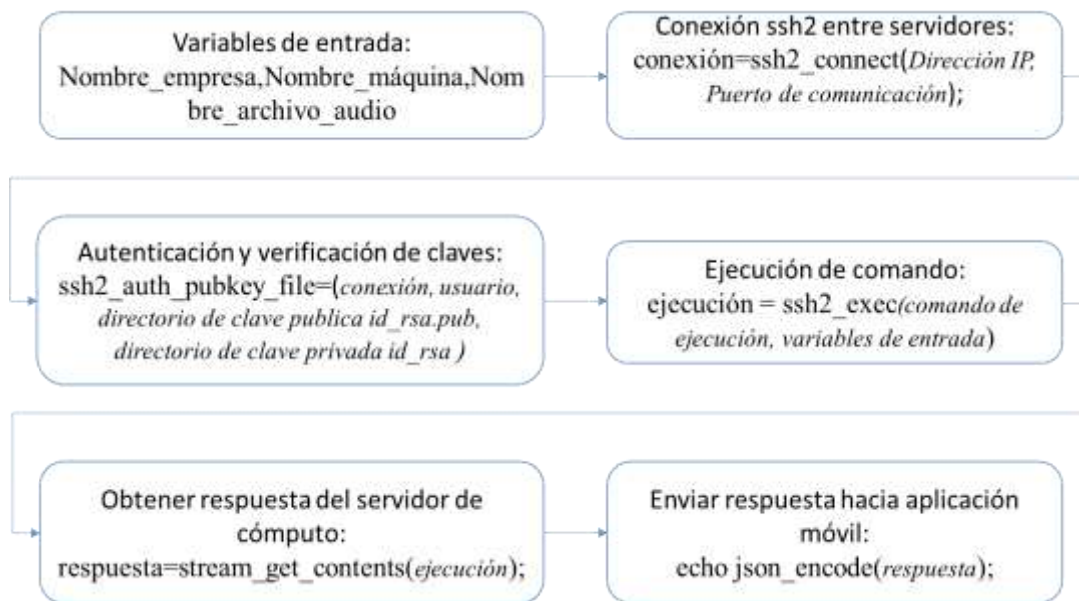


Figura 22 Flujograma del servicio de presentación.

Fuente: Autor

2) Servicio de directorios

El servicio permite crear los respectivos directorios en el servidor de cómputo para mantener organizados los archivos de audio que el usuario desea almacenar y procesar. Este servicio necesita como entrada el nombre de la empresa a la cual pertenece el usuario y el nombre de la máquina que desea analizar. La información es enviada hacia el servidor de cómputo mediante el protocolo de seguridad SSH y además se envía el comando a ejecutar en el servidor remoto para la creación de los directorios. En la Figura 23 se puede observar el flujograma del servicio.

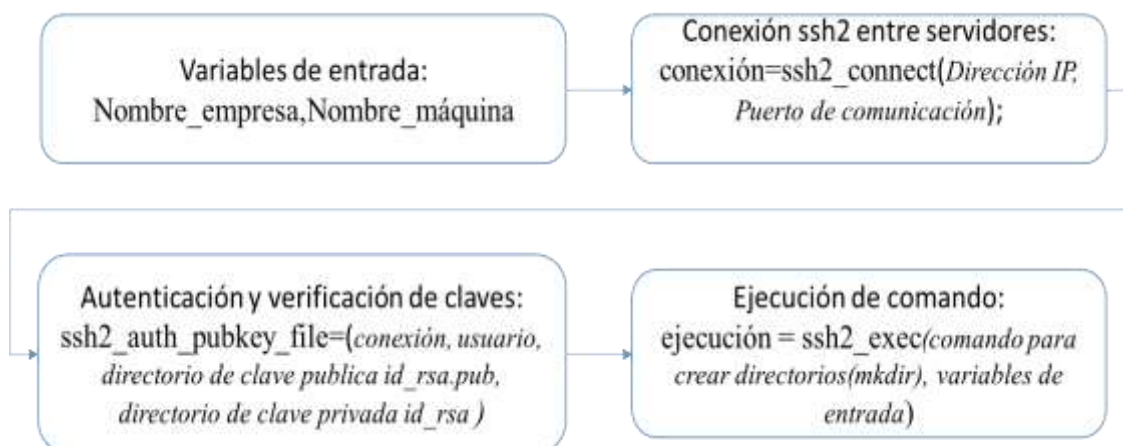


Figura 23 Flujograma del servicio de directorios.

Fuente: Autor

3) Servicio de almacenamiento

El servicio de almacenamiento de la señal de audio grabada por el dispositivo móvil, solicita información de la ruta donde desea almacenar el audio y el archivo generado con formato *.wav*. Mediante el comando *'ssh2_scp_send'* de *PHP* se copia el archivo de audio generado desde la ruta del dispositivo móvil hacia el servidor de cómputo remoto usando el protocolo de comunicación *SCP*⁴. En la Figura 24 se observa el flujo del proceso de almacenamiento de la señal de audio capturada.

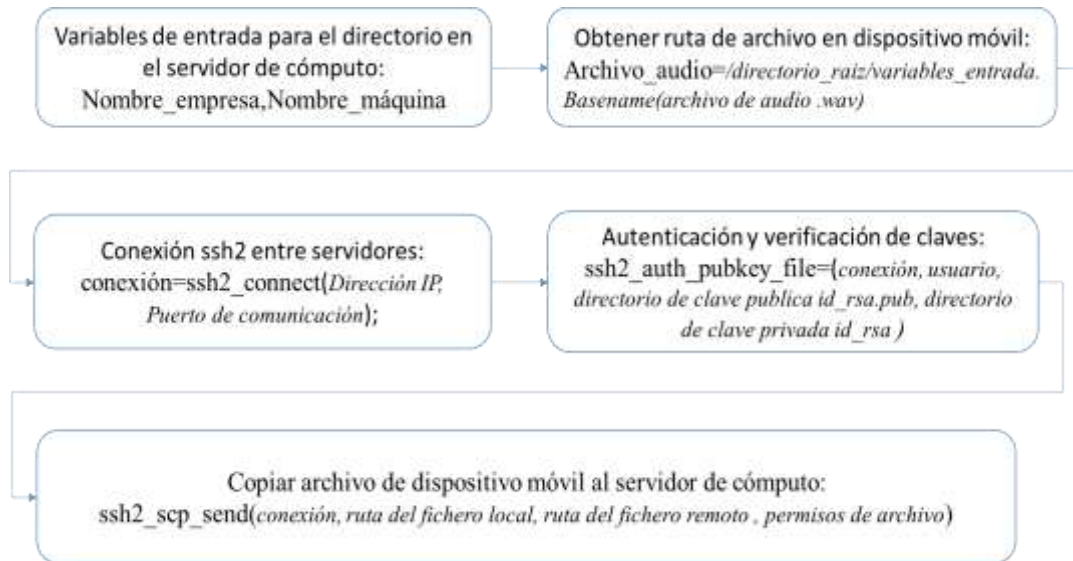


Figura 24 Flujograma del servicio de almacenamiento.

Fuente: Autor

4) Servicio de registro o inicio de sesión de usuario

El servicio dispone de un gestor de base de datos *MySQL* para almacenar la información personal del cliente. Desde la aplicación móvil se envían los datos proporcionados por el usuario hacia el servidor web usando el protocolo *HTTP* mediante el método *POST*. En el servidor web se realiza una conexión hacia el gestor de base de datos y la información es almacenada en tablas. En la Figura 25 se puede observar el flujograma del proceso.

⁴ **Secure Copy Protocol SCP:** Es un medio de transferencia segura de archivos informáticos entre un host local y otro remoto usando el protocolo *SSH*[51].

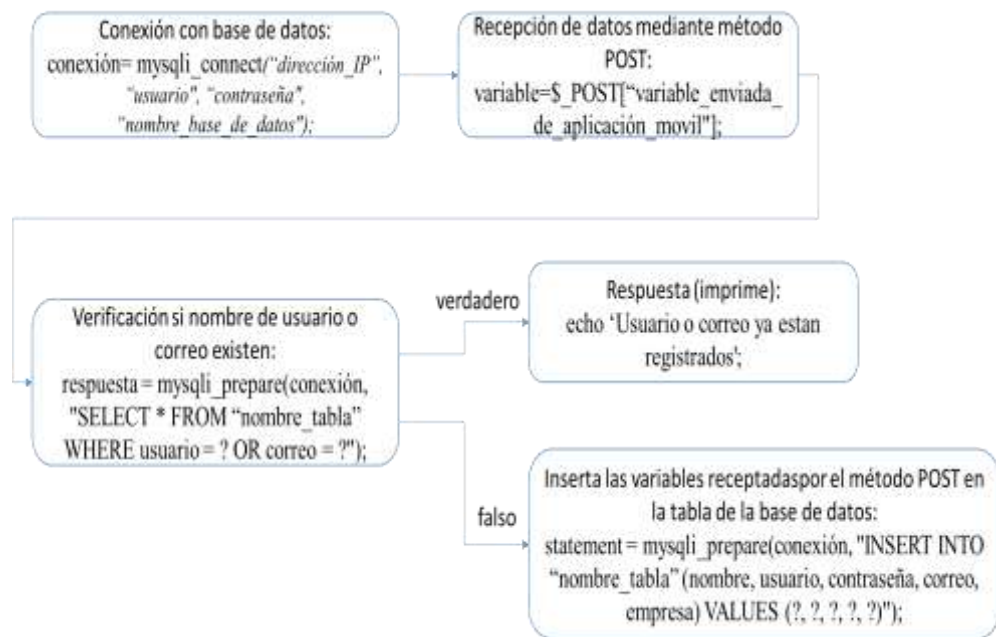


Figura 25 Flujograma del servicio de registro o inicio de sesión de usuario.

Fuente: Autor

Para el inicio de sesión de usuario el proceso es similar, la aplicación móvil envía los datos de nombre de usuario y contraseña al servidor web. En el servidor web se realiza una conexión hacia la base de datos y se realiza una consulta si los datos se encuentran registrados en la tabla de la base de datos. La respuesta es de tipo booleana y es enviada nuevamente hacia la aplicación mediante mensajería JSON. Si la respuesta es verdadera es decir se encuentra registrado, entonces el usuario podrá acceder el resto de servicios que proporciona el sistema.

