

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**DETECCIÓN DE EVENTOS DE CONTAMINACIÓN ACÚSTICA  
INDUSTRIAL BASADO EN UNA RED DE SENSORES  
DESARROLLO DE WEB SERVICE DE NOTIFICACIONES PARA  
INFORMAR POR REDES SOCIALES DATOS DE RED DE  
SENSORES ACÚSTICOS**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
TELECOMUNICACIONES**

**PABLO ESTEBAN LLUMIQUINGA VILLENAS**

**pablo.llumiquinga@epn.edu.ec**

**DIRECTOR: DR. TARQUINO SÁNCHEZ ALMEIDA**

**tarquino.sanchez@epn.edu.ec**

**DMQ, octubre 2022**

## CERTIFICACIONES

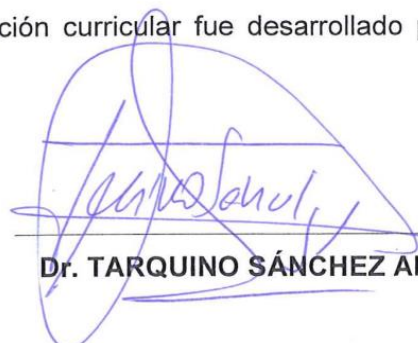
Yo, Pablo Llumiyinga declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**PABLO LLUMIQUINGA**

Certifico que el presente trabajo de integración curricular fue desarrollado por PABLO LLUMIQUINGA, bajo mi supervisión.



---

**Dr. TARQUINO SÁNCHEZ ALMEIDA**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

PABLO LLUMIQUINGA

DR. TARQUINO SÁNCHEZ ALMEIDA

## **DEDICATORIA**

De manera especial dedico este Trabajo de Integración Curricular a Dios, mis queridos padres, hermanas y a mi amada novia, quienes me han apoyado incondicionalmente a lo largo de mi vida personal y académica.

## **AGRADECIMIENTO**

Agradezco infinitamente a mis padres, hermanas y novia, quienes han sido pilar fundamental para la realización del presente proyecto ya que gracias a su apoyo a sido posible culminar esta etapa importante de mi vida.

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VIII
ABSTRACT .....	IX
1 INTRODUCCIÓN.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos .....	2
1.3 Alcance .....	2
1.4 Marco teórico .....	3
<b>1.4.1 Normativa para la contaminación acústica.....</b>	<b>4</b>
<b>1.4.2 Google Forms .....</b>	<b>7</b>
<b>1.4.3 Tamaño de muestra para encuestas .....</b>	<b>7</b>
<b>1.4.4 Web Service .....</b>	<b>8</b>
<b>1.4.5 Python .....</b>	<b>9</b>
2 METODOLOGÍA.....	11
2.1 Elaboración de Encuestas.....	12
<b>2.1.1 Tipo de empresa .....</b>	<b>12</b>
<b>2.1.2 Tipo de Zona .....</b>	<b>13</b>
<b>2.1.3 Puesto de trabajo.....</b>	<b>13</b>
<b>2.1.4 Horarios de trabajo rotativos .....</b>	<b>14</b>
<b>2.1.5 Periodo de Tiempo .....</b>	<b>14</b>
<b>2.1.6 Datos sobre ruido .....</b>	<b>15</b>
<b>2.1.7 Cantidad de personal .....</b>	<b>16</b>
<b>2.1.8 Cantidad de personal por área .....</b>	<b>16</b>
<b>2.1.9 Cantidad de maquinaria .....</b>	<b>16</b>
<b>2.1.10 Operación de Maquinaria .....</b>	<b>17</b>
<b>2.1.11 Tipo de Red Social.....</b>	<b>17</b>
<b>2.1.12 Tipo de información para la publicación.....</b>	<b>18</b>
<b>2.1.13 Intervalo de tiempo para recibir las Notificaciones.....</b>	<b>18</b>
<b>2.1.14 Momento para recibir las Notificaciones .....</b>	<b>19</b>
<b>2.1.15 Alertas de niveles de ruido .....</b>	<b>19</b>

2.1.16	<b>Emisiones de nivel de ruido</b>	20
2.1.17	<b>Nivel de Riesgo de la salud</b>	20
2.1.18	<b>Niveles superiores de ruido</b>	21
2.1.19	<b>Posibles Soluciones en las notificaciones</b>	22
2.1.20	<b>Afecciones por ruido</b>	22
2.1.21	<b>Afecciones por niveles riesgo</b>	22
2.1.22	<b>Problemas debido al ruido</b>	23
2.1.23	<b>Costo de sistema de red de sensores con notificaciones</b>	23
2.2	<b>Calculo para tamaño de muestra para aplicar de encuesta</b>	24
2.3	<b>Presentación de Encuesta</b>	25
2.4	<b>Instalación de librerías de Python</b>	25
2.4.1	<b>Instalación de datetime</b>	25
2.4.2	<b>Instalación de Pandas</b>	25
2.4.3	<b>Instalación de PyAutoGUI</b>	26
2.4.4	<b>Instalación de matplotlib</b>	26
2.4.5	<b>Instalación de selenium</b>	26
2.4.6	<b>Instalación de webdriver</b>	26
2.4.7	<b>Instalación de folium</b>	26
2.5	<b>Diseño de código en Python para Web Service de notificaciones</b>	27
2.5.1	<b>Importación de librerías de Python a código</b>	28
2.5.2	<b>Entrada de parámetros de día</b>	28
2.5.3	<b>Creación de variables de sensores y carpetas de notificaciones</b>	29
2.5.4	<b>Apertura de navegador en WhatsApp</b>	30
2.5.5	<b>Lectura de base de datos</b>	31
2.5.6	<b>Parámetros de inicio y final del día</b>	31
2.5.7	<b>Filtración de base de datos</b>	32
2.5.8	<b>Creación de variables para lazos</b>	32
2.5.9	<b>Lazo primario de análisis</b>	33
2.5.10	<b>Análisis por sensor</b>	33
2.5.11	<b>Análisis para alertas por sensor</b>	34
2.5.12	<b>Rango de gráficas de alerta</b>	35
2.5.13	<b>Graficador de alertas</b>	37
2.5.14	<b>Notificación de alertas</b>	39
2.5.15	<b>Subfunción de Notificaciones para Whatsapp</b>	39
2.5.16	<b>Cierre de navegador de Whatsapp</b>	41
2.5.17	<b>Graficador de Notificaciones diarias de los niveles de sonido</b>	41

<b>2.5.18</b>	<b>Notificaciones diarias de los niveles de sonido</b> .....	<b>44</b>
<b>2.5.19</b>	<b>Graficador de Notificaciones diarias de los clústeres</b> .....	<b>44</b>
<b>2.5.20</b>	<b>Notificaciones diarias de los clústeres</b> .....	<b>46</b>
<b>2.5.21</b>	<b>Graficador y notificación de pico máximo de sonido</b> .....	<b>51</b>
<b>2.5.22</b>	<b>Eliminación de carpeta de notificaciones</b> .....	<b>53</b>
2.4	Pruebas de Web Service de notificaciones para uso adecuado .....	53
2.5	Presentación de código.....	54
3	<b>RESULTADOS, CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>55</b>
3.1	Resultados .....	55
<b>3.1.1</b>	<b>Resultados de las Encuestas</b> .....	<b>55</b>
<b>3.1.2</b>	<b>Resultados del Web Service de Notificaciones</b> .....	<b>57</b>
3.2	Conclusiones.....	70
3.3	Recomendaciones.....	71
4	<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>72</b>
5	<b>ANEXOS</b> .....	<b>74</b>
5.1	ANEXO I.....	75
5.2	ANEXO II.....	79
5.3	ANEXO III.....	82
5.4	ANEXO IV .....	82
5.5	ANEXO V .....	83



## RESUMEN

El presente estudio trata sobre el diseño y elaboración de un Web Service de notificaciones para informar por redes sociales datos provenientes de una red de sensores acústicos tomados del Sistema de Monitorio Ambiental (EMS) que pertenecen a la empresa australiana ARCS Group del proyecto TULIP, utilizando para ello, la herramienta de software Python. En el primer capítulo, se presenta la base teórica para el desarrollo del proyecto como son definiciones, aspectos del estudio de mercado, tamaño de la muestra, normativas sobre niveles acústicos y el software a utilizar. En el segundo capítulo, se detallará las etapas a seguir en la elaboración de encuestas para establecer los parámetros para las notificaciones, el desarrollo de la programación del Web Service y las pruebas del funcionamiento. El tercer capítulo contiene los resultados y análisis obtenidos de la aplicación de las encuestas, funcionamiento de Web Service, conclusiones y recomendaciones que se obtuvo al finalizar este componente.

**PALABRAS CLAVE:** Web Service, redes sociales, sensores acústicos, EMS, Python.

## **ABSTRACT**

This study deals with the design and elaboration of a notification Web Service of notifications to inform through social networks data coming from a network of acoustic sensors taken from the Environmental Monitoring System (EMS) that belong to the Australian company ARCS Group of the TULIP project, using for this, the Python software tool. The first chapter presents the theoretical bases for the development of the project, such as definitions, aspects of the market study, sample size formula, standards on acoustic levels and the software to be used. In the second chapter, the stages to follow in the elaboration of a survey to establish the parameters for the notifications, the development of the programming of the Web Service and performance tests. The third chapter details the results and analysis obtained from the application of the surveys, operation of the Web Service, conclusions and recommendations that were obtained at the end of this component.

**KEYWORDS:** Web Service, social networks, acoustic sensors, EMS, Python.

# 1 INTRODUCCIÓN

El desarrollo científico y tecnológico a través del tiempo ha dejado en evidencia ciertas problemáticas como es el caso de la contaminación ambiental. Entre las mayores consecuencias están: la contaminación del aire, suelo y agua, dependiendo del contaminante puede ser química, radioactiva, térmica, acústica, visual, lumínica, electromagnética, microbiológica y genéticos. Por ello, se ve la necesidad de regular estos factores por medio de normativas, ya que estas pueden traer efectos nocivos a corto, mediano o largo plazo en el medio ambiente y por lo tanto afectar la salud de los seres humanos. Este problema incide principalmente en los lugares más poblados y donde se da mayor movilidad.

A pesar de la existencia de normativas que regulen los diferentes tipos de contaminación, éstas, no se toman en cuenta o no se le da mayor importancia a la contaminación acústica, considerando que es un problema que puede causar paulatinamente riesgos en la salud de las personas dependiendo de la frecuencia de exposición y el nivel de presión sonora. Por tal motivo es fundamental conocer estos riesgos y dar posibles soluciones, entre las formas de monitoreo se puede plantear o proponer contratar periódicamente a empresas especializadas con respecto a la toma y regulación de niveles acústicos o también se puede disponer del uso de equipos de medición de ruido como sonómetros, sin embargo, para el uso de estos dispositivos es necesaria la presencia de personal capacitado para realizar dichas mediciones y mantenimientos programados, por ello, se puede optar por medidas preventivas para proteger contra la contaminación acústica.

Una solución eficiente, para automatizar y mejorar los procesos en el monitoreo de niveles acústicos es el presente proyecto, que consiste en colocar una red de sensores acústicos para detectar eventos de contaminación acústica, sistemas de comunicación inalámbricos, algoritmos de procesamientos de datos y visualización de los resultados mediante un Web Service de notificaciones en redes sociales basado en normativas.

El proyecto se ha dividido en tres componentes: la captura y transporte de datos; el procesamiento de datos y presentación en una interfaz gráfica; y la visualización y análisis de información para la toma de decisiones. En este estudio, se desarrollara el tercer componente del proyecto, que consiste la elaboración de encuestas para establecer los parámetros de visualización de gráficas, desarrollo de un Web Service de notificaciones que permita comunicar por medio de publicaciones de gráficas de eventos de alertas sonaras en la red social Whatsapp, además se presenta las mediciones diarias de la red de sensores acústicos tomados del Sistema de Monitorio Ambiental (EMS) que pertenecen

a la empresa australiana ARCS Group del proyecto TULIP y los clústeres generados por la segunda componente de procesamiento de datos en la red social Facebook.

## **1.1 Objetivo general**

Diseñar y elaborar un Web Service de notificaciones utilizando el programa Python, el mismo que servirá para informar los datos obtenidos previamente por la red de sensores acústicos a los usuarios, mediante redes sociales.

## **1.2 Objetivos específicos**

- Elaborar encuestas sobre Información Acústica para la obtención de parámetros para las notificaciones.
- Desarrollar un Web Service de notificación para comunicar a los usuarios sobre alertas por riesgo acústico.
- Aplicar el Web Service de notificación para recopilación diaria de señales acústicas para usuarios en redes sociales.

## **1.3 Alcance**

Para el presente proyecto se deberá elaborar encuestas mediante el software administrativo Google Forms, con el propósito de establecer los parámetros de visualización de las notificaciones del Web Service como son: periodo de tiempo, intervalo entre notificaciones, tipo de información, momento de publicación, tipo de red social y niveles de riesgo. Para el planteamiento de las preguntas de los cuestionarios se analizará la normativa proporcionada por la Organización Mundial de la Salud (OMS), sobre los riesgos acústicos para la salud humana, para la ejecución de la encuesta se determinará el tamaño de muestra de población; considerando el lugar de aplicación donde se requiere instalar la red de sensores acústicos y la recogida de información a monitorear.

Con la información obtenida mediante las encuestas se conocerá cuáles son los parámetros para las notificaciones, que permitirá realizar gráficas y publicaciones mediante el uso del Web Service de notificaciones.

El Web Service de notificaciones, se creará mediante la herramienta de software Python, primero procesará la información proveniente de la segunda componente del proyecto que previamente procesaría la información de la red de sensores acústicos; el programa después con los datos acústicos creará las gráficas y las notificaciones de eventos de alertas de sonido que pueda afectar a la salud humana con base en la normativa de la

OMS y posteriormente publicara estas, mediante la red social Whatsapp en un grupo cerrado creado para el proyecto.

A continuación, el Web Service de notificaciones recopilará la información diaria acústica, información de clústeres y pico máximo en el día para realizar las notificaciones de los datos obtenidos por la red de sensores hacia la red social Facebook en una página creada para el proyecto.

Las notificaciones servirán para informar a los diferentes niveles jerárquicos de organizaciones para la toma de decisiones.

Debido a que no se logró obtener información a tiempo correspondiente al primer componente del proyecto, referente a la captura de datos y transporte hacia un repositorio digital (base de datos), recogida por la implementación de una red de sensores acústicos en una empresa nacional, se actualizó y aprobó el Plan de Trabajo de Integración Curricular, para que en su lugar se permita utilizar la base de datos obtenida de una red de sensores acústicos tomados del Sistema de Monitorio Ambiental (EMS) que pertenecen a la empresa australiana ARCS Group del proyecto TULIP, lo cual nos indica que la red de sensores se encuentra ubicados en Australia en ambientes exteriores y permitiendo el avance del proyecto planteado.

Ya que no es posible aplicar las encuestas en el entorno donde se obtuvieron los datos, se tomará como referencia la empresa "*Indufrit Company S.A.*", ubicada en la ciudad de Quito, considerando que éste proyecto puede ser planteado en un futuro para la planta de producción de la empresa; esta industria tiene mandos de dirección a nivel gerencial y operativo, por lo que, se realizaran dos tipos de encuesta dirigidas a los diferentes mandos para obtener toda la información posible para el desarrollo del Web Service, objeto de este estudio.

## **1.4 Marco teórico**

Con el paso del tiempo las industrias, el transporte y la población global han ido creciendo, sin embargo, el ruido no se mostraba como un factor de importancia de manera general, pero debido al desarrollo científico e industrial de la humanidad se han demostrado los efectos dañinos que causa el ruido en la salud humana; ahí, parte la necesidad de estudiar y regular a través de normativas pertinentes que se encuentran en función del tipo de lugar y ambiente para catalogar las mediciones máximas con las cuales se debe regir, evitando así los problemas en la población a causa de ruido.

Para poder informar de los efectos acústicos a la población se da paso a la creación de una Web Service de Notificación usando software; las notificaciones tendrán establecidos parámetros basados en encuestas creadas por un software de formulación.

#### **1.4.1 Normativa para la contaminación acústica**

La contaminación acústica está presente en la cotidianidad de la sociedad actual, por tal motivo las autoridades deben controlar y eliminar este problema por el bien de la ciudadanía, basándose en el monitoreo de las diversas fuentes acústicas además de respetar las normas.

##### **1.4.1.1 Norma Técnica ISO 9612 (Acústica – Determinación de la exposición al ruido laboral, Método de ingeniería)**

ISO 9612 detalla una forma para medir el nivel de ruido en un entorno de trabajo y calcular el nivel de exposición de los trabajadores al ruido, para el desarrollo de esta norma se tiene como referencia las normativas técnicas: ISO 1999, ISO/TEC 98-3, IEC 60942:2003, IEC 61252 y IEC 61672-1:2002; se especifican tres etapas para la medición [1].

- **Análisis de la labor**

Se planifica la estrategia en función de la información obtenida de los trabajadores y el entorno de laboral.

- **Selección de la estrategia de medición y la medición**

Se detalla la situación laboral, el número de trabajadores, la intención de las mediciones, el tiempo de la jornada laboral y la cantidad de información obtenida, en función de esto se plantean las estrategias de medición: medición basada en la tarea, medición basada en el trabajo y medición de una jornada completa. Las mediciones se deben realizar con un valor principal y el nivel de presión sonora continua.

- **Tratamiento de errores e incertidumbres**

Se trata de compensar los errores e incertidumbres, en función de las fuentes de errores como son:

- Las variaciones en el trabajo diario, las condiciones de operación, la incertidumbre en la selección del lugar.
- Los instrumentos y la calibración
- La posición del micrófono

- Las contribuciones falsas
- Un análisis del trabajo mal hecho o no realizado.

#### 1.4.1.2 Guía de la Organización Mundial de la Salud (OMS)

La OMS presento una guía para la comunidad con pautas para el ruido ambiental, como son la medición y prevención de la contaminación acústica a nivel mundial. En la actualidad existen múltiples fuentes de contaminación acústica sean en áreas urbanas o rurales, que son un problema por la gran cantidad de fuentes de ruido en áreas urbanas en especial, por lo cual mediante un estudio y la colaboración de varios participantes se ha desarrollado la Tabla 1, donde se puede encontrar valores de referencia para el ruido en ambientes específicos [2].

**Tabla 1.** Valores de referencia para el ruido en entornos específicos [2]

<b>Ambientes específicos</b>	<b>Efecto (s) crítico (s) en la salud</b>	<b>LAeq [dB (A)]</b>	<b>Tiempo base [horas]</b>	<b>LAmáx rápido [dB]</b>
Sala de estar al aire libre	Molestias graves, diurnas y nocturnas	55	16	-
	Molestia moderada, diurna y nocturna	50	16	-
Vivienda, interior	Inteligibilidad del habla y molestias moderadas, durante el día y la noche	35	16	
Dormitorios interiores	Alteración del sueño, durante la noche.	30	8	45
Dormitorios exteriores	Alteración del sueño, ventana abierta (valores exteriores)	45	8	60
Salas de clases de la escuela y preescolares, en interiores	Inteligibilidad del habla, alteración de la extracción de información, comunicación de mensajes	35	Durante clase	-
Preescolares dormitorios, interior	Alteración del sueño	30	Tiempo de sueño	45

Patio de la escuela exterior	Molestia (fuente externa)	55	Durante el juego	-
Sala de hospital habitaciones, interiores	Alteración del sueño, durante la noche	30	8	40
	Alteración del sueño, durante el día y la noche.	30	16	-
Hospitales, tratamiento habitaciones, interiores	Interferencia con el descanso y la recuperación.	# 1		
Industrial, áreas comerciales y de tráfico, interiores y al aire libre	La discapacidad auditiva	70	24	110
Ceremonias, festivales y entretenimiento eventos	Discapacidad auditiva (usuarios: <5veces / año)	100	4	110
Direcciones públicas, en interiores y al aire libre	La discapacidad auditiva	85	1	110
Música y otros sonidos a través de auriculares / audífonos	Discapacidad auditiva (valor de campo libre)	85 # 4	1	110
Sonidos de impulso de juguetes, fuegos artificiales y armas de fuego	Discapacidad auditiva (adultos)	-	-	140 # 2
	Discapacidad auditiva (niños)	-	-	120 # 2
Al aire libre en un parque y zonas recreativas	Interrupción de la tranquilidad	# 3		

# 1: El nivel más bajo posible.

# 2: Pico de presión sonora (no LAF, máx.) medida a 100 mm del oído.

# 3: Deben preservarse las áreas exteriores silenciosas existentes y la proporción de ruido intrusivo el sonido de fondo natural debe mantenerse bajo.

# 4: Con auriculares, adaptados a valores de campo libre.



### **1.4.2 Google Forms**

Google Forms es una herramienta útil de software administrativo que nos permite realizar encuestas a compañeros, amigos, grupo de personas o población objetiva para recopilar información de forma ágil y sencilla. Una ventaja es que el formulario de Google puede vincularse a una hoja de cálculo permitiendo que la información obtenida se pueda analizar de la mejor forma estadística.

Otra de las ventajas es que se puede compartir la encuesta con cualquier otro usuario solamente enviando el link generado.

### **1.4.3 Tamaño de muestra para encuestas**

Para la aplicación de una encuesta se requiere solamente tomar una muestra de las personas que representen a la población total a ser investigada y tomar el nombre y tamaño de muestra, además se utiliza para investigación de mercado y representa el número de encuestas con respuestas completas recibidas. Esta muestra no se toma de forma aleatoria sino en función de diferentes factores estadísticos que se debe considerar:

- **Tamaño de la población**

Se debe considerar el total de las personas que conforman el grupo de población que se desea estudiar, para ello se debe saber claramente quién forma parte y quién no. Este valor de tamaño de población puede ser conocido o estimado.

- **Margen de error**

En toda investigación siempre se presentan errores, por lo que se determina un grado de error que sea aceptable para el estudio. El margen de error se puede dar en valores medios, entre la media de la muestra y la media de la población a estudiar, normalmente se establece un margen de error de  $\pm 5\%$ .

#### **Desviación estándar**

Es la variación de las respuestas individuales entre sí y la media, cuando la desviación estándar es baja las respuestas están cerca de la media y si es más alta las respuestas están dispersas, una desviación estándar buena para garantizar un tamaño de muestra que represente a la población se considera en 0,5.

- **Nivel de confianza**

Se establece a cuánta confianza se desea tener en que la media real se encontrará dentro del margen de error. Los intervalos de confianza más comunes son de un 90 %, 95 % y 99 % de confianza [3].

- **Determinar la puntuación Z**

Se convierte el nivel de confianza en una puntuación Z. Estas son las puntuaciones Z para los niveles de confianza más comunes [3]:

90 % – Puntuación Z = 1,645

95 % – Puntuación Z = 1,96

99% – Puntuación Z = 2,576

#### **1.4.3.1 Uso de la fórmula del tamaño de la muestra finita**

Se introduce la puntuación Z, la desviación estándar y el intervalo de confianza en una fórmula para obtener el tamaño de muestra cómo se presenta en la ecuación 1:

$$n = \frac{N * Z^2 * \sigma^2}{(N-1) * e^2 + Z^2 * \sigma^2} \quad (1)$$

Donde:

n= Tamaño de la muestra

N= Tamaño de la población

Z= Corresponde al valor dado del nivel de confianza

$\sigma$ = Desviación estándar

e= Error en la proporción de la muestra

Esta ecuación se utiliza para un tamaño de población finita. Si el universo a estudiar es muy pequeño, simplemente se asume el universo [4].

#### **1.4.4 Web Service**

Un Web Service es un interfaz entre dos máquinas electrónicas para comunicarse entre sí, estas máquinas pueden ser aplicaciones o sistemas donde existen protocolos abiertos y estándares para intercambiar datos. Las aplicaciones cuyos lenguajes de programa sean diferentes y funcionen en diferentes plataformas utilizan Web Service para intercambiar información en la red, por ejemplo, entre Windows y Python o JAVA y Linux que tienen estándares abiertos.

La ventaja principal de Web Service es que la comunicación no depende de una plataforma determinada, por lo que el cliente y el servidor mantendrán mínimas similitudes para poder comunicarse. Por lo tanto, la tecnología Web Service recurre a formatos estandarizados que interpretan todos los sistemas [5].

#### **1.4.4.1 Arquitectura de los Web Services**

Una de las principales características de esta arquitectura, es que los servicios que se ofrecen tienen que ser relacionados y acoplados al mínimo entre sí, y la comunicación entre servicios debe tener un mecanismo ligero como APIs (Application Programming Interfaces) basadas en HTTP.

Los sistemas de comunicación agnósticos como estos nos permiten desarrollar Web Service con el lenguaje y las herramientas que se ajuste a las demandas del proyecto, pero en este caso se debe escoger un lenguaje de alto nivel para las características particulares que se requiere aplicar, Python es el software que nos permite avanzar rápidamente ya que es un lenguaje de tipado dinámico y se acomoda al servicio a implementar.

Python es ampliamente utilizado, por lo que se tiene una gran base de programadores, combina múltiples paradigmas de programación como el orientado a objetos y el funcional. Además, su sintaxis está orientada a escribir menos líneas de código, pero más claras que otros lenguajes, lo que facilita enormemente su mantenimiento. Por otro lado, existen excelentes frameworks, bibliotecas muy asentadas y orientadas al desarrollo de APIs REST y sistemas de comunicación basados en HTTP, lo que lo hace perfecto para la construcción de microservicios [6].

#### **1.4.5 Python**

El software fue creado por Dutch en 1991 utilizando los lenguajes ABC, donde la mayoría de las funciones de ABC se convierten en Python y evitar confundirlos con C, ya que Python es gratuito como también sus extensiones y vienen con descargas básicas [7].

Python es considerado como entre los modelos de diseño funcional y orientado a objetos, debido al uso de características similares a los elementos del lenguaje C como son bibliotecas iteradas, definiciones, funciones y objetos que se usan muy fácilmente en la programación.

Se caracteriza por su escritura dinámica, debido a que consigue reasignar variables a diferentes tipos de datos, esto hace que Python sea muy flexible; se diferencia de otros lenguajes de programación que se escriben estáticamente.

