

**UNIVERSIDAD DE EL SALVADOR**  
**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**  
**DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**



**TRABAJO DE GRADUACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE SISTEMA PARA MONITOREO  
VOLCÁNICO DE VARIABLES MULTIPARAMÉTRICAS SÍSMICAS Y EMISIONES  
DE DIÓXIDO DE AZUFRE EN EL VOLCÁN DE SANTA ANA.**

**PARA OPTAR AL GRADO DE:**

**INGENIERO DE SISTEMAS INFORMÁTICOS**

**PRESENTAN:**

**CÁCERES BARRIENTOS, LUIS ALBERTO**  
**CÁRCAMO GONZÁLEZ, CARLOS ALEXANDER**  
**PADILLA HERNÁNDEZ, MARVIN RENÉ**  
**ZALDAÑA LEMUS, MELVIN GUILLERMO**

**DOCENTE DIRECTOR:**

**ING. JOSÉ ROBERTO COLÓN VILLALTA**

**OCTUBRE DE 2017**

**SANTA ANA**

**EL SALVADOR**

**CENTROAMÉRICA**

**UNIVERSIDAD DE EL SALVADOR**  
**AUTORIDADES CENTRALES**

**RECTOR**

Maestro Roger Armando Arias Alvarado

**VICE-RECTOR ACADÉMICO**

Doctor Manuel de Jesús Joya

**VICE-RECTOR ADMINISTRATIVO (INTERINO)**

Ingeniero Carlos Armando Villalta

**SECRETARIO GENERAL**

Licenciado Cristóbal Hernán Ríos Benítez

**DEFENSORA DE LOS DERECHOS UNIVERSITARIOS**

Maestra Claudia María Melgar de Zambrana

**FISCAL GENERAL**

Licenciado Rafael Humberto Peña Marín

**UNIVERSIDAD DE EL SALVADOR**  
**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**  
**AUTORIDADES**

**DECANO**

Maestro Raúl Ernesto Azcúnaga López

**VICEDECANO**

Ingeniero Roberto Carlos Sigüenza Campos

**SECRETARIO DE LA FACULTAD**

Licenciado David Alfonso Mata Aldana

**JEFE DEL DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**

Ingeniero Douglas García Rodezno

# AGRADECIMIENTOS

“A Dios, por darme fortaleza, entendimiento y perseverancia en cada etapa de mi carrera hasta culminarla.

Agradecimientos especiales a mi madre Sonia del Carmen Barrientos, que a pesar de la distancia siempre estuvo conmigo dándome su confianza y apoyo incondicional para que pudiera convertirme en la persona que soy hoy en día.

Gracias también a mis hermanas y a mi novia que estuvieron apoyándome de uno u otra manera durante mis estudios y la realización de mi trabajo de grado.

Para finalizar agradezco a mis compañeros y amigos con los cuales compartí muchas tardes de estudio, ocio y aventuras en nuestro paso por la universidad.”

*Luis Cáceres*

“Primeramente, a Dios, a quien doy la gloria y la honra de todos mis logros y de todas mis victorias. Que siempre ha estado a mi lado protegiéndome de todo mal, librándome de todo peligro, y brindándome todas las herramientas necesarias para llegar a este punto de mi vida. Gracias por tu grandeza y tu misericordia.

A mi madre, Aída González, que siempre estuvo a mi lado, dándome las palabras de aliento que necesitaba para seguir adelante, y diciéndome lo que no quería escuchar cuando perdía el rumbo. Que siempre estuvo presente desde el día que me dio la vida hasta la fecha, impidiendo que me faltara algo; se merece este triunfo, y los que me queden por vivir. Gracias por tu amor y por tu esfuerzo.

A mi hermana, Katherinne Paola González, que desde que nació, se convirtió en mi mayor motor de felicidad. Que me ha enseñado más de lo que yo haya podido transmitirle. Que con su cariño y amor, hace que me despierte motivado todos los días a ser una mejor persona, para convertirme en el mejor modelo a seguir para ella. Gracias por llegar a mi vida.

A mi familia y amigos, que incondicionalmente estuvieron apoyándome, algunos desde cerca, otros desde el extranjero; siempre vieron lo mejor de mí y estuvieron pendientes para ayudar en lo que hiciera falta. Gracias por su apoyo y cariño.

A mis líderes y compañeros de trabajo, que estuvieron presentes en mi última etapa de preparación académica. Que me entrenaron desde el inicio de mi vida laboral y me dieron las herramientas necesarias para forjarme profesionalmente. Que tuvieron la paciencia necesaria para ayudarme a crecer y llegar lejos. Gracias por creer en mí.

A mis compañeros de equipo de tesis, que evitaron que nos diéramos por vencidos, cuando el camino de volvió difícil y oscuro. Que pusieron su máximo empeño junto a mí para finalizar este proyecto de la mejor manera. Gracias por su entrega.”

*Carlos Cárcamo*

“Doy gracias a Dios por la culminación de un gran éxito en mi vida, gracias a mi Madre Marlene Hernández quien me ha apoyado toda la vida, brindándome sus valiosos consejos, alentándome a nunca rendirme en la vida, inculcándome valiosas enseñanzas que nunca olvidare. Gracias a mi Padre Oscar Padilla, quien me mostro valiosas experiencias, ser una persona de valor con grandes motivaciones, orientándome a caminar sobre los senderos de la vida y nunca rendirme ante cualquier inclemencia, mostrándome la dirección correcta a continuar. Gracias a mi hermano Oscar A. Padilla que desde lejos me ha apoyado, motivándome a continuar y culminar mis estudios.

Gracias a mis compañeros de tesis Luis, Carlos y Melvin, con los cuales demostraron grandes habilidades comunicativas, trabajo en equipo; gran empeño por concluir lo que empezamos, por su valioso apoyo ante las diferentes problemáticas presentadas y su gran aporte para construir este proyecto.

Gracias a muchas amistades que se forjaron dentro de los años de estudio, por medio de los cuales nos motivamos con un continuo aprendizaje, forjando grandes lazos afectivos, creando metas y sueños, manteniendo cada uno su propósito y ayudándome a nunca mostrar flaqueza ante cualquier dificultad que se presente.

Muchas gracias a todas las personas que me apoyaron a seguir adelante, a concluir una etapa más de la vida, finalizar un sueño y poder ser un profesional de éxito.”

*Marvin Padilla*

“A Dios, por regalarme la sabiduría y fuerzas necesarias para poder terminar este proyecto.

A mi Madre Eva Zaldaña, la mejor mamá del mundo, porque cada vez que intente rendirme ella estuvo ahí dándome la fuerza, el ánimo, el amor, todo lo necesario para seguir adelante, gracias por estar ahí siempre para mí.

A mi familia porque siempre me brindaron el amor, ayuda y paciencia incondicional gracias por estar ahí siempre y apoyarme todo el tiempo en mis decisiones.

A mis compañeros de tesis Carlos, Marvin y Luis por el apoyo y el compromiso durante todo este proceso en cual todos tuvimos que sacrificar parte de nuestros compromisos, tiempos y recurso para lograr nuestra meta.

Y finalmente amigos que siempre estuvieron pendientes y me dieron siempre su apoyo incondicional.”

*Melvin Zaldaña*

# AGRADECIMIENTOS DEL EQUIPO

“No podemos dejar escapar la oportunidad de reconocer el apoyo brindado por personas que estuvieron involucradas en el desarrollo de este trabajo de grado, que nos brindaron su tiempo y conocimientos que ayudaron a garantizar el éxito de este proyecto. De no haber sido por ellos, nunca hubiéramos podido finalizar nuestro sistema; ya que su aporte fue necesario e indispensable.

Al Licenciado Rodolfo Olmos, que nos apoyó desde el inicio del proyecto, brindándonos de toda su experiencia en el área de vulcanología. Gracias por todos los sábados que sacrificó a nuestro lado, por su amor a la ciencia y su deseo de crear un sistema de software innovador, pionero en el área, que ayude a entender el comportamiento volcánico, para poder ayudar a mucha gente. Gracias por su paciencia, por su tiempo, su conocimiento y por su incondicional apoyo.

A Griselda Marroquín, y todo su equipo de trabajo de la Unidad de Gestión de Datos e Información del Ministerio de Medio Ambiente y Recursos Naturales (MARN); que nos brindaron su cálida ayuda, brindándonos las herramientas necesarias para conectar nuestro sistema con su red sísmológica EartWorm. Gracias por todas las gestiones y por su profesionalismo.