3.3 SERVIDOR DE CÓMPUTO

Para el tratamiento de la señal de audio almacenada en el servidor de cómputo se utilizó el software de procesamiento Python versión 3.5, por sus características de programación de alto nivel, uso de librerías científicas, y por formar parte de un proyecto de software libre. La aplicación móvil solicitará a Python información estadística y datos para graficar un histograma de la señal de audio que contiene el archivo *.wav*.

Cada script solicita primero información de directorios en el servidor donde se encuentra ubicado el archivo de sonido que se desea procesar. Una vez conocida la ubicación se procede a cargar el archivo para luego extraer información estadística y los datos pertenecientes al histograma generado. La información es devuelta hacia la

aplicación móvil a través del servidor web. Este proceso está apegado a las tendencias de SOA debido a que los scripts necesitan solamente las entradas solicitadas independientemente de la lógica de la aplicación móvil o servidor web. Cada script procesa datos y devuelve información perteneciente a un servicio. Se desarrollaron 4 scripts los cuales son: procesar.py, process_inf.py, process_inf_sound.py, intersec.py. A continuación, se especifica el desarrollo de cada uno de los programas:

El script procesar.py devuelve información estadística y datos del histograma del archivo de audio almacenado como señal de la máquina en condiciones normales. El flujograma de este proceso se puede observar en la Figura 26. Para la extracción de datos estadísticos se utiliza librerías de Python tales como numpy, sys, scipy.stats, matplotlib.

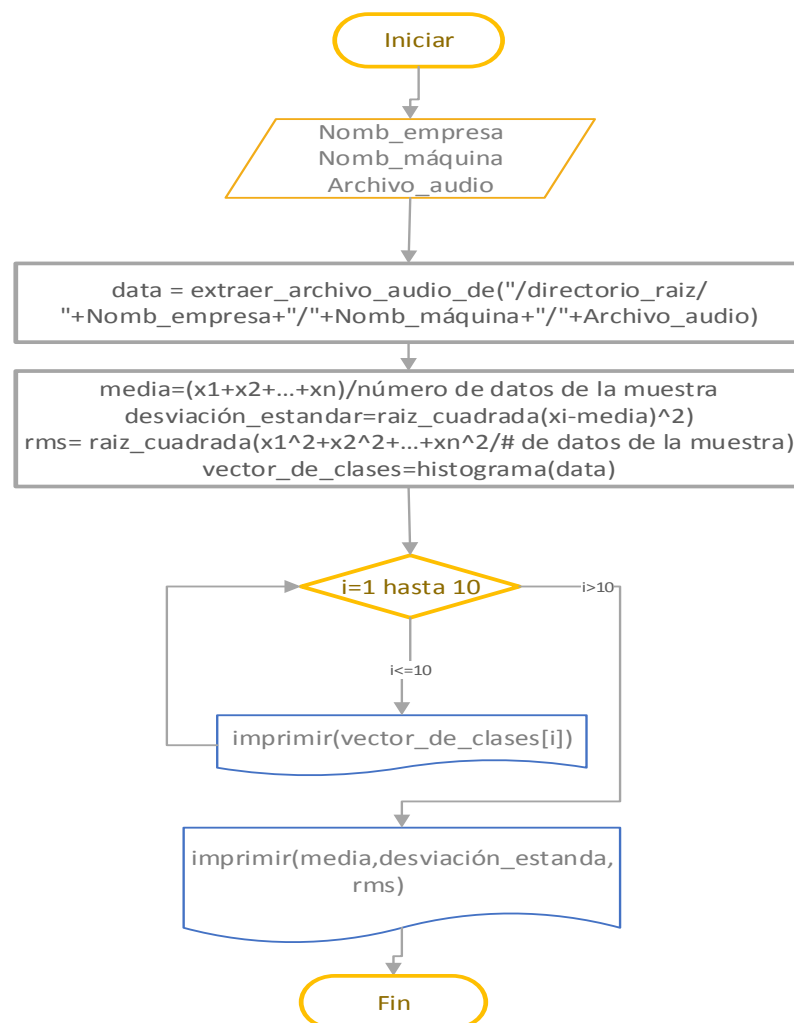


Figura 26 *Flujograma del script de python procesar.*

Fuente: *Autor*

El script `process_inf.py` devuelve información estadística y datos de un histograma del vector de soporte total en estado de funcionamiento normal de la máquina. El flujograma de este proceso se puede observar en la Figura 27.

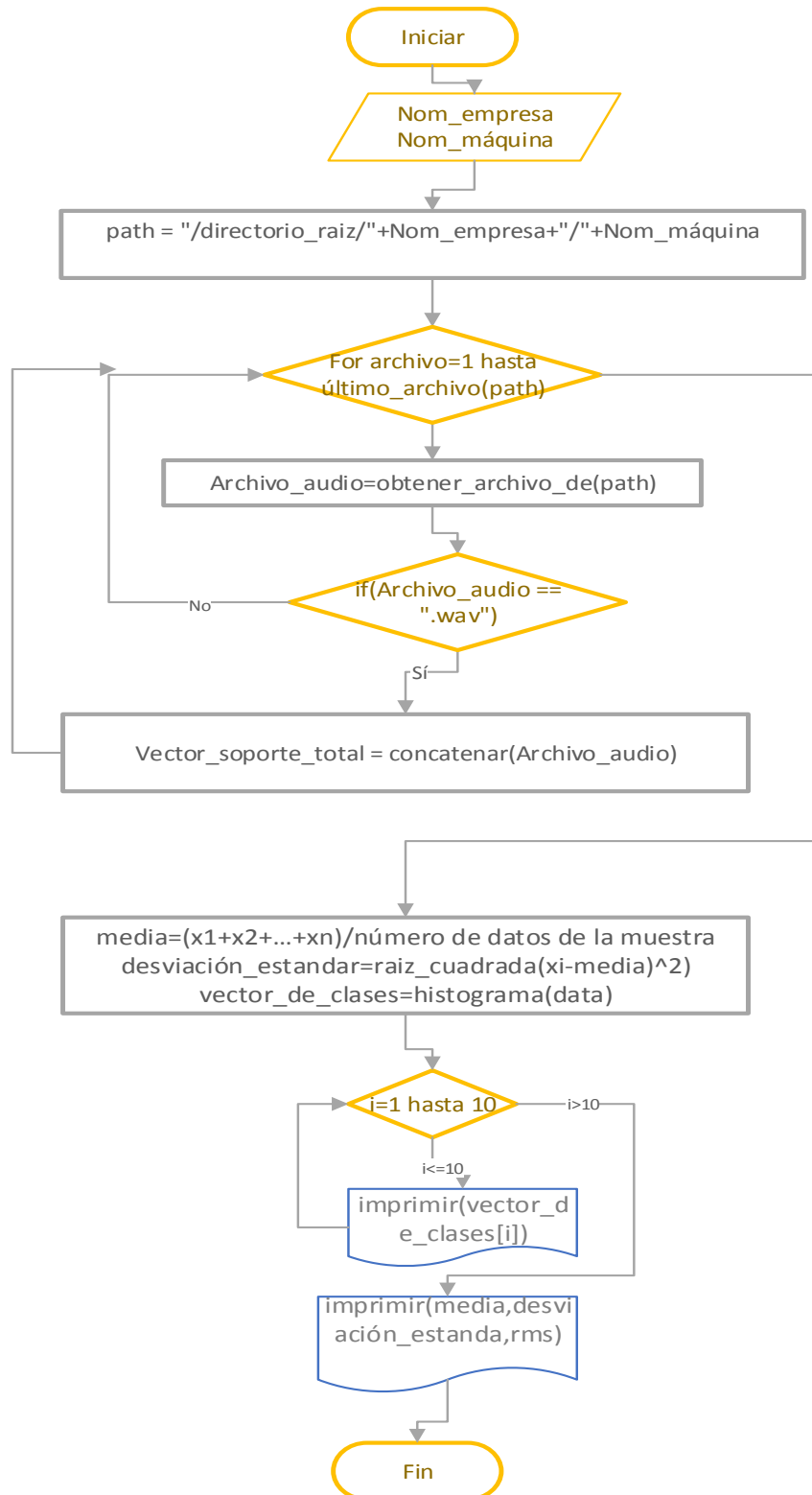


Figura 27 Flujograma del script de python `process_inf`.

Fuente: **Autor**

El script `process_inf_sound.py` procesa el archivo de audio con presunción de fallos, este archivo se encuentra alojado en una carpeta denominada *'for_inference'* dentro del árbol de directorios de la empresa a la cual pertenece el usuario. Se imprime en la aplicación móvil los datos estadísticos como: la media, desviación estándar y además de información del histograma generado. El flujograma es similar al utilizado en el script `procesar.py` donde únicamente cambian las variables de entrada. Por último, el script `intersec.py` solicita información de los directorios de almacenamiento de las señales de audio grabadas: en condiciones de funcionamiento normal y con posible fallo en el funcionamiento de la máquina. Con el fin de realizar una comparación estadística e inferir el estado actual de la máquina analizada. Para este proceso se hace uso de los flujogramas anteriormente utilizados para el procesamiento de audio tanto del vector de soporte total como de la señal con presunción de fallo.

CAPÍTULO 4: FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

En esta sección se presenta el funcionamiento de la aplicación móvil para detección de fallos (*por sus siglas en inglés fault detection application FDAPP*). Se detalla cuáles son los pasos que debería seguir el usuario para consumir los servicios propuestos por el sistema. Como ya se ha venido mencionando el sistema está compuesto por dos partes relevantes: el almacenamiento de señales en estado de funcionamiento normal de la máquina y, por otro lado, la inferencia de la condición de la máquina. Además, se muestra el servicio de registro o inicio de sesión de usuario y la creación de directorios específicos para cada usuario o empresa registrado.