Como sabemos, las cadenas son la parte importante de cualquiera de las diversas estructuras de datos. En Python, para crear una cadena, se puede usar comillas simples o comillas dobles, las cadenas se utilizan para almacenar varios registros como texto, nombres y variables. Es una secuencia que puede realizar alfabetos y otros elementos. Si hablamos de una secuencia de cadena como una sola palabra se registrará como una lista con la información de todos y cada uno de los elementos, también podemos usar la indexación con cadena y trabajar en posiciones particulares de letras [8].

#### **1.4.5.1 Manejo de archivos en Python**

Python utiliza objetos para interactuar con archivos externos, estos objetos pueden ser cualquier tipo de archivo de audio, texto, correos electrónicos, documentos de Excel, bases de datos, etc.

#### **1.4.5.2 Bibliotecas y Paquetes de Python**

La biblioteca estándar de Python es una colección de sintaxis y semántica fieles al lenguaje de Python con una distribución centralizada.

Python posee la capacidad de tener más de doscientos módulos principales, que se encuentran en el corazón de la biblioteca estándar, pero además de esto también puede acceder a una colección cada vez mayor de varios miles de componentes del Python Package Index (PyPI) [8].

Para instalar estas componentes se utiliza el comando *pip install paquetes*, PIP (Preferred Installer Program), la misma que se utiliza para instalar paquetes de software en lenguaje de Python y viene incluido en el software.

## 2 METODOLOGÍA

El desarrollo de Web Service de notificaciones tiene como propósito informar a los usuarios de redes sociales sobre el nivel de presión sonora de una red de sensores acústicos, para lo cual se simulará la entrada de datos de los sensores acústicos por medio de una base de datos tomados del Sistema de Monitoreo Ambiental (EMS) que pertenecen a la empresa australiana ARCS Group del proyecto TULIP, la cual será procesada en la segunda componente del proyecto entregando una base de datos con información de fecha y tiempo de las mediciones realizadas, sensores, sonido promedio, sonido pico, temperatura, humedad, coordenadas de los sensores y clústeres en función de ruido, temperatura y humedad.

Debido a que no es posible aplicar la encuesta en el lugar donde se tomaron los datos, se tomará como referencia la aplicación en la empresa Indufrit Company S.A. que próximamente tendrá una red de sensores acústicos.

Para la creación de las gráficas para las notificaciones de alertas, mediciones diarias de los sensores, clústeres y pico máximo en el día es necesario establecer parámetros con la información sobre los valores de referencia para ruido en entornos específicos de la OMS y la Norma ISO 9612 sobre Acústica por medio de las encuestas a la gente de los mandos de la empresa como son nivel gerencial y operativo, la cual con lleva a una serie de etapas:

- Elaboración de encuestas
- Cálculo del tamaño de la muestra para la aplicación de encuesta
- Presentación de la encuesta

Con los resultados de la encuesta establecidos se requiere la creación de un Web Service de notificaciones el mismo que se realizará con el software Python debido a que nos permite avanzar rápido por su lenguaje de tipado dinámico y se adapta al servicio a implementar, lo que nos lleva a otra serie de etapas las cuales son:

- Instalación de librerías de Python

- Diseño de código en Python para Web Service de notificaciones
- Pruebas de Web Service de notificaciones para uso adecuado
- Presentación de código

A continuación, se explicará el proceso llevado en cada una de las etapas ya mencionadas.

## **2.1 Elaboración de Encuestas**

Para conocer los requerimientos de las gráficas sobre los niveles acústicos es necesario la aplicación de una encuesta para obtener de manera estadística los parámetros de visualización que se deben representar según las actividades y necesidades de los usuarios en relación al ruido.

Esta encuesta será diseñada con el fin de poder aplicar a la empresa Indufrit Company S.A. lo que servirá como guía para los parámetros de visualización, dicha empresa tiene mandos de nivel gerencial y operativo, por lo que se realizara dos tipos de encuesta dirigidas a los diferentes mandos para que puedan ser contestadas sin inconveniente alguno.

Las preguntas elaboradas para la encuesta están enfocadas en base a la información de las normas acústicas antes planteadas, con lo cual determinaremos los parámetros de visualización como son periodo de tiempo en función de hornada laboral, intervalo de tiempo de entre notificaciones, el tipo de información para la publicación, momento de publicación, tipo de red social donde quiere recibir la notificación y niveles de riesgo.

Las encuestas serán de forma anónima, la información recolectada será con fines académicos

A continuación, se detallará cada pregunta

### **2.1.1 Tipo de empresa**

En esta parte de la encuesta se preguntará a los usuarios el tipo de industria, las industrias listadas son lugares donde es posible implementar la red de sensores acústicos, en caso de conocer otros lugares se colocó la función de otros.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se puede observar en la Figura 1.

1. Indique a que se dedica la empresa: \*

- Manufactura de productos alimenticios
- Manufactura de bienes
- Construcción
- Agropecuaria
- Petroquímica
- Otra...

**Figura 1.** Pregunta de Tipo de empresa

### 2.1.2 Tipo de Zona

Esta pregunta se formula para conocer la ubicación de la empresa ya que en función de su ubicación los niveles acústicos tienen sus respectivos límites

Esta pregunta va dirigida a nivel gerencial debido que es posible que el nivel operativo no conozca de estos datos, como se indica en la Figura 2.

2. La empresa está ubicada en: \*

- Zona residencial
- Zona residencial mixta
- Zona Industrial

**Figura 2.** Pregunta de Tipo de Zona

### 2.1.3 Puesto de trabajo

Para la aplicación de la encuesta en función de la muestra de población de la empresa es necesario conocer la sección donde trabaja y especificar su puesto de trabajo, con esto también se hace un control de identificación para conocer si los colaboradores llenaron la encuesta en función de los mandos a los que pertenecen.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se detalla en la Figura 3.

3. Seleccione su puesto de trabajo: \*

Gerencial

Operativo

Especificar su puesto de trabajo: \*

Texto de respuesta breve

**Figura 3.** Preguntas de puesto de trabajo

#### **2.1.4 Horarios de trabajo rotativos**

Para conocer la cantidad de personal que tiene un horario variable se realiza la pregunta de esta manera se considerará los niveles de riesgo en función de la cantidad de personal y el momento en el cual se puede realizar las notificaciones.

Esta pregunta se hace a la encuesta de nivel operativo debido que es posible que el nivel gerencial no tenga horarios rotativos, como se indica en la Figura 4.

3. ¿Trabaja en horarios rotativos? \*

Sí

No

**Figura 4.** Pregunta de Horarios de trabajo rotativos

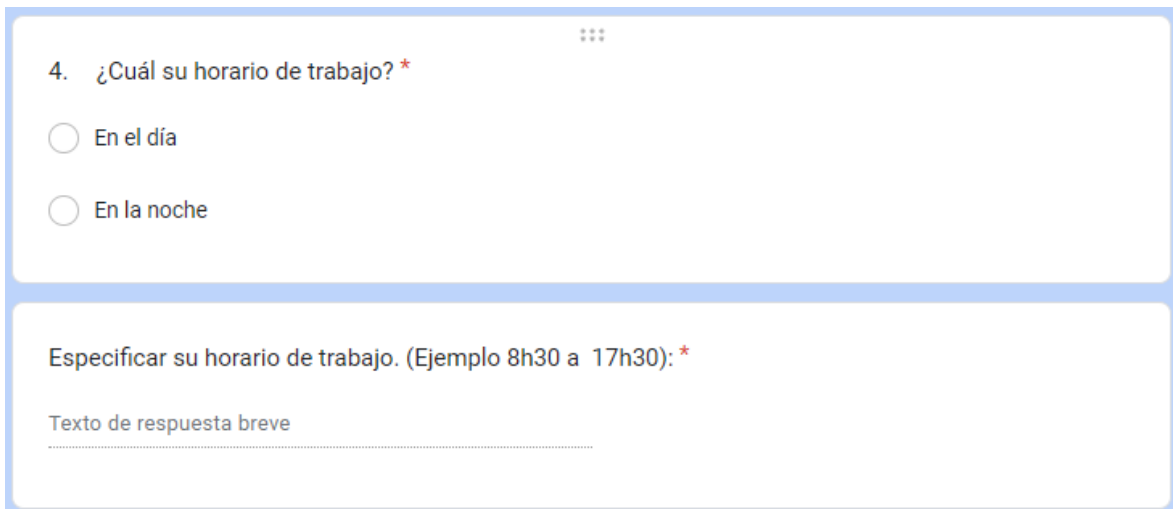
#### **2.1.5 Periodo de Tiempo**

Según las normas sobre ruido se toma en cuenta los horarios, debido a que en el día y la noche no se realiza la misma cantidad de actividades laborales, esto se debe a que la cantidad de ruido debe disminuir para no afectar con el descanso de la población.

Se establece el horario de trabajo para conocer los intervalos de tiempo para los niveles de ruido, este periodo servirá para ser representado en las notificaciones según el horario de trabajo que se establezca, incluyendo descansos y almuerzos.



Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se observa en la Figura 5.



4. ¿Cuál su horario de trabajo? \*

En el día

En la noche

Especificar su horario de trabajo. (Ejemplo 8h30 a 17h30): \*

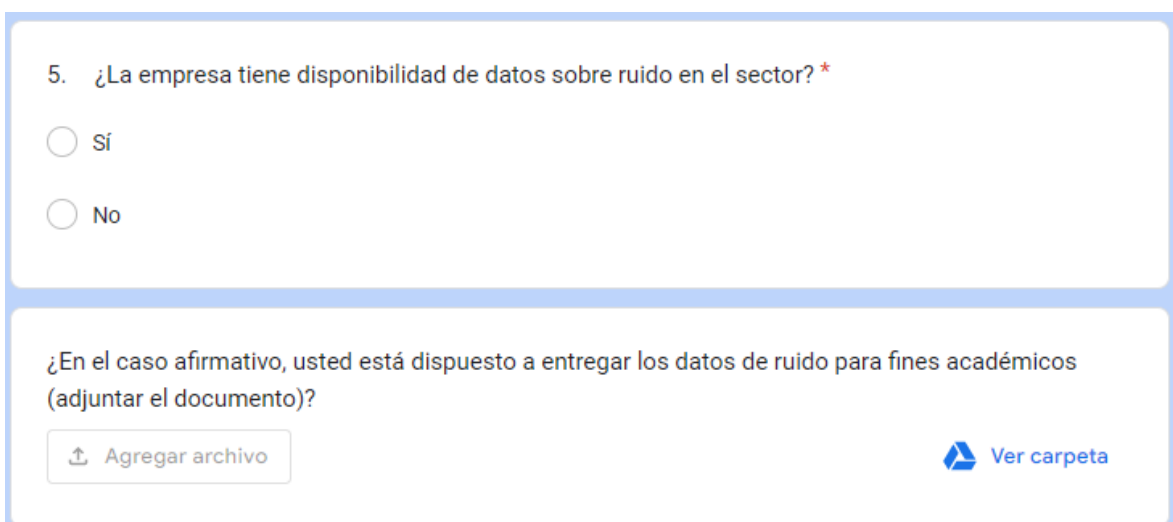
Texto de respuesta breve

**Figura 5.** Preguntas de Periodo de Tiempo

#### 2.1.6 Datos sobre ruido

Es posible que la empresa posea datos acústicos, por lo tanto, se requiere conocer si el nivel gerencial dispone de esa información de ser el caso de poseerla, puede compartir dichos datos para poder regular.

Esta pregunta se hace solo a la encuesta de nivel gerencial debido que es posible que el nivel operativo no conozca de estos datos, como muestra la Figura 6.



5. ¿La empresa tiene disponibilidad de datos sobre ruido en el sector? \*

Sí

No

¿En el caso afirmativo, usted está dispuesto a entregar los datos de ruido para fines académicos (adjuntar el documento)?

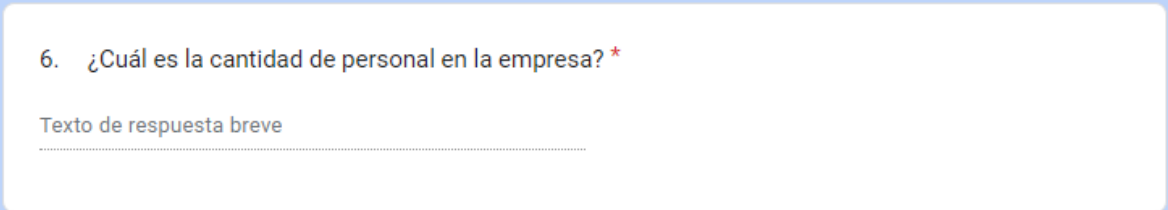
[Ver carpeta](#)

**Figura 6.** Preguntas de Datos sobre ruido

### 2.1.7 Cantidad de personal

Se fórmula para conocer la cantidad de personal de la empresa.

Esta pregunta se hace solo a la encuesta de nivel gerencial debido que es posible que el nivel operativo no conozca de estos datos, como muestra Figura 7.



6. ¿Cuál es la cantidad de personal en la empresa? \*

Texto de respuesta breve

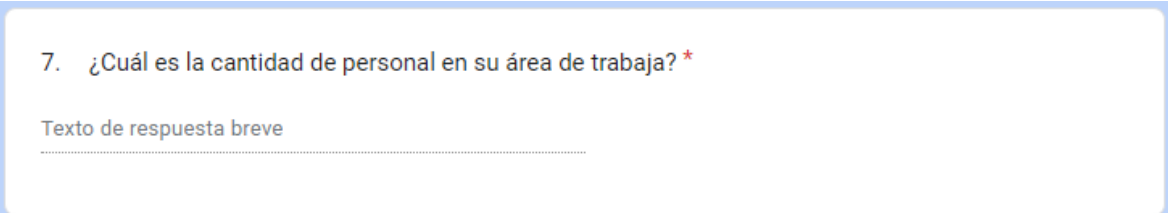
.....

**Figura 7.** Pregunta de Cantidad de Personal

### 2.1.8 Cantidad de personal por área

La cantidad de ruido de una empresa puede depender de varios factores como el número de trabajadores.

Esta pregunta se hace solo a la encuesta de nivel gerencial debido que los mandos gerenciales son los únicos que saben la cantidad de personal de su propia área, como se observa en la Figura 8.



7. ¿Cuál es la cantidad de personal en su área de trabaja? \*

Texto de respuesta breve

.....

**Figura 8.** Pregunta de cantidad de personal por área

### 2.1.9 Cantidad de maquinaria

La cantidad de ruido en una empresa puede depender de varios factores como el número de maquinaria.

Esta pregunta se hace solo a la encuesta de nivel gerencial debido que los mandos gerenciales son los únicos que saben la cantidad de maquinaria en la empresa, como se indica en la Figura 9.

8. ¿Cuál es la cantidad de máquinas en la empresa? \*

Texto de respuesta breve

**Figura 9.** Pregunta de Cantidad de Maquinaria

### 2.1.10 Operación de Maquinaria

En este apartado se debe conocer quienes manipulan máquinas en la empresa y que tipo de maquinaria opera, ya que estas son las personas más propensas y se pueden ver afectadas por el ruido y serán los usuarios que estén más interesados que llegue la notificación.

Esta pregunta se hace solo a la encuesta de nivel operativo debido que los mandos gerenciales no manipulan maquinaria en la empresa, como muestra la Figura 10.

5. ¿Usted es operario de una máquina? \*

Si

No

Responder en caso que su respuesta anterior fue SI, especificar la maquinaria de la cual se encarga:

Texto de respuesta breve

**Figura 10.** Preguntas de Operación de Maquinaria

### 2.1.11 Tipo de Red Social

En esta parte de la encuesta se pregunta sobre el tipo de red social donde se desea recibir las notificaciones, aquí se puede escoger más de una red social si el usuario dispone, con esto se establece la red o redes sociales por la cual se hará el Web Service de notificaciones.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se detalla en la Figura 11.

9. Para el servicio de notificaciones, ¿en cuál red social prefiere recibir las publicaciones? \*

Facebook

Twitter

Instagram

**Figura 11.** Pregunta de Tipo de Red Social

### 2.1.12 Tipo de información para la publicación

Se procede a preguntar cómo desea recibir las notificaciones con información gráfica, escrita o combinada; se determinará la información que será publicada en las notificaciones.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se observa en la Figura 12.

10. Para el servicio de notificaciones en las redes sociales, ¿qué tipo de información desea en la publicación? \*

Información gráfica

Información escrita

Información gráfica y escrita

**Figura 12.** Pregunta de Tipo de información para la publicación

### 2.1.13 Intervalo de tiempo para recibir las Notificaciones

En esta sección se pregunta el intervalo de tiempo que se requiere recibir las notificaciones, con esto se determina el periodo de tiempo de acumulación para las notificaciones.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se demuestra en la Figura 13.

11. Para el servicio de notificaciones en las redes sociales, ¿cómo desearía recibir las publicaciones? \*

Diaria

Semanal

Mensual

**Figura 13.** Pregunta de Intervalo de tiempo para recibir las Notificaciones

#### **2.1.14 Momento para recibir las Notificaciones**

Se pregunta el intervalo de tiempo que se requiere recibir las notificaciones, con esto establece cuando se va hacer las notificaciones.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como indica la Figura 14.

12. Dentro de su horario laboral, ¿cuál es la hora de preferencia en la que desearía recibir la notificación de ruido?. Ejemplo (12h00 a 13h00) \*

Texto de respuesta breve

**Figura 14.** Pregunta de Momento para recibir las Notificaciones

#### **2.1.15 Alertas de niveles de ruido**

En este apartado se pregunta si desea recibir notificaciones de alerta instantánea por picos de ruido que pueden causar daños a la salud de las personas.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se observa en la Figura 15.

13. En caso de alerta en los niveles de ruido, desea recibir información en tiempo real (instantáneamente) \*

Sí

No

**Figura 15.** Pregunta de Alertas de niveles de ruido

### 2.1.16 Emisiones de nivel de ruido

En función de normas anteriormente establecidas se determinó que los niveles de ruido acústico se deben medir en decibeles en función del nivel de ruido o nivel de presión sonora continuo equivalente tanto de la frecuencia o tiempo, por lo tanto, esto se formula a los trabajadores de la empresa para saber sus necesidades.

En este contexto se establecerá las unidades para el periodo de tiempo y niveles de ruido.

Esta pregunta se hace solo a la encuesta de nivel gerencial debido que los mandos gerenciales son los que nos disponen como desean la información, como se muestra en la Figura 16.

14. Ecuador regula las emisiones de ruido en decibeles (dB), pero también se puede establecer en Nivel de Presión Sonora Continuo Equivalente (NPS eq), escoger la opción con la cual desea que se represente el ruido de la empresa. \*

Niveles de ruido (dB) vs. Tiempo

NPS eq [dB(A)] vs Tiempo

Niveles de ruido (dB) vs Frecuencia

NPS eq [dB(A)] vs Frecuencia

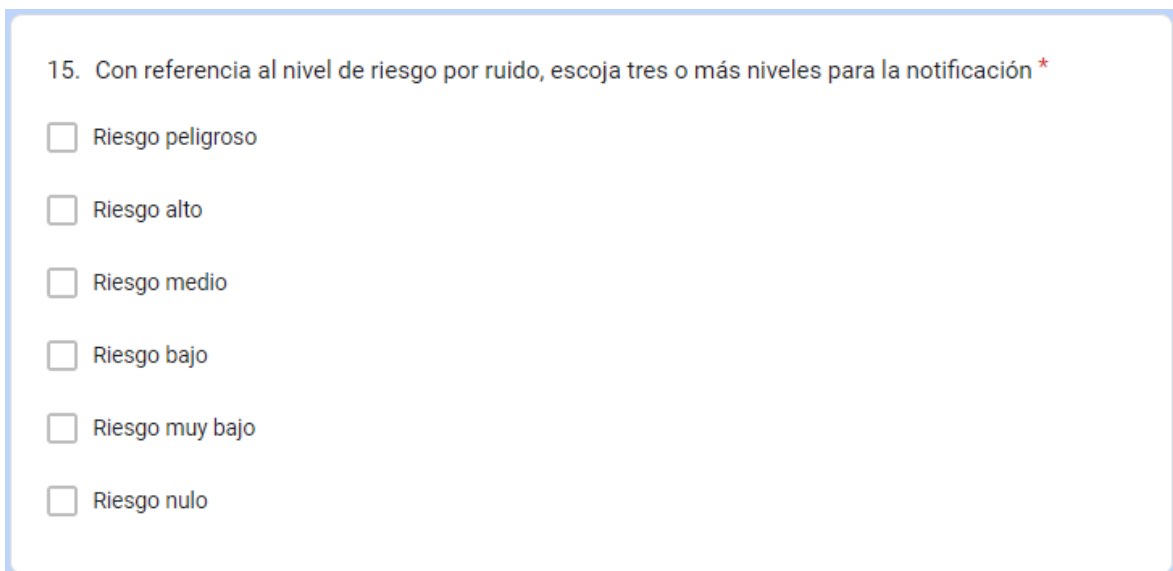
**Figura 16.** Pregunta de emisiones de nivel de ruido

### 2.1.17 Nivel de Riesgo de la salud

Esta pregunta nos permitirá determinar según los encuestados los niveles de riesgo que están interesados en conocer por medio de las notificaciones y así poder establecer niveles

de alerta para las notificaciones donde es posible escoger más de una opción, estos niveles se encuentran establecidos en función del lugar, entornos establecidos previamente en los datos de la OMS.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se puede mostrar en la Figura 17.



15. Con referencia al nivel de riesgo por ruido, escoja tres o más niveles para la notificación \*

- Riesgo peligroso
- Riesgo alto
- Riesgo medio
- Riesgo bajo
- Riesgo muy bajo
- Riesgo nulo

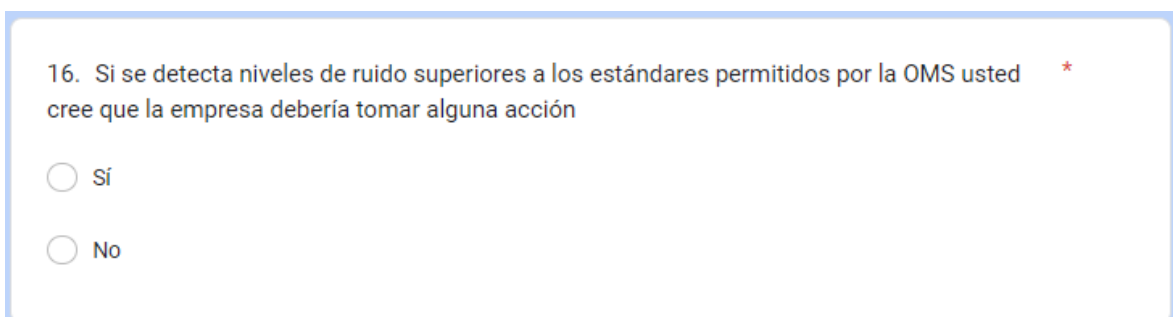
**Figura 17.** Pregunta de nivel de riesgo para la salud

### 2.1.18 Niveles superiores de ruido

En esta sección se pregunta si la empresa requiere tomar acciones en caso de detectar niveles superiores de ruido que pueden causar daños a la salud de las personas.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo

Esto se observa en la Figura 18.



16. Si se detecta niveles de ruido superiores a los estándares permitidos por la OMS usted cree que la empresa debería tomar alguna acción \*

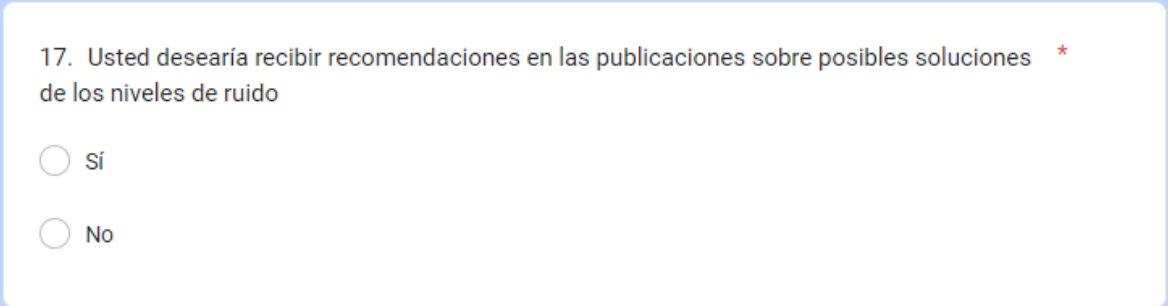
- Sí
- No

**Figura 18.** Pregunta de nivel de superiores de ruido

### 2.1.19 Posibles Soluciones en las notificaciones

Se pregunta si se requiere recibir soluciones de los niveles de ruido en las notificaciones.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como indica la Figura 19.



17. Usted desearía recibir recomendaciones en las publicaciones sobre posibles soluciones \*  
de los niveles de ruido

Sí

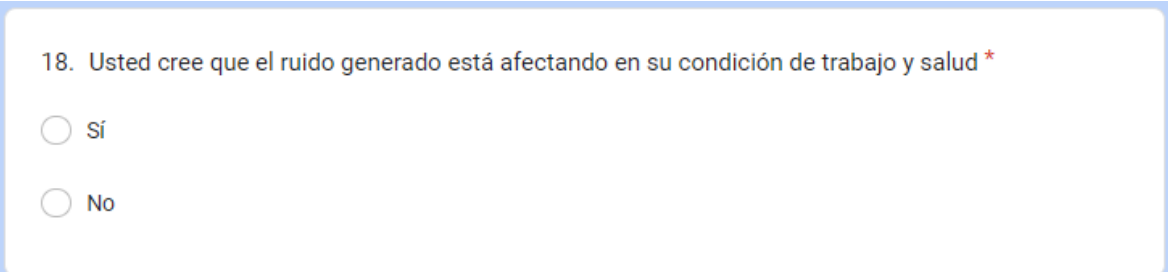
No

**Figura 19.** Pregunta de Posibles Soluciones en las notificaciones

### 2.1.20 Afecciones por ruido

Este apartado sirve para saber si el usuario se vio afectado en su salud debido al ruido.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como muestra la Figura 20.



18. Usted cree que el ruido generado está afectando en su condición de trabajo y salud \*

Sí

No

**Figura 20.** Pregunta de Afecciones de ruido

### 2.1.21 Afecciones por niveles riesgo

Sirve para conocer si el usuario informa sobre en que el nivel de riesgo por ruido cree que se encuentra.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como indica la Figura 21.