Al Ingeniero Roberto Colón, nuestro asesor de trabajo de grado; que fungió de gran manera sus papel, aconsejándonos cuando más los necesitábamos, y enseñándonos a gestionar y tratar con todas las partes involucradas en nuestro proyecto. Gracias por su tiempo, y por todos los conocimientos transmitidos a lo largo del proyecto”

*Luis, Carlos, Marvin y Melvin*

# ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	29
CAPÍTULO I: ESTUDIO PRELIMINAR .....	30
1. ESTUDIO PRELIMINAR.....	31
1.1. ANTECEDENTES.....	31
1.1.1. Red de monitoreo vulcanológico del Servicio Nacional de Estudios Territoriales.....	31
1.1.2. Red de monitoreo del Instituto de Vulcanología IV-UES.....	33
1.1.3. Red de monitoreo BIRA-IASB.....	35
1.2. MARCO CONCEPTUAL.....	37
1.2.1. Volcanes .....	37
1.2.1.1. Tipos de actividad volcánica .....	38
1.2.1.2. Productos volcánicos .....	39
1.2.1.3. Tipos de relieve volcánico .....	40
1.2.1.4. Origen y tipos de magma .....	40
1.2.1.5. Diferenciación magmática .....	41
1.2.1.6. La vulcanología y los vulcanólogos.....	42
1.2.1.7. Observación de los volcanes .....	42
1.2.1.8. Terremotos y la deformación de suelos .....	43
1.2.2. Sismología .....	44
1.2.2.1. La investigación sismológica.....	44

1.2.2.2.	Onda Sísmica.....	45
1.2.2.3.	Onda P.....	46
1.2.2.4.	Onda S.....	47
1.2.2.5.	Ondas superficiales.....	47
1.2.2.6.	Tremor volcánico.....	49
1.2.2.7.	Explosiones.....	49
1.2.2.8.	Sismos y fallas .....	50
1.2.2.9.	Foco sísmico .....	51
1.2.2.10.	Clasificación de los sismos .....	53
1.2.2.11.	RSAM - Medida de amplitud sísmica en tiempo real.....	54
1.2.2.12.	Funcionamiento del RSAM.....	55
1.2.3.	Gases volcánicos .....	57
1.2.3.1.	Vapor de agua (H <sub>2</sub> O).....	58
1.2.3.2.	Dióxido de azufre (SO <sub>2</sub> ) .....	59
1.2.3.3.	Sulfuro de hidrógeno (H <sub>2</sub> S) .....	60
1.2.3.4.	Dióxido de carbono (CO <sub>2</sub> ) .....	61
1.2.3.5.	Cloruro de hidrógeno (HCl) .....	61
1.2.3.6.	Fluoruro de hidrógeno (HF) .....	62
1.2.3.7.	Metano (CH <sub>4</sub> ) .....	62
1.2.3.8.	Hidrógeno (H <sub>2</sub> ) .....	62
1.2.3.9.	Helio (He).....	63
1.2.3.10.	Nitrógeno (N <sub>2</sub> ) .....	63
1.2.3.11.	Importancia de los flujos de SO <sub>2</sub> en el monitoreo volcánico..	63
1.2.3.12.	Mediciones de flujos de SO <sub>2</sub> volcánicos.....	64
1.2.3.13.	Emisión de gases en el volcán de Santa Ana .....	64

1.2.4.	Herramientas estadísticas y matemáticas.....	66
1.2.4.1.	Análisis de series temporales.....	66
1.2.4.2.	Transformada rápida de Fourier.....	69
1.3.	PLANTEAMIENTO DEL PROBLEMA.....	73
1.3.1.	Situación problema .....	73
1.3.2.	Formulación del problema.....	74
1.3.2.1.	Dificultad en la recopilación de los datos requeridos .....	75
1.3.2.2.	Descentralización de la información.....	75
1.3.2.3.	Procesamiento lento y sujeto a errores humanos .....	75
1.3.2.4.	Cálculo y análisis de los indicadores limitados.....	76
1.3.2.5.	Información sin posibilidad de ser compartida .....	76
1.3.2.6.	Complicaciones para consultar datos históricos .....	76
1.4.	JUSTIFICACIÓN.....	77
1.5.	OBJETIVOS.....	79
1.5.1.	Objetivo general .....	79
1.5.2.	Objetivos específicos .....	79
1.6.	ALCANCES .....	80
1.7.	LIMITACIONES .....	82
1.8.	METODOLOGÍA DE DESARROLLO.....	83
1.9.	PLANIFICACIÓN DE RECURSOS A UTILIZAR.....	87
1.9.1.	Recursos humanos .....	87
1.9.2.	Otros recursos.....	87
1.9.3.	Recursos de hardware .....	87
1.9.4.	Recursos de software .....	88
	CAPÍTULO II: DETERMINACIÓN DE REQUERIMIENTOS.....	89

2.	DETERMINACIÓN DE REQUERIMIENTOS .....	90
2.1.	REQUERIMIENTOS INFORMÁTICOS.....	90
2.1.1.	Requerimientos funcionales .....	90
2.1.2.	Requerimientos no funcionales .....	91
2.2.	REQUERIMIENTOS DE DESARROLLO.....	92
2.2.1.	Recurso humano.....	92
2.2.2.	Requerimientos legales.....	94
2.2.2.1.	Licencias .....	94
2.2.3.	Requerimientos de software.....	95
2.2.3.1.	Lenguaje de programación.....	95
2.2.3.2.	Entorno de desarrollo .....	96
2.2.3.3.	Framework .....	96
2.2.3.4.	Gestor de bases de datos .....	97
2.2.3.5.	Servidor de aplicaciones web.....	98
2.2.3.6.	Aplicación de escritorio .....	99
2.2.3.7.	Entorno de java .....	99
2.2.3.8.	Sistema operativo .....	100
2.2.3.9.	Herramientas utilitarias y ofimática .....	100
2.2.4.	Requerimientos de hardware .....	101
2.2.5.	Requerimientos de diseño de sistemas .....	102
2.2.5.1.	Técnicas de análisis.....	102
2.2.5.2.	Técnicas de programación.....	103
2.2.5.3.	Técnicas de diseño .....	105
2.3.	REQUERIMIENTOS OPERATIVOS.....	107
2.3.1.	Recurso humano.....	107

2.3.2.	Requerimientos legales.....	108
2.3.2.1.	Licencias .....	108
2.3.3.	Requerimientos de software.....	108
2.3.3.1.	Requerimientos generales de software del cliente.....	108
2.3.3.2.	Requerimientos generales de software del servidor .....	109
2.3.3.3.	Sistema operativo .....	109
2.3.3.4.	Earthworm.....	110
2.3.3.5.	Novac program.....	113
2.3.4.	Requerimientos de hardware .....	115
2.3.4.1.	Requerimientos generales de hardware.....	115
2.3.4.2.	Amazon web services .....	116
2.4.	PROPUESTA DE SOLUCIÓN.....	118
2.4.1.	Diagrama de enfoque del sistema propuesto.....	118
2.4.2.	Descripción del enfoque del sistema propuesto.....	119
2.4.2.1.	Medio ambiente.....	119
2.4.2.2.	Frontera.....	120
2.4.2.3.	Entrada.....	120
2.4.2.4.	Salida .....	121
2.4.2.5.	Procesos .....	123
2.4.2.6.	Control.....	124
2.4.3.	Diagramas de casos de uso (MAVAMSI).....	126
2.4.3.1.	Diagrama general (MAVAMSI).....	126
2.4.3.2.	Subsistema gestión de accesos.....	127
2.4.3.3.	Subsistema gestión de vulcanología.....	128
2.4.3.4.	Subsistema análisis de gases .....	129

2.4.3.5.	Subsistema análisis de sismicidad.....	129
2.4.3.6.	Subsistema gestión de notificaciones .....	130
2.4.1.	Diagramas de casos de uso (MAVAMSI_BOT1).....	131
2.4.1.1.	Diagrama general (MAVAMSI_BOT1).....	131
2.4.2.	Diagramas de casos de uso (MAVAMSI_BOT2).....	132
2.4.2.1.	Diagrama general (MAVAMSI_BOT2).....	132
CAPÍTULO III: DISEÑO DE LA SOLUCIÓN.....		133
3.	DISEÑO DE LA SOLUCIÓN.....	134
3.1.	Diseño de Arquitectura de Software .....	135
3.1.1.	Software In House de la Solución .....	138
3.1.1.1.	Máquina Virtual (AMAZON_VM) .....	139
3.1.1.2.	Procesos automáticos .....	147
3.1.1.3.	Aplicación Web .....	152
3.1.2.	Software de Terceros.....	155
3.1.2.1.	Servidores de terceros .....	155
3.1.2.2.	Aplicaciones de terceros .....	159
3.1.3.	Fuentes de Datos.....	161
3.1.3.1.	NOVAC Program y Emisiones de Dióxido de Azufre .....	162
3.1.3.2.	Earthworm y RSAM.....	172
3.2.	DISEÑO DE BASE DE DATOS .....	180
3.2.1.	Data Warehouse .....	180
3.2.2.	Modelo Entidad Relación .....	183
3.2.3.	Tablas y Campos .....	184
3.2.4.	Sistema de Gestor de Base de Datos .....	192
3.3.	DISEÑO DE COMPONENTES DE SOFTWARE.....	199