4.1 INTERFAZ DE FDAPP

Para realizar las respectivas pruebas de funcionamiento de FDAPP, se instaló la aplicación móvil en un Smartphone Samsung J7[®] con la versión de Android 7.0. Una vez instalada la aplicación en el dispositivo móvil se la identifica con un icono en el menú principal como se puede observar en la Figura 28.

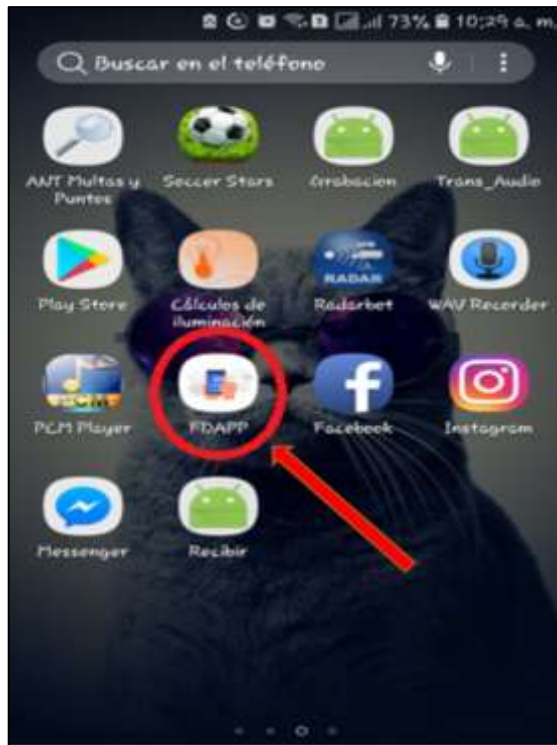


Figura 28 Icono de la aplicación FDAPP.
Fuente: *Autor*

Una vez que el usuario ingresa a la aplicación, tiene dos opciones: registrarse si es el caso de un nuevo usuario o si ya está registrado ingresar los datos de nombre de usuario y contraseña. En la Figura 29 se observa la interfaz de las opciones de registro o inicio de sesión.

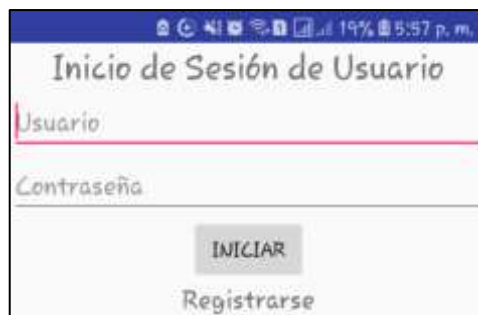


Figura 29 Registro o inicio de sesión.
Fuente: *Autor*

4.2 REGISTRO O INICIO DE SESIÓN DE USUARIO

El registro de usuario es primordial para que el usuario pueda acceder a los servicios que ofrece el sistema. Cabe mencionar que la información ingresada sirve como referencia para la creación de directorios en el servidor de cómputo. Los datos que se solicitan son: nombres personales, nombre de usuario, contraseña, confirmación de contraseña, nombre de la empresa a la cual pertenece y un correo

electrónico. Esta opción cuenta con funciones de validación de datos, es decir si el usuario deja casilleros en blanco, las contraseñas no coinciden, registra un nombre de usuario o correo ya registrado, la aplicación no le permite realizar el registro con éxito y le muestra un mensaje de alerta. El usuario deberá realizar los cambios correspondientes para poder continuar con el registro. En la Figura 30 se puede observar el proceso de registro de usuario. Cuando ya se encuentre registrado, el usuario podrá acceder al sistema ingresando el nombre de usuario y contraseña en los campos correspondientes como se observa en la Figura 29.



Figura 30 Registro de usuario y validación de datos ingresados en caso de espacios en blanco, contraseñas incorrectas e usuario ya registrado.

Fuente: *Autor*

4.3 MENÚ PRINCIPAL

Dentro del menú principal el usuario tiene dos opciones: la primera es capturar señales de audio en estado normal de la máquina y la segunda obtener la inferencia del estado de la máquina mediante comparación de histogramas. A

continuación, se detalla específicamente los pasos a seguir para el consumo de dichos servicios:

➤ **Captura de señales de audio en estado normal de la máquina**

Esta opción le permite al usuario lo siguiente: registrar el nombre de la máquina que desea analizar, generar una base de datos con los nombres de las máquinas registradas y mostrar la base de datos generada. Una vez que el usuario registre el nombre de la máquina o a su vez si ya se encuentra registrada, selecciona el nombre de la máquina, la aplicación le lleva hacia una nueva actividad donde se procederá a grabar la señal de audio. Al finalizar la grabación de audio, la aplicación le lleva hacia la siguiente actividad donde podrá obtener los resultados estadísticos de la señal de audio. De igual forma tiene la opción de validación de datos, para evitar registrar nombres ya existentes en la base de datos y, además evitar ingresar espacios en blanco. En la Figura 31 se puede observar la interfaz de la aplicación móvil para la opción capturar señal del estado de la máquina.

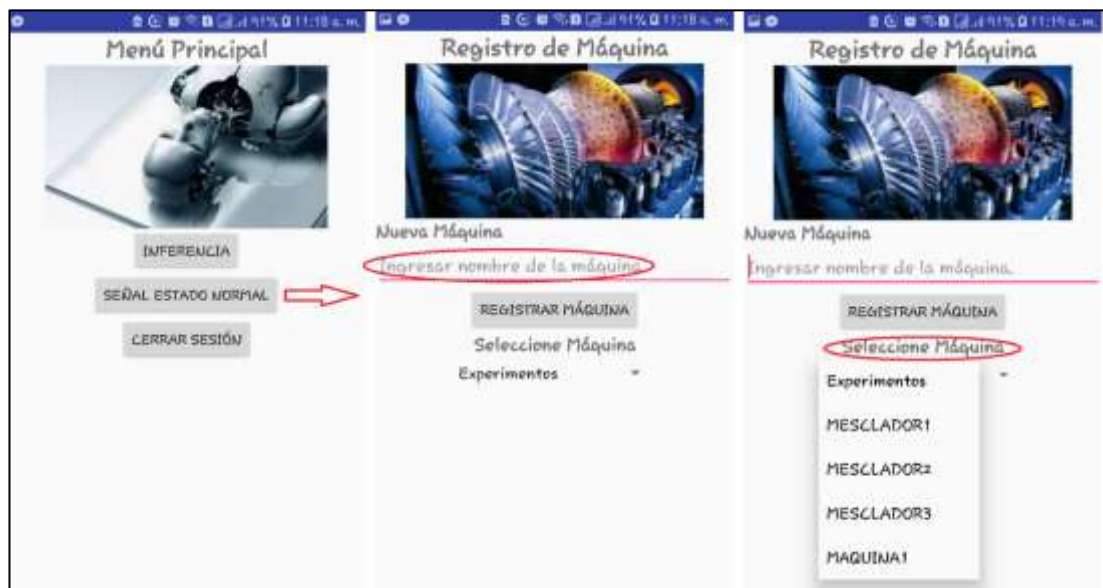


Figura 31 *Proceso del registro o selección de la máquina que desea analizar.*

Fuente: *Autor*

➤ **Inferencia del estado de la máquina**

Para el proceso de inferencia el usuario deberá primero seleccionar la máquina que desea analizar. Esto le permite al sistema identificar las señales de estado normal de dicha máquina almacenadas en el servidor de cómputo. Una vez seleccionada la máquina se procederá a grabar la señal de audio cuando se presuma un fallo de funcionamiento. Al finalizar la grabación, la aplicación le llevará hacia

una siguiente actividad donde podrá obtener los resultados estadísticos y de inferencia de la máquina. En la Figura 32 se observa la interfaz con el proceso de inferencia del estado de la máquina.



Figura 32 Proceso para obtener el estado de inferencia de la máquina.
Fuente: *Autor*

4.4 DIRECTORIOS EN EL SERVIDOR DE CÓMPUTO

Esta sección le permite al usuario tener toda su información de forma organizada y segura en el servidor de procesamiento. La creación de estos directorios es automática, no es necesario que el desarrollador genere dichas carpetas. El sistema identifica si el nombre de la empresa registrada por el usuario es nuevo, crea la carpeta con el nombre de la empresa. Si la empresa ya se encuentra registrada solamente se asigna al usuario permisos para que pueda continuar almacenando información en el directorio. En la Figura 33 se puede observar los directorios creados en el servidor de cómputo.

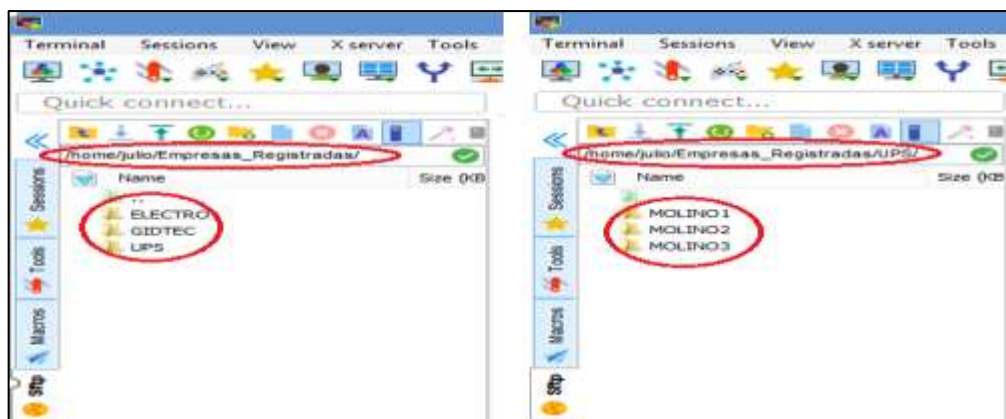


Figura 33 Directorios en el servidor de cómputo.
Fuente: *Autor*

4.5 RESULTADOS OBTENIDOS

Para comprobar el funcionamiento del sistema propuesto se utilizó el banco simulador de fallos perteneciente al Grupo de Investigación y Desarrollo en Tecnologías Industriales GIDTEC de la Universidad Politécnica Salesiana. El banco contiene un sistema mecánico rotativo [48], que puede ser utilizado con distintas configuraciones combinando elementos mecánicos e instrumentación necesaria dependiendo del experimento que se desarrolle. El banco se utiliza como fuente de señales acústicas con dos configuraciones: la primera en estado normal de funcionamiento, y la segunda con fallos en la máquina. En la Figura 34 se observa el proceso de capturas de señal de audio del banco de vibraciones.