19. A qué nivel de riesgo por ruido cree que se encuentra expuesto \*

Riesgo alto

Riesgo medio

Riesgo bajo

**Figura 21.** Pregunta de Afecciones de nivel de riesgo

### 2.1.22 Problemas debido al ruido

Esta pregunta permite saber específicamente del usuario el tipo de problemas por ruido que adolece o le puede estar afectando, puede escoger más de una opción, en caso de ser afectado por algún problema que no se menciona en la pregunta puede poner en la opción otros.

Esta pregunta se hace tanto a la encuesta de nivel gerencial y operativo, como se observa en Figura 22.

20. Usted ha tenido algún problema por el ruido, ¿qué problemas a tenido? \*

Falta de sueño

Dolor de cabeza

Bajo rendimiento en el trabajo

Ninguno

Otra...

**Figura 22.** Pregunta de Problemas debido al Ruido

### 2.1.23 Costo de sistema de red de sensores con notificaciones

Esta pregunta solo se realiza en la encuesta a nivel gerencial debido que ellos son los encargados de empresa y son capaces de darle un valor económico al sistema de la red de sensores con notificaciones, como se determina en la Figura 23.

21. ¿Cuánto estaría dispuesta la empresa a pagar por tener este sistema (Dólares)? \*

Texto de respuesta breve

Figura 23. Pregunta de Costo del Sistema

## 2.2 Cálculo para tamaño de muestra para aplicar encuesta

La muestra que se seleccionó para las encuestas está en función del tamaño de la población que existe en la empresa Indufrit Company S.A. es de 15 personas, esta empresa nos servirá como guía para establecer los parámetros para las notificaciones y para el cálculo del tamaño de la muestra se utiliza la ecuación 1 planteada anteriormente:

Donde:

n= Tamaño de la muestra

N= Tamaño de la población (15)

Z= Corresponde al valor dado del nivel de confianza del 95% que es igual a 1,96

$\sigma$ = Desviación estándar buena (0,5)

e= Error en la proporción de la muestra (0,05)

$$n = \frac{N * Z^2 * \sigma^2}{(N-1) * e^2 + Z^2 * \sigma^2} \quad (1)$$

$$n = \frac{15 * (1,96)^2 * (0,5)^2}{(15 - 1) * (0,05)^2 + (1,96)^2 * (0,5)^2}$$

$$n = 14$$

Se deben realizar 14 encuestas, partiendo de una población de 15, pero debido a que el valor de la muestra es muy pequeño se considera realizar la encuesta a los 15 trabajadores de la empresa.

## 2.3 Presentación de Encuesta

Se procede a cargar las preguntas planteadas en el software administrativo de encuestas ya antes mencionado Google Forms.

Las dos encuestas realizadas tanto para el nivel gerencial y el nivel operativo se encuentran en los siguientes links:

- **Encuesta nivel Gerencial:**

[https://docs.google.com/forms/d/e/1FAIpQLSeQdVNoY5uQ9XsJEpH\\_KKZVKUZcDW8QgG75SzqMm3CRMvDsPg/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSeQdVNoY5uQ9XsJEpH_KKZVKUZcDW8QgG75SzqMm3CRMvDsPg/viewform?usp=sf_link)

- **Encuesta nivel Operativo:**

[https://docs.google.com/forms/d/e/1FAIpQLSdUmBX5SuDt0iExbqa0NjnmIv0NNRt39xOM0jBQVi8BthWjXA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdUmBX5SuDt0iExbqa0NjnmIv0NNRt39xOM0jBQVi8BthWjXA/viewform?usp=sf_link)

También se puede observar las encuestas en los ANEXO I Y II

## 2.4 Instalación de librerías de Python

Al instalar el software Python este viene con varias librerías por defecto, pero para la elaboración del Web Service para notificaciones se necesitarán la instalación de otras librerías que no vienen por defecto.

A continuación, se procederá a explicar los comandos para la instalación de las diferentes librerías de Python que necesitamos.

### 2.4.1 Instalación de datetime

Después de obtenerse el pip instalado, en una terminal se escribirá

```
pip install datetime
```

El módulo datetime proporciona clases para manipular fechas y horas, zonas horarias concretas que representan la base de datos de zonas horarias de la IANA y funciones generales relacionadas con el calendario [9].

### 2.4.2 Instalación de Pandas

Después de obtenerse el pip instalado, en una terminal se escribirá

```
pip install pandas
```

Pandas es una biblioteca de Python utilizada para trabajar con conjuntos de datos, tiene funciones para analizar, limpiar, explorar y manipular datos. El nombre "Pandas" tiene una

referencia tanto a "Panel Data" como a "Python Data Analysis" y fue creado por Wes McKinney en 2008 [10].

### **2.4.3 Instalación de PyAutoGUI**

Después de obtenerse el pip instalado, en una terminal se escribirá

```
pip install pyautogui
```

PyAutoGUI permite que sus scripts de Python controlen el mouse y el teclado para automatizar las interacciones con otras aplicaciones. La API está diseñada para ser simple. PyAutoGUI funciona en Windows, macOS y Linux, y se ejecuta en Python 2 y 3 [11].

### **2.4.4 Instalación de matplotlib**

Después de obtenerse el pip instalado, en una terminal se escribirá

```
pip install matplotlib
```

Esta librería se utiliza para análisis matemáticos

### **2.4.5 Instalación de selenium**

Después de obtenerse el pip instalado, en una terminal se escribirá:

```
pip install selenium
```

Selenium es una herramienta de código abierto, utilizado para realizar sitios web automatizados. Tiene la capacidad de hacer más trabajo de automatización para simplificar procesos ayudando a los ingenieros de pruebas [12].

### **2.4.6 Instalación de webdriver**

Después de obtenerse el pip instalado, en una terminal se escribirá

```
pip install webdriver_manager
```

Los enlaces de Selenium Python proporcionan una API conveniente para acceder a Selenium WebDrivers como Firefox, Ie, Chrome, Remote, Microsoft Edge [13].

### **2.4.7 Instalación de folium**

Después de obtenerse el pip instalado, en una terminal se escribirá

```
pip install folium
```

Folium facilita la visualización de datos que han sido manipulados en Python en un mapa interactivo. Permite tanto el enlace de datos a un mapa para visualizaciones como marcadores en el mapa [14].

## **2.5 Diseño de código en Python para Web Service de notificaciones**

El siguiente código servirá como Web Service de notificaciones el cual seguirá los parámetros establecidos por las normativas previamente indicadas y la guía de los resultados las encuestas realizadas a la empresa Indufrit Company S.A debido a que no se pudo realizar las encuestas en el lugar de donde se toman los datos de los sensores acústicos EMS ubicado en Australia.

La base de datos entrara en un procesamiento de datos correspondiente a la segunda componente del proyecto, donde se nos entregara como resultado la base de datos incluyendo columnas correspondientes al valor de clústeres y coordenadas de ubicación de los sensores; los clústeres están en creados en función del ruido, temperatura y humedad.

El código solicitará un día entre las fechas de la base de datos del EMS de las fechas comprendidas del año 2019 al 2020 que se encuentra en el ANEXO III. Después el código simulará la entrada de datos de la red de sensores acústicos por medio de la base.

Posteriormente el código implementará el proceso de notificaciones de alerta con información gráfica y escrita, para lo cual ingresará a la red social Whatsapp en un grupo creado para que los usuarios en el cual el código analizará la base datos de 6AM a las 2 PM del día ingresado y procederá hacer notificaciones, si los niveles acústicos son mayores o iguales 110 dB que están establecidos por la OMS para lugares externos que en este caso se encuentran los sensores, en la notificación se informara en que sensor ocurrió la alarma y la fecha con la hora del evento.

Después, el código al terminar de analizar la base de datos en el lapso de tiempo antes dicho, este procederá a ingresar a la red social Facebook donde realizará notificaciones diarias de los niveles de sonido en función del tiempo y de los clústeres en función del tiempo de cada sensor entre las 6AM y 2PM con información gráfica y escrita, además se indicará los picos de alerta sonido que se produjo y los peligros determinados por el valor de los clústeres a lo largo del día simulado.

En caso de no existir eventos en algún sensor, no se publicará su notificación, cabe mencionar que las notificaciones diarias y alerta se almacenarán en carpetas con el día y sensor.

Finalmente, el código elaborara un mapa marcando el lugar del sensor donde se produjo el mayor pico del sonido a lo largo del día simulado e ingresara a la red social Facebook

para realizar su respectiva notificación con información gráfica y escrita de ruido, temperatura y humedad.

A continuación, se explicará cada parte del código.

### 2.5.1 Importación de librerías de Python a código

En esta parte del código se procede a la importación de las librerías como son `datetime`, `pandas`, `pyautogui`, `matplotlib`, `selenium` y `folium` que se detalló previamente; las librerías de `time`, `shutil`, `webbrowser` y `pathlib` vienen por defecto instaladas en Python.

La librería `time` nos permite entender variables de tiempo como son fechas días y horas también permite hacer las pausas necesarias para que el navegador de internet procese los datos automáticos ingresados, por otro lado, la librería `shutil` y `pathlib` permite el manejo de archivos y carpetas además de rutas de los sistemas de archivos; la librería `webbrowser` permite controlar un navegador web que en este caso es Chrome a nuestra conveniencia. Debido a que necesitamos manipular botones en lenguaje HTML es necesario el uso `Selenium.webdriver.chrome.service` y `selenium.webdriver.common.by` que nos permitan también activar de forma remota el navegador, pero podremos usar estos botones en lenguaje HTML. Todo lo explicado se puede muestra en la Figura 24.

```
#Importar Librerias
import datetime
import time
import shutil
import folium
import pandas as pd
import pyautogui as pau
import webbrowser as wb
import gc
import matplotlib.pyplot as plt
from pathlib import Path
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
```

**Figura 24.** Importación de librerías Python en el código

### 2.5.2 Entrada de parámetros de día

En este apartado se procede al ingreso de datos del día que se encontrarán en formato días/mes/año, este dato estará en formato `string(str)` que es un formato de texto por lo cual se usara la función de `datetime.strptime()` para convertirlo en formato `datetime.datetime` que es un formato de tiempo y finalmente se usara `pd.to_datetime(dia).date()` de `pandas` para poder convertir el formato de tiempo en un formato de `datetime.date` que es un formato de solo fechas, como indica la Figura 25.

```
#Entrada de dia
dia = datetime.datetime.strptime(input('Digite el dia: '), '%d/%m/%y')
dia = pd.to_datetime(dia).date()
```

**Figura 25.** Entrada de parámetros de día

### 2.5.3 Creación de variables de sensores y carpetas de notificaciones

En esta sección se crea las variables que se almacenaran en formato str los nombres de los 10 sensores acústicos de la EMS y también se crearan las carpetas de notificaciones por sensor que tendrán siguiente formato de nombre como ejemplo *dia\_15/08/20\_ EMS Charlestown Public School*, para lo cual se usara la función *path (nombre de carpeta). mkdir ()* lo cual permitirá la creación de las carpetas en el directorio donde esta nuestro código de Python, la función *exist\_ok=True* permite sobreescritura en caso de que la carpeta ya exista.

Estas carpetas guardaran las gráficas de notificaciones de alerta en el día y sensor más la gráfica de notificación del día.

Se procede a crear la carpeta *Notificadora* donde también se almacenarán las notificaciones que nos servirá para cargar las gráficas de las notificaciones de alerta, como se observa en las Figuras 26 y 27.

```
#Entrada de dia
dia = datetime.datetime.strptime(input('Digite el dia: '), '%d/%m/%y')
dia = pd.to_datetime(dia).date()
#Creacion de variables de sensores y carpetas para guardar notifiaciones
sensor1 = "EMS Charlestown Public School"
sensor2 = "EMS Charlestown Square North Piazza Bus Stop"
sensor3 = "EMS Charlestown Square South Piazza"
sensor4 = "EMS Five Islands Roundabout Speers Point"
sensor5 = "EMS Gari Street Charlestown"
sensor6 = "EMS Lake Macquarie Art Gallery Jetty"
sensor7 = "EMS Pacific Hwy Crn Charlestown Road"
sensor8 = "EMS Pacific Hwy Crn Frederick Street Charlestown"
sensor9 = "EMS Speers Point Pool"
sensor10 = "EMS Whitebridge High School"
```

**Figura 26.** Creación de variables de sensores

```

Path('Notificadora').mkdir(exist_ok=True)
carp_tx1 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor1
Path('{0}'.format(carp_tx1)).mkdir(exist_ok=True)
carp_tx2 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor2
Path('{0}'.format(carp_tx2)).mkdir(exist_ok=True)
carp_tx3 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor3
Path('{0}'.format(carp_tx3)).mkdir(exist_ok=True)
carp_tx4 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor4
Path('{0}'.format(carp_tx4)).mkdir(exist_ok=True)
carp_tx5 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor5
Path('{0}'.format(carp_tx5)).mkdir(exist_ok=True)
carp_tx6 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor6
Path('{0}'.format(carp_tx6)).mkdir(exist_ok=True)
carp_tx7 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor7
Path('{0}'.format(carp_tx7)).mkdir(exist_ok=True)
carp_tx8 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor8
Path('{0}'.format(carp_tx8)).mkdir(exist_ok=True)
carp_tx9 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor9
Path('{0}'.format(carp_tx9)).mkdir(exist_ok=True)
carp_tx10 = "dia_" + dia.strftime("%d%m%y")+"_"+sensor10
Path('{0}'.format(carp_tx10)).mkdir(exist_ok=True)

```

Figura 27. Creación de variables de carpetas

#### 2.5.4 Apertura de navegador en WhatsApp

En esta parte se procede a abrir el navegador web Chrome en el *web.whatsapp.com* para empezar a realizar las notificaciones de alerta, en la primera línea se emplea para guardar el URL en una variable que permita que se abra el web Whatsapp en un grupo específico, el código de grupo se puede obtener en la parte de información de grupo y enlace de invitación al grupo como se muestra en la Figura 28.



Figura 28. Código de grupo

La línea que se encuentra comentada nos permite guardar el URL en una variable para abrir el navegador en web Whatsapp en una persona específica solamente se necesita el número de telefónico.



Después se usa la función de webbrowser conocida como *wb.open ()* que nos permite abrir el URL almacenado en la variable antes explicada, como se indica en la Figura 29.

```
#Abriendo Navegador para Whatapp
url2 = "https://web.whatsapp.com/accept?code=CyqU8yvcl6c7UtzmlwGSTzp"
#url2 = "https://web.whatsapp.com/send?phone=+5939951102697"
wb.open(url2)
```

**Figura 29.** Abriendo navegador para Whatsapp

### 2.5.5 Lectura de base de datos

Se procede a la lectura de la base de datos de la EMS, primero se escribe un mensaje usando la función *print* para saber si ya está leyendo la base de datos, usando la función *pd.read\_excel(base de datos)* de pandas, leemos y guardamos la base de datos en una variable, se debe escribir su ubicación en la computadora.

Después para mejor el análisis procedemos a dividir en nuevas columnas de datos las fechas con las horas de la base datos para lo cual usamos la función *pd.to\_datetime (columna de fechas y horas).dt. date* y *pd.to\_datetime (columna de fechas y horas).dt. time* dividiendo en columnas de solo fechas y horas en formatos *datetime. date* y *datetime. Time*, como se indica en la Figura 30.

```
#Lectura de la base de datos
print("Leyendo base de datos ....")
ah = "D:\Documentos\TESIS\core\environmental-monitoring-system-historical.xlsx"
dts = pd.read_excel(ah)
dts['Dates'] = pd.to_datetime(dts['Time']).dt.date
dts['Times'] = pd.to_datetime(dts['Time']).dt.time
```

**Figura 30.** Lectura de base de datos

### 2.5.6 Parámetros de inicio y final del día

En esta parte se crean variables para almacenar el tiempo de inicio y final del día en función de los parámetros establecidos por la encuesta, para lo cual se utiliza la función *datetime.time (horas, minutos, segundos)* de *datetime*, estos parámetros están en formato *datetime.time*, como muestra la Figura 31.

```
#Creacion de variables para fijar tiempo de inicio y final de dia
timeini = datetime.time(6,0)
# Para dia completo timeini = datetime.time(0,0)
timefin = datetime.time(14,0)
#Para dia completo timefin = datetime.time(23,59,59)
```

**Figura 31.** Parámetros de inicio y final del día

### 2.5.7 Filtración de base de datos

En esta parte del código se procede a filtrar la base de datos a nuestra conveniencia para lo cual se filtra en función del día ingresado con el tiempo de inicio y final del día, para lo cual se crea una lista de tipo dataframe de base de datos usando la función `dts.loc()`, todo esto se muestra en la Figura 32, el cual debido al anterior filtrado tiene sus datos *index* al azar por lo que se debe realizar un reseteo de los datos de la lista usando la función `lista.reset_index(drop=True, inplace=True)`, esto se hace con el fin de obtener un mejor análisis del listado filtrado, los *index* son una columna de números que corresponde al indicador de cada dato del dataframe creado por la base de datos leída, como muestra la Figura 32.

```
#Filtrado de datos
lista = dts.loc[(dts['Dates'] == dia) & (dts['Times'] >= timeini)
               & (dts['Times'] <= timefin)]
lista.reset_index(drop=True, inplace=True)
```

Figura 32. Filtración de base de datos

### 2.5.8 Creación de variables para lazos

Se crea las diferentes variables antes de entrar a los diferentes lazos creados para el análisis de datos para notificaciones de alerta, todas estas variables se explicarán en los próximos postulados.

La variable *maximus* usando la función de `lista.index.stop - 1` nos permite determinar el número total de *index* del nuevo listado filtrado para poder analizar en los diferentes lazos

Lo mencionado se puede observar en las Figuras 33 y 34.

```
#Revisión de datos para casos de notificaciones de alerta
#Creación de variables para lazos
maximus = lista.index.stop - 1
coind1 = []
coind2 = []
coind3 = []
coind4 = []
coind5 = []
coind6 = []
coind7 = []
coind8 = []
coind9 = []
coind10 = []
```

Figura 33. Creación de variables para los diferentes lazos 1

```

contador = 0
maxidx1 = 0
maxidx2 = 0
maxidx3 = 0
maxidx4 = 0
maxidx5 = 0
maxidx6 = 0
maxidx7 = 0
maxidx8 = 0
maxidx9 = 0
maxidx10 = 0
cronos1 = " "
cronos2 = " "
cronos3 = " "
cronos4 = " "
cronos5 = " "
cronos6 = " "
cronos7 = " "
cronos8 = " "
cronos9 = " "
cronos10 = " "

```

**Figura 34.** Creación de variables para los diferentes lazos 2

### 2.5.9 Lazo primario de análisis

En esta parte del código se aplica un lazo *for* para poder analizar cada uno de los datos del listado filtrado previamente por el código; al momento del filtrado los datos se ordenaron de tal manera se establece que los últimos datos son las primeras horas del día y los primeros datos son los últimos del día por lo cual se realiza el análisis usando los *index* de cada dato del mayor al menor.

La variable *real* nos permite ver los *index* como si fueran de menor a mayor.

La variable *sensor\_a* usando la función *lista.loc(i, 'Device Name')* permite saber por medio de los *índex*, en la columna de los 10 sensores que pertenece estos datos para después poder hacer otro análisis, como se indica en la Figura 35.

```

# Lazo primario de analisis
for i in range(maximus,-1,-1):
    real = maximus - i
    sensor_a= lista.loc[i, 'Device Name']

```

**Figura 35.** Lazo primario de análisis

### 2.5.10 Análisis por sensor

En esta parte del código va dentro del lazo primario, aquí se analiza usando la función *if* si la variable *sensor\_a* es igual a un sensor

Si es igual pasa a las sus respectivas condiciones donde se usa las variables previamente creadas, las variables *coind1* nos permite almacenar los diferentes *index* que corresponden al primer sensor, la variable *maxidx1* por cada lazo se resetea e inicia para poder usar la función *len()* y saber el número de elementos totales de la variable *coind1* que nos servirá después para otro análisis.

En caso de no cumplir la condición, el análisis pasa a otro comando *elif* hasta que coincida con su respectivo sensor lo cual significa que esta parte se repite para los diferentes sensores y tiene sus respectivas variables, como muestra la Figura 36.

```
# Lazo primario de analisis
for i in range(maximus,-1,-1):
    real = maximus - i
    sensor_a= lista.loc[i,'Device Name']
    #Sensor 1
    if sensor_a == sensor1:
        coind1 = coind1 +[i]
        maxidx1 = 0
        maxidx1 = len(coind1)
```

Figura 36. Tipo de sensor

### 2.5.11 Análisis para alertas por sensor

Este apartado del código va dentro de las condiciones de la función *if* previamente indicada, aquí se hace el análisis para las alertas de sonido usando un comando *if*, en caso de que el dato correspondiente a los picos de sonido con su respectivo *index* sea mayor o igual a 110 dB previamente explicados como momentos en los que se debe generar la alerta de sonido.

En caso de que el valor sea mayor o igual, aumentara el valor de la variable *contador* que después será explicada su función, también se guardara el valor de la fecha en la cual ocurrió la alerta de ruido en la variable *date\_a* usando su respectivo *index* con la función *lista.loc [i, 'Time']* de dataframe, de igual manera se guardara el valor en dB del sonido en el cual ocurrió la alerta de ruido en la variable *n\_alerta\_0* usando su respectivo *index* con la función *lista.loc [i, 'Peak Sound (dBA)].item()* de dataframe, también guardara el valor de las horas en la cual ocurrió la alerta de ruido en la variable *timer\_0* usando su respectivo *index* con la función *lista.loc [i, 'Times'].strftime("%H:%M:%S")* de dataframe dando un formato str del tiempo, se crean las variables *n\_alerta1* y *n\_alerta2* de tipo filas, estas se utilizaran después para graficar por lo que el primer valor que se carga será el de la variable *n\_alerta0* y se crea la variable *timer* de tipo filas que será utilizado después para graficar y su primer valor en cargarse será el de *timer\_0*.

En caso de no cumplir la condición significa que el nivel de ruido está dentro del rango donde no produce daños por lo cual no es necesario la creación de alertas y salta al inicio del lazo for primario para analizar el siguiente dato.

Todo lo explicado en este apartado se repite en los diferentes comandos if de sensores y tiene sus respectivas variables, como se muestra en la Figura 37.

```
#Sensor 1
if sensor_a == sensor1:
    coind1 = coind1 + [i]
    maxidx1 = 0
    maxidx1 = len(coind1)
    #IF de alertas
    if lista.loc[i, 'Peak Sound (dBA)'] >= 110:
        contador = contador + 1
        date_a= lista.loc[i, 'Time']
        n_alerta_0 = lista.loc[i, 'Peak Sound (dBA)'].item()
        timer_0 = lista.loc[i, 'Times'].strftime("%H:%M:%S")
        n_alerta_1 = [n_alerta_0]
        n_alerta_2 = [n_alerta_0]
        timer = [timer_0]
```

Figura 37. Alerta de alerta por sensor

### 2.5.12 Rango de gráficas de alerta

Esta parte del código va dentro de las condiciones de la función *if* para alertas previamente mostradas, el análisis para los rangos de gráficas de alertas de sonido nos permite graficar los anteriores 5 valores de nivel de sonido y el momento de ocurrir un pico de alerta de sonido, pudiendo existir casos donde el primer valor analizado sea una alerta lo cual significa que no hay valores anteriores para hacer la gráfica de alerta, para lo que se hace el uso de varios comandos *if* y *elif* donde usando la variable *maxidx1* correspondiente al número de datos totales analizados a este sensor, podemos ver si existen valores anteriores para poder realizar las gráficas.

Una vez determinado cuantos valores anteriores existen por medio de los comandos *if* este accede a un lazo *for* que encuentra en un rango de los valores *index* anteriores pertenecientes a este sensor, lo que nos permite cargar los valores de sonido anteriores a la alerta de sonido en la variable *n\_alerta\_1* de tipo fila la cual nos permitirá hacer la primera gráfica de sonido en la alerta.

El siguiente comando *if* y *else* tiene la función de graficar las anteriores alertas de sonido para lo cual carga los valores alertas sonido y en caso de no ser alerta coloca un cero para que este no interfiera en las gráficas, estos valores se cargan en la variable *n\_alerta\_2* de tipo fila la cual nos permitirá hacer la segunda gráfica en la alerta.

Después cargamos los valores de tiempo anteriores a la alerta de sonido en la variable *timer* de tipo fila la cual nos permite hacer las dos gráficas en la alerta.

Antes de pasar, al graficar limitamos el eje y de la gráfica usando máximos y mínimos de la variable *n\_alerta\_1* para que se vea más estética las gráficas para lo cual usamos los comandos *max ()* y *min ()*.

Todo lo explicado en este apartado se repite en los diferentes comandos if de sensores y tiene sus respectivas variables, como se muestran en la Figuras 38 y 39.

```
#IF de alertas
if lista.loc[i,'Peak Sound (dBA)'] >= 110:
    contador = contador + 1
    date_a= lista.loc[i,'Time']
    n_alerta_0 = lista.loc[i,'Peak Sound (dBA)'].item()
    timer_0 = lista.loc[i,'Times'].strftime("%H:%M:%S")
    n_alerta_1 = [n_alerta_0]
    n_alerta_2 = [n_alerta_0]
    timer = [timer_0]
    if maxidx1 > 5:
        for m in [coind1[maxidx1-2], coind1[maxidx1-3], coind1[maxidx1-4],
                  coind1[maxidx1-5], coind1[maxidx1-6]]:
            j = lista.loc[m,'Peak Sound (dBA)']
            j = j.item()
            n_alerta_1 = [j]+ n_alerta_1
            if j >= 110 :
                n_alerta_2 = [j]+ n_alerta_2
            else:
                n_alerta_2 = [0]+ n_alerta_2
            k = lista.loc[m,'Times'].strftime("%H:%M:%S")
            timer = [k]+ timer
    elif maxidx1 > 4:
        for m in [coind1[maxidx1-2], coind1[maxidx1-3], coind1[maxidx1-4],
                  coind1[maxidx1-5]]:
            j = lista.loc[m,'Peak Sound (dBA)']
            j = j.item()
            n_alerta_1 = [j]+ n_alerta_1
            if j >= 110 :
                n_alerta_2 = [j]+ n_alerta_2
            else:
                n_alerta_2 = [0]+ n_alerta_2
            k = lista.loc[m,'Times'].strftime("%H:%M:%S")
            timer = [k]+ timer
```

**Figura 38.** Rango de gráficas de alerta 1

```

elif maxidx1 > 3:
    for m in [coind1[maxidx1-2], coind1[maxidx1-3],coind1[maxidx1-4]]:
        j = lista.loc[m,'Peak Sound (dBA)']
        j = j.item()
        n_alerta_1 = [j]+ n_alerta_1
        if j >= 110 :
            n_alerta_2 = [j]+ n_alerta_2
        else:
            n_alerta_2 = [0]+ n_alerta_2
        k = lista.loc[m,'Times'].strftime("%H:%M:%S")
        timer = [k]+ timer
elif maxidx1 > 2:
    for m in [coind1[maxidx1-2], coind1[maxidx1-3]]:
        j = lista.loc[m,'Peak Sound (dBA)']
        j = j.item()
        n_alerta_1 = [j]+ n_alerta_1
        if j >= 110 :
            n_alerta_2 = [j]+ n_alerta_2
        else:
            n_alerta_2 = [0]+ n_alerta_2
        k = lista.loc[m,'Times'].strftime("%H:%M:%S")
        timer = [k]+ timer
elif maxidx1 > 1:
    j = lista.loc[coind1[maxidx1-2],'Peak Sound (dBA)']
    j = j.item()
    n_alerta_1 = [j]+ n_alerta_1
    if j >= 110 :
        n_alerta_2 = [j]+ n_alerta_2
    else:
        n_alerta_2 = [0]+ n_alerta_2
    k = lista.loc[coind1[maxidx1-2],'Times'].strftime("%H:%M:%S")
    timer = [k]+ timer
liminy = min(n_alerta_1) - 1
limaxy = max(n_alerta_1) + 1

```

Figura 39. Rango de gráficas de alerta 2

### 2.5.13 Graficador de alertas

Esta parte del código va dentro de las condiciones de la función *if* para alertas previamente mostradas después del apartado de rango de gráficas de alerta, en esta parte creamos las variables *figs* y *ax1* en las cuales se almacenarán los comandos para plotear la gráfica de alerta.