3.3.1.	Entornos y Lenguaje de Programación .....	199
3.3.1.1.	Java.....	201
3.3.1.2.	Java Virtual Machine .....	203
3.3.1.3.	Java Runtime Environment .....	205
3.3.1.4.	Java Developer Kit .....	206
3.3.1.5.	NetBeans IDE .....	207
3.3.2.	Diseño y Desarrollo de Componentes In House .....	209
3.3.2.1.	MAVAMSI_BOT2 .....	210
3.3.2.2.	MAVAMSI_BOT1 .....	214
3.3.2.3.	MAVAMSI Aplicación web.....	219
3.3.3.	Diagramas de Flujo .....	223
3.3.3.1.	Diagrama de flujo de MAVAMSI_BOT2 .....	224
3.3.3.2.	Diagramas de flujo de MAVAMSI_BOT2.....	227
3.3.3.3.	Diagramas de flujo de MAVAMSI.....	234
3.3.4.	Diagramas de Clases.....	250
3.3.4.1.	Clases de MAVAMSI_BOT1 .....	251
3.3.4.2.	Clases MAVAMSI.....	254
3.4.	DISEÑO DE INTERFACES Y NAVEGACIÓN .....	280
3.4.1.	MAVAMSI, el sitio web.....	280
3.4.2.	Mapa de Navegación .....	284
3.4.3.	Diseño de Vistas .....	286
3.5.	DISEÑO DE SEGURIDAD.....	288
3.5.1.	Inicio de Sesión.....	288
3.5.2.	Roles y permisos.....	289
CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN.....		292

4.	DESARROLLO DE LA SOLUCIÓN .....	293
4.1.	PLAN DE ACTIVIDADES DE DESARROLLO INICIAL.....	294
4.2.	DESARROLLO DE MAVAMSI_BOT2 .....	298
4.2.1.	Recursos necesarios.....	298
4.2.2.	Actividades previas y de preparación.....	298
4.2.3.	Actividades de desarrollo .....	298
4.2.4.	Actividades de pruebas de desarrollo .....	299
4.2.5.	Retos surgidos durante desarrollo y pruebas.....	299
4.2.6.	Actividades de ajuste de desarrollo.....	300
4.2.7.	Limitantes de desarrollo .....	300
4.3.	DESARROLLO DE MAVAMSI_BOT1 .....	301
4.3.1.	Recursos necesarios.....	301
4.3.2.	Actividades previas y de preparación.....	301
4.3.3.	Actividades de desarrollo .....	302
4.3.4.	Actividades de pruebas de desarrollo .....	302
4.3.5.	Retos surgidos durante desarrollo y pruebas.....	302
4.3.6.	Actividades de ajuste de desarrollo.....	306
4.3.7.	Limitantes de desarrollo .....	306
4.4.	DESARROLLO DE MAVAMSI.....	307
4.4.1.	Recursos necesarios.....	307
4.4.2.	Actividades previas y de preparación.....	307
4.4.3.	Actividades de desarrollo .....	307
4.4.4.	Actividades de pruebas de desarrollo .....	308
4.4.5.	Retos surgidos durante desarrollo y pruebas.....	308
4.4.6.	Actividades de ajuste de desarrollo.....	309

4.4.7.	Limitantes de desarrollo .....	309
4.5.	PLAN DE ACTIVIDADES DE DESARROLLO REAL.....	310
CAPÍTULO 5: PLAN DE IMPLEMENTACIÓN.....		316
5.	PLAN DE IMPLEMENTACIÓN .....	317
5.1.	METODOLOGÍA DE IMPLEMENTACIÓN.....	319
5.1.1.	Preparación.....	319
5.1.1.1.	Objetivos de la etapa .....	319
5.1.1.2.	Actividades de la etapa .....	319
5.1.2.	Instalación y acondicionamiento .....	320
5.1.2.1.	Objetivo de la etapa .....	320
5.1.2.2.	Actividades de la etapa .....	320
5.1.3.	Pruebas al sistema.....	320
5.1.3.1.	Objetivo de la etapa .....	320
5.1.3.2.	Actividades de la etapa .....	321
5.1.4.	Capacitación de usuarios .....	322
5.1.4.1.	Objetivo de la etapa .....	322
5.1.4.2.	Actividades de la etapa .....	322
5.2.	ACTIVIDADES DE IMPLEMENTACIÓN DE SOFTWARE .....	323
5.3.	CONFIGURACIÓN DE SERVIDORES.....	325
5.3.1.	Configuración de Máquina Virtual .....	325
5.3.1.1.	Sistema Operativo.....	326
5.3.1.2.	Puesta en Marcha y Monitoreo de Instancia de Máquina Virtual 328	
5.3.1.3.	Configuración de red .....	330
5.3.2.	Configuración de Máquina en Estación de Observación.....	331

5.4.	CONFIGURACIÓN DE SERVIDORES DE SOFTWARE.....	333
5.4.1.	FileZilla.....	333
5.4.1.1.	Instalación de FileZilla Server .....	334
5.4.1.2.	Configuración de usuario FTP.....	334
5.4.1.3.	Configuración de estructura de carpetas en servidor.....	336
5.4.2.	GlassFish.....	337
5.4.2.1.	Instalación de GlassFish .....	337
5.4.3.	MySQL .....	338
5.4.3.1.	Instalación de MySQL Server.....	339
5.4.3.2.	Instalación de MySQL Workbench .....	339
5.4.3.3.	Creación de Instancia de Base de Datos .....	341
5.4.3.4.	Implementación de Base de Datos .....	343
5.5.	IMPLEMENTACIÓN DE SOFTWARE DE TERCEROS.....	346
5.5.1.	Novac Program .....	346
5.5.1.1.	Configuración de Red.....	346
5.5.1.2.	Configuración de ScanDoas en Novac Program.....	348
5.5.2.	EarthWorm .....	352
5.5.2.1.	Actividades previas y de preparación.....	352
5.5.2.2.	Actividades de Implementación y Configuración.....	353
5.5.2.3.	Actividades de Pruebas de Implementación .....	353
5.5.2.4.	Retos Surgidos durante Pruebas e Implementación de EarthWorm	354
5.5.2.5.	Ajustes a configuración inicial .....	354
5.5.2.6.	Limitantes de Implementación de EarthWorm.....	355
5.6.	IMPLEMENTACIÓN DE SOFTWARE IN HOUSE .....	356

5.6.1.	MAVAMSI_BOT2 .....	356
5.6.1.1.	Implementación de MAVAMSI_BOT2 .....	356
5.6.1.2.	Configuración de NovacTrigger.....	357
5.6.1.3.	Configuración de FTPRobotScheduler.....	358
5.6.2.	MAVAMSI_BOT1 .....	358
5.6.2.1.	Configuración de rutas de carpetas de archivos .....	359
5.6.3.	MAVAMSI.....	361
5.6.3.1.	Despliegue de MAVAMSI en GlassFish.....	361
5.6.3.2.	Configuración de JDBC en GlassFish.....	362
5.7.	ACTIVIDADES DE PLAN DE IMPLEMENTACIÓN .....	365
	CONCLUSIÓN.....	368
	REFERENCIAS BIBLIOGRÁFICAS.....	369
	ANEXOS .....	377
	ANEXO 1: SOLICITUD DE INFORMACIÓN DE MOVIMIENTO SÍSMICO AL MARN .....	378
	ANEXO 2: CARTA DE COMPROMISO CON MARN .....	379
	ANEXO 3: CARTA DE CONFORMIDAD DE FINALIZACIÓN DE APLICACIÓN .....	380
	ANEXO 4: VIDEOS DE MANUALES DE USUARIO.....	381
	ANEXO 5: EVIDENCIAS DE EQUIPO DE TRABAJO EN DESARROLLO DEL PROYECTO.....	383