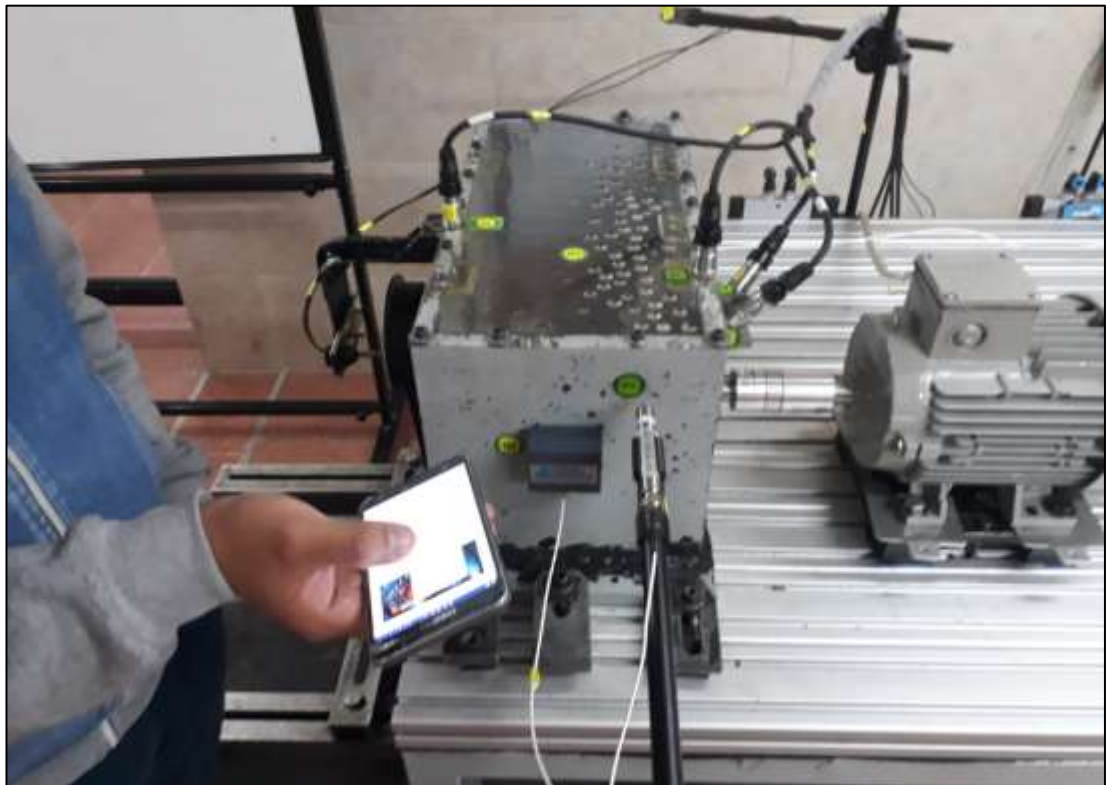


Figura 34 *Adquisición de señales acústicas en el banco de vibraciones de la Universidad Politécnica Salesiana.*

Fuente: *Autor*

A continuación, se verifica el funcionamiento de los servicios propuestos en la arquitectura del sistema (Sección 2).

➤ **Servicio de registro de Usuario**

Se registró 6 clientes pertenecientes a 3 empresas distintas, para el inicio de sesión el cliente necesita ingresar su nombre de usuario y contraseña. Cabe recalcar que, si los usuarios pertenecen a la misma empresa, la información de registro de

máquinas y señales de audio almacenada en el servidor de cómputo serán visibles entre estos usuarios. En la Figura 35 se puede observar el registro de usuarios en el servidor web. Esta información estará disponible solamente para el administrador del servicio web.

```

Database changed
mysql> select * from user \G
***** 1. row *****
      id: 8
      name: Julio Fernando Sigüencia
      username: julio93
      password: 3093
      email: jsigüenciau@est.ups.edu.ec
      business: UPS
***** 2. row *****
      id: 9
      name: Marco Nieto
      username: marcol
      password: 1234
      email: marco@ups.com
      business: UPS
***** 3. row *****
      id: 10
      name: Mauricio Tipán
      username: mauricio1
      password: 1234
      email: mauricio1@ups.com
      business: UPS
***** 4. row *****
      id: 11
      name: Diego Cabrera
      username: diegol
      password: 1234
      email: diegol@gidtec.com
      business: GIDTEC
***** 5. row *****
      id: 12
      name: Klever Sigüencia
      username: klever1
      password: 1234
      email: klever1@electro.com
      business: ELECTRO
***** 6. row *****
      id: 13
      name: Marlon Chazi
  
```

Figura 35 Registro de usuarios en la base de datos del servidor web.
Fuente: **Autor**

➤ **Servicio de almacenamiento**

Las grabaciones tomadas con el dispositivo móvil son almacenadas en el servidor de cómputo dentro de los directorios específicos para cada cliente. En la Figura 36 se puede observar los archivos de audio almacenados.

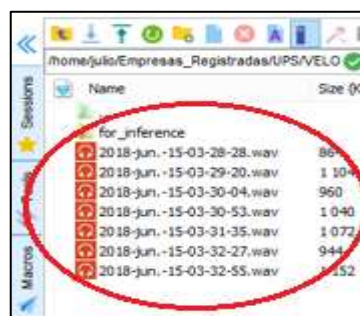


Figura 36 Almacenamiento de grabaciones de audio en el servidor de cómputo.
Fuente: **Autor**

➤ **Servicio de procesamiento**

El banco de vibraciones posee 3 velocidades distintas y cada una genera un sonido diferente. Las velocidades son:

- Velocidad baja: 360 rpm
- Velocidad media: 720 rpm
- Velocidad alta: 960 rpm

Se procedió a generar la base de datos con señales de audio en estado normal de la máquina. La captura de datos se realizó durante 5 días tomando 3 muestras diarias a diferentes velocidades. El sonido capturado se almacena en formato .wav sin comprimir, codificado en PCM de 16 bits. La frecuencia de muestreo es de 44.1kHz equivalente a la grabación de audio con calidad de CD. El tiempo de grabación de audio fue de 10 segundos según recomendaciones dadas en [2], con lo cual se obtiene los siguientes resultados:

❖ Velocidad baja

Para este tipo de prueba se capturo 5 señales de audio. En las Figuras 37 y 38 se puede observar los valores estadísticos y el histograma generado a partir del archivo de audio .wav grabado en el dispositivo móvil. Debido a que son grabados en condiciones normales existe un cierto grado de similitud entre valores estadísticos e histograma de ambos archivos de audio.

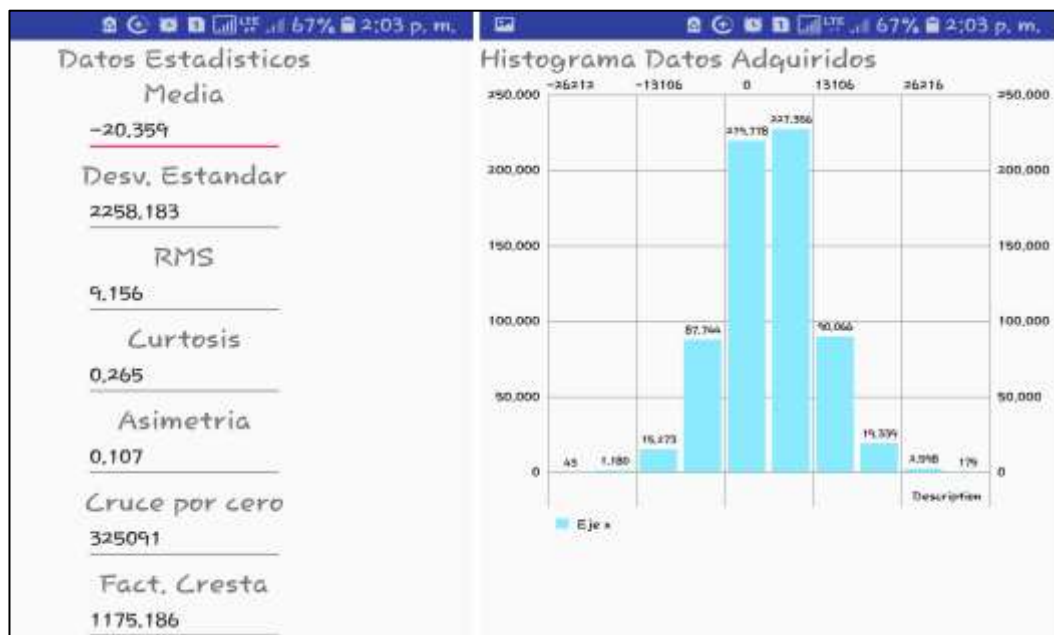


Figura 37 Resultados obtenidos de la primera muestra de audio capturada a velocidad baja en el banco de vibraciones.

Fuente: *Autor*



Figura 38 Resultados obtenidos de la última muestra de audio capturada a velocidad baja en el banco de vibraciones.

Fuente: *Autor*

❖ Velocidad media

De igual manera se capturo 5 señales de audio. Para esta velocidad el nivel de sonido aumenta. En las Figuras 39 y 40 se puede observar los valores estadísticos y el histograma generado a partir del archivo de audio .wav grabado en el dispositivo móvil.