El comando *plt.rcParams ("figure. figsize")* nos permite dar el tamaño a la gráfica el momento de guardar como imagen.

Usando el comando *ax1.plot()* elaboramos una gráfica de líneas correspondiente a la primera gráfica de la alerta donde su eje x, y son las variables *timer* y *n\_alerta\_1* previamente explicadas, tiene marcador de tipo punto y leyenda con el nombre de "Sonido".

Usando el comando `ax1.scatter()` realizamos una gráfica de puntos correspondiente a la segunda gráfica de la alerta donde su eje x, y son las variables `timer` y `n_alerta_2` previamente explicadas, tiene marcador de color rojo aumentado el tamaño en 40 veces el tamaño de un punto normal y leyenda con el nombre de "Alerta de Sonido".

El comando `ax1.set_ylim ()` nos permite cargar los valores máximos y mínimos que obtuvimos previamente como límites en el eje y de la gráfica, la variable `leg` al almacenar el comando `ax1.legend()` que nos sirve para configurar el tamaño de letra y la ubicación de la leyenda en la gráfica.

El comando `plt.title ()` nos permite crear el título de la gráfica y también se especifica el tamaño del título, el título tendrá información del día y el sensor en donde se produjo la alerta de sonido, el comando `plt.grid ()` nos permite activar la rejilla de la gráfica que en este caso es verde de líneas entre cortadas y tamaño de línea de 0,5.

Los comandos `plt.xticks()` y `plt.yticks()` nos permiten cargar el tamaño de letra de los ejes, los comandos `plt.xlabel()` y `plt.ylabel()` nos permiten cargar el título de los ejes y el tamaño de letra.

Los comandos `plt.savefig()` nos permite guardar la gráfica creada en el computador, el primer comando la guarda en la carpeta de signada por día y sensor y el segundo comando la guarda en la carpeta notificadora para después realizar su respectiva notificación.

Todo lo explicado en este apartado se repite en los diferentes comandos if de sensores y tiene sus respectivas variables como se observa en las Figuras 40 y 41.

```
#Graficador de alertas de dia del sensor 1
figs = []
ax1 = []
figs = plt.figure()
ax1 = figs.add_subplot(111)
plt.rcParams["figure.figsize"] = (20, 12)
ax1.plot(timer,n_alerta_1,marker = '.',label = "Sonido")
ax1.scatter(timer,n_alerta_2,color= 'r',s=400,label = "Alerta de Sonido")
ax1.set_ylim(liminy,limaxy)
leg = ax1.legend(fontsize=16,bbox_to_anchor=(0.2, 1))
plt.title("Alerta de sonido medido el dia "
          + str(date_a.strftime("%d/%m/%y %H:%M:%S"))
          + " en el sensor " + sensor1,
          fontweight = 20, fontsize = 20)
```

**Figura 40.** Graficador de alertas 1



```

plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)
plt.xticks(fontweight = 15, fontsize = 15)
plt.yticks(fontweight = 15, fontsize = 15)
plt.ylabel("Sonido (dBA)", fontweight = 20, fontsize = 20)
plt.xlabel("Tiempo(hh:mm:ss)", fontweight = 20, fontsize = 20)
plt.savefig("D:\Documentos\TESIS\core\dia_{0}_{1}\webservice_alerta_{2}.png"
            .format(dia.strftime("%d%m%y"), sensor1, maximus-i))
plt.savefig(r"D:\Documentos\TESIS\core\notificadora\webservice_alerta_{0}.png"
            .format(real))

```

**Figura 41.** Graficador de alertas 2

### 2.5.14 Notificación de alertas

En este apartado del código va dentro de las condiciones de la función *if* para alertas previamente mostradas después del apartado de graficador de alerta, procedemos a guardar los valores de tiempo, cuando ocurrió las alertas de sonido en la variable *cronos1* la cual nos servirá después para las notificaciones al final del día.

Procedemos a llamar a la subfunción para notificaciones de WhatsApp denominada *notwpp()* en la cual cargamos el comentario para notificación de alerta donde se usa el *date\_a* y *sensor\_a* previamente ingresados, la gráfica de la alerta de sonido y la variable *contador*.

Al terminar la subfunción esta sale de la función *if* de alertas y regresa al lazo *for* primario y procede hacer esto con cada dato analizado.

Todo lo explicado en este apartado se repite en los diferentes comandos *if* de sensores y tiene sus respectivas variables, como se muestra la Figura 42.

```

#Notificacion
alertaf1 = str(date_a.strftime("%H:%M:%S"))
cronos1 = cronos1 + alertaf1 + ", "
notwpp("ALERTA !!! nivel alto de sonido superior a 110 dB"
      + " detectado en el sensor " + sensor_a + " en el dia "
      + str(date_a.strftime("%d/%m/%y %H:%M:%S")),
      "webservice_alerta_{0}.png".format(real), contador)

```

**Figura 42.** Notificación de alerta

### 2.5.15 Subfunción de Notificaciones para Whatsapp

En esta parte del código se ha creado la subfunción para las notificaciones en el grupo de Whatsapp, donde utilizando el comando *if* con la variable *contador* previamente ingresada se escoge la forma de cargar la imagen de la gráfica de alerta previamente guardada, la variable *contador* nos permite identificar si es la primera gráfica de alerta en subirse o no ya que al momento de ingresar en el navegador del Whatsapp se necesita el botón de cargar pero solo puede ser activado después de 12 activaciones automáticas del botón Tab para

lo cual usamos el comando *pau.press()* de la librería PyAutoGUI y el comando *time.sleep()* para crear retardos al momento de la activación automática del botón; en caso de no ser la primera grafica para llegar al botón de cargar se necesita de 2 activaciones automáticas del botón Tab.

Usando la activación automática del comando *pau.press()* activamos la flecha arriba y enter para escoger la carga de imagen, esperamos un momento con el comando *time.sleep()* para que se abra la ventana de abrir archivos y después usando el comando *pau.typewrite()* de la librería PyAutoGUI escribimos automáticamente el nombre de gráfica de alerta previamente creada, la ventana de abrir archivo se encuentra apuntando a la carpeta Notificadora previamente creada donde están ubicadas las gráficas, esperamos un momento con el comando *time.sleep()* para que se escriba el nombre y después usando *pau.press()* presionamos automáticamente enter para cargar la gráfica.

Usando *pau.typewrite()* cargamos el comentario de gráfica, esperamos un momento con el comando *time.sleep()* para que se escriba el comentario y después usando *pau.press()* presionamos automáticamente enter para completar que se publique la notificación.

Finalmente esperamos durante un momento usando el comando *time.sleep()* para que se cargue completamente la notificación en el Whatsapp y así la subfunción retorne, como se observa en la Figura 43.

```

#Notificador de Whatapp
def notwpp(comentario_w, imagen, conta):
    #Whatsapp

    #Comentar
    if conta == 1:
        for i in range(12):
            time.sleep(0.35)
            pau.press("tab")
            pau.press("enter")
        else:
            for i in range(2):
                time.sleep(0.35)
                pau.press("tab")
                pau.press("enter")
            for j in range(1):
                pau.press("up")
                pau.press("enter")
            time.sleep(1.5)
            pau.typewrite(imagen)
            time.sleep(0.5)
            pau.press("enter")
            time.sleep(1.5)
            pau.typewrite(comentario_w)
            time.sleep(0.5)
            pau.press("enter")
            time.sleep(6)
    return

```

Figura 43. Subfunción de notificaciones de Whatsapp

### 2.5.16 Cierre de navegador de Whatsapp

Después de analizar los datos de 6AM a 2PM del día ingresado y por sensor para alertas de sonido se procede a cerrar el navegador, para lo cual usamos el comando *pau.hotkey()* de la librería de PyAutoGUI que nos permite hacer activación de teclas combinadas para poder activar automáticamente el comando ALT+F4 para cerrar ventanas, esperamos un momento usando *time.sleep()* y presionamos automáticamente enter con *pau.press()* para cerrar el navegador, como se observa en la Figura 44.

```

#Cerrar navegador de Whatapp
pau.hotkey("alt", "f4")
time.sleep(0.35)
pau.press("enter")

```

Figura 44. Cierre de navegador de Whatsapp

### 2.5.17 Graficador de Notificaciones diarias de los niveles de sonido

En esta parte del código usando un comando *if* y *else* con la variable *maxidx1* previamente explicada identificamos si a lo largo del día existió alguna medición de sonido en el sensor,

en caso de haber valores se procede haciendo gráfico, guardado y notificación de la gráfica del día ingresa y en caso de no haber datos se procede a eliminar la carpeta creada previamente para las notificaciones de alerta por día y sensor ya que esta carpeta está vacía para lo cual usamos el comando `path().rmdir()` para eliminar carpetas vacías.

En el caso de encontrar datos, debemos crear la gráfica para lo cual de la base de datos cargados creamos dos listas de tipo dataframe usando la función `dfs.loc()` para filtrado, aquí filtramos los datos por el día, sensor, tiempo de inicio de día y final para los datos donde existieron alertas de sonido para la primera lista de alerta.

Para la segunda lista de sonido donde el filtrado es el mismo proceso, pero aquí se abarca todas las mediciones de sonido por día, sensor, tiempo de inicio de día y final.

Creamos variables para los límites del *eje* y en función del primer listado, se crea las variables `figs2` y `ax2` en las cuales se almacenarán los comandos para plotear la gráfica para notificación diaria, el comando `plt.rcParams ("figure. figsize")` nos permite dar el tamaño a la gráfica para el momento de guardado como imagen.

Usando el comando `plt.plot()` graficamos una gráfica de líneas correspondiente a la segunda lista de la alerta donde su *eje* x, y son la columna de fechas del listado y la columna de picos de sonido del listado, de color verde con marcador de tipo punto y leyenda con el nombre de "Sonido".

Usando el comando `plt.scatter()` graficamos una gráfica de puntos correspondiente a la primera lista de alerta donde su *eje* x, y son la columna de fechas del listado y la columna de picos de sonido del listado con alertas, tiene marcador de color rojo y leyenda con el nombre de "Alerta de Sonido".

El comando `plt.ylim()` nos permite cargar los valores máximos y mínimos que obtuvimos previamente como límites en el *eje* y de la gráfica, la variable `leg` al almacena el comando `ax2.legend()` que nos sirve para configurar el tamaño de letra y la ubicación de la leyenda en la gráfica.

El comando `plot.title ()` nos permite crear el título de la gráfica que está en función del día y el sensor en donde se produjo la alerta de sonido, el comando `plot.grid ()` nos permite activar la rejilla de la gráfica que en este caso es verde de líneas entre cortadas y tamaño de línea de 0,5.

Los comandos `plt.xticks()` y `plt.yticks()` nos permiten cargar el tamaño de letra de los ejes, el comando `plt.xticks()` también nos permite cambiar los nombres de las variables de las

columnas de fechas por los nombres de las variables de columnas de tiempo, los comandos `plt.xlabel()` y `plt.ylabel()` nos permiten cargar el título de los ejes y el tamaño de letra.

Los comandos `plt.savefig()` nos permite guardar la gráfica creada en el computador, el primer comando la guarda en la carpeta de signada por día y sensor y el segundo comando la guarda en la carpeta por defecto que apunta la red social Facebook para después realizar su respectiva notificación.

Todo esto se puede apreciar en las Figuras 45 y 46; todo lo explicado en este apartado se repite para los diferentes sensores y tiene sus respectivas variables.

```
#Graficador de final de dia del sensor 1
if maxidx1 >= 1:
    #Grafica 1
    lista2_1 = dts.loc[(dts['Device Name'] == sensor1) & (dts['Dates'] == dia)
                      & (dts['Peak Sound (dBA)'] >= 110)
                      & (dts['Times'] >= timeini) & (dts['Times'] <= timefin)]
    lista2_2 = dts.loc[(dts['Device Name'] == sensor1) & (dts['Dates'] == dia)
                      & (dts['Times'] >= timeini) & (dts['Times'] <= timefin)]
    liminy2 = min(lista2_2['Peak Sound (dBA)']) - 1
    limaxy2 = max(lista2_2['Peak Sound (dBA)']) + 1
    figs2 = plt.figure()
    ax2 = figs2.add_subplot(111)
    plt.rcParams['figure.figsize'] = (20, 12)
    plt.plot(lista2_2['Time'], lista2_2['Peak Sound (dBA)'], color='g',
            marker='.', label="Sonido")
    plt.scatter(lista2_1['Time'], lista2_1['Peak Sound (dBA)'], color='r',
              s=60, label="Alerta de Sonido")
    plt.ylim(liminy2, limaxy2)
    plt.xticks(lista2_2['Time'], lista2_2['Times'], rotation = 90,
              fontweight = 15, fontsize = 15)
    plt.yticks(fontweight = 15, fontsize = 15)
    leg = ax2.legend(fontsize=16, bbox_to_anchor=(0.2, 1))
    plt.title("Nivel de Sonido (dBA) vs tiempo medido el dia "
            + str(dia.strftime("%d/%m/%y")) + " en el sensor "
            + sensor1, fontweight = 20, fontsize = 20)
    plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)
    plt.ylabel("Sonido Pico (dBA)", fontweight = 20, fontsize = 20)
    plt.xlabel("Tiempo(hh:mm:ss)", fontweight = 20, fontsize = 20)
    #Guardar 1
    plt.savefig("D:\Documentos\TESIS\core\dia_{0}_{1}\webservice_finaldia.png"
              .format(dia.strftime("%d/%m/%y"), sensor1))
    plt.savefig(r"C:\Users\esteb\webservice_finaldia_1.png")
```

**Figura 45.** Graficador de notificaciones diarias de niveles de sonido

```
else:
    Path('{0}'.format(carp_tx1)).rmdir()
```

**Figura 46.** Eliminación de carpetas vacías

### 2.5.18 Notificaciones diarias de los niveles de sonido

En el siguiente apartado se encuentra en dentro del comando *if* y *else* con la variable *maxidx1* después de la parte graficador de notificaciones diarias de los niveles de sonido, en esta parte se crea el comentario que ira con la gráfica previamente realizada donde el comentario se forma por dos partes la primera parte del mensaje nos dice cuando se realizó esta revisión para lo cual usa el día ingresado y el sensor donde se tomó las mediciones, la segunda parte está en función de un comando *if* donde en función de la variable *cronos1* que se explicó previamente nos permitirá saber si a lo largo del día hubo alertas de sonido en caso de haber la segunda parte del mensaje nos informara a que horas del día se produjeron estas alertas y que se informe a las autoridades, en caso de no haber el mensaje nos informara sobre que los niveles de sonido son bajos en peligro.

Después procedemos a llamar a la subfunción para notificaciones de Facebook denominada *notfcb()* en la cual cargamos el nombre de la gráfica y el comentario.

Todo lo explicado en este apartado se repite para los diferentes sensores y tiene sus respectivas variables, como muestra la Figura 47.

```
#Notificacion 1
Comentario = "Nivel de Sonido Revisado el dia "
+ str(dia.strftime("%d/%m/%y")) + " en el sensor "
+sensor1+" esta notificacion a sido enviada a las "
+ str(timefin.strftime("%H:%M:%S"))+"\n"
if cronos1 == ' ':
    Comentario = Comentario
    + "Los niveles de sonido se encontrar"
    +"en niveles bajos de peligro acustico"
else:
    Comentario = Comentario + "Alerta de sonido detectada a las "
    + cronos1 + " por favor tomar estas alertas en cuenta, "
    +"en caso de algun problema por niveles de sonido "
    +"comunicarse con las autoridades encargadas"
ima_nombre = "webservice_finaldia_1.png"
notfcb(Comentario, ima_nombre)
```

Figura 47. Notificación diaria de niveles de sonido

### 2.5.19 Graficador de Notificaciones diarias de los clústeres

Este apartado se encuentra dentro del comando *if* y *else* con la variable *maxidx1* después de la parte de notificaciones diarias de los niveles de sonido, donde en esta parte creamos la gráfica para la notificación de clústeres en función del tiempo para lo cual usamos la segunda lista de sonido filtrada previamente utilizada.

Se crea las variables *figs2\_1* y *ax2\_1* en las cuales se almacenarán los comandos para plotear la gráfica para notificación diaria, el comando *plt.rcParams* ("*figure. figsize*") nos permite dar el tamaño a la gráfica para el momento de guardado como imagen.

Usando el comando *plt.plot()* graficamos una gráfica de líneas correspondiente a la segunda lista del sonido donde su *eje x*, y son la columna de fechas del listado y la columna de clústeres del listado, tiene marcador de color rojo y líneas punteadas leyenda.

El comando *plt.title ()* nos permite crear el título de la gráfica que está en función del sensor en donde se registraron los clústeres, el comando *plt.grid ()* nos permite activar la rejilla de la gráfica que en este caso es verde de líneas entre cortadas y tamaño de línea de 0,5.

Los comandos *plt.xticks()* y *plt.yticks()* nos permiten cargar el tamaño de letra de los ejes, el comando *plt.xticks()* también nos permite cambiar los nombres de las variables de las columnas de fechas por los nombres de las variables de columnas de tiempo, los comandos *plt.xlabel()* y *plt.ylabel()* nos permiten cargar el título de los ejes y el tamaño de letra.

Los comandos *plt.savefig()* nos permite guardar la gráfica creada en el computador, el primer comando la guarda en la carpeta de signada por día y sensor y el segundo comando la guarda en la carpeta por defecto que apunta la red social Facebook para después realizar su respectiva notificación.

Todo lo explicado en este apartado se repite para los diferentes sensores y tiene sus respectivas variables. Todo esto se puede apreciar en la Figura 48.

```
#Grafica 2
figs2_1 = plt.figure()
ax2_1 = figs2_1.add_subplot(111)
plt.rcParams['figure.figsize'] = (20, 12)
plt.plot(lista2_2['Time'],lista2_2['KMeans_Clusters'],'ro',
         lista2_2['Time'],lista2_2['KMeans_Clusters'],'b--')
plt.xticks(lista2_2['Time'],lista2_2['Times'],rotation = 90,
           fontweight = 15, fontsize = 15)
plt.yticks(fontweight = 15, fontsize = 15)
plt.title(sensor1, fontweight = 20, fontsize = 20)
plt.xlabel("Tiempo(hh:mm:ss)",fontweight = 20, fontsize = 20)
plt.ylabel('Clúster',fontweight = 20, fontsize = 20)
plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)
#Guardar 2
plt.savefig("D:\Documentos\TESIS\core\dia_{0}_{1}\webservice_cluster.png"
          .format(dia.strftime("%d%m%y"),sensor1))
plt.savefig(r"C:\Users\esteb\webservice_finaldia_1_2.png")
```

**Figura 48.** Graficador de notificaciones diarias de clústeres

### 2.5.20 Notificaciones diarias de los clústeres

En el siguiente apartado se encuentra en dentro del comando *if* y *else* con la variable *maxidx1* después de la parte graficador de notificaciones diarias de los clústeres, en esta parte se crea el comentario que ira con la gráfica previamente realizada donde el comentario se forma por el día ingresado y el sensor donde se tomó las mediciones, incluyendo la explicación del valor de los clústeres con respecto a la temperatura humedad y ruido.

Después procedemos a llamar a la subfunción para notificaciones de Facebook denominada *notfcb()* en la cual cargamos el nombre de la gráfica y el comentario.

Al terminar la subfunción esta sale de la función *if* y pasa a un apartado similar al notificador diario, pero correspondiente al siguiente sensor

Todo lo explicado en este apartado se repite para los diferentes sensores y tiene sus respectivas variables, como muestra la Figura 49.

```
#Notificacion 2
Comentario = "Análisis de clusteres generados por algoritmo K-Means "
+"para Temperatura, Humedad y Ruido en el día "
+ str(dia.strftime("%d/%m/%y"))+" en el sensor "+sensor1
+" esta notificacion a sido enviada a las "
+ str(timefin.strftime("%H:%M:%S"))+" donde:\n"
+"Si el cluster es 0 en la temperatura no hay cambios peligrosos,"
+"la humedad tiene concentracion alta tener cuidado en lugares cerrados "
+"con poca ventilacion y el ruido tienen niveles altos "
+"pero sin consecuencias\n"
+"Si el cluster es 1 en la temperatura no hay cambios peligrosos,"
+"la humedad tiene concentracion alta tener cuidado en lugares cerrados "
+"con poca ventilacion y el ruido tienen niveles altos "
+"puede causar lesiones al oido humano\n"
+"Si el cluster 2 por lo cual en la temperatura hay cambios peligrosos "
+"llegando al umbral de tolerancia humano 55°C,"
+"la humedad tiene concentracion alta tener cuidado en lugares cerrados "
+"con poca ventilacion y el ruido tienen niveles altos "
+"exposiciones cortas al ruido causa lesiones al oido humano\n"
+"Si el cluster 3 por lo cual en la temperatura no hay cambios peligrosos,"
+"la humedad tiene concentracion alta tener cuidado en lugares cerrados "
+"con poca ventilacion y el ruido tienen niveles peligrosos de exposicion "
+"puede causar lesiones irreversibles al oido humano "
ima_nombre = "webservice_finaldia_1_2.png"
notfcb(Comentario, ima_nombre)
```

**Figura 49.** Notificación diaria de clústeres



### 2.5.20.1 Subfunción para Notificaciones para Facebook

En esta aparte del código se creó la subfunción para las notificaciones en Facebook en una página de Facebook business donde usando el comentario y el nombre de la gráfica previamente ingresados se publicará la notificación.

Todo esto se explica en los siguientes apartados.

#### 2.5.20.1.1 Creación de variables para acceso a red social, usuario y contraseña

En el siguiente apartado del código se crean las variables donde se les asignará la dirección URL, usuario y contraseña de la página de Facebook, como se indica en la Figura 50.

```
def notfcb(comentario, imagen):  
    #Facebook  
  
    #Carga de datos de acceso  
    url1 = "https://m.facebook.com/home.php?tbua=1"  
    usr = 'pab.villenas12@gmail.com'  
    psw = '3512EPCo'
```

Figura 50. Creación de variables dirección, usuario y contraseña.

#### 2.5.20.1.2 Creación de selectores

Los selectores son códigos en lenguaje HTML contruidos con etiquetas, atributos y valores que identifican elementos web dentro de una página web.

La creación de selectores puede ser de tipo CSS y Xpath que trabajan de forma similar, su única diferencia es en su forma de construcción.

Para nuestro código los selectores nos permitirán buscar y utilizar en la página de Facebook los botones de inicio de sesión, para publicación, para cargar imagen, para ingresar a otras páginas, barras de texto de usuario, texto de contraseña, texto de comentario de imagen así cargaremos de manera automática las variables anteriormente creadas para acceder a la red social, realizar la notificación y finalmente publicarla.

Todos los selectores se los obtiene de la herramienta de desarrollador del navegador de página web de la red social de Facebook, como se observa en la Figura 51.

```

#Selectores
btton_ini = '#login_password_step_element > button'
btton_acc = '/html/body/div[1]/div/div[2]/div/div[2]/div[6]/div/a/div'
btton_pag = '/html/body/div[1]/div/div[4]/div/div/div/a[2]/div'
btton_pag_2 = '/html/body/div[1]/div/div[4]/div/div/div/div[4]'
btton_not = '/html/body/div[1]/div/div[4]/div/div[1]/div/div[3]'
+ '/div/div[2]/div[2]/div[2]/div/div/div[2]/div[3]'
btton_ima = '/html/body/div[2]/div[1]/div/div[5]/div[3]/form'
+ '/div[6]/div/button[1]'
btton_publish = '#composer-main-view-id > div.acw > div > div > button'
sel_usr = '#m_login_email'
sel_psw = '#m_login_password'
comen_post = '/html/body/div[2]/div[1]/div/div[5]/div[3]'
+ '/form/div[2]/textarea'

```

**Figura 51.** Selectores y Xpath

### 2.5.20.1.3 Acción automática para acceso a navegador de Chrome e ingreso a la dirección electrónica.

Antes de poder continuar con el código se debe descargar el archivo ejecutable chromedriver.exe que se lo puede encontrar en la página de drivers del navegador de Chrome.

Después se le asigna a la variable *naveg* el servicio del archivo ejecutable del webdriver del navegador Chrome por medio de la función *Service* importada de la librería *selenium.webdriver*.

La línea que se encuentra comentada permite asignar a la variable *naveg* el mismo servicio, pero para navegar en Microsoft Edge y teniendo igualmente su archivo ejecutable driver para este navegador.

A continuación, el comando *naveg.start()* nos permite abrir automáticamente el navegador de Chrome. Después se le asigna a la variable *driver* el control del navegador por medio de la función *webdriver.Remote()* importada de la librería de selenium, con la siguiente iteración *driver.maximize\_window()* sirve para maximizar la pantalla para poder apreciar mejor al navegador.