# ÍNDICE DE FIGURAS

Figura 1.1 - Instituto de Vulcanología IV-UES.....	35
Figura 1.2 - Propagación de ondas sísmicas en el interior de la Tierra .....	45
Figura 1.3 - Ejemplos de falla inversa, normal y lateral.....	46
Figura 1.4 - Movimiento y propagación de ondas sísmicas. ....	48
Figura 1.5 - Distribución de sismicidad. ....	51
Figura 1.6 - Representación de Foco Sísmico. ....	52
Figura 1.7 - Cráter del volcán Ixmiquilpan de Santa Ana.....	65
Figura 1.8 - Dominio temporal, y Análisis espectral aplicado.....	69
Figura 1.9 - Análisis tiempo-frecuencia mediante transformada wavelet. ....	71
Figura 1.10 - Escalograma 3D de la serie temporal de la emisión difusa de dióxido de carbono el volcán San Vicente, en 2002. ....	72
Figura 1.11 - Representación de la metodología del ciclo de vida de software en V.....	85
Figura 2.1 - Diagrama de enfoque del sistema propuesto. ....	118
Figura 2.2 - Diagrama de caso de uso “General (MAVAMSI)”.....	126
Figura 2.3 - Diagrama de caso de uso “Gestión de Accesos”.....	127
Figura 2.4 - Diagrama de caso de uso “Gestión de Vulcanología”.....	128
Figura 2.5 - Diagrama de caso de uso “Análisis de Gases”.....	129
Figura 2.6 - Diagrama de caso de uso “Análisis de Sismicidad”.....	129
Figura 2.7 - Diagrama de caso de uso “Gestión de Notificaciones”.....	130
Figura 2.8 - Diagrama de caso de uso “General (MAVAMSI_BOT1)”.....	131
Figura 2.9 - Diagrama de caso de uso “General (MAVAMSI_BOT2)”.....	132
Figura 3.1 - Diagrama de componentes de software .....	135
Figura 3.2- Símbolos de diagrama de componentes.....	136
Figura 3.3 - Diagrama de componentes de software In House .....	139
Figura 3.4 - Diagrama de funcionamiento de MAVAMSI_BOT2 .....	149

Figura 3.5 - Diagrama de funcionamiento de MAVAMSI_BOT1 .....	151
Figura 3.6 - Diagrama de servidores de terceros .....	155
Figura 3.7 - Pantalla principal de Novac Program.....	164
Figura 3.8 - Diagrama de módulos de Earthworm.....	175
Figura 3.9 - Modelo Entidad Relación de Base de Datos de la Sistema .....	183
Figura 3.10 - Elementos que interactúan con MAVAMSI_BOT2.....	213
Figura 3.11 - Estructura de carpetas de archivos de emisión de gases en servidor .....	215
Figura 3.12 - Estructura de carpetas de archivos de RSAM en servidor.....	216
Figura 3.13 - Diagrama de flujo de proceso de transmisión de archivos de mediciones de emisión de gases vía FTP .....	224
Figura 3.14 - Diagrama de flujo de proceso de obtención de archivos de mediciones de RSAM vía Earthworm.....	227
Figura 3.15 - Diagrama de flujo de proceso de inserción de mediciones en base de datos y envío de alertas por correo electrónico.....	228
Figura 3.16 - Diagrama de flujo de proceso de inicio de sesión en MAVAMSI	234
Figura 3.17 – Diagrama de flujo de proceso de despliegue de gráfico de concentración de gas .....	235
Figura 3.18 - Diagrama de flujo de proceso despliegue de gráfico de contracción de gas por rango .....	238
Figura 3.19 - Diagrama de flujo de proceso de despliegue de gráfico de RSAM .....	241
Figura 3.20 - Diagrama de flujo de proceso despliegue de gráfico de RSAM por rango .....	244
Figura 3.21 - Diagrama de flujo de proceso despliegue de gráfico de RSAM vs Emisiones de gas .....	247
Figura 3.22 - Diagrama de clase Archivo .....	251
Figura 3.23 - Diagrama de clase Conexion.....	251
Figura 3.24 - Diagrama de clase Gas .....	251
Figura 3.25 - Diagrama de clase MainView .....	252

Figura 3.26 - Diagrama de clase MAVAMSIBOT .....	253
Figura 3.27 - Diagrama de clase Volcan .....	253
Figura 3.28 - Diagrama de clase Property .....	254
Figura 3.29 - Diagrama de clase ServicioEmail .....	254
Figura 3.30 - Diagrama de paquetes de clases de aplicación web MAVAMSI	255
Figura 3.31 - Diagrama de clases de paquete Entidades, de proyecto MAVAMSI	256
.....	
Figura 3.32 - Diagrama de clase LogRSAM.....	257
Figura 3.33 - Diagrama de clase Estacion .....	258
Figura 3.34 - Diagrama de clase Volcan .....	258
Figura 3.35 - Diagrama de clase Log .....	259
Figura 3.36 - Diagrama de clase Gas .....	260
Figura 3.37 - Diagrama de clase GasEstacion.....	260
Figura 3.38 - Diagrama de clase Menu .....	261
Figura 3.39 - Diagrama de clase Parametros.....	261
Figura 3.40 - Diagrama de clase Persona.....	262
Figura 3.41 - Diagrama de clase Usuario.....	262
Figura 3.42 - Diagrama de clase Rol.....	263
Figura 3.43 - Diagrama de clase RolMenu.....	263
Figura 3.44 - Diagrama de clase UsuarioControlller .....	264
Figura 3.45 - Diagrama de clase RolMenuController .....	265
Figura 3.46 - Diagrama de clase LogController.....	266
Figura 3.47 - Diagrama de clase LogRSAMController .....	267
Figura 3.48 - Diagrama de clase GasController .....	268
Figura 3.49 - Diagrama de clase EstacionController.....	268
Figura 3.50 - Diagrama de clase GasEstacionController .....	269
Figura 3.51 - Diagrama de clase VolcanController.....	270
Figura 3.52 - Diagrama de clase ComparativoController .....	271
Figura 3.53 - Diagrama de clase DatosGas .....	272
Figura 3.54 - Diagrama de clase DatosRSAM .....	273

Figura 3.55 - Diagrama de clase DatosGasRango.....	273
Figura 3.56 - Diagrama de clase ServicioCorreo .....	273
Figura 3.57 - Diagrama de clase DatosZonasHorarias .....	274
Figura 3.58 - Diagrama de clases de paquete Facades de proyecto MAVAMSI .....	275
Figura 3.59 - Diagrama de clase AbstractFacade .....	276
Figura 3.60 - Diagrama de clase EstacionFacade .....	276
Figura 3.61 - Diagrama de clase GasEstacionFacade .....	276
Figura 3.62 - Diagrama de clase GasFacade .....	276
Figura 3.63 - Diagrama de clase LogFacade .....	277
Figura 3.64 - Diagrama de clase LogRSAMFacade .....	277
Figura 3.65 - Diagrama de clase MenuFacade .....	277
Figura 3.66 - Diagrama de clase ParametrosFacade.....	277
Figura 3.67 Diagrama de clase PersonaFacade .....	278
Figura 3.68 - Diagrama de clase RolFacade.....	278
Figura 3.69 - Diagrama de clase RolMenuFacade.....	278
Figura 3.70 - Diagrama de clase TipoVolcanFacade .....	278
Figura 3.71 - Diagrama de clase UsuarioFacade.....	279
Figura 3.72 - Diagrama de clase VolcanFacade .....	279
Figura 3.73 - Logo de MAVAMSI .....	281
Figura 3.74 - Mapa de Navegación de MAVAMSI.....	285
Figura 3.75 - Diseño de plantilla de vistas de MAVAMSI .....	286
Figura 3.76 - Pantalla de inicio de MAVAMSI .....	287
Figura 3.77 - Módulo de vista de inicio de sesión a MAVAMSI.....	288
Figura 4.1 - Diagrama de plan de actividades de desarrollo inicial de aplicaciones in house.....	294
Figura 4.2 - Diagrama de plan de actividades de desarrollo real de aplicaciones in house.....	310
Figura 5.1 - Metodología de Implementación.....	319

Figura 5.2 - Elección de sistema operativo Windows Server de instancia de máquina virtual en Amazon EC2 .....	327
Figura 5.3 - Elección de sistema operativo Linux de instancia de máquina virtual en Amazon EC2 .....	327
Figura 5.4 - Monitoreo de instancia de máquina virtual.....	328
Figura 5.5 - Monitoreo de recursos de instancia de maquina virtual .....	329
Figura 5.6 - Vista de máquina virtual AMAZON_VM .....	330
Figura 5.7 - Configuraciones de red de máquina virtual.....	331
Figura 5.8 - Administración de suarios en FileZilla Server .....	335
Figura 5.9 - Configuración de carpetas compartidas en FileZilla .....	336
Figura 5.10 - GlassFish ejecutándose en máquina virtual .....	338
Figura 5.11 - MySQL Workbench instalado en AMAZON_VM .....	341
Figura 5.12 - Creación de instancia de conexión a la base de datos .....	343
Figura 5.13 - Ingeniería Inversa de Workbench para importación de esquema de base de datos MAVAMSI .....	345
Figura 5.14 - Esquema MAVAMSI importado en base de datos .....	345
Figura 5.15 - Configuración de red para conexión de ScanDoas.....	347
Figura 5.16 - Configuración de modelo de dispositivo ScanDoas en Novac Program.....	348
Figura 5.17 - Confiuración de evaluación maestra de ScanDoas en Novac Program.....	349
Figura 5.18 - Configuración de comunicación entre ScanDoas y Novac Program .....	350
Figura 5.19 - Configuración de carpeta de salida de archivos de texto de mediciones de emision de gases en Novac Program.....	351
Figura 5.20 - Configuración finalizada de Novac Program.....	352
Figura 5.21 - Configuración de NovacTrigger .....	357
Figura 5.22 - Configuración de FTPRobotScheduler .....	358
Figura 5.23 - Ejecución de MAVAMSI_BOT1 en máquina virtual .....	359
Figura 5.24 - Estructura de carpetas para archivos de emisiones de gases ...	360