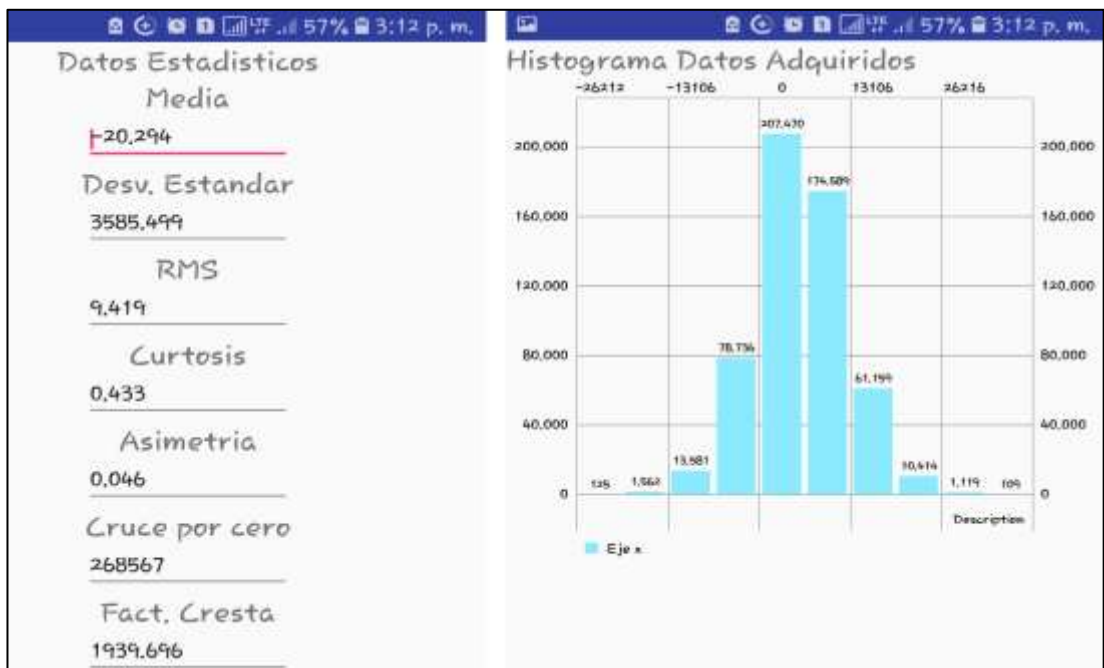


Figura 39 Resultados obtenidos de la primera muestra de audio capturada a velocidad media en el banco de vibraciones.

Fuente: *Autor*



Figura 40 Resultados obtenidos de la última muestra de audio capturada a velocidad media en el banco de vibraciones.

Fuente: **Autor**

❖ Velocidad alta

Por último, se capturo las señales cuando la máquina funcionaba a su más alta velocidad generando los resultados estadísticos que se puede observar en las Figuras 41 y 42.



Figura 41 Resultados obtenidos de la primera muestra de audio capturada a velocidad alta en el banco de vibraciones.

Fuente: **Autor**



Figura 42 Resultados obtenidos de la última muestra de audio capturada a velocidad alta en el banco de vibraciones.
Fuente: *Autor*

➤ Servicio del estado de inferencia de la máquina

Generadas las firmas de audio en estado de funcionamiento normal es posible obtener el estado de la máquina. Como ya se había mencionado anteriormente se realiza una comparación entre histogramas del vector de soporte total de las firmas de audio en estado normal y la señal de audio en un estado desconocido de la máquina. El nuevo estado se obtuvo al cambiar las configuraciones del banco de vibraciones para modelar un daño severo. Para el experimento se colocó piñones que presentaban crack o grietas.

Una vez configurado el banco se obtuvo los siguientes resultados:

❖ Velocidad Baja

En la Figura 43 se puede observar los valores de la media y desviación estándar de los datos acústicos en condiciones normales y los datos de una señal con fallo del banco de vibraciones. Además, el factor de similitud y los histogramas generados entre las señales de estado normal y fallo.

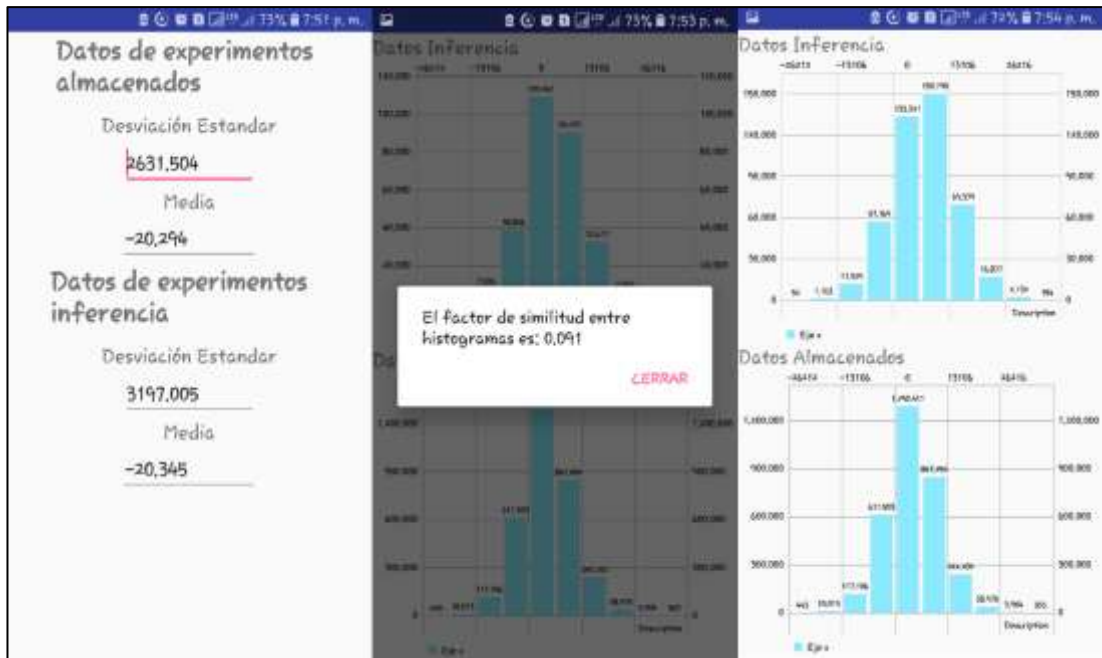


Figura 43 Resultados de inferencia del banco de vibraciones a velocidad baja con fallo.
Fuente: Autor

La Figura 44 muestra los resultados obtenidos al realizar el experimento del estado de inferencia con una señal de audio en condiciones normales de funcionamiento. Los datos que se pueden observar son similares a los de la figura anterior.

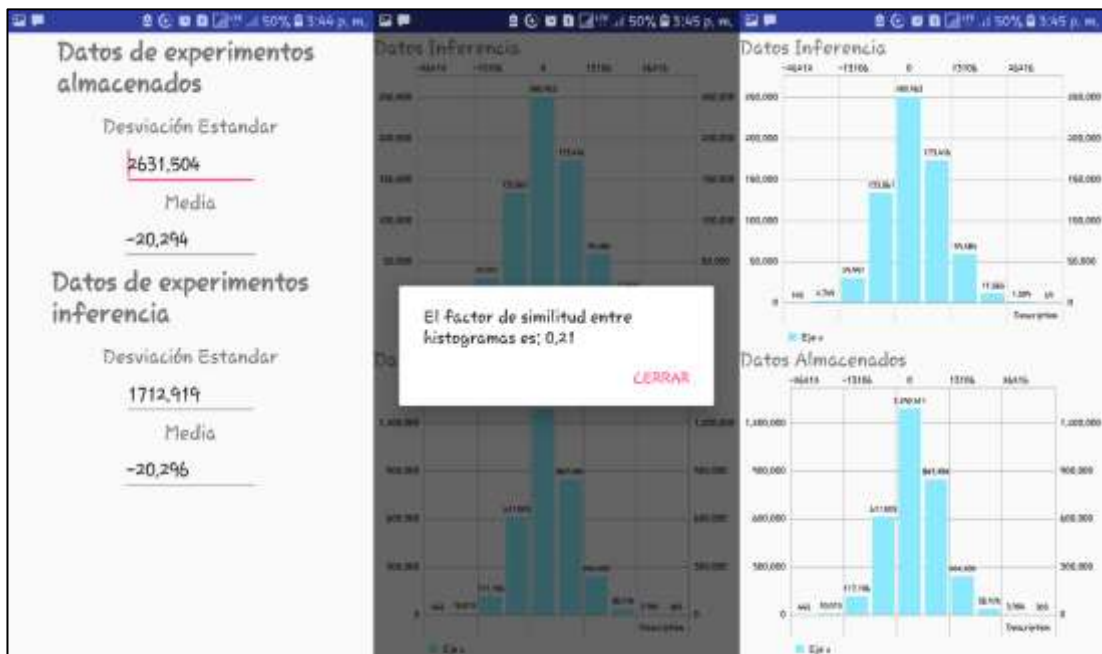


Figura 44 Resultados de inferencia del banco de vibraciones a velocidad baja en estado de funcionamiento normal.
Fuente: Autor

❖ Velocidad Media

Los resultados de inferencia cuando el banco de vibraciones funciona a velocidad media se puede observar a continuación en la Figura 45.



Figura 45 Resultados de inferencia del banco de vibraciones a velocidad media con fallo.

Fuente: **Autor**

En la Figura 46 se observa los resultados obtenidos para el experimento velocidad media con una señal de estado normal de funcionamiento.