En la siguiente parte usando el comando *driver.get()* se carga la dirección URL que se desea acceder, que previamente se guardó en la variable *url1*, como se muestra en la Figura 52.

```

#Abrir navegador
naveg = Service('D:\Documentos\TESIS\core\chromedriver.exe')
#naveg = Service('D:\Documentos\TESIS\core\msedgedriver.exe')
naveg.start()
driver = webdriver.Remote(naveg.service_url)
driver.maximize_window()
driver.get(url1)

```

**Figura 52.** Acceso al navegador y carga de la dirección electrónica

#### 2.5.20.1.4 Acción automática para carga de datos de usuario y acceso a red social Facebook

En esta parte del apartado se procede al uso de los selectores anteriormente creados para lo cual se usará la función *find\_element(By)* importa se *selenium.webdriver.common.by* para selectores de tipo CSS que nos permite apuntar hacia los selectores los datos de acceso como son usuario y contraseña con la función *send\_keys()* y finalmente la activación del selector que posee el botón de inicio de sesión por medio de un click automático.

La función *time.sleep()* nos permite hacer un tiempo de pausa para que la página de la red social Facebook procese los datos ingresados y así los siguientes apartados del código no se activen innecesariamente antes de la apertura del muro de notificaciones.

Después usando la función *find\_element(By)* importa se *selenium.webdriver.common.by* para selectores de tipo Xpath nos permita ingresar a la página business para activar el botón de notificaciones con imágenes por medio de un click automático, como se indica en la Figura 53.

```

driver.find_element(By.CSS_SELECTOR,sel_usr).send_keys(usr)
driver.find_element(By.CSS_SELECTOR,sel_psw).send_keys(psw)
driver.find_element(By.CSS_SELECTOR,bton_ini).click()
time.sleep(4)
driver.find_element(By.XPATH,bton_acc).click()
time.sleep(4)
driver.find_element(By.XPATH,bton_pag).click()
time.sleep(4)
driver.find_element(By.XPATH,bton_pag_2).click()
time.sleep(4)
driver.find_element(By.XPATH,bton_not).click()
time.sleep(4)
driver.find_element(By.XPATH,bton_ima).click()

```

**Figura 53.** Carga de datos de usuario de empresa y acceso a la red social Facebook

#### 2.5.20.1.5 Acción para cargar la imagen

En esta parte de código se ejecuta un tiempo de espera con la función *time.sleep()* para que la ventana de cargar de imágenes se abra normalmente sin problemas, después usando la función *pau.typewrite()* se cargara el nombre de gráfica y finalmente usando

*pau.press()* presionara enter para cargar la gráfica, por defecto la página de Facebook abrirá la ventana de carga en la carpeta de usuarios del computador donde ya se encuentran las imágenes de las gráficas del día.

Se establece en periodo de retardo para que la gráfica se cargue bien en la notificación con la función *time.sleep()*, como se muestra en la Figura 54.

```
#Cargar imagen
time.sleep(5)
pau.typewrite(imagen)
pau.press("enter")
time.sleep(4.5)
```

**Figura 54.** Carga de gráfica

#### 2.5.20.1.6 Acción automática para publicación de la notificación

En esta parte del código nos disponemos a la colocación del comentario de la imagen de la gráfica, para lo cual usamos el selector que corresponde a la barra de texto del comentario con la función *find\_element(By)* importa se *selenium.webdriver.common.by* para selectores de tipo Xpath donde se colocara el mensaje que creamos y después usando el selector del botón de publicación la función *find\_element(By)* importa se *selenium.webdriver.common.by* para selectores de tipo CSS se daría un click automático permitiendo que se notifique en el muro de la página.

Se establece en periodo de retardo para que la publicación se cargue bien en el muro con la función *time.sleep()*, como se observa en la Figura 55.

```
#Comentar
driver.find_element(By.XPATH,comen_post).send_keys(comentario)
driver.find_element(By.CSS_SELECTOR,bton_publish).click()
time.sleep(7)
```

**Figura 55.** Código para comentar

#### 2.5.20.1.7 Acción para cerrado de página del navegador

Ya finalizada la publicación, el navegador se encuentra abierto entonces debemos cerrar el navegador para lo cual a las variables que se les cargo tanto el control y el acceso se les debe ordenar apagado de sus funciones, cerrando el navegador de Chrome y así la subfunción retorne, como se indica en la Figura 56.

```
#Cerrar navegador de Facebook
driver.quit()
navig.send_remote_shutdown_command()
return
```

**Figura 56.** Acciones para cerrado de navegador

### 2.5.21 Graficador y notificación de pico máximo de sonido

Una vez realizada las notificaciones diarias de niveles de sonido y clústeres para cada sensor, procedemos a realizar la gráfica del mapa marcando el lugar del sensor donde se dio el mayor pico de sonido en el día.

Procedemos creando el comentario para la notificación donde se colocará el día simulado, el tiempo de final de día y el sensor donde se dio el mayor pico de sonido, después colocamos datos sonido pico, temperatura, humedad y clúster correspondiente a este evento.

Creamos el mapa para lo cual utilizamos la función *folium.map()* de la librería folium previamente explicada donde colocamos las coordenados en latitud y longitud donde queremos centrar el mapa y la cantidad de acerca para poder apreciar el lugar donde está el sensor, después utilizamos la función *folium.marker()* de la liberara folium para colocar los datos del marcador para lo cual colocamos las coordenadas en latitud y longitud de la ubicación del sensor donde se produjo el mayor pico de sonido, y finalmente agregamos estos datos a la variable *mapa* donde guardamos antes los valores del mapa utilizando la función *add\_to()*.

Guardamos el mapa usando la función *mapa.save()*, esta función guarda los mapas de este tipo en *archivos.html* para poder acceder e interactuar con el mapa.

Debido a que no se puede cargar imágenes con formato HTML procedemos hacer una transformación para que la imagen tenga formato PNG para poder hacer su respectiva notificación; procedemos a acceso a navegador de Chrome e ingreso a la dirección donde está la gráfica del mapa en formato HTML, utilizamos la variable *naveg* para darle el servicio del archivo ejecutable del webdriver del navegador Chrome por medio de la función *Service* importada de la librería *selenium.webdriver*.

A continuación, el comando *naveg.star()* nos permite abrir automáticamente el navegador de Chrome. Después se le asigna a la variable *driver* el control del navegador por medio de la función *webdriver.Remote()* importada de la librería de selenium, con la siguiente iteración *driver.maximize\_window()* sirve para maximizar la pantalla para poder apreciar mejor al navegador.

En la siguiente parte usando el comando *driver.get()* se carga la dirección de la gráfica del mapa en formato HTML que se desea acceder, que previamente se guardó con el nombre *mapa.html*.

Esperamos un tiempo usando la función `time.sleep()` para que se abra completamente la gráfica del mapa que creamos, procedemos a realizar una captura de pantalla y guardado del mapa en una imagen en formato PNG usando el comando `driver.save_screenshot()` y cerramos el navegador para lo cual a las variables que se les cargo tanto el control y el acceso se les debe ordenar apagado de sus funciones, cerrando el navegador de Chrome.

Finalmente procedemos a llamar a la subfunción para notificaciones de Facebook denominada `notfcb()` en la cual cargamos el nombre de la gráfica del mapa en formato PNG y el comentario; todo lo explicado se puede ver en la Figura 57.

```
#Lazo secundario para mapa
p_max = lista['Peak Sound (dBA)'].max()
for k in range(maximus,-1,-1):
    if lista.loc[k,'Peak Sound (dBA)'] == p_max:
        #Comentario 3
        Comentario = "Deteccion de Pico de Sonido Maximo a Lo Largo del dia "
        + str(dia.strftime("%d/%m/%y"))+ " a Las "+ str(lista.loc[k,'Times'])
        + " en el sensor "+str(lista.loc[k,'Device Name'])
        + " esta notificacion a sido enviada a Las "
        + str(timefin.strftime("%H:%M:%S"))+"\n"
        + "-----Alerta-----\n"
        + "Sonido Pico (dBA) "
        +str(lista.loc[k,'Peak Sound (dBA)'])+"\n"
        + "Temperatura (°C) "
        +str(lista.loc[k,'Temperature (°C)'])+"\n"
        + "Humedad (%)"
        +str(lista.loc[k,'Humidity (%)'])+"\n"
        + "KMeans_Cluster "
        +str(lista.loc[k,'KMeans_Clusters'])+"\n"
        + "-----"
        #Grafica 3
        mapa = folium.Map(location=[lista.loc[k,'Latitud'],
                                   lista.loc[k,'Longitud']], zoom_start=15)
        folium.Marker(location=[lista.loc[k,'Latitud'],
                                lista.loc[k,'Longitud']]).add_to(mapa)
        mapa.save("mapa_sensor.html")
        naveg = Service('D:\Documentos\TESIS\core\chromedriver.exe')
        naveg.start()
        driver = webdriver.Remote(naveg.service_url)
        driver.maximize_window()
        driver.get("file:///D:/Documentos/TESIS/core/mapa_sensor.html")
        time.sleep(4)
        driver.save_screenshot(r"C:\Users\esteb\mapa_alerta.png")
        driver.quit()
        naveg.send_remote_shutdown_command()
        #Notificacion 3
        ima_nombre = "mapa_alerta.png"
        notfcb(Comentario, ima_nombre)
```

**Figura 57.** Graficador y notificación de mapa de mayor pico de sonido

### 2.5.22 Eliminación de carpeta de notificaciones

Una vez realizada las notificaciones del mapa de mayor pico de sonido, usando el comando `shutil.rmtree()` para eliminar carpetas que posean archivos, eliminamos la carpeta Notificadora donde se guardaban las gráficas de todas las publicaciones de alertas ya que cada alerta también esta guardado en su propia carpeta por sensor y día, como se muestra en la Figura 58.

```
shutil.rmtree('Notificadora')
```

Figura 58. Eliminación de carpetas de notificaciones

## 2.4 Pruebas de Web Service de notificaciones para uso adecuado

Al realizar pruebas con el código en casos de haber muchas notificaciones de alerta de sonido el espacio de memoria de la computadora se fuerza y apaga el código.

Se coloco la librería Garbage Collector interface que nos permite liberar espacio de memoria por medio de la eliminación de basura generada de las notificaciones anteriores, como se observa en la Figura 59.

```
#Importar Librerías
import datetime
import time
import shutil
import folium
import pandas as pd
import pyautogui as pau
import webbrowser as wb
import matplotlib.pyplot as plt
import gc
from pathlib import Path
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
```

Figura 59. Importación de Garbage Collector

También se utilizó varios comandos en los procesos de elaboración de graficas para cada sensor para cerrar y eliminar gráficas usando las funciones `plt.cla()`, `plt.clf()` y `plt.close()`, cerrar interfaz usando la función `plt.ioff()` y liberar espacio de memoria aplicando `del figs` y `gc.collect()` de la librería de Garbage Collector interface, como se indica en la Figura 60.

```

plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)
plt.xticks(fontweight = 15, fontsize = 15)
plt.yticks(fontweight = 15, fontsize = 15)
plt.ylabel("Sonido (dBA)", fontweight = 20, fontsize = 20)
plt.xlabel("Tiempo(hh:mm:ss)", fontweight = 20, fontsize = 20)
plt.ioff()
plt.savefig("D:\Documentos\TESIS\core\dia_{0}_{1}\webservice_alerta_{2}.png"
            .format(dia.strftime("%d%m%y"),sensor1,maximus-i))
plt.savefig(r"D:\Documentos\TESIS\core\notificadora\webservice_alerta_{0}.png"
            .format(real))

plt.cla()
plt.clf()
plt.close()
del figs
gc.collect()

```

**Figura 60.** Comandos para reducir memoria

Se presentaron problemas en la ploteada de la primera grafica esto se debe a un problema propio del software Python al momento de guarda la imagen por lo cual se creó un comando para guardar una imagen con datos vacíos para que haga la falla antes de la entrada de datos en el lazo for primario para evitar el problema, como se indica en la Figura 61.

```

#Grafica cero
lista0 = dts.loc[(dts['Device Name'] == sensor1) & (dts['Dates'] == dia)]
figs0 = plt.figure()
ax0 = figs0.add_subplot(111)
plt.rcParams["figure.figsize"] = (20, 12)
plt.plot(lista0['Time'],lista0['Peak Sound (dBA)'])
plt.scatter(lista0['Time'],lista0['Peak Sound (dBA)'])
plt.ioff()
plt.savefig(r"D:\Documentos\TESIS\core\notificadora\graficacero")
plt.cla()
plt.clf()
plt.close()
del figs0
gc.collect()

```

**Figura 61.** Gráfica cero

## 2.5 Presentación de código

El código completo se encuentra en el ANEXO IV.



## **3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES**

### **3.1 Resultados**

En esta parte se presentan los resultados de las encuestas realizadas y la aplicación del código de Web Services de Notificaciones.

#### **3.1.1 Resultados de las Encuestas**

Para la obtención de los resultados de las encuestas estas se aplicaron a la empresa Indufrit Company S.A. la misma que nos colaboró, a continuación, se mostrarán los resultados obtenidos de la encuesta a nivel gerencial de empresa y operativo.

Se recibieron tres respuestas por parte del nivel gerencial y 11 respuestas por parte del nivel operativo, manifestando que la empresa es una industria de manufactura de productos alimenticios y la empresa se encuentra ubicada en una zona residencial.

Los resultados mostraron que la encuesta de nivel gerencial fue contestada por personal de nivel gerencial y la encuesta de nivel operativo fue respondida por personal de nivel operativo y una persona de nivel gerencial.

Las personas que llenaron la encuesta de nivel general fueron el gerente general, gerente de producción y jefe de mantenimiento, las personas que llenaron la encuesta de nivel operativo fueron el segundo jefe de producción, 1 asistente de gerencia, 7 empacadores, 1 operador de empacadora, 1 operador de la línea de fritura y 1 operador del área de amasado.

El número de trabajadores de la empresa es de 15 personas, los 3 de nivel gerencial y 12 de nivel operativo, por área de trabajo el gerente general tiene personal de 3 personas, el gerente de producción tiene 5 personas y el jefe de mantenimiento tiene 3 personas.

El horario de trabajo de nivel gerencial y operativo es en el día en el horario de 6 AM a 2 PM, el operador de empacadora trabaja en horario rotativo.

La empresa no cuenta con datos sobre nivel de ruido en el sector por lo cual ellos quieren tener datos en caso de que lo requieran.

Poseen un total de 8 maquinarias industriales que conforman la línea de amasado y laminado, la línea de fritura continua y la línea de empacado, que son manejadas por 3 personas de nivel operativo y los 2 de nivel gerencial

Manifiestan que les gustaría recibir notificaciones por medio de la Red social Facebook con información escrita y gráfica de niveles de ruido (dB) vs Tiempo con datos de riesgo entre

bajo, medio y alto, a nivel general el 66.7% correspondiente a 2 personas quiere recibir notificaciones diarias a las 2PM y el 33.3% correspondiente a 1 persona quiere recibir notificaciones semanales a la 1PM, mientras que a nivel operativo todos quieren recibir notificaciones diarias a las 2 PM.

En caso de alertas en los niveles de ruido tanto nivel gerencial y operativo les gustaría recibir notificaciones al instante.

Tanto nivel gerencial y operativo les gustaría que la empresa tome acciones en caso de detectar niveles de ruido superiores a los permitidos por la OMS y recibir posibles soluciones contra el ruido.

En el caso de nivel general, el 66.7% correspondiente a 2 personas tienen afecciones como dolor de cabeza debido al ruido y el 33.3% correspondiente a 1 persona no tiene afecciones por el ruido, en el caso de nivel operativo el 54.5% correspondiente a 6 personas tiene afecciones como falta de sueño, dolor de cabeza y bajo rendimiento debido al ruido mientras que el 45.5% correspondiente a 5 personas no tiene afecciones por el ruido.

En el caso de nivel general, todos creen que se encuentran expuestos a niveles de ruido bajo en la empresa, en el caso de nivel operativo el 90.9% correspondiente a 10 personas creen que están expuestas a niveles de ruido bajo en la empresa mientras que el 9.1% correspondiente a 1 persona se siente expuesta a un nivel de ruido medio en la empresa.

El gerente de producción está dispuesto a pagar por el sistema alrededor de \$400 mientras que el resto de gerentes está dispuesto a pagar \$200.

Todos los datos obtenidos se encuentran en el ANEXO V

Con esto resultados obtenidos y las normativas anteriormente indicadas se procede a generar los parámetros para el Web Service de Notificaciones como son las notificaciones de alerta con información gráfica y escrita en niveles de ruido (dB) vs Tiempo en la red social Whatsapp en un grupo cerrado, donde se analizará los datos de 6AM a las 2 PM y procederá hacer notificaciones si los niveles acústicos son mayores o iguales 85 dB como dice la OMS para lugares externos, en la notificación se informara en que sensor ocurrió la alerta con su fecha y hora.

Para el momento de llegar 2PM, se harán notificaciones diarias con información gráfica y escrita en niveles de ruido (dB) vs Tiempo en la red social Facebook para cada sensor entre las 6AM y 12PM indicando la alerta que se produjo a lo largo del día.

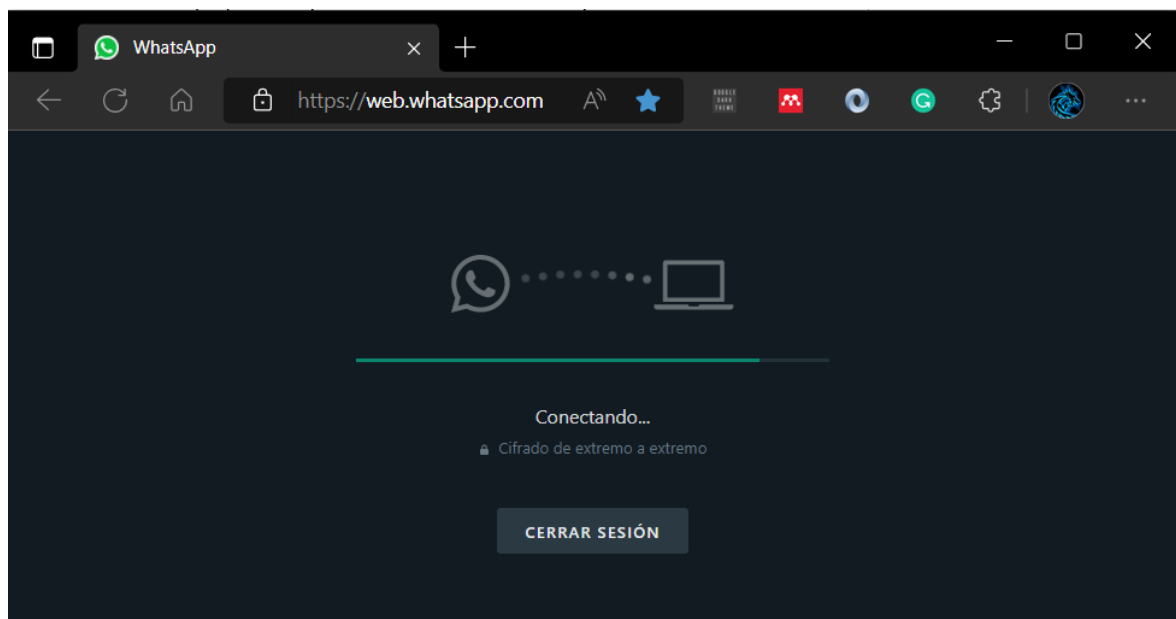
### 3.1.2 Resultados del Web Service de Notificaciones

Para la obtención de los resultados de las Web Services de Notificaciones se procede a ingresar un día dentro del rango de la base de datos del EMS en este caso será 14/07/20 y se procederá activar el código, primero leyendo la base de datos como se muestra en la Figura 62.