Figura 5.25 - Estructura de carpetas para archivos de RSAM ..... 361

Figura 5.26 - Implementación de MAVAMSI en GlassFish ..... 362

Figura 5.27 - Configuración de pool de conexión a base de datos desde GlassFish  
..... 363

Figura 5.28 - Propiedades de pool de conexión a base de datos desde GlassFish  
..... 363

Figura 5.29 - MAVAMSI implementada correctamente ..... 364

Figura 5.30 – Diagrama de actividades del plan de implementación de la solución  
..... 365

# ÍNDICE DE CUADROS

Cuadro 1.1 - Técnicas de Monitoreo utilizadas por SNET. ....	32
Cuadro 2.1 - Requerimientos funcionales del sistema. ....	90
Cuadro 2.2 - Requerimientos no funcionales del sistema. ....	91
Cuadro 2.3 - Competencias de los integrantes del equipo de desarrollo. ....	92
Cuadro 2.4 - Funciones de los integrantes del equipo de desarrollo. ....	93
Cuadro 2.5 - Hardware y sus especificaciones del equipo de desarrollo. ....	102
Cuadro 2.6 - Perfil Cargo-Puesto del Administrador del Sistema.....	107
Cuadro 2.7 - Requerimientos mínimos de software en máquina cliente. ....	109
Cuadro 2.8 - Requerimientos mínimos de software en máquina servidor. ....	109
Cuadro 2.9 - Requerimientos mínimos de hardware en maquina cliente.....	115
Cuadro 2.10 - Requerimientos mínimos de hardware en maquina servidor. ..	116
Cuadro 3.1 - Definición de tabla ESTACION .....	184
Cuadro 3.2 - Definición de tabla GAS .....	184
Cuadro 3.3 – Definición de tabla GASESTACION .....	185
Cuadro 3.4 - Definición de tabla HISTORIALLOG .....	185
Cuadro 3.5 - Definición de tabla HISTORIALRSAM.....	186
Cuadro 3.6 - Definción de tabla LOG .....	186
Cuadro 3.7 - Definición de tabla LOGRSAM.....	189
Cuadro 3.8 - Definción de tabla MENU .....	189
Cuadro 3.9 - Definición de tabla PARAMETROS.....	190
Cuadro 3.10 - Definición de tabla PERSONA .....	190
Cuadro 3.11 - Definición de tabla ROL .....	190
Cuadro 3.12- Definición de tabla ROLMENU .....	191
Cuadro 3.13 - Definición de tabla TIPOVOLCAN.....	191
Cuadro 3.14 - Definición de tabla USUARIO .....	191
Cuadro 3.15 - Definición de tabla USUARIOALERTA.....	192

Cuadro 3.16 - Definición de tabla VOLCAN .....	192
Cuadro 5.1 - Componentes de software a implementar .....	324
Cuadro 5.2 - Componentes a implementar de MAVAMSI_BOT2.....	356
Cuadro 5.3 - Parámetros de archivo de configuración de MAVAMSI_BOT2 ..	357

# ÍNDICE DE TABLAS

Tabla 1.1 - Recursos de hardware a utilizar.....	87
Tabla 1.2 - Recursos de software a utilizar. ....	88
Tabla 3.1 - Comparación entre proveedores de servicios en la nube .....	145
Tabla 3.2 - Definición de campos de archivo de mediciones de dióxido de azufre .....	170
Tabla 3.3 - Generalidades de archivo de dióxido de azufre .....	171
Tabla 3.4 - Definición de campos de archivo de mediciones de RSAM .....	178
Tabla 3.5 - Generalidades de archivo de RSAM .....	179
Tabla 3.6 - Comparación entre sistemas gestores de base de datos .....	193
Tabla 3.7 - Comparación entre lenguajes de programación .....	200
Tabla 3.8 - Roles y Permisos de MAVAMSI.....	291
Tabla 5.1 - Actividades de capacitación .....	322

# INTRODUCCIÓN

El Salvador, es uno de los países con mayor exposición a eventos sísmicos y volcánicos de la región. Ya que formamos parte del Cinturón de Fuego del Pacífico, la cadena volcánica más densa del mundo. Específicamente, nuestra ciudad, cuenta con el volcán más alto del país, el Volcán Ilamatepec.

Actualmente, han existido diferentes proyectos de software, que se encargan de monitorizar la actividad emergente de este coloso, los cuales, han ayudado a conocer su comportamiento. Sin embargo, los resultados arrojados por los diferentes aplicativos, miden variables independientes que nunca llegan a relacionarse con otras, y de esta forma, enriquecer más el estudio del mismo y buscar reacciones inmediatas antes posibles tragedias.

El presente documento, es un proyecto de software que busca interrelacionar dos de las variables más representativas en la actividad volcánica: la actividad sísmica y las emisiones de dióxido de azufre; quienes, a lo largo de la historia, han demostrado tener una intrínseca relación entre sí en actividades volcánicas pasadas.

A continuación, iniciamos con un estudio preliminar, en el cual se brinda un acercamiento a la problemática que se desea abordar, así como el Marco Teórico necesario para comprender los fundamentos vulcanológicos indispensables para el diseño y desarrollo del proyecto. Luego, se describen los requerimientos identificados, que guiarán el correcto diseño de la solución a realizar; así como nuestra propuesta de diseño de solución que busque cubrir las necesidades identificadas. Finalizamos con el plan de implementación, pasando por todos los retos que nos tocó vencer durante el desarrollo del proyecto.

---

# **CAPÍTULO I: ESTUDIO PRELIMINAR**

---

# 1. ESTUDIO PRELIMINAR

## 1.1. ANTECEDENTES

### *1.1.1. Red de monitoreo vulcanológico del Servicio Nacional de Estudios Territoriales.*

El Servicio Nacional de Estudios Territoriales, mejor conocido como *SNET* por sus siglas, es una institución dirigida por el Ministerio de Medio Ambiente y Recursos Humanos de El Salvador (MARN).

El SNET tiene como propósito la realización de estudios y monitoreo ambientales, que tengan relación directa e indirecta con la dinámica territorial y la probabilidad de que se concreten pérdidas y daños económicos, sociales y ambientales. El SNET busca brindar las referencias territoriales en cuanto a las amenazas, vulnerabilidades, capacidades y oportunidades, a fin de mejorar la toma de decisiones para el desarrollo sostenible y la seguridad humana.

“En esta red es donde se desarrolla y opera Sistemas de Alerta Temprana contra erupciones volcánicas. Para ello, se realizan trabajos de instrumentación, monitoreo sismo-volcánico, observación visual por parte de personal local y vulcanólogos especializados, medición de temperatura en fumarolas, fisuras, pozos y cuerpos de agua, medición química de gases, aguas y cenizas, mediciones geodésicas (GPS, EDM), utilización de sensores remotos (fotografías aéreas, imágenes de satélite, imágenes de radar, entre otros).

También es competencia de esta área, realizar monitoreo de erupciones volcánicas (sismicidad volcánica, tremor) y evaluación de amenaza, por medio de los métodos arriba detallados, coordinar y operar el sistema de alerta a la aviación civil, contra nubes de ceniza volcánica en coordinación con Sismología y Meteorología; fomentar, supervisar, coordinar y ejecutar estudios de: amenaza

volcánica en el marco nacional y local, vulcanismo histórico, régimen tectónico antes y después de erupciones (en cooperación con el IGN), construir y mantener bancos de datos de: vulcanismo histórico y reciente, datos vulcanológicos y afectación por gases volcánicos.

La elaboración de Comunicados de alerta, boletines informativos extraordinarios y ordinarios con periodicidad mensual y anual, mapas de Amenazas Volcánicas y Mapas de Riesgo Volcánico, es de crucial importancia a fin de brindar información rápida y oportuna a las autoridades del gobierno central, gobiernos locales y público en general sobre cualquier condición de amenaza.” (SNET, 2016)

Las técnicas de monitoreo que el SNET utiliza para el monitoreo de los principales volcanes activos de El Salvador se pueden observar en el siguiente cuadro.