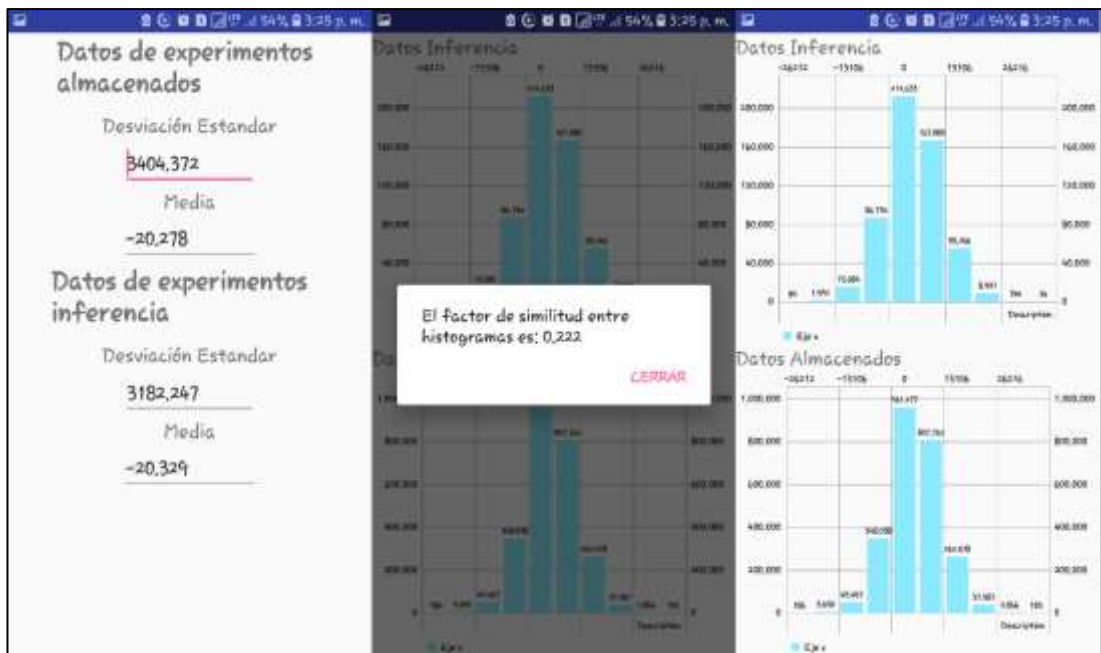


Figura 46 Resultados de inferencia del banco de vibraciones a velocidad media con una señal de estado de funcionamiento normal.

Fuente: **Autor**

❖ Velocidad Alta

Por último, para el caso de velocidad alta, los resultados de inferencia con una señal de fallo se los puede observar en la Figura 47.

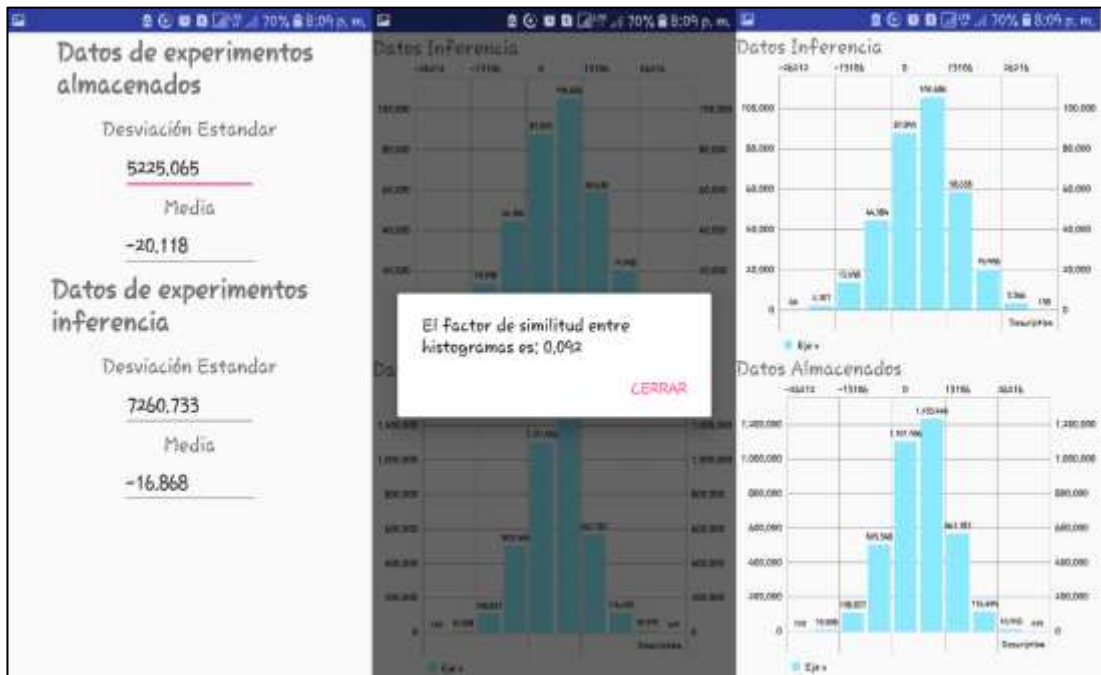


Figura 47 Resultados de inferencia del banco de vibraciones a velocidad alta con fallo.

Fuente: *Autor*

La figura 48 muestra el resultado del experimento a velocidad alta del banco de vibraciones con una señal de estado normal de funcionamiento.

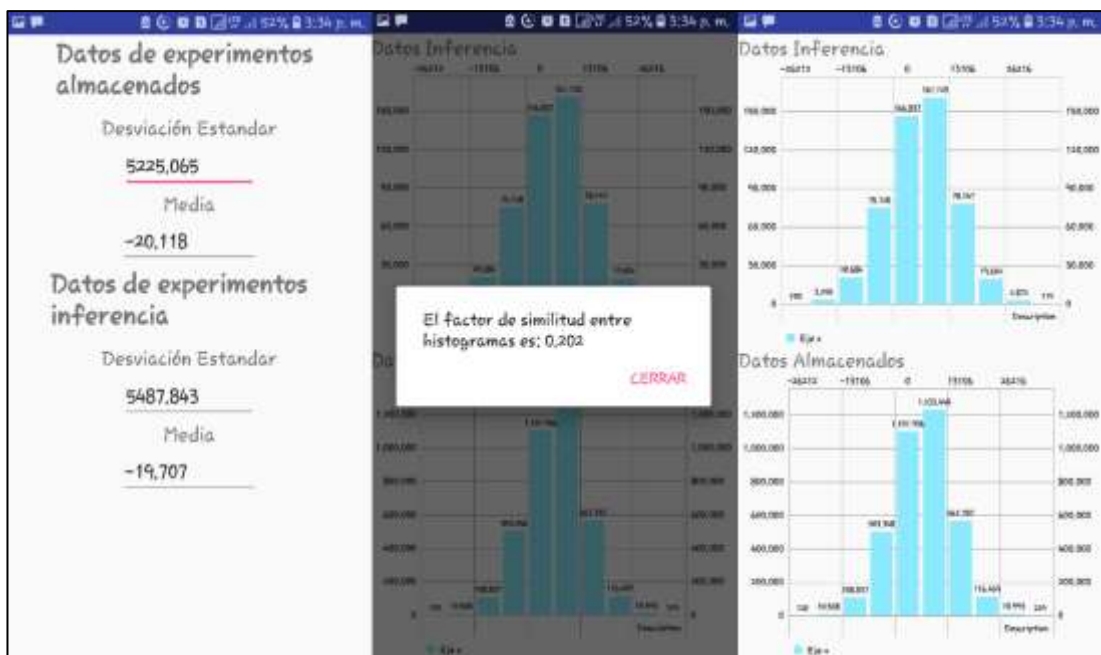


Figura 48 Resultados de inferencia con una señal de estado normal de funcionamiento del banco de vibraciones a velocidad alta.

Fuente: *Autor*

4.6 ANÁLISIS DE RESULTADOS

En las gráficas obtenidas anteriormente se muestran los valores digitalizados del sonido de la máquina mediante histogramas. Además, valores estadísticos que permiten interpretar el comportamiento de las gráficas de acuerdo a los distintos escenarios experimentados. Los valores se obtienen a partir de la señal analógica temporal de audio. Digitalizando estos valores se obtiene un rango de -32768 a 32768 bits debido a que la captura de datos se la realiza con una resolución de 16 bits y una frecuencia de muestreo de 44100 Hz. La cantidad de datos capturados depende de la frecuencia de muestreo es decir se están tomando 44100 muestras por segundo. En la Figura 49 se puede observar la representación gráfica de las amplitudes en función del tiempo de la señal de audio de la máquina. La señal de audio original está compuesta por la mezcla de múltiples fuentes acústicas de los distintos componentes que conforma la máquina. Al realizar el histograma de los datos se observa que existe un comportamiento con distribución normal, debido al gran número de datos procesados con señales desde diferentes fuentes acústicas (componentes de la máquina) y obedeciendo al Teorema de Limite Central que dice “*Cuando el tamaño de la muestra es lo suficientemente grande, la distribución de las medias sigue aproximadamente una distribución normal*”[49].

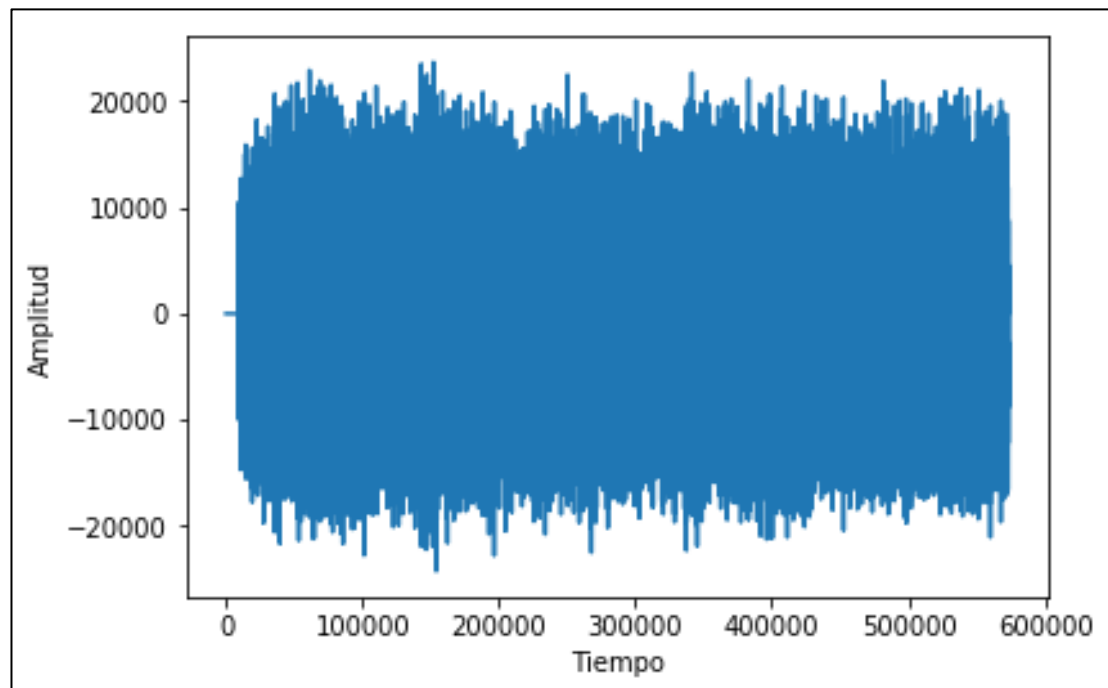


Figura 49 Señal de audio de amplitudes en función del tiempo del banco de vibraciones a velocidad alta.