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64-bit]
Type "help", "copyright", "credits" or "license()" for more information
>>>
===== RESTART: D:\Documentos\TESIS\core\webservice_notificado
Digite el dia: 14/07/20
Leyendo base de datos ....
```

**Figura 62.** Ingreso de día en la base de datos

Después se activa la función de notificaciones de alerta donde empieza con la apertura del navegador en la página de Whatsapp, como se muestra en la Figura 63.



**Figura 63.** Apertura de página de Whatsapp

Continúa accediendo directo al chat del grupo cerrado que se llama pruebas webservice donde hace las notificaciones de alerta en caso de que en el día en el intervalo de 6 AM y 2 PM se produzcan eventos de alertas de sonido en cualquier sensor, esto se puede ver a continuación en las Figuras 64, 65 y 66 se ve las notificaciones de alerta que se generaron a lo largo del día simulado.

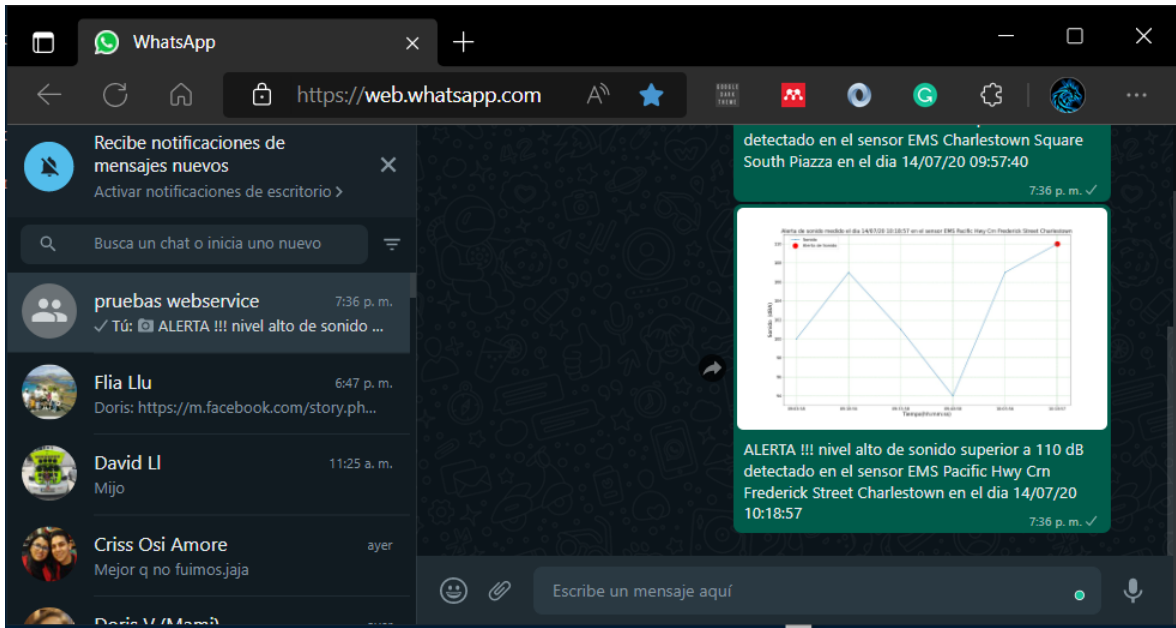


Figura 64. Notificaciones de alertas por Whatsapp

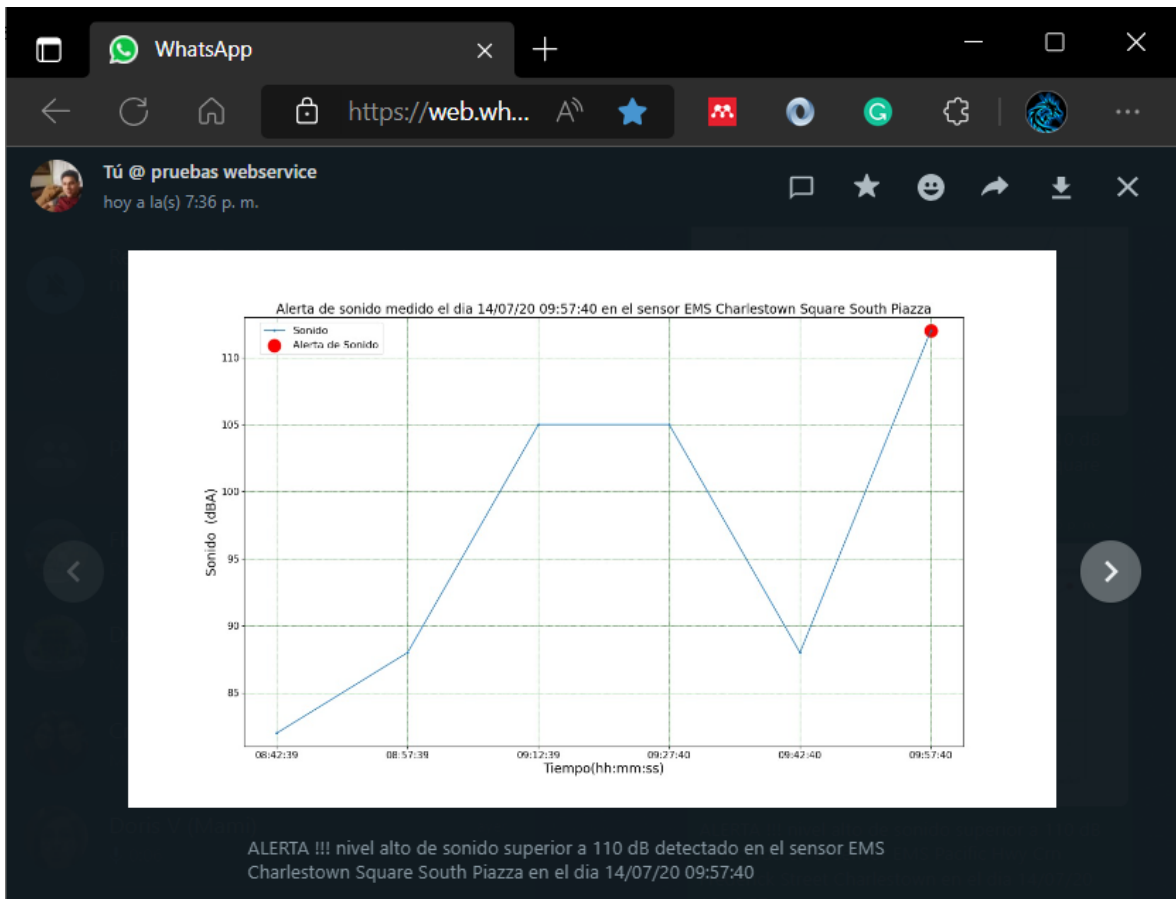
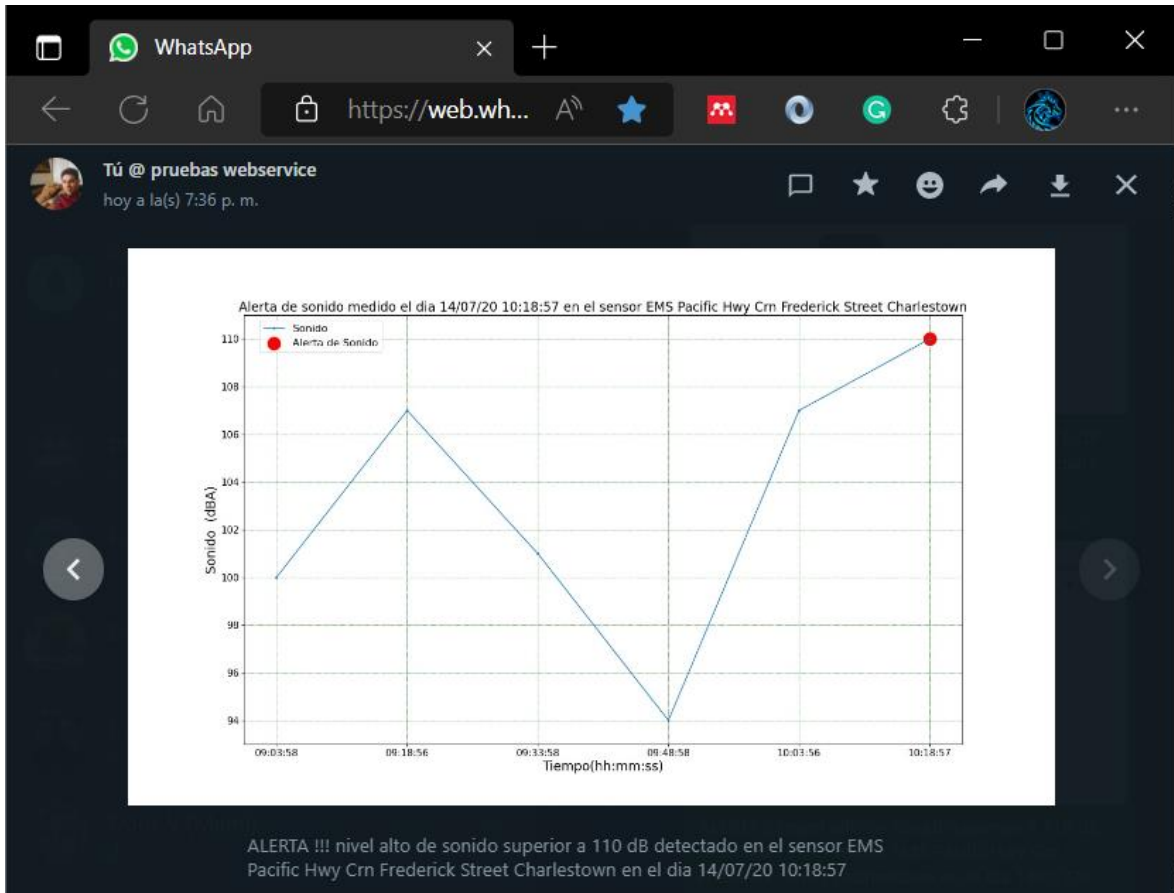
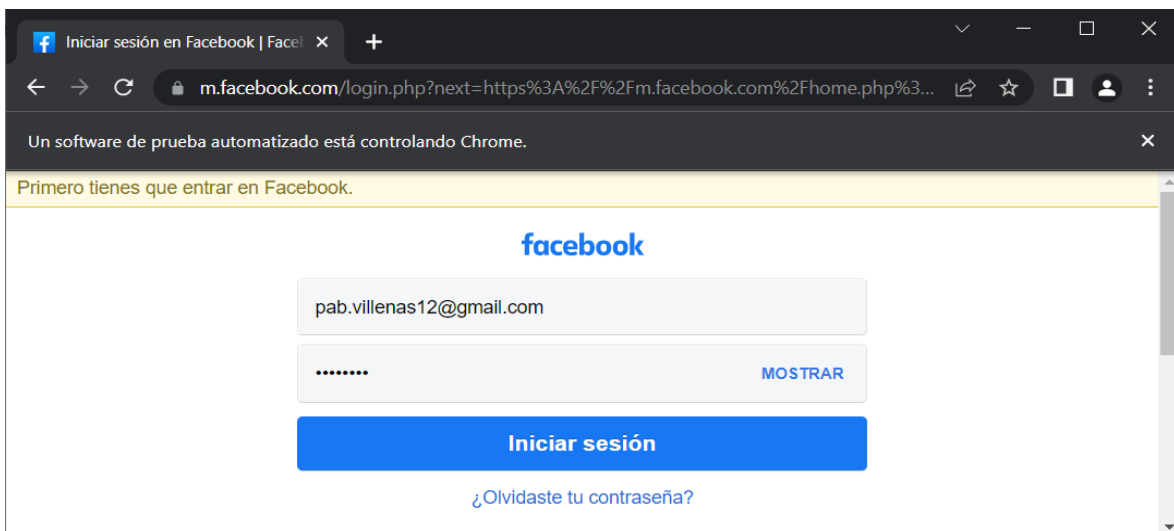


Figura 65. Notificación de alerta largo del día 1



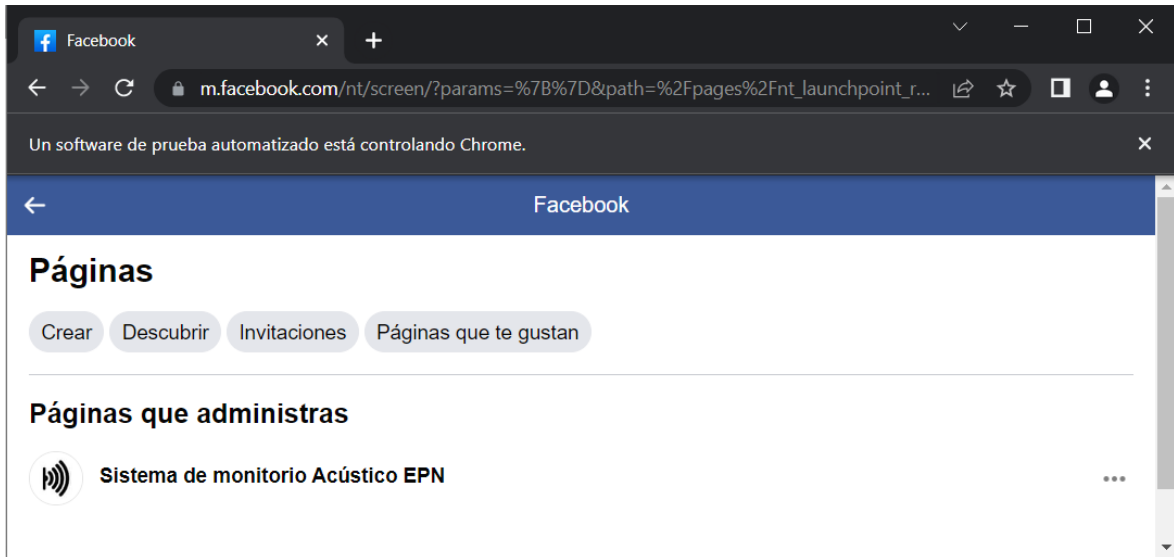
**Figura 66.** Notificación de alerta largo del día 2

Una vez completada la gestión en el día entre las 6 AM y 2 PM empieza con las notificaciones diarias para niveles de sonido y clústeres de cada sensor para lo cual accede al navegador de Chrome por medio del driver ejecutable e ingresa a la red social Facebook y carga los datos para iniciar la sesión como se puede ver a continuación en la Figura 67.

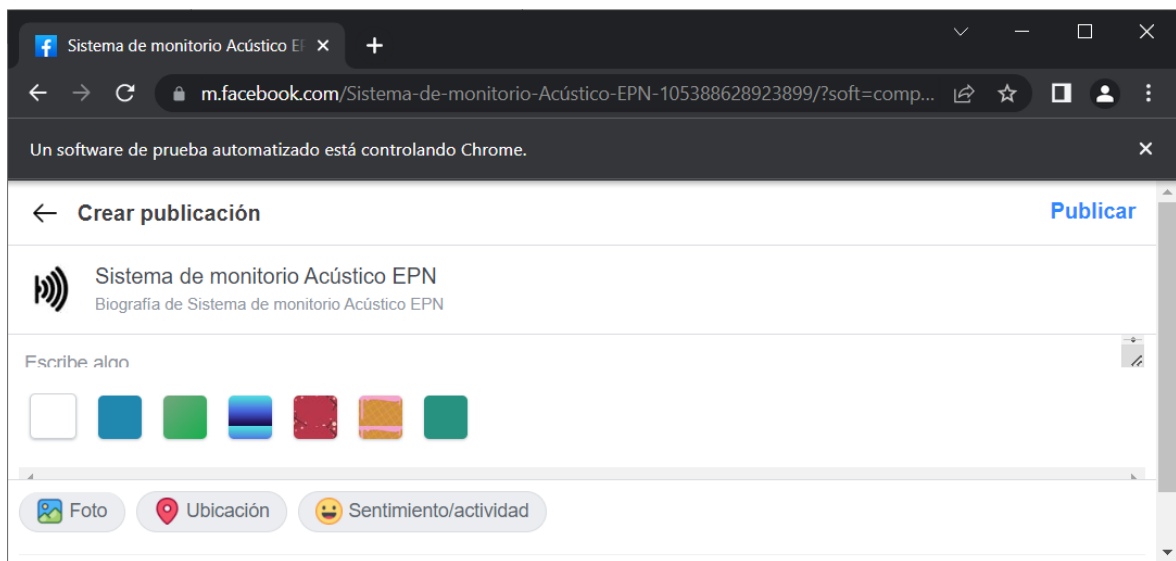


**Figura 67.** Accediendo a Facebook

Entra en la página Facebook business creada para las notificaciones denominada Sistema de monitorio Acústico EPN y procede a abrir el creador de publicaciones esto se lo puede ver en las Figuras 68 y 69.

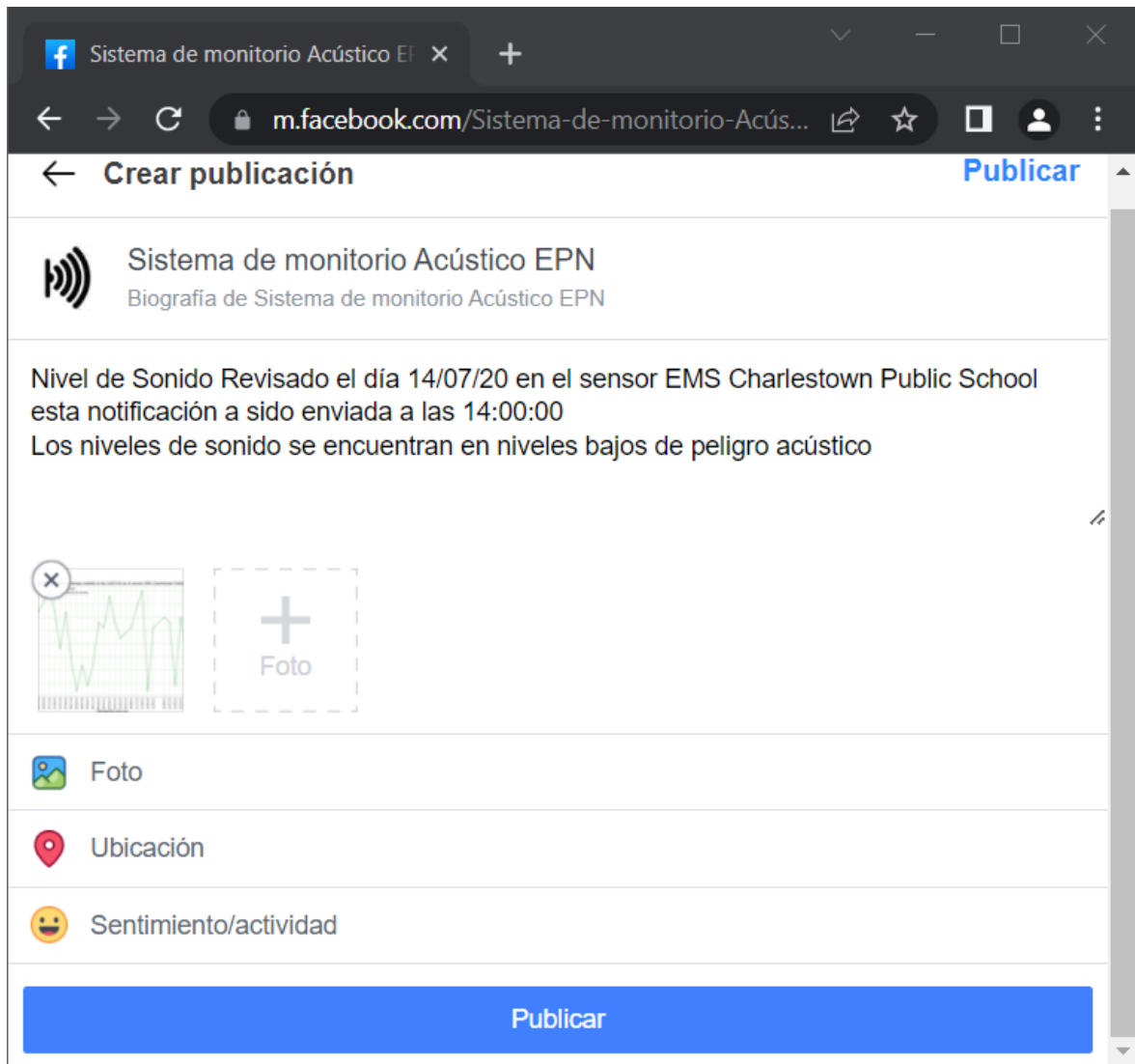


**Figura 68.** Accediendo a página de Sistema de monitoreo



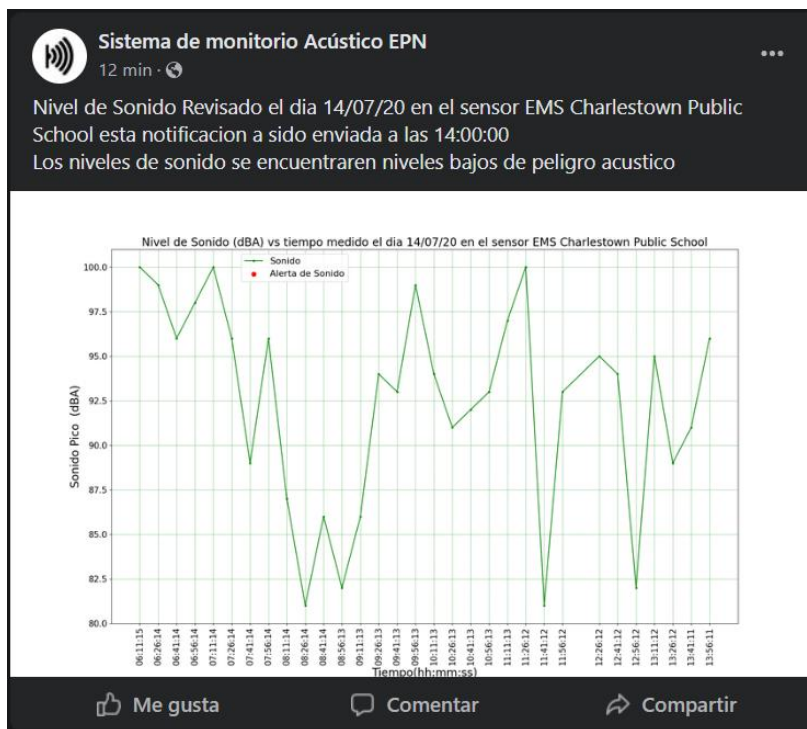
**Figura 69.** Accediendo al creador de publicaciones

Procede a abrir la ventana para cargar la imagen y carga la primera grafica del día correspondiente al sensor, después coloca el mensaje de comentario y publica, todo esto se puede observar en las Figura 70.

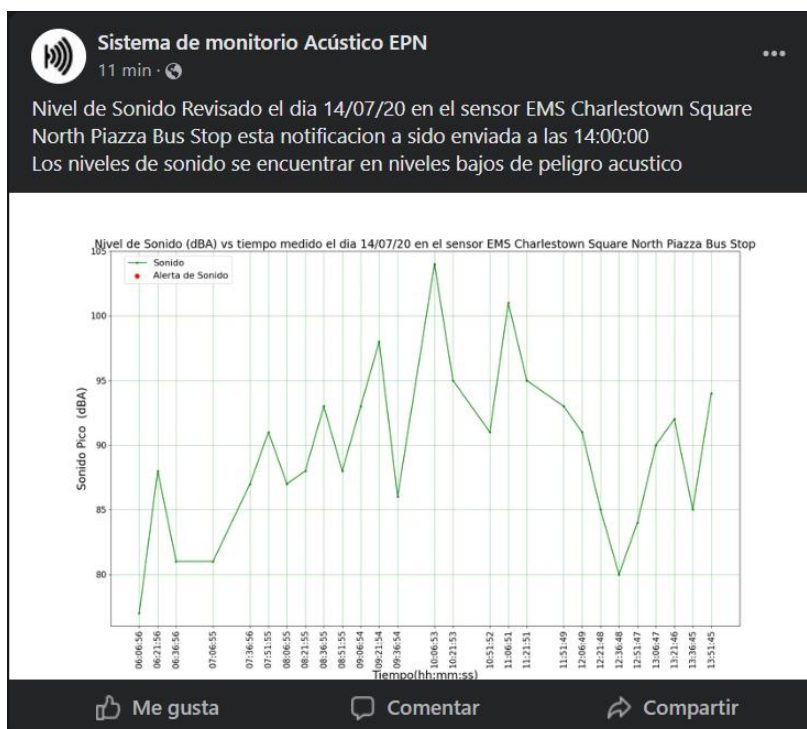


**Figura 70.** Escribe comentario y publica

En las Figuras 71, 72, 73, 74, 75 y 76 se puede observar las notificaciones diarias de niveles de sonido generadas para el día simulado.

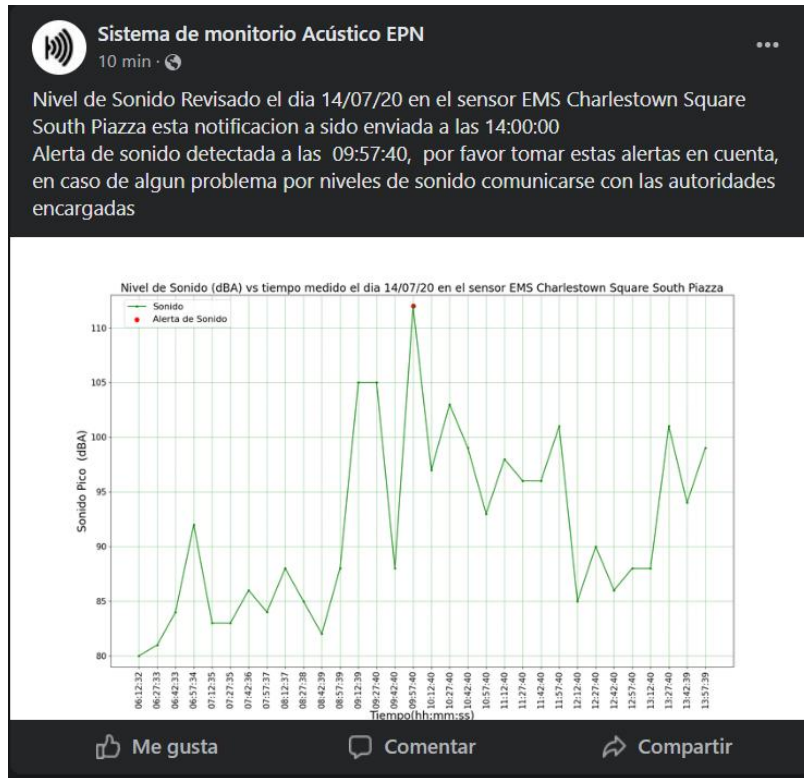


**Figura 71.** Notificación diaria de niveles de sonido publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Public School

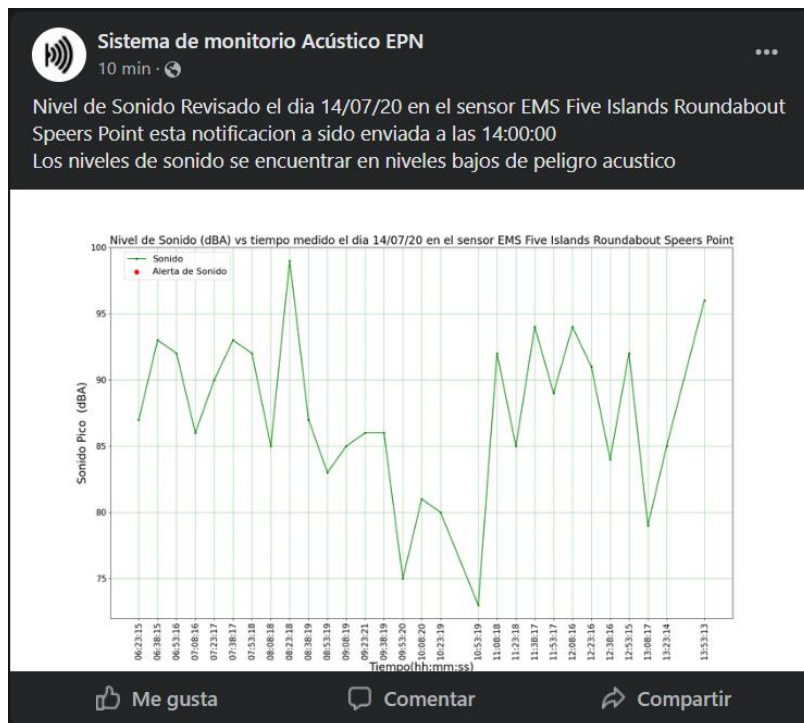


**Figura 72.** Notificación diaria de niveles de sonido publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Square North Piazza Bus Stop

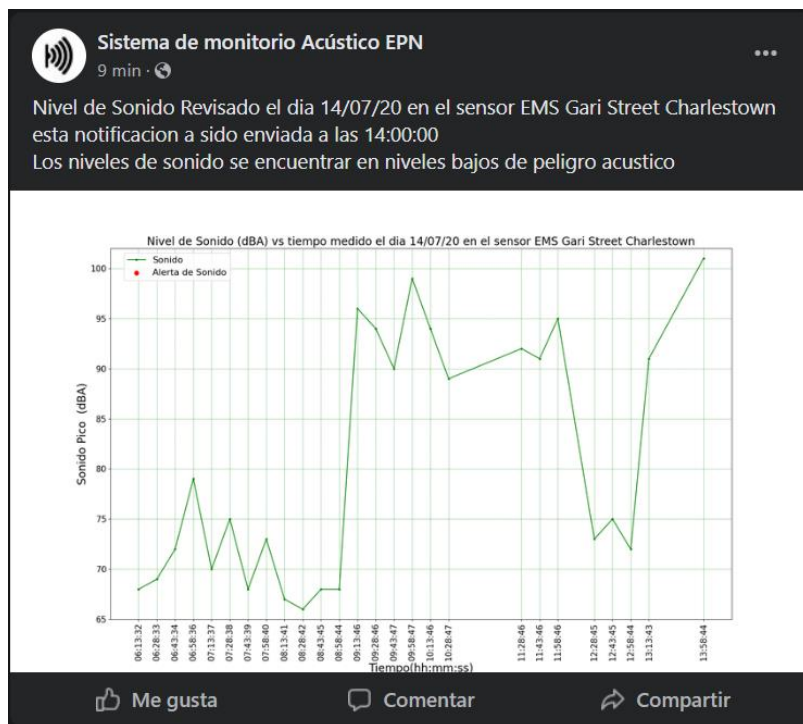




**Figura 73.** Notificación diaria de niveles de sonido publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Square South Piazza



**Figura 74.** Notificación diaria de niveles de sonido publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Five Islands Roundabout Speers Point

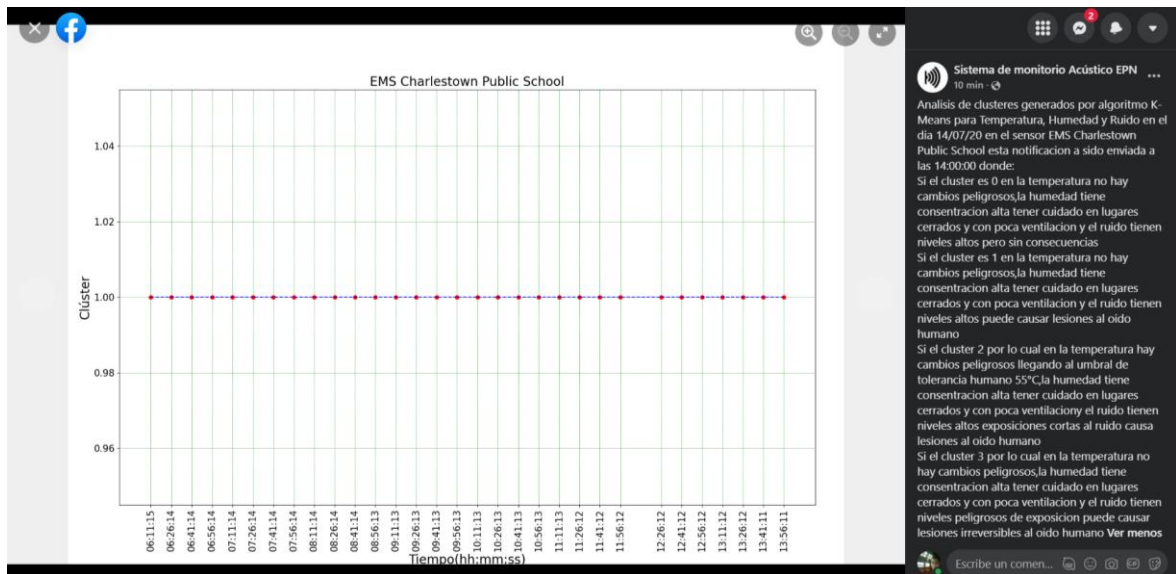


**Figura 75.** Notificación diaria de niveles de sonido publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Gari Street Charlestown

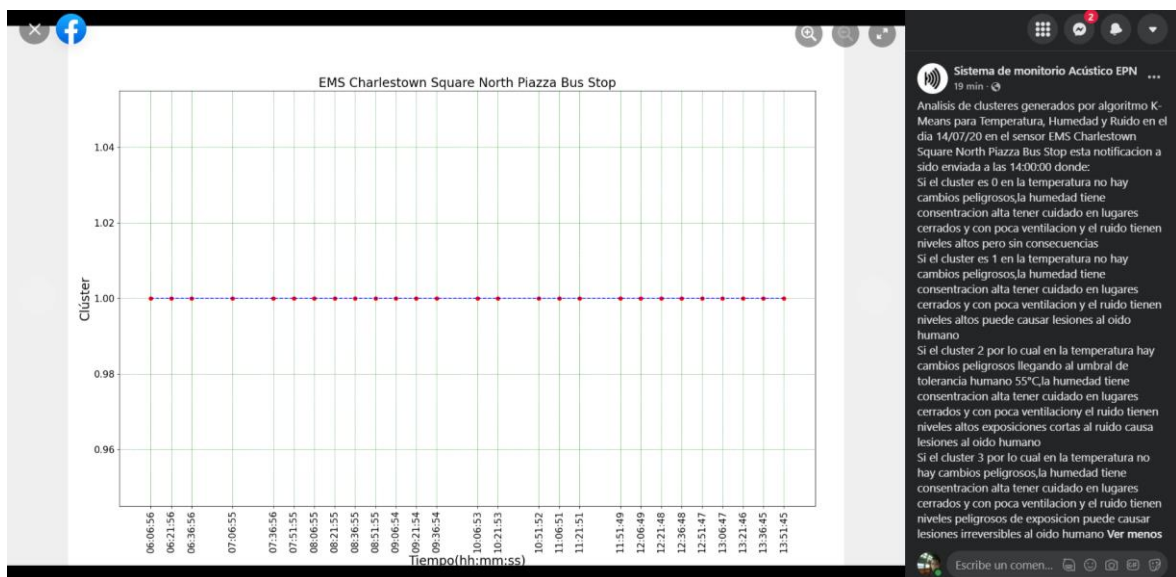


**Figura 76.** Notificación diaria de niveles de sonido publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Pacific Hwy Crn Fredrick Street Charlestown

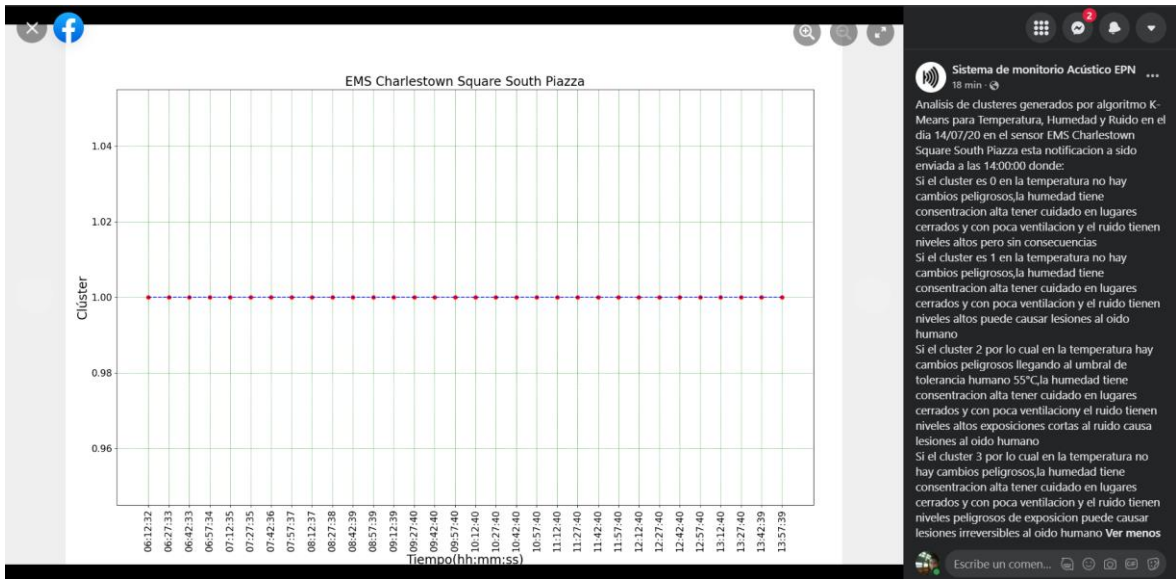
En la Figura 77, Figura 78, Figura 79, Figura 80, Figura 81 y Figura 82 se puede observar las notificaciones diarias de clústeres generadas para el día simulado



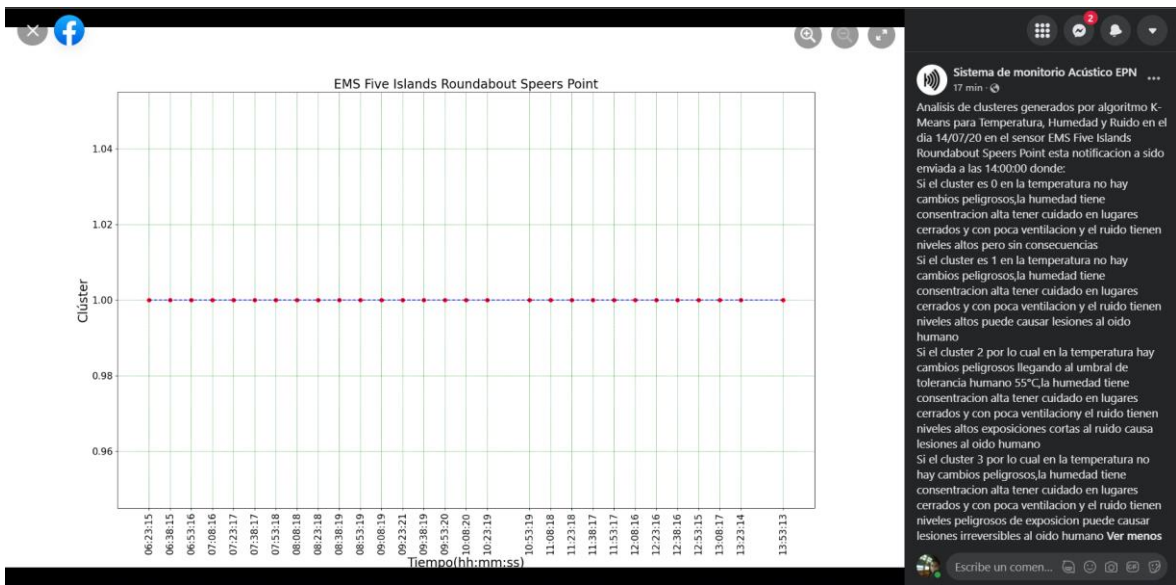
**Figura 77.** Notificación diaria de clúster publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Public School



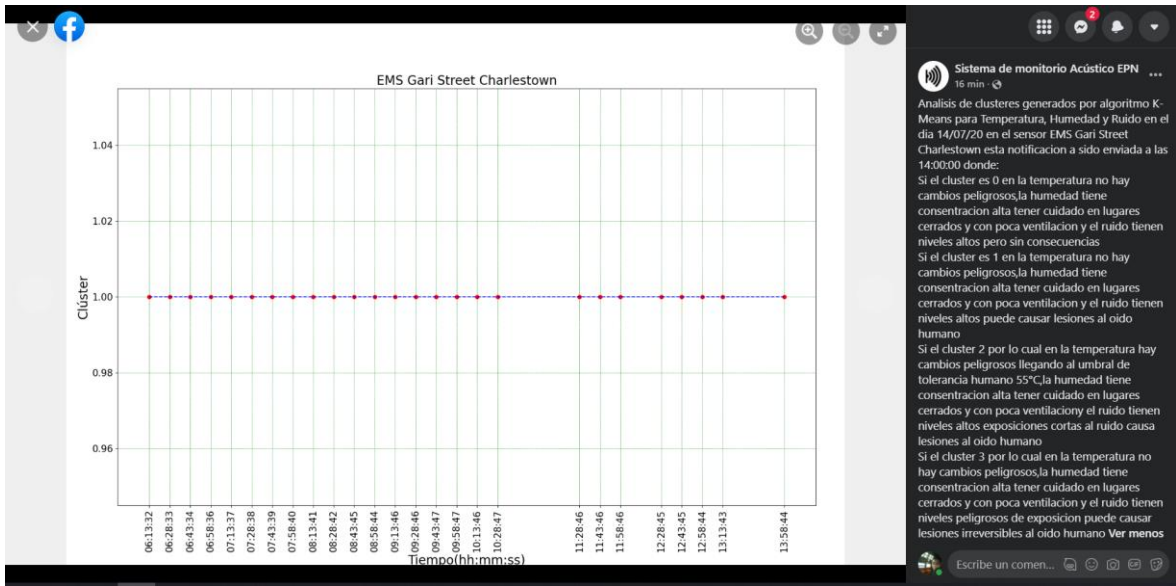
**Figura 78.** Notificación diaria de clúster publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Square North Piazza Bus Stop



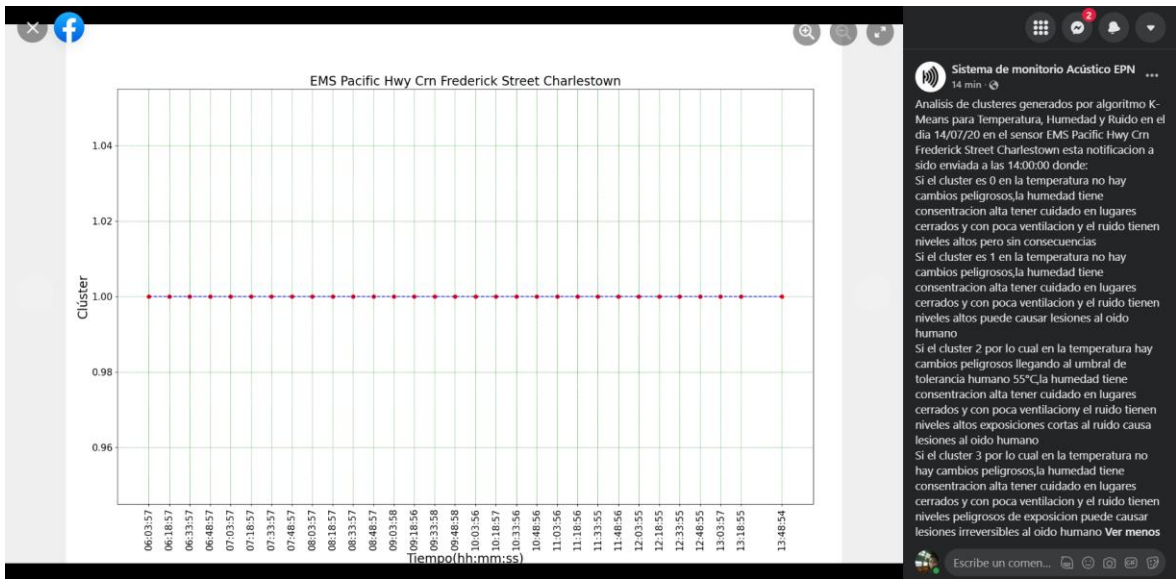
**Figura 79.** Notificación diaria de clúster publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Square South Piazza



**Figura 80.** Notificación diaria de clúster publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Five Islands Roundabout Speers Point

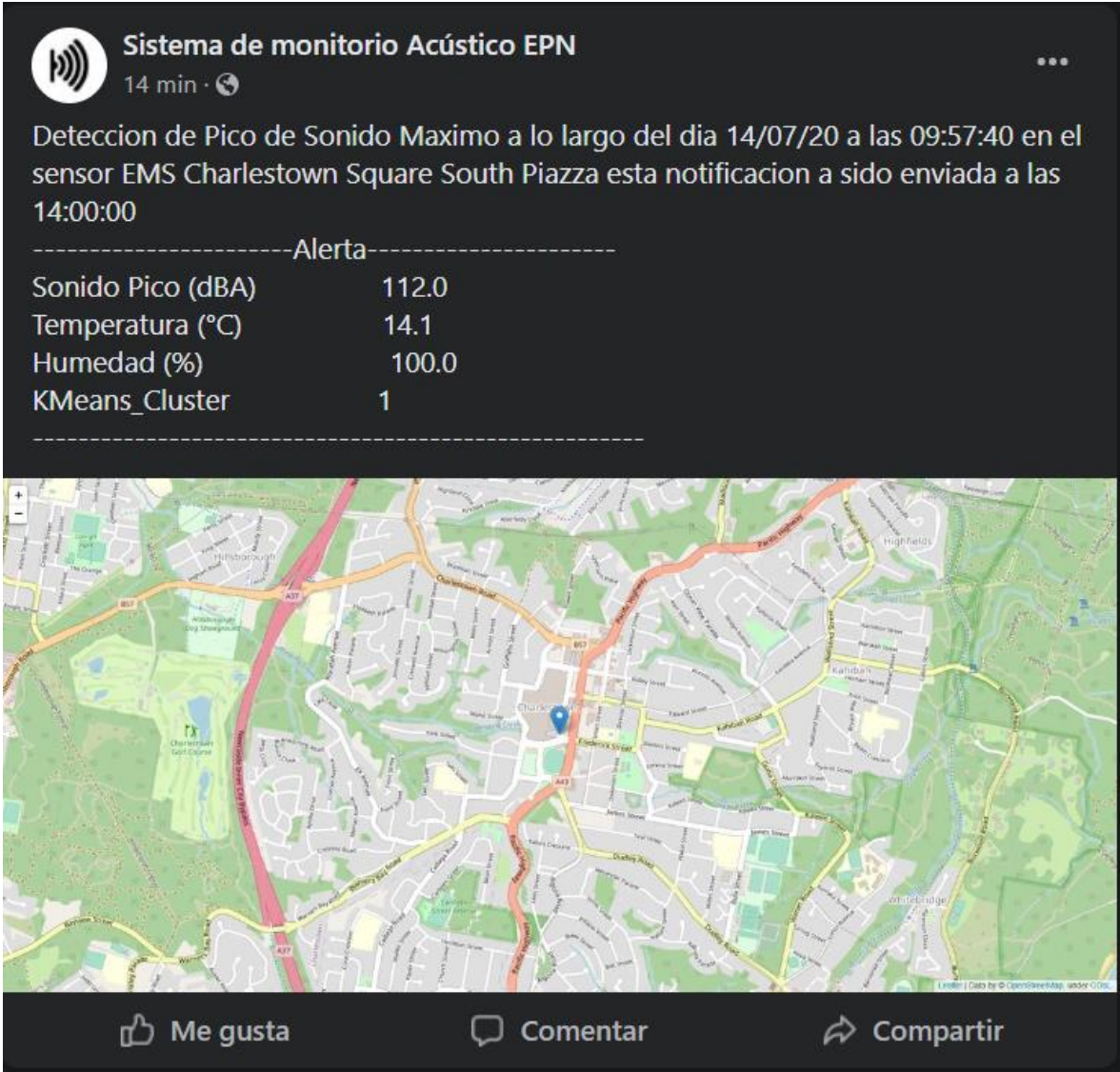


**Figura 81.** Notificación diaria de clúster publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Gari Street Charlestown



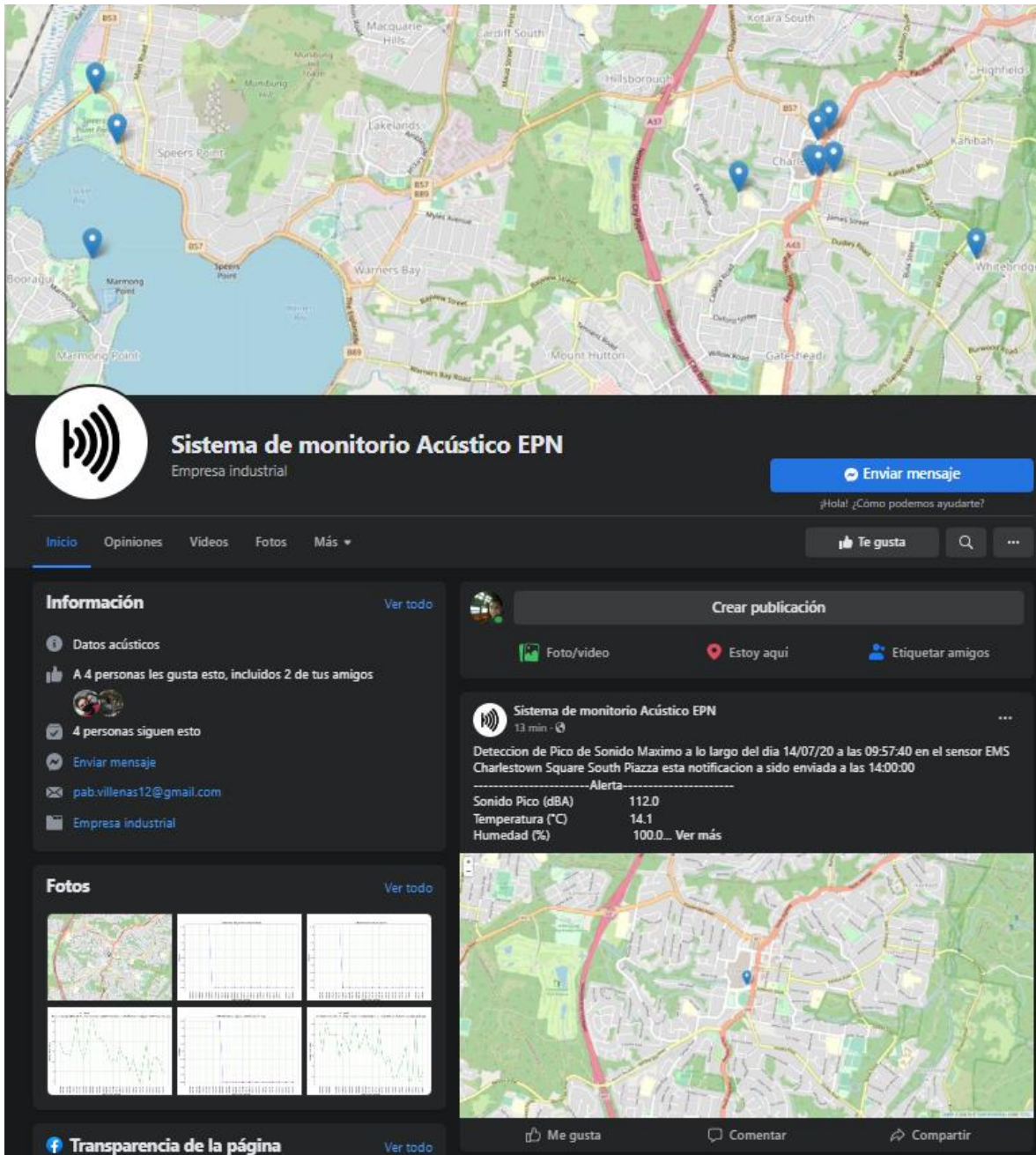
**Figura 82.** Notificación diaria de clúster publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Pacific Hwy Crn Fredrick Street Charlestown

En la Figura 83 se puede observar la notificación del mapa marcando el lugar del sensor donde se produjo el pico máximo de sonido a lo largo del día simulado.



**Figura 83.** Notificación del mapa marcando el lugar del sensor donde se produjo el mayor pico del sonido a lo largo del día simulado publicada en Facebook correspondiente al día 14/07/20 en el sensor EMS Charlestown Square South Piazza

En la Figura 84 podemos ver como se ve las notificaciones en la página de Facebook business de Sistemas de Monitoreo Acústico EPN.



**Figura 84.** Notificación diaria publicada en página de Facebook business

En caso de necesitar las gráficas del día tanto de alertas como notificaciones diarias y clústeres se puede encontrar en las carpetas que el código crea para guardar las notificaciones por día y sensor que se encuentra en el directorio donde se encuentra el código de Python, esto se puede ver en las Figura 85.

dia_140720_EMS Pacific Hwy Crn Frederick Street Charlestown	14/8/2022 08:24 PM	Carpeta de archivos
dia_140720_EMS Gari Street Charlestown	14/8/2022 08:22 PM	Carpeta de archivos
dia_140720_EMS Five Islands Roundabout Speers Point	14/8/2022 08:09 PM	Carpeta de archivos
dia_140720_EMS Charlestown Square South Piazza	14/8/2022 08:07 PM	Carpeta de archivos
dia_140720_EMS Charlestown Square North Piazza Bus Stop	14/8/2022 08:05 PM	Carpeta de archivos
dia_140720_EMS Charlestown Public School	14/8/2022 08:03 PM	Carpeta de archivos

**Figura 85.** Carpetas con notificaciones

Finalmente se comprueba el correcto funcionamiento del código de Python para Web Service de notificaciones de una red de sensores acústicos EMS.

### 3.2 Conclusiones

Las encuestas realizadas a los mandos gerencial y operativo de la empresa Indufrit Company S.A. lograron establecer varios de los parámetros de visualización para las notificaciones como son periodo de tiempo, intervalo entre notificaciones, el tipo de información, momento de publicación y tipo de red social, el resto de parámetros se determinaron en función de las normativas de la OMS, donde las notificaciones de alerta, diarias, clústeres y pico máximo tendrán información gráfica y escrita de datos que se analizará de 6AM a las 2 PM del día ingresado, con alertas notificadas por la red social Whatsapp cuando los niveles acústicos son mayores o iguales 110 dB establecidos por la OMS para lugares externos; para las notificaciones de diarias y clústeres las notificaciones serán por la red social Facebook informando los datos acumulados para el día en niveles de sonido y clústeres; para las notificaciones de pico máximo se elaborara un mapa con lugar del sensor donde se produjo el mayor pico del sonido a lo largo del día, notificando con la red social Facebook.

Utilizando el código diseñado de Web Service para Notificaciones se logró publicar en la red social de Whatsapp en el grupo cerrado creado para el proyecto donde los usuarios de los mandos de la empresa Indufrit Company S.A. podrán ver las notificaciones de alerta en caso de que se produzcan, podrán ver información gráfica del pico de sonido que sobrepasa los 110 dB con los 5 anterior valores analizados de sonido en función del tiempo, con información escrita del momento en el que ocurrió el evento y el sensor donde se produjo, con lo cual se podrá tomar medidas correctivas para evitar el nivel del ruido.