*Cuadro 1.1 - Técnicas de Monitoreo utilizadas por SNET.*

TÉCNICAS DE MONITOREO UTILIZADAS POR SNET EN LOS PRINCIPALES VOLCANES ACTIVOS DE EL SALVADOR						
TÉCNICA / DEPARTAMENTO	SANTA ANA	SAN MIGUEL	SAN SALVADOR	IZALCO	SAN VICENTE	ILOPANGO
SÍSMICO						
GASES (SCANDOAS SO2)						
HIDROGEOQUÍMICO						
TEMPERATURA						
IN-SITU						
SENSORES						
TERMOGRAFÍA(IR)						
VISUAL						
OBSERVACIONES DE CAMPO						
CÁMARA WEB						
DEFORMACIÓN GPS (PLAN PILOTO)						

*Fuente: Servicio Nacional de Estudios Territoriales - SNET. (2016). Servicio Geológico.*

Por otra parte, se mantiene coordinación con el Grupo de Investigación Vulcanológica de la Universidad Nacional de El Salvador, quien está a cargo del monitoreo geoquímico de gases difusos, para el intercambio y análisis de datos.

### ***1.1.2. Red de monitoreo del Instituto de Vulcanología IV-UES***

“A raíz de los terremotos ocurridos en los meses de enero y febrero del año 2001; la Universidad de El Salvador (UES) y el Ministerio del Medio Ambiente y Recursos Naturales (MARN) solicitaron al gobierno español, a través de la Agencia Española de Cooperación Internacional (AECI) la asistencia técnica del Instituto Tecnológico y de Energías Renovables (ITER) organismo dependiente del Cabildo Insular de Tenerife (Islas Canarias, España) para evaluar el impacto que estos fuertes sismos pudieran ejercer sobre los principales sistemas volcánicos salvadoreños, y materializar acciones destinadas a contribuir a la reducción del riesgo volcánico en El Salvador.

Este esfuerzo fue concretado a través del proyecto “Apoyo a la Vigilancia Volcánica en El Salvador”, financiado por AECI entre los años 2001 y 2002.

La asistencia técnica continuó durante los años 2003 al 2005 dentro del marco del proyecto “Prevención de Riesgos Sísmicos y Volcánicos en El Salvador” financiado por el Fondo Salvadoreño de Cooperación Mixta España – El Salvador, Fondo Mixto 2003. El máximo organismo responsable de la ejecución de este proyecto fue la UES, teniendo como contraparte al ITER.

Aun cuando es ampliamente aceptada por la comunidad científica internacional la relevancia extrema de los gases en el fenómeno volcánico, ya que estos afectan profundamente el ascenso de magma y la actividad eruptiva, solo recientemente el desarrollo de nuevas tecnologías ha permitido la realización de medidas rutinarias de diferentes tipos de gases que son emitidos por los volcanes.

Un claro objetivo en la monitorización o vigilancia de los gases volcánicos es determinar cambios en la liberación de ciertos gases procedentes del volcán que pueden indicar cambios de las condiciones del sistema volcánico. Estos cambios en la emisión y en la composición química de los gases volcánicos pueden utilizarse

junto a otras técnicas más convencionales, para mejorar y optimizar la sistemática de detección de señales de alerta temprana sobre futuras y posibles erupciones volcánicas, y en consecuencia mejorar nuestro conocimiento científico sobre el comportamiento de nuestros volcanes.

El Instituto de Vulcanología de la Universidad de El Salvador desarrolla labores de vigilancia y monitoreo volcánico, con el objetivo de dar respuesta a las comunidades, en cuanto a la prevención y mitigación del riesgo volcánico.

Algunos de los resultados logrados por la institución son los siguientes:

- Desgasificación difusa de dióxido de carbono ( $\text{CO}_2$ ) y sulfuro de hidrógeno ( $\text{H}_2\text{S}$ ) junto a parámetros meteorológicos (temperatura, humedad, etc.) monitorizados por medio de estaciones de la Red Geoquímicas Salvadoreñas REGEOSAL).
- Registro continuo de la emisión difusa de  $\text{CO}_2$  y  $\text{H}_2\text{S}$  en los volcanes de San Miguel, San Salvador, San Vicente y Santa Ana-Izalco-Coatepeque mediante el empleo de sensores de infrarrojo y electroquímicos.
- Se cuantifica la emisión de gases volcánicos como el  $\text{SO}_2$  utilizando técnicas de espectroscopia óptica. Esta técnica mide la luz solar absorbida por las moléculas del gas en estudio cuando la luz solar pasa a través de éstas.” (Universidad de El Salvador, 2016)

El Instituto de Vulcanología de la Universidad de El Salvador (IV-UES) está compuesto por un conjunto de personas profesionales comprometidas con la prevención y mitigación del riesgo volcánico, en la siguiente fotografía algunos de los integrantes del IV-UES.

*Figura 1.1 - Instituto de Vulcanología IV-UES*



*Fuente: Sol, M. (02 de febrero de 2016). El Salvador: Un volcán Activo. El Universitario.*

### **1.1.3. Red de monitoreo BIRA-IASB**

“Creado en 1964, sus principales tareas son la investigación y servicio público en aeronomía espacial, que es la física y la química de la atmosfera de la Tierra, otros planetas y del espacio exterior. Sus científicos se basan en modelos del globo terrestre, aire, instrumentos espaciales y modelos por computadora.

BIRA-IASB se centra en:

- Química y Física de atmósferas y Física del Espacio.
- Diseña y construye instrumentos para monitorear atmósferas y el entorno espacial.
- Opera experimentos belgas a bordo de la Estación Espacial Internacional y otros satélites.
- Participa en programas internacionales de medición.

- Compara observaciones con simulaciones numéricas para validar y mejorar nuestro conocimiento.
- Resulta conocimientos científicos en los servicios en beneficio de la sociedad.
- Difunde este conocimiento a través de publicaciones, servicios web y de difusión pública.” (Belgian Federal Scientific Research Institute, 2016)

## **1.2. MARCO CONCEPTUAL**

“La vulcanología es una rama de la geología que se dedica al estudio de los volcanes y lo relativo a sus erupciones, estructura, y origen. También estudia los efectos que los fenómenos volcánicos ejercen sobre la atmósfera e hidrosfera terrestre, así como el aporte de elementos químicos sobre la corteza terrestre y la distribución de los yacimientos minerales ligados a ellos. Se dedica a la clasificación de los productos volcánicos y las estructuras que imprimen una morfología típica de los terrenos para prevenir los riesgos geológicos de origen volcánico, mediante la emisión de pronósticos.” (SNET, 2016)

Antes de poder llevar a cabo el desarrollo de un sistema de información para el análisis de variables que integran esta ciencia, se debe tener una base teórica sobre sismología, como se forman los sismos, su clasificación, la relación que tienen con los volcanes y sus fenómenos geológicos.

Es por esto que el orden definido para el marco teórico al inicio aborda la temática de los sismos en general no solamente los provocados por volcanes. Luego dentro del estudio se incluyen los gases emanados por los volcanes, estos constituyen la mayor parte del volumen de las erupciones volcánicas, el vapor de agua es el gas más común, pero se contemplan otros gases importantes para la vulcanología. Por último, se hace una referencia a la teoría que sustenta el cálculo matemático de los indicadores que se utilizan para el seguimiento y pronóstico de fenómenos volcánicos.

### ***1.2.1. Volcanes***

“Un volcán es un conducto que pone en comunicación la parte superior de la corteza sólida con los niveles inferiores de la misma. Es también una estructura por la cual emergen el magma (roca fundida) en forma de lava y gases del interior del planeta.

Los volcanes se forman cuando el material caliente del interior de la Tierra asciende y se derrama sobre la corteza. Este material caliente, llamado magma, puede provenir de dos fuentes; del material derretido de la corteza en subducción o provenir de mucho más adentro del planeta.

En una zona del interior de la tierra o aumenta la temperatura o disminuye la presión., esto hace que se forme un magma. El magma suele contener materiales sólidos y un contenido en gases. Ahora, en su camino a la superficie, el magma, suele acumularse en una especie de depósito llamado cámara magmática. Si esta cámara se encuentra llena, los nuevos aportes de magma desencadenarán una erupción ya que se incrementa la presión en la cámara magmática y sus paredes se dilatan y fracturan. Mientras, los gases escapan y arrastran el resto del magma.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, pág. 4)

### ***1.2.1.1. Tipos de actividad volcánica***

“El ascenso del magma ocurre generalmente en episodios de actividad violenta denominados erupciones, la cuales pueden variar en, duración y frecuencia; siendo desde conductos de corrientes de lava hasta explosiones extremadamente destructivas. Pueden producirse dos tipos de actividad.