Fuente: *Autor*

➤ Captura de señales en estado normal de funcionamiento

A continuación, en la Tabla 2 se muestran los valores estadísticos de las grabaciones de audio capturadas. Estos datos pertenecen a señales de estado normal de funcionamiento de la máquina.

Tabla 2 Resultados estadísticos de las señales de audio en estado normal de funcionamiento de la máquina.

Fuente: Autor

Resultados Estadísticos de las Grabaciones de Audio del Banco de Vibraciones								
		Media	Desviación Estándar	RMS	Curtosis	Asimetría	Cruce por cero	Factor de cresta
Velocidad Baja	1°	-20,359	2258,183	9,156	0,256	0,107	325091	1175,186
	2°	-20,354	2849,57	7,568	0,354	0,127	325328	1851,612
	3°	-20,354	2692,522	8,012	0,267	0,13	325143	1737,893
	4°	-20,227	2590,2	6,786	0,255	0,13	301130	1956,97
	5°	-20,157	2743,429	11,215	0,358	0,14	273042	1268,301
Velocidad Media	1°	-20,294	3585,499	9,419	0,133	0,046	268567	1939,696
	2°	-20,288	3321,051	10,277	0,155	0,05	244959	1632,134
	3°	-20,158	3611,467	11,414	0,161	0,056	240569	1478,886
	4°	-20,273	3295,582	9,417	0,277	0,11	240168	1768,504
	5°	-20,257	3448,347	10,658	0,23	0,041	268691	1590,542
Velocidad Alta	1°	-19,414	5710,799	6,611	-0,052	-0,034	216498	3586,901
	2°	-20,394	5589,371	7,726	-0,012	-0,025	278907	2979,032
	3°	-19,852	5247,325	10,967	-0,097	-0,021	241179	1929,242
	4°	-20,385	5253,954	10,027	-0,165	-0,062	269466	2005,948
	5°	-19,972	5158,019	8,824	-0,13	-0,051	261131	2199,229

En la tabla 3 se muestran los valores promediados de las medidas estadísticas obtenidas en los experimentos realizados. El valor de la media es aproximadamente -20 para los tres experimentos en condiciones normales de funcionamiento de la máquina. Si se observa en la figura 49 se puede notar un offset de señal respecto a cero y luego una variación de amplitudes desde cero hacia el valor mínimo y hacia el valor máximo. Debido a este comportamiento la media de los datos se sitúa en un valor cercano a cero.

Tabla 3 Valores estadísticos promedio de las señales de audio en estado normal de funcionamiento de la máquina.

Fuente: Autor

Valores estadísticos promedios de la tabla 2.							
	Media	Desviación Estándar	RMS	Curtosis	Asimetría	Cruce por cero	Factor de cresta
Vel. Baja	-20,2902	2626,7808	8,5474	0,298	0,1268	309946,8	1597,992
Vel. Media	-20,254	3452,3892	10,237	0,1912	0,0606	252590,8	1681,952
Vel. Alta	-20,0034	5391,8936	8,831	-0,0912	-0,0386	253436,2	2540,07

En la Tabla 2 se puede observar una variación mínima de los resultados obtenidos, debido a que todos los casos experimentados se los realizó sin carga, ni variación de las configuraciones del banco de vibraciones. La tabla 3, está compuesta por los valores promedio de las medidas estadísticas resultados de la experimentación. Para el caso de velocidad baja, según los histogramas presentados en las figuras 37 y 38 se observa una concentración de datos cercanos a la media y una forma más apuntada de la campana de gauss formada por la distribución de los datos. A velocidad media la forma del histograma reduce su forma apuntada y esto se comprueba a razón de que la desviación estándar tiene un leve aumento como se observa en las figuras 39 y 40. El valor de la media presenta una variación mínima, por lo cual no hay desplazamiento de los datos. Por último, a velocidad alta la media se mantiene en -20 variando únicamente su parte decimal. La desviación estándar aumenta considerablemente y se ve reflejado en las figuras 41 y 42, los histogramas se ensanchan y aumentan el área de las colas en la campana de gauss.

El valor rms se mantiene entre 8 y 10 e indica la energía desprendida de la señal. Un aumento considerable de este valor podría indicar defectos en el sistema. La curtosis se encuentra en rangos normales y se comporta de acuerdo a la forma de la distribución normal, mientras más apuntada sea la forma de la distribución la curtosis se incrementará. El valor de asimetría debe ser cercano a cero, un incremento indicara mayor área en la cola de la distribución. El valor de factor de cresta oscila entre los 1500 y 2500. Si existiera picos de la señal en el dominio del tiempo darán como resultado un incremento en el factor de cresta.

➤ **Inferencia del estado de la máquina**

Los resultados obtenidos muestran valores favorables para identificar los fallos configurados en el banco de vibraciones. Para el caso de velocidad baja, el valor de la media y desviación estándar de las señales de estado normal y señal con fallo no presentan cambios considerables en sus cifras, sin embargo, el factor de similitud da como resultado 0.091, resultando un indicador de fallo en una de las partes de la máquina que se encuentra en movimiento. Para diferenciar el funcionamiento del sistema de detección de fallas se realizó el experimento con una señal de estado normal, mostrando como resultado valores estadísticos similares y un

factor de similitud de 0.21, valor que se distancia de cero e indicador de un funcionamiento normal del banco de vibraciones.

Para el experimento a velocidad media, los valores estadísticos desviación estándar y media muestran una considerable variación entre las señales de estado normal y la señal con fallo como se puede observar en la figura 45. El factor de similitud para la velocidad media es igual a 0.197, también cercano a cero e indicador de fallo. En este caso la inferencia con una señal de estado normal de funcionamiento da como resultado 0.222. Existe un valor menor en los factores de similitud, sin embargo, en el caso de la señal de estado normal el factor se distancia más del cero. Los valores estadísticos a diferencia del experimento realizado con una señal con fallo, son similares entre el vector de soporte y la señal analizada como se observa en la Figura 46.

Por último, para el experimento a velocidad alta hay una variación considerable en los valores estadísticos como se observa en la Figura 47 y un factor de similitud igual a 0.092, siendo este valor el más cercano a cero, aseverando un fallo inminente en la máquina. La Figura 48 muestra los resultados de inferencia con una señal de estado normal a velocidad alta. El factor de similitud da como resultado 0.202, que a diferencia del caso analizado con una señal de fallo se encuentra más distante del cero e indicador de un buen funcionamiento de la máquina. Los valores estadísticos son muy cercanos entre el vector de soporte y la señal analizada.

Los valores obtenidos del factor de similitud tanto para el caso de una señal de audio con fallo como para una señal de audio en estado normal muestran una diferencia mínima. En el caso de una señal en estado normal los datos almacenados superan a los datos de la señal capturada para el análisis, pero la media se mantiene y no existe una mayor variación de los datos. Sin embargo, para la inferencia de una señal con fallo se logra notar un desplazamiento de la media respecto a la media de los datos en estado normal. Este desplazamiento hace que el factor de similitud sea cercano a cero pues el área de intersección entre histogramas se reduce.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

La identificación temprana de fallos en las diferentes máquinas que son parte de un proceso evita pérdidas de producción en las empresas. Lograr identificar tempranamente e incluso predecir fallos reduce las paradas innecesarias en la línea de producción como también alarga el tiempo de vida útil de las máquinas. En la actualidad la mayoría de los procesos son controlados en base a identificar variables que afectan directamente al proceso. Sin embargo, la nueva tendencia de la industria es el almacenamiento de la información. La combinación de los sofisticados elementos de adquisición de datos, tecnologías de la información y procesadores de cómputo de alto rendimiento, permiten introducir modelos que se han desarrollado únicamente en los laboratorios.

En base a los criterios mencionados anteriormente se propuso un sistema para la detección de fallos de máquinas rotativas basándose en una arquitectura orientada a servicios. La finalidad fue conseguir un sistema no invasivo que pueda formar parte de las técnicas de mantenimiento en la industria. Mediante el uso de dispositivos móviles se logró capturar las señales de audio y enviarlas al servidor de cómputo a través del internet para el procesamiento e inferencia del estado de la máquina. Los sofisticados dispositivos móviles que disponen la mayoría de los usuarios presentan

características de hardware robustas que fueron aprovechadas para adquirir el sonido que genera la máquina.

El nivel sonido que emiten las máquinas cuando están en funcionamiento es muy alto y difícil de distinguir por los técnicos cuando existe un fallo leve. Se logra notar una diferencia por el ser humano cuando la máquina presenta un fallo severo. El sistema de detección de fallas propuesto le permite al técnico observar el comportamiento de la máquina en base a un análisis estadístico de la señal de audio que emite y tomar acciones de mantenimiento en el caso de fallo para evitar que la máquina alcance un daño severo. Los valores obtenidos en el análisis de resultados demuestran que un fallo leve a mayor movimiento altera el sonido que emite la máquina y al no ser detectado podría provocar daños en otros componentes de la máquina o incluso alterar el proceso que se encuentra realizando. El análisis de datos permite detectar fallas tempranas y evitar paradas innecesarias como también prolongar el tiempo de vida útil de las máquinas.

Se recomienda utilizar dispositivos móviles de última generación, debido a que presentan características de hardware robustas que facultan el funcionamiento del sistema de detección de fallos propuesto.

El sistema de detección de fallas está basado en una arquitectura SOA, lo que permite agregar servicios que puedan ser implementados a futuro. Una vez almacenada la información es posible realizar procesamiento de la información con diferentes herramientas matemáticas, estadísticas, machine learning, etc., para diagnóstico de fallos más específicos. Además, llevar a cabo experimentaciones a nivel industrial como una técnica de mantenimiento para las distintas máquinas que son parte de un proceso.