Al momento de terminar de analizar el día simulado, el Web Service para Notificaciones consiguió realizar las notificaciones diarias, clústeres y pico máximo en la red social Facebook en la página business creada para el proyecto para los usuarios de los mandos de la empresa Indufrit Company S.A, donde podrán ver información gráfica de los niveles de sonido y clústeres en función del tiempo tanto de los eventos con niveles normales y



alertas analizadas de 6AM a las 2PM para el día ingresa, con información escrita del momento en que se realiza la notificación, para las notificaciones diarias en caso de producirse eventos de alerta verán los momentos en los que estos ocurrieron; en el caso de los clústeres verán la información con el significado de cada valor clúster y su relación con el ruido, temperatura y humedad para tomar su respectiva precaución; para el pico máximo se verá información del lugar donde se produjo el evento, momento, nivel de sonido, temperatura y humedad.

### **3.3 Recomendaciones**

Realizar más estudios en cuanto a de formas de acceder a redes sociales por medio de más APIs.

Realizar un programa que puede actualizar el archivo ejecutable del navegador web ya que se puede desactualizar cada vez que el navegador web se actualiza.

Realizar un mejoraras al Web Service para lograr un análisis tanto semanal o mensualmente sobre la contaminación acústica en la ubicación de los sensores acústicos, para determinar eventos posibles y evitar daños a la salud humana de las personas que pasan por los lugares donde están ubicados los sensores.

## 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] ISO 9612:2009. ISO (2009 April) [Online]. Available: <https://www.iso.org/standard/41718.html> [Último acceso: 2021 11 15].
- [2] Berglund, Birgitta, Lindvall, Thomas, Schwela, Dietrich H & World Health Organization. Occupational and Environmental Health Team (1999). Guidelines for community noise. World Health Organization [Online]. Available: <https://apps.who.int/iris/handle/10665/66217> [Último acceso: 2021 11 15].
- [3] Calculadora de tamaño de muestra y guía completa. Qualtrics (2020 May 21) [Online]. Available: <https://www.qualtrics.com/es-la/gestion-de-la-experiencia/investigacion/ calcular-tomano-muestra/> [Último acceso: 2021 11 15].
- [4] Cálculo del tamaño de la muestra. Share and Discover Knowledge on SlideShare [Online]. Available: <https://es.slideshare.net/MiriamR1983f/clculo-del-tamao-de-la-muestra-35348541> [Último acceso: 2021 11 15].
- [5] M. Zaforas. Como implementar microservicios con Python en minutos. [Online]. Available: <https://www.paradigmadigital.com/dev/implementar-microservicios-python-minutos/> [Último acceso: 2021 11 15].
- [6] A. Kumar y S. P. Panda, "A Survey: How Python Pitches in IT-World", 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon),2019, pp. 248-251, Doi: 10.1109/COMITCon.2019.8862251.
- [7] B. Overland, *Python without fear*, 1<sup>st</sup> ed. Addison-Wesley Professional, 2017.
- [8] X. He, L. Xu, X. Zhang, R. Hao, Y. Feng y B. Xu, "PyART: Python API Recommendation in Real-Time", 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion),2021, pp. 246-247, Doi: 10.1109/ICSE-Companion52605.2021.00114.
- [9] datetime — Basic date and time types. Python. [Online]. Available: <https://docs.python.org/3/library/datetime.html> [Último acceso: 2021 11 25].

- [10] Pandas Introduction. W3Schools Online Web Tutorials [Online]. Available: [https://www.w3schools.com/python/pandas/pandas\\_intro.asp](https://www.w3schools.com/python/pandas/pandas_intro.asp) [Último acceso: 2021 11 25].
- [11] Welcome to PyAutoGUI's documentation. PyAutoGUI. [Online]. Available: <https://pyautogui.readthedocs.io/en/latest/> [Último acceso: 2021 11 25].
- [12] V. Neethidevan, G. Chandrasekaran, " Web automation using selenium web driver python", International Journal of Recent Technology and Engineering, vol. 7, pp. 845-847, March 2019.
- [13] Python: abrir enlaces usando selenium. Acervo Lima. [Online]. Available: <https://es.acervolima.com/python-abrir-enlaces-usando-selenium/> [Último acceso: 2022 07 29].
- [14] Folium — Folium 0.12.1 documentation. [Online]. Available: <http://python-visualization.github.io/folium/> [Último acceso: 2022 07 29].

## **5 ANEXOS**

ANEXO I. Encuesta sobre niveles acústicos para nivel gerencial para empresa

ANEXO II. Encuesta sobre niveles acústicos para nivel operativo para empresa

ANEXO III. Base de Datos del EMS

ANEXO IV. Programa en Python denominado webservice\_notificador.py

ANEXO V. Resultados de las encuestas sobre niveles acústicos para nivel gerencial y nivel operativo

## 5.1 ANEXO I

### Encuesta de Niveles de Ruido 1

Esta encuesta es de carácter anónimo, se utilizará para desarrollar notificaciones de una red de sensores acústicos que medirá los niveles de ruido en la empresa. Esta información será usada con fines académicos y mejoramiento en la empresa

#### \*Obligatorio

1. Indique a que se dedica la empresa:\*

Manufactura de productos alimenticios

Manufactura de bienes

Construcción

Agropecuaria

Petroquímica

Otros:

2. La empresa está ubicada en:\*

Zona residencial

Zona residencial mixta

Zona Industrial

3. Seleccione su puesto de trabajo:\*

Gerencial

Operativo

Especificar su puesto de trabajo:\*

Tu respuesta

4. ¿Cuál su horario de trabajo?\*

En el día

En la noche

Especificar su horario de trabajo. (Ejemplo 8h30 a 17h30):\*

Tu respuesta

5. ¿La empresa tiene disponibilidad de datos sobre ruido en el sector?\*

Sí

No

¿En el caso afirmativo, usted está dispuesto a entregar los datos de ruido para fines académicos (adjuntar el documento)?

[Agregar archivo](#)

6. ¿Cuál es la cantidad de personal en la empresa?\*

Tu respuesta

7. ¿Cuál es la cantidad de personal en su área de trabaja?\*

Tu respuesta

8. ¿Cuál es la cantidad de máquinas en la empresa?\*

Tu respuesta

9. Para el servicio de notificaciones, ¿en cuál red social prefiere recibir las publicaciones?\*

Facebook

Twitter

Instagram

10. Para el servicio de notificaciones en las redes sociales, ¿qué tipo de información desea en la publicación?\*

Información gráfica

Información escrita

Información gráfica y escrita

11. Para el servicio de notificaciones en las redes sociales, ¿cómo desearía recibir las publicaciones?\*

Diaria

Semanal

Mensual

12. Dentro de su horario laboral, ¿cuál es la hora de preferencia en la que desearía recibir la notificación de ruido? Ejemplo (12h00 a 13h00)\*

Tu respuesta

13. En caso de alerta en los niveles de ruido, desea recibir información en tiempo real (instantáneamente)\*

Sí

No

14. Ecuador regula las emisiones de ruido en decibeles (dB), pero también se puede establecer en Nivel de Presión Sonora Continuo Equivalente (NPS eq), escoger la opción con la cual desea que se represente el ruido de la empresa.\*

Niveles de ruido (dB) vs. Tiempo

NPS eq [dB(A)] vs Tiempo

Niveles de ruido (dB) vs Frecuencia

NPS eq [dB(A)] vs Frecuencia

15. Con referencia al nivel de riesgo por ruido, escoja tres o más niveles para la notificación\*

Riesgo peligroso

Riesgo alto

Riesgo medio

Riesgo bajo

Riesgo muy bajo

Riesgo nulo

16. Si se detecta niveles de ruido superiores a los estándares permitidos por la OMS usted cree que la empresa debería tomar alguna acción\*

Sí

No

17. Usted desearía recibir recomendaciones en las publicaciones sobre posibles soluciones de los niveles de ruido\*

Sí

No

18. Usted cree que el ruido generado está afectando en su condición de trabajo y salud\*

Sí

No

19. A qué nivel de riesgo por ruido cree que se encuentra expuesto\*

Riesgo alto

Riesgo medio

Riesgo bajo

20. Usted ha tenido algún problema por el ruido, ¿qué problemas a tenido?\*

Falta de sueño

Dolor de cabeza

Bajo rendimiento en el trabajo

Ninguno

Otros:

21. ¿Cuánto estaría dispuesta la empresa a pagar por tener este sistema (Dólares)?\*

Tu respuesta

GRACIAS

Se agradece por su tiempo y colaboración en la resolución de la encuesta.



## 5.2 ANEXO II

### Encuesta de Niveles de Ruido 2

Esta encuesta es de carácter anónimo, se utilizará para desarrollar notificaciones de una red de sensores acústicos que medirá los niveles de ruido en la empresa. Esta información será usada con fines académicos y mejoramiento en la empresa

**\*Obligatorio**

1. Indique a que se dedica la empresa:\*

Manufactura de productos alimenticios

Manufactura de bienes

Construcción

Agropecuaria

Petroquímica

Otros:

2. Seleccione su puesto de trabajo:\*

Gerencial

Operativo

Especificar su puesto de trabajo:\*

Tu respuesta

3. ¿Trabaja en horarios rotativos?\*

Sí

No

4. ¿Cuál su horario de trabajo?\*

En el día

En la noche

Especificar su horario de trabajo. Ejemplo de 8h30 a 17h30.\*

Tu respuesta

5. ¿Usted es operario de una máquina?\*

Si

No

Responder en caso que su respuesta anterior fue SI, especificar la maquinaria de la cual se encarga:

Tu respuesta

6. Para el servicio de notificaciones, ¿en cuál red social prefiere recibir las publicaciones?\*

Facebook

Twitter

Instagram

7. Para el servicio de notificaciones en las redes sociales, ¿qué tipo de información desea que posea la publicación?\*

Información gráfica

Información escrita

Información gráfica y escrita

8. Para el servicio de notificaciones en las redes sociales, ¿cómo desearía recibir las publicaciones?\*

Diaria

Semanal

Mensual

9. Dentro de su horario laboral, ¿cuál es la hora de preferencia en la que desearía recibir la notificación de ruido? Ejemplo de 12h00 a 13h00\*

Tu respuesta

10. En caso de alerta en los niveles de ruido, desea recibir información en tiempo real (instantáneamente)\*

Sí

No

11. Con referencia al nivel de riesgo por ruido, escoja tres o más niveles para la notificación\*

Riesgo peligroso

Riesgo alto

Riesgo medio

Riesgo bajo

Riesgo muy bajo

Riesgo nulo

12. Si se detecta niveles de ruido superiores a los estándares permitidos por la OMS usted cree que la empresa debería tomar alguna acción\*

Sí

No

13. Usted desearía recibir recomendaciones en las publicaciones sobre posibles soluciones de los niveles de ruido\*

Sí

No

14. Usted cree que el ruido generado está afectando en su condición de trabajo y salud\*

Sí

No

15. A qué nivel de riesgo por ruido cree que se encuentra expuesto\*

Riesgo alto

Riesgo medio

Riesgo bajo

16. Usted ha tenido algún problema por el ruido, ¿qué problemas a tenido?\*

Falta de sueño

Dolor de cabeza

Bajo rendimiento en el trabajo

Ninguno

Otros:

### **5.3 ANEXO III**

<https://www.dropbox.com/scl/fi/2nsqhp7zldw73qc0lf9t2/environmental-monitoring-system-historical.xlsx?dl=0&rlkey=fhox754okie6m67z6fsla883i>

### **5.4 ANEXO IV**

[https://www.dropbox.com/s/4gpw93woou2o7vw/webservice\\_notificador\\_dias\\_pro.py?dl=0](https://www.dropbox.com/s/4gpw93woou2o7vw/webservice_notificador_dias_pro.py?dl=0)

## 5.5 ANEXO V



Especificar su puesto de trabajo:

3 respuestas

Gerente de Producción

Gerente general

Jefe de mantenimiento

4. ¿Cuál su horario de trabajo?

 Copiar

3 respuestas

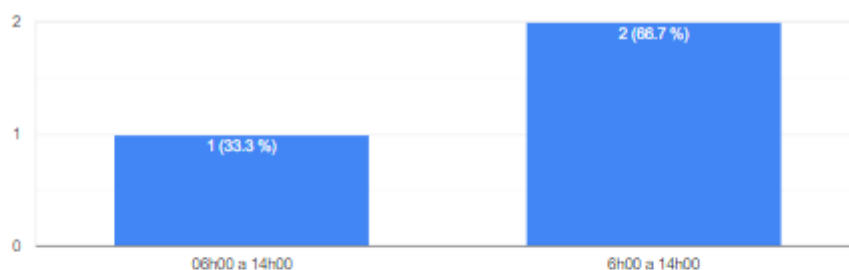


● En el día  
● En la noche

Especificar su horario de trabajo. (Ejemplo 8h30 a 17h30):

 Copiar

3 respuestas



5. ¿La empresa tiene disponibilidad de datos sobre ruido en el sector?

 Copiar

3 respuestas



● Si  
● No

¿En el caso afirmativo, usted está dispuesto a entregar los datos de ruido para fines académicos (adjuntar el documento)?

[Ver carpeta](#)

0 respuestas

Todavía no hay respuestas para esta pregunta.

6. ¿Cuál es la cantidad de personal en la empresa?

[Copiar](#)

3 respuestas



7. ¿Cuál es la cantidad de personal en su área de trabaja?

3 respuestas

5

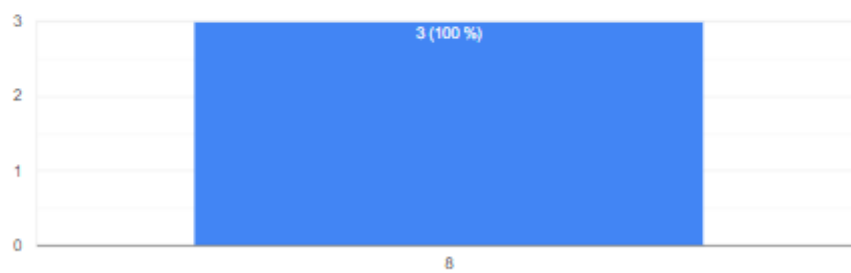
3

6

8. ¿Cuál es la cantidad de máquinas en la empresa?

[Copiar](#)

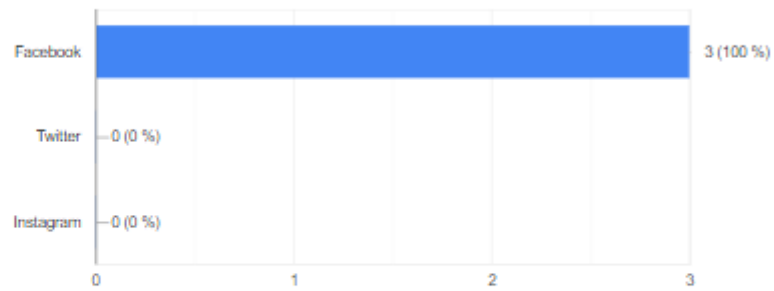
3 respuestas



9. Para el servicio de notificaciones, ¿en cuál red social prefiere recibir las publicaciones?

 Copiar

3 respuestas



10. Para el servicio de notificaciones en las redes sociales, ¿qué tipo de información desea en la publicación?

 Copiar

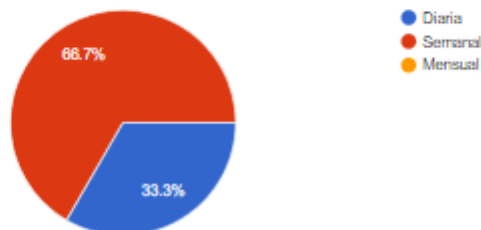
3 respuestas



11. Para el servicio de notificaciones en las redes sociales, ¿cómo desearía recibir las publicaciones?

 Copiar

3 respuestas

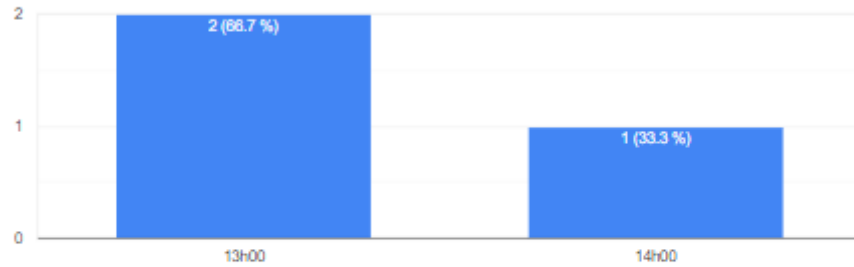




12. Dentro de su horario laboral, ¿cuál es la hora de preferencia en la que desearía recibir la notificación de ruido?. Ejemplo (12h00 a 13h00)

[Copiar](#)

3 respuestas



13. En caso de alerta en los niveles de ruido, desea recibir información en tiempo real (instantáneamente)

[Copiar](#)

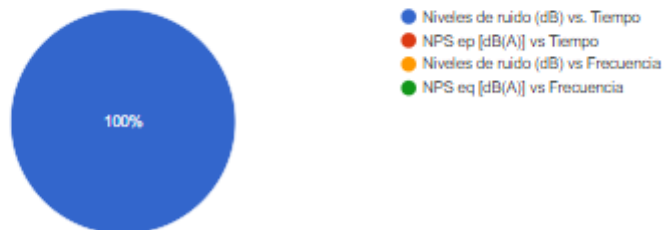
3 respuestas



14. Ecuador regula las emisiones de ruido en decibeles (dB), pero también se puede establecer en Nivel de Presión Sonora Continuo Equivalente (NPS eq), escoger la opción con la cual desea que se represente el ruido de la empresa.

[Copiar](#)

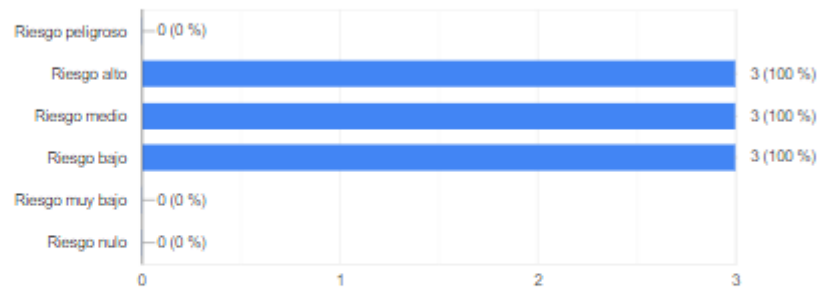
3 respuestas



15. Con referencia al nivel de riesgo por ruido, escoja tres o más niveles para la notificación

 Copiar

3 respuestas




16. Si se detecta niveles de ruido superiores a los estándares permitidos por la OMS usted cree que la empresa debería tomar alguna acción

 Copiar

3 respuestas



17. Usted desearía recibir recomendaciones en las publicaciones sobre posibles soluciones de los niveles de ruido

 Copiar

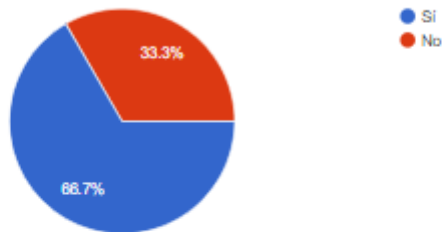
3 respuestas



18. Usted cree que el ruido generado está afectando en su condición de trabajo y salud

[Copiar](#)

3 respuestas



19. A qué nivel de riesgo por ruido cree que se encuentra expuesto

[Copiar](#)

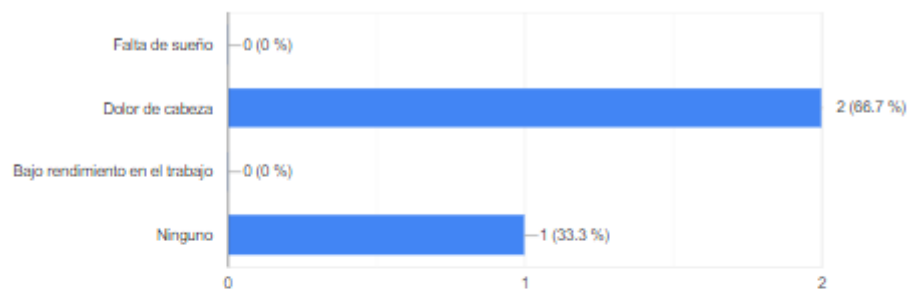
3 respuestas



20. Usted ha tenido algún problema por el ruido, ¿qué problemas a tenido?

[Copiar](#)

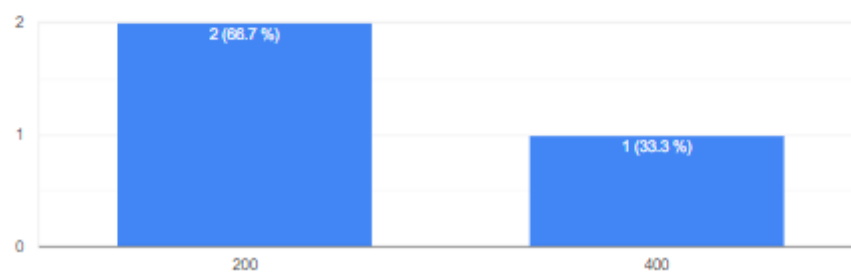
3 respuestas



21. ¿Cuánto estaría dispuesta la empresa a pagar por tener este sistema (Dólares)?

[Copiar](#)

3 respuestas



11 respuestas



Se aceptan respuestas

Resumen

Pregunta

Individual

1. Indique a que se dedica la empresa:

Copiar

11 respuestas

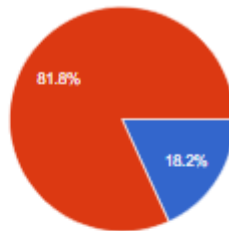


- Manufactura de productos alimenticios
- Manufactura de bienes
- Construcción
- Agropecuaria
- Petroquímica

2. Seleccione su puesto de trabajo:

Copiar

11 respuestas

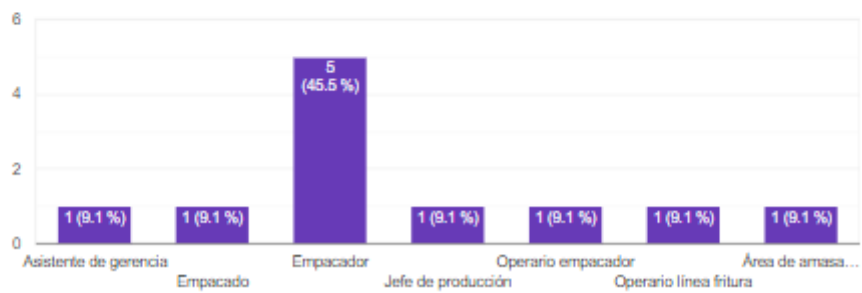


- Gerencial
- Operativo

Especificar su puesto de trabajo:

Copiar

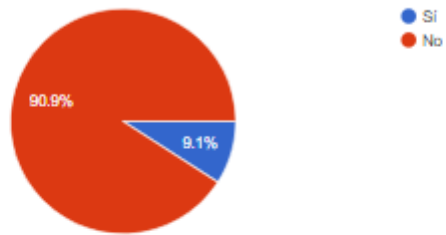
11 respuestas



3. ¿Trabaja en horarios rotativos?

 Copiar

11 respuestas



4. ¿Cuál su horario de trabajo?

 Copiar

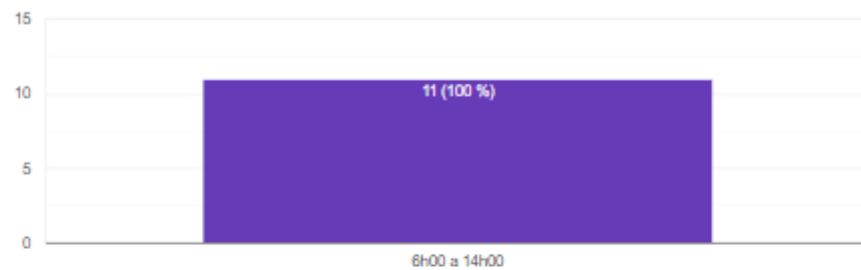
11 respuestas




Especificar su horario de trabajo. Ejemplo de 8h30 a 17h30.

 Copiar

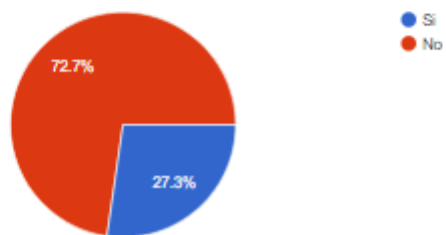
11 respuestas



5. ¿Usted es operario de una máquina?

 Copiar

11 respuestas



Responder en caso que su respuesta anterior fue SI, especificar la maquinaria de la cual se encarga:

3 respuestas

Línea de empackado

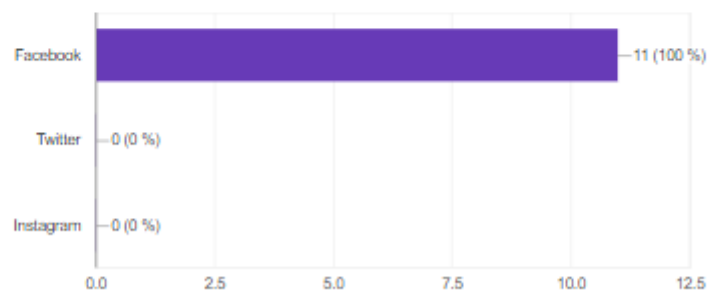
Freidora continua

Amasadora y laminadoras

6. Para el servicio de notificaciones, ¿en cuál red social prefiere recibir las publicaciones?

 Copiar

11 respuestas



7. Para el servicio de notificaciones en las redes sociales, ¿qué tipo de información desea que posea la publicación?

 Copiar

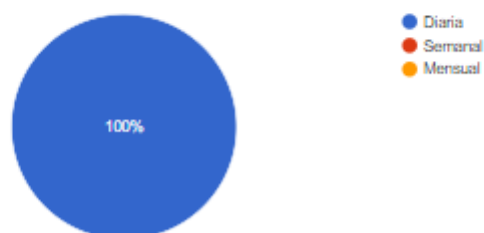
11 respuestas



8. Para el servicio de notificaciones en las redes sociales, ¿cómo desearía recibir las publicaciones?

 Copiar

11 respuestas



9. Dentro de su horario laboral, ¿cuál es la hora de preferencia en la que desearía recibir la notificación de ruido?. Ejemplo de 12h00 a 13h00

 Copiar

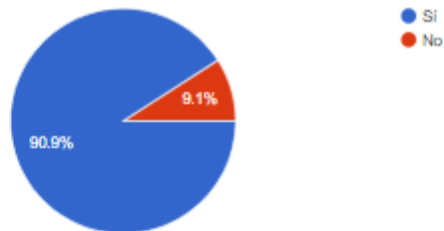
11 respuestas



10. En caso de alerta en los niveles de ruido, desea recibir información en tiempo real (instantáneamente)

 Copiar

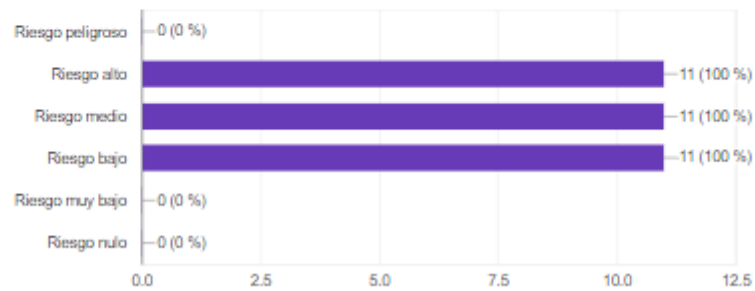
11 respuestas



11. Con referencia al nivel de riesgo por ruido, escoja tres o más niveles para la notificación

 Copiar

11 respuestas



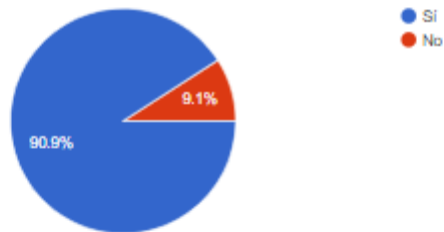
12. Si se detecta niveles de ruido superiores a los estándares permitidos por la OMS usted cree que la empresa debería tomar alguna acción [Copiar](#)

11 respuestas



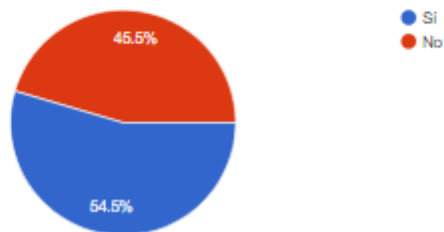
13. Usted desearía recibir recomendaciones en las publicaciones sobre posibles soluciones de los niveles de ruido [Copiar](#)

11 respuestas



14. Usted cree que el ruido generado está afectando en su condición de trabajo y salud [Copiar](#)

11 respuestas

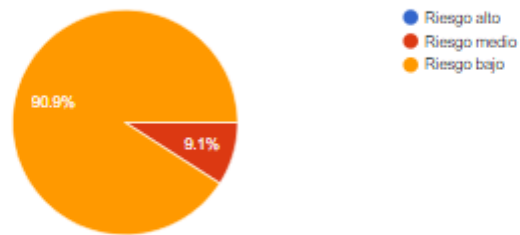




15. A qué nivel de riesgo por ruido cree que se encuentra expuesto

[Copiar](#)

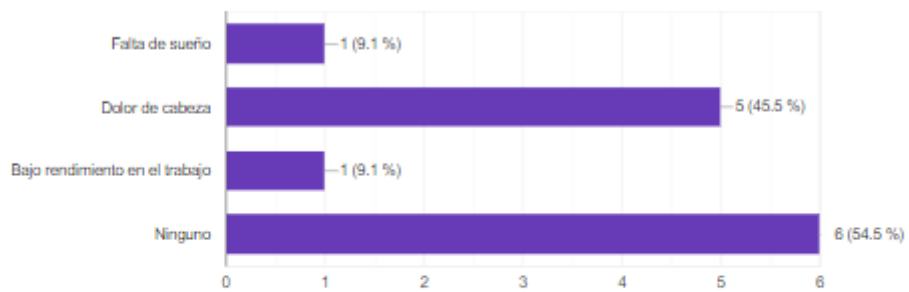
11 respuestas



16. Usted ha tenido algún problema por el ruido, ¿qué problemas a tenido?

[Copiar](#)

11 respuestas



GRACIAS