La actividad ***Efusiva*** que se caracteriza básicamente porque:

- La lava es basáltica y sale a altas temperaturas, siendo así poco viscosa. Al ser tan líquida, forma ríos de lava que bajan muy rápido por la ladera del volcán destruyéndolo todo a su paso.
- Los gases no se acumulan. La poca viscosidad de la lava facilita su ascenso de manera gradual. Esto reduce el número y la intensidad de las explosiones.
- Se forman pocos piroclastos porque la lava está a alta temperatura y cualquier material sólido se funde. Hay pocas explosiones debido al tipo de lava y la escasez de gases.

La actividad **Explosiva** que se caracteriza básicamente porque:

- La lava es muy viscosa, solidifica y obstruye los conductos de salida, lo que forma explosiones muy violentas que pueden llegar a volar el cráter entero formando una caldera.
- Los gases no se acumulan, aumentan la presión y provocan fuertes explosiones que proyectan al aire gran cantidad de piroclastos.
- Se originan abundantes piroclastos. Pueden formarse nubes bajas que se desplazan a gran velocidad sobre el suelo y arrastran todo lo que encuentran a su paso. Son las nubes ardientes.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, pág. 5)

### **1.2.1.2. Productos volcánicos**

“En una erupción volcánica el volcán expulsa:

- **Gases:** Son los primeros productos en llegar a la superficie. El componente más abundante de estos gases es el vapor de agua. Pero también hay vapor de agua.
- **Lavas:** son materiales fundidos que fluyen por la boca del volcán formando coladas a veces más lentas, a veces más rápidas. Depende de la cantidad de sílice que tengan pueden ser más o menos viscosas:
  - a) Cuanta más sílice, más viscosa será la lava y formará una actividad volcánica explosiva.
  - b) Cuanto menos, menos viscosa será la lava y se producirá una actividad efusiva.
- **Piroclastos:** son los materiales sólidos proyectados al aire por el escape violento de gases. Según su tamaño se clasifican en:
  - a) Ceniza: tamaño inferior a 2mm (son los de menor tamaño)
  - b) Lapilli: son de 2mm a 65mm.
  - c) Bombas o bloques: son de tamaño superior a 65mm y tienen una forma muy irregular.

- d) Si tienen forma más o menos redondeada se llaman bombas.”  
(Lippert, Sevillano Reimers, & Millán Flaile, 2011, pág. 5)

### **1.2.1.3. Tipos de relieve volcánico**

Edificios volcánicos:

- **Escudo:** es un cono volcánico de base muy amplia, formado por una actividad efusiva, con lavas muy fluidas que solidificaron lejos del cráter.
- **Estratovolcán:** cono formado por capas aleatorias de piroclastos y magmas solidificados. Suelen ser más consistentes y muy altos.
- **Domo:** es un edificio volcánico que en la última erupción producida en éste la lava solidificó nada más salir de la chimenea formando una cúpula en el cráter.

Calderas:

- **Caldera de Colapso:** es creada por el hundimiento del edificio volcánico por la acumulación de materiales sobre la cámara magmática.
- **Caldera de Explosión:** formada por una actividad explosiva, en una de las explosiones la parte superior del cono se destruyó.
- **Caldera de Erosión:** formada por la retirada de los materiales que cubrían la cámara magmática.” (Lippert, Sevillano Reimers, & Millán Flaile, pág. 6)

### **1.2.1.4. Origen y tipos de magma**

“El magma es un fundido de minerales (especialmente de silicatos y volátiles). Este fundido de minerales es un líquido muy complejo y se puede presentar muy variado, desde los fluidos a gran temperatura (superior a 1000C°) que forman los basaltos, hasta los espesos y relativamente fríos (600 a 700C°) que cristalizarán con la composición química de los granitos.

Es frecuente que el magma tenga una tendencia a ascender a través de las rocas hacia la superficie y que algún tiempo después, el magma comience a enfriarse y

acabe cristalizado en rocas magmáticas. Como el crecimiento de los cristales lleva tiempo, si el enfriamiento del magma es lento y en profundidad origina rocas plutónicas que presentan un aspecto granuloso; si es más rápido y se origina cerca o sobre la superficie serán rocas volcánicas, que se caracteriza por tener cristales microscópicos o por presentar un aspecto vítreo como la obsidiana.

Químicamente los elementos más abundantes en los magmas son aquellos que se originan en la corteza terrestre: O, Si, Al, Fe, Mg, Ca, Na y K, que ocupan el 99% del volumen.

Si ordenamos los magmas según sus componentes en sílice, éstos se clasifican en:

- Los magmas ácidos que poseen un 66% de sílice.
- Los magmas intermedios o neutros que poseen un 52 o 66% de sílice.
- Los magmas básicos que poseen un 45 o 52% de sílice.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, págs. 7-8)

#### ***1.2.1.5. Diferenciación magmática***

“La otra posible evolución magmática se produce cuando por algún motivo los minerales dejan de estar en contacto con el magma y se producen rocas de distinta composición.

El caso más frecuente de este punto es la diferenciación por gravedad, es decir, el descenso de los cristales.

La diferenciación magmática puede diversificar un magma homogéneo en varias rocas finales de composiciones muy distintas:

- ***Magma ácido:*** con rocas ricas en sílice de composición equivalente a la de los granitos. Este tipo de magma produce rocas plutónicas (granitos) y rocas volcánicas (riolitas). Los granitos se caracterizan porque son claros, granulados, con feldespato, cuarzo y biotita. Puede formarse por dos tipos de

orígenes: Por la anatexia: fusión total de la composición inicial de componentes marinos. Y por diferenciación magmática a partir de magmas básicos en presencia de restos transformados de rocas metamórficas. Las riolitas son equivalentes a los granitos con abundancia en vidrio.

- **Magma intermedio:** con rocas de una mezcla magmática entre el magma ácido y básico que se caracteriza por ausencia de cuarzo y olivino. Los minerales esenciales son las plagioclasas calco sódicas y las piroxenas.
- **Magma básico:** con rocas pobres en sílice de una composición equivalente a la de los basaltos. Puede contener dos tipos de rocas: los gabros y los basaltos. Los gabros son rocas plutónicas, textura granuda y claras. Sus minerales esenciales son las plagioclasas y augitas. Los basaltos forman un 80% de las lavas volcánicas. Son oscuros, pesados y pueden presentar fenocristales. Sus minerales esenciales son las plagioclasas y los piroxenos.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, pág. 8)

#### ***1.2.1.6. La vulcanología y los vulcanólogos***

“La vulcanología es el estudio de los volcanes. Un vulcanólogo es aquel que se encarga de estudiar los volcanes y sus fenómenos. Los vulcanólogos estudian los volcanes, especialmente los que están activos. Aquí observan la frecuencia e intensidad de las erupciones, analizan los materiales expulsados por el volcán y recogen cenizas, muestras de magma y otros materiales procedentes de interior del volcán. Una de las principales finalidades de este estudio es la predicción de futuras erupciones, aunque hasta ahora no se ha logrado predecir las concretamente.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, pág. 9)

#### ***1.2.1.7. Observación de los volcanes***

“Cada volcán tiene su propio carácter. La función principal de los vulcanólogos es vigilar a los volcanes. Donde se puede hacer esto mejor es en los observatorios vulcanológicos. Aquí se acumulan todos los datos tomados por los instrumentos de medición colocados alrededor del volcán. El primer observatorio de este tipo fue

construido junto al Vesubio en 1841. Hoy en día hay más de 70 de estas estaciones de control distribuidas por todo el mundo.

Si un científico examina un volcán, primero debe ocuparse con su historia. ¿Cuándo fue la última vez que el volcán entró en erupción? ¿Estas erupciones se producen en series de tiempo regulares? ¿Qué intensidad tienen sus erupciones? Para poder evaluar la probabilidad de futuras erupciones es imprescindible hacer un inventario. Tras la reconstrucción de antiguas erupciones se pueden elaborar modelos y simulaciones teóricas, de cómo podría ser una explosión futura del volcán. Pero para una supervisión exacta y para una predicción de erupción, deben de ser instalados numerosos instrumentos que puedan registrar cambios geofísicos y geoquímicos.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, pág. 11)

#### ***1.2.1.8. Terremotos y la deformación de suelos***

“En el 95% de los casos, un terremoto es antecedente a la explosión volcánica. Por lo tanto, el instrumento más importante para prevenir las erupciones es el sismógrafo. Este aparato es capaz de detectar el más mínimo movimiento vertical u horizontal del suelo. Un volcán activo hay toda una red de estos instrumentos. Gracias a la ayuda de los sismógrafos los vulcanólogos pueden determinar el hipocentro y su camino a la superficie terrestre. A partir de esto se puede determinar cuándo será la siguiente erupción volcánica y qué intensidad tendrá.