REFERENCIAS BIBLIOGRÁFICAS

- [1] X. Ren, M. Ong, G. Allan, V. Kadiramanathan, H. Thompson, and P. Fleming, “A Service Oriented Architecture for Integration of Fault Diagnostics,” Springer, Berlin, Heidelberg, 2005, pp. 146–157.
- [2] N. K. Verma, J. V. Singh, M. Gupta, S. Dixit, R. K. Sevakula, and A. Salour, “Windows mobile and tablet app for acoustic signature based machine health monitoring,” in *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, 2014, pp. 1–6.
- [3] F. Jammes and H. Smit, “Service-oriented architectures for devices - the SIRENA view,” in *INDIN '05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pp. 140–147.
- [4] T. Erl, *SOA : principles of service design*. Prentice Hall, 2008.
- [5] L. Auer, N. Kryvinska, C. Strauss, and E. Belov, “Software-based Business Applications/Tools to Assess Complex SOA Investments - The Cross-Vendor Comparative Analysis,” in *2011 International Conference on Complex, Intelligent, and Software Intensive Systems*, 2011, pp. 397–401.
- [6] J. Barata, L. Ribeiro, and A. Colombo, “Diagnosis using Service Oriented Architectures (SOA),” *Ind. Informatics, 2007 5th IEEE Int. Conf.*, vol. 2, pp. 1203–1208, 2007.
- [7] H. A. ElMaraghy, “Flexible and reconfigurable manufacturing systems paradigms,” *Int. J. Flex. Manuf. Syst.*, vol. 17, no. 4, pp. 261–276, Oct. 2005.
- [8] J. Barata, L. Camarinha-Matos, and G. Cândido, “A multiagent-based control system applied to an educational shop floor,” *Robot. Comput. Integr. Manuf.*, vol. 24, no. 5, pp. 597–605, Oct. 2008.
- [9] P. Jiang, M. Elag, P. Kumar, S. D. Peckham, L. Marini, and L. Rui, “A service-oriented architecture for coupling web service models using the Basic Model Interface (BMI),” *Environ. Model. Softw.*, vol. 92, pp. 107–118, 2017.
- [10] M. Cerrada, F. Pacheco, R. V. Sanchez, D. Cabrera, J. Macancela, and P. Lucero, “SOA Based Integrated Software to Develop Fault Diagnosis Models Using Machine Learning in Rotating Machinery,” in *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, 2017, pp. 28–37.

- [11] U. Demir, E. Ghaleb, and H. K. Ekenel, “A Face Recognition Based Multiplayer Mobile Game Application,” Springer, Berlin, Heidelberg, 2014, pp. 214–223.
- [12] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, “Future Networking Challenges: The Case of Mobile Augmented Reality,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 1796–1807.
- [13] J. Cohen, “Embedded speech recognition applications in mobile phones: Status, trends, and challenges,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 5352–5355.
- [14] W. S. Ng, W. T. Ang, S. Viswanathan, and C. K. Tham, “Experiences on developing SOA based mobile healthcare services,” *2009 IEEE Asia-Pacific Serv. Comput. Conf.*, pp. 498–501, 2009.
- [15] J. U. Meyer, “Open SOA health web platform for mobile medical apps: Connecting securely mobile devices with distributed electronic health records and medical systems,” *19th IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA 2014*, 2014.
- [16] C. De La and T. R. González, “Arquitectura SOA con Tecnología Microsoft.”
- [17] IBM Knowledge Center, “IBM Knowledge Center - Informática distribuida.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSAL2T_9.1.0/com.ibm.ci.cs.tx.doc/concepts/c_wht_is_distd_comptg.html. [Accessed: 13-Apr-2018].
- [18] A. R. De Soto and E. Cuervo Fernández, “Nuevas Tendencias en Sistemas de Información: Procesos y Servicios,” *Pecvnia*, vol. 2, pp. 129–158, 2006.
- [19] M. A. Bernabé-Poveda and C. M. López-Vázquez, *Fundamentos de las Infraestructuras de Datos Espaciales (IDE) - Miguel A. Bernabé-Poveda y Carlos M. López-Vázquez - Google Libros*, Biblioteca Online SL. Madrid, 2012.
- [20] M. R. Morales roldán and D. N. Thompson, “Aplicaciones móviles nativas orientadas a servicios y recursos de bibliotecas universitarias,” *Uned.Ac.Cr*, pp. 1–12, 2013.
- [21] D. Triboan, L. Chen, F. Chen, and Z. Wang, “Towards a service-oriented architecture for a mobile assistive system with real-time environmental sensing,” *Tsinghua Sci. Technol.*, vol. 21, no. 6, pp. 581–597, Dec. 2016.

- [22] M. Navarro and J. M. Corchado, “Aplicación multiagente en entornos móviles para la toma de datos en zonas de aparcamiento,” *Dep. Informática y Automática Univ. Salamanca*, p. 107.
- [23] I. Corporation, “El Desarrollo De Aplicaciones Moviles Nativas,” p. 10, 1012.
- [24] C. Aulds, *Linux Apache Web Server Administration*. John Wiley & Sons, 2006.
- [25] “Versión 2.4 de la documentación del Servidor de HTTP Apache - Servidor Apache HTTP Versión 2.4.” [Online]. Available: <http://httpd.apache.org/docs/2.4/>. [Accessed: 16-Apr-2018].
- [26] “SSH (Secure Shell) Página de inicio | SSH.COM.” [Online]. Available: <https://www.ssh.com/ssh/>. [Accessed: 14-Apr-2018].
- [27] SSH Communications Security ., “Protocolo SSH - Inicio de sesión remoto seguro y transferencia de archivos | SSH.COM.” [Online]. Available: <https://www.ssh.com/ssh/protocol/>. [Accessed: 14-Apr-2018].
- [28] D. J. Barrett, R. Silverman, and : O ’reilly, “SSH, The Secure Shell: The Definitive Guide,” 2001.
- [29] W. Willems and I. Gutiérrez García, *Una introducción a la criptografía de clave pública (2a. ed.)*. Universidad del Norte, 2011.
- [30] L. M. Crespo Martínez and F. A. C. Herías, “INTRODUCCIÓN A TCP/IP Sistemas de Transporte de Datos.”
- [31] “IBM Knowledge Center - Protocolos TCP/IP.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/com.ibm.aix.networkcomm/tcpip_protocols.htm. [Accessed: 15-Apr-2018].
- [32] S. Garcia Garrido, *Ingeniería de mantenimiento Manual práctico para la gestión eficaz del mantenimiento industrial*, RENOVETEC. Madrid, 2012.
- [33] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, “A review of process fault detection and diagnosis: Part III: Process history based methods,” *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 327–346, Mar. 2003.
- [34] Z. Chen, C. Li, and R.-V. Sánchez, *Multi-layer neural network with deep belief network for gearbox fault diagnosis*, vol. 17, no. 5. JVE International Ltd., 2015.
- [35] M. Orman, P. Rzeszucinski, A. Tkaczyk, K. Krishnamoorthi, C. T. Pinto, and M. Sulowicz, “Bearing fault detection with the use of acoustic signals recorded by a hand-held mobile phone,” in *2015 International Conference on*

- Condition Assessment Techniques in Electrical Systems (CATCON)*, 2015, pp. 252–256.
- [36] M. Rodríguez, “Transductores y sensores: propiedades y características técnicas,” *Revista digital INESEM*, Granada, 2016.
- [37] E. MartínezMartines and A. Santoyo Serrano, *Fundamentos de Telecomunicaciones y Redes*. 2012.
- [38] C. Guisande González, *Tratamiento de datos*. Díaz de Santos, 2006.
- [39] G. White, *Introducción al Análisis de Vibraciones*, Azima DLI. U S A, 2010.
- [40] Z. Chen, C. Li, and R.-V. Sanchez, “Gearbox Fault Identification and Classification with Convolutional Neural Networks,” *Shock Vib.*, vol. 2015, pp. 1–10, Oct. 2015.
- [41] J. L. R. E. W. / K. E. Y. / R. H. M. / S. L. M. Devore, *Probabilidad Y Estadística Para Ingenierías Y Ciencias*. 2008.
- [42] J. Hernández, “Cálculo simplificado del valor medio y eficaz de una forma de onda,” *Técnica Ind.*, no. January, 2007.
- [43] R. Behar Gutiérrez and P. Grima Cintas, *55 respuestas a dudas típicas de estadística*. Diaz de Santos, 2010.
- [44] J. C. Vesga Ferreira, G. Granados Acuña, and J. E. Sierra Carrillo, “Optimization of a multi-service network over a PLC channel under MmQoS,” *Ing. y Desarro.*, 2015.
- [45] M. J. Swain and D. H. Ballard, “Color indexing,” *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, Nov. 1991.
- [46] Núñez A.C., “Información al consumidor de los atributos de un producto de supermercado mediante una aplicación móvil: ‘FoodAdvice,’” p. 62, 2015.
- [47] Hometech, “Micrófonos externos para grabación de audio en dispositivos móviles – home:TECH.” [Online]. Available: <https://hometech.com.mx/microfonos-externos-para-grabacion-de-audio-en-dispositivos-moviles/>. [Accessed: 26-Apr-2018].
- [48] J. Jimbo, J. Armando, S. Vivar, and J. Gabriel, “Diseño y construcción de un banco didáctico para la medición de vibraciones mecánicas en los Laboratorios de Instrumentación de la Universidad Politécnica Salesiana sede Cuenca,” 2012.
- [49] S. Minitab, “El teorema del límite central.”

- [50] F. Maciá Pérez, *Administración de servicios de Internet: de la teoría a la práctica*. Publicaciones de la Universidad de Alicante, 2008.
- [51] American Top Tools, “SCP (Secure Copy),” *Blog att*, 2015. [Online]. Available:
<http://www.att.es/producto/goanywhere/products/director/connectivity/scp.html>. [Accessed: 06-Jun-2018].