Además de los terremotos, es frecuente que la deformación de suelos junto al volcán, preceda a la explosión. Estas deformaciones indican la subida de la actividad magmática en el interior del volcán. Se incrementa la presión y el volcán se abulta. Como solo raras veces estos cambios son detectados a simple vista, son necesarios instrumentos que sean capaces de registrar el más mínimo cambio con máxima precisión. Un clinómetro mide los cambios del ángulo de inclinación en la ladera de la montaña. Cada volcán tiene grietas más pequeñas o más grandes. Cuando una montaña comienza a deformarse, el tamaño de las grietas suele cambiar. El posible contraerse o espaciarse de las grietas, se mide mediante un extensómetro o fisurómetro. Otras formas para averiguar la deformación del suelo

son los aparatos automáticos de medición de distancia. Ambos métodos pueden determinar con máxima precisión los cambios del movimiento del suelo.

Los vulcanólogos estudian sobre todo las vibraciones y las deformaciones. Pero hay otros factores que nos dicen algo sobre la actividad de un volcán: Gravitación, magnetismo, acústica, geoquímica.” (Lippert, Sevillano Reimers, & Millán Flaile, 2011, págs. 4-12)

## ***1.2.2. Sismología***

### ***1.2.2.1. La investigación sismológica***

“El conocimiento, teorías y modelos sobre la estructura y los procesos en el interior de la Tierra se han obtenido a partir de la observación de fenómenos que los mismos procesos generan.

Uno de los procesos, el relacionado con la generación y propagación de ondas sísmicas ha sido objeto de amplia investigación a nivel global. La investigación en sismología se ha dividido fundamentalmente en dos categorías.

- El estudio de la propagación de las ondas y la estructura de la Tierra asociada: identificación de las diferentes capas (corteza, manto y núcleo) y su heterogeneidad, las diferencias entre continente y océano, las zonas de subducción, las propiedades de los materiales (inelásticas y anisotropías), entre otros.
- El estudio de la fuente y sus fenómenos asociados: tipificación y localización de fuentes, energía liberada, geometría, área y desplazamiento de las fallas, estudios de predicción, entre otros.

Gran parte de la observación sismológica se hace de manera instrumental. A partir de registros sísmicos instrumentales se obtienen resultados cuantitativos con base en las siguientes relaciones, fenómenos internos como el fallamiento,

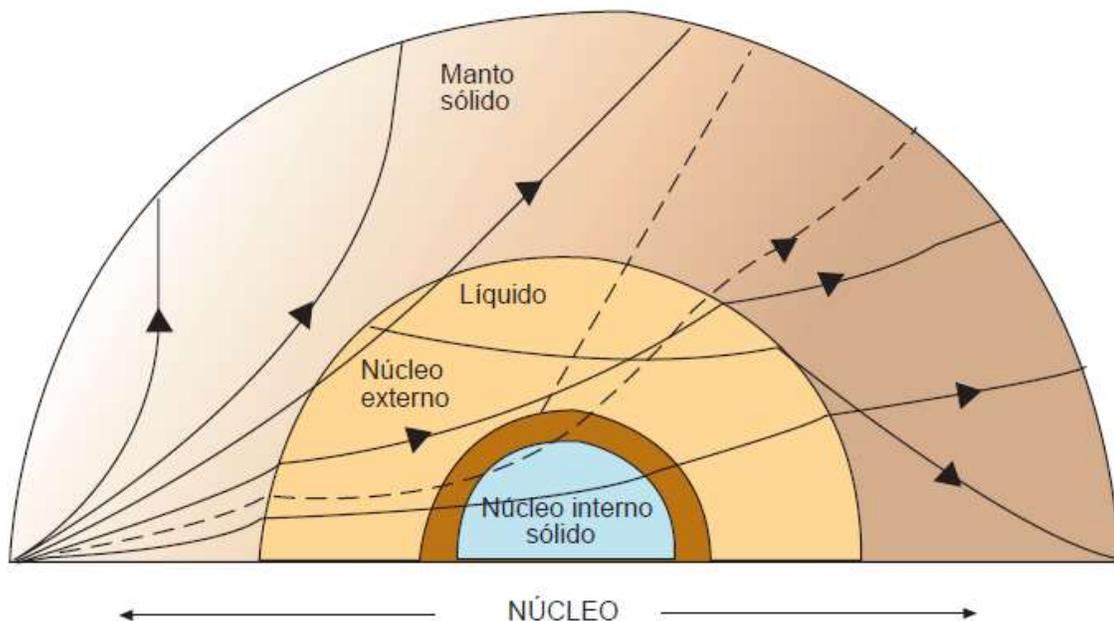
movimiento del magma, explosión minera, circulación hidráulica, y fenómenos externos como el viento, la presión atmosférica, las ondas y mareas oceánicas y el ruido cultural involucran movimientos rápidos que producen ondas sísmicas detectables.” (Vera Lizcano, 2003, págs. 6-7)

### 1.2.2.2. Onda Sísmica

“Una onda sísmica puede compararse a cuando se arroja una piedra a un pozo de agua, en esta acción se observan ondas que divergen del punto donde cayó la piedra, pero éstas también se transmiten en profundidad divergiendo desde el mismo punto.

Un fenómeno comparable a éste es el que ocurre en la Tierra al producirse un sismo, es decir, desde el foco o lugar de ruptura del equilibrio elástico se transmiten ondas en todas direcciones en el interior y superficie de la Tierra, tal como se ve en la siguiente figura.

Figura 1.2 - Propagación de ondas sísmicas en el interior de la Tierra



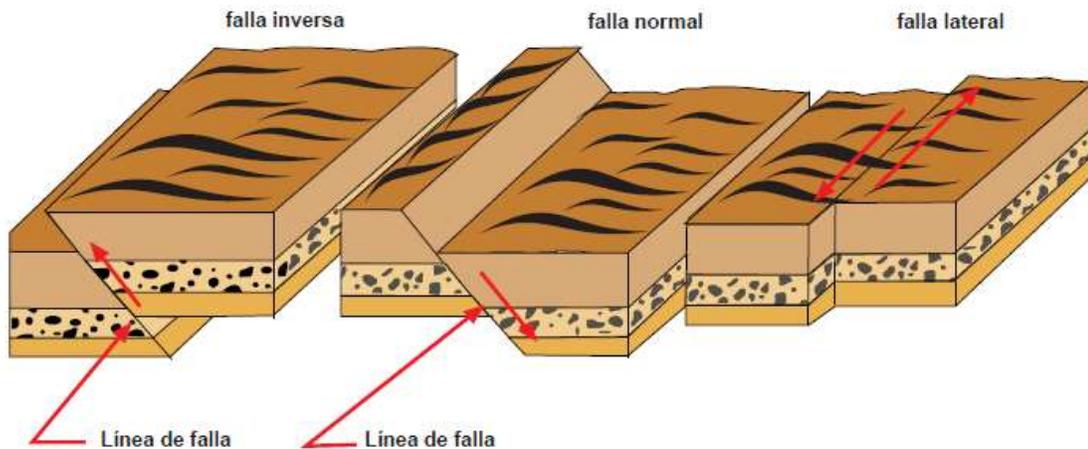
Fuente: Lorca Mella, E., & Recabarren Herrera, M. (1994). *Terremotos y Tsunamis o Maremotos*, pág. 14.

Para la mayoría de los sismos superficiales (0 a 70 km de profundidad), el mecanismo de generación de ondas elásticas es una fractura o ruptura del material en la región. En otras palabras, los esfuerzos superan el límite de ruptura del material en esa región, por lo tanto, se fractura, produciéndose lo que se denomina comúnmente como "falla" y esta "falla" es la que genera las ondas sísmicas.

Una "falla" se puede definir como el movimiento relativo entre bloques de la corteza, y esquemáticamente se pueden observar varios tipos en la Figura 1.3.

Una vez que se ha sobrepasado el punto de ruptura del material en una región (es decir, se ha producido una falla), se generan fundamentalmente tres tipos de ondas sísmicas." (Lorca Mella & Recabarren Herrera, 1994, págs. 13-15)

*Figura 1.3 - Ejemplos de falla inversa, normal y lateral.*



*Fuente: Lorca Mella, E., & Recabarren Herrera, M. (1994). Terremotos y Tsunamis o Maremotos, pág. 15.*

### **1.2.2.3. Onda P**

“Corresponde a la onda primaria o longitudinal. Se caracteriza porque el movimiento de las partículas en el medio tiene la misma dirección que la propagación de la onda. La onda P, como las ondas sonoras, son capaces de transmitirse en rocas y, además, por medios líquidos. Debido a su naturaleza,

parecida a la de las ondas sonoras, es posible escuchar parte de ellas cuando la frecuencia que poseen está contenida en el rango audible (superior a aproximadamente 15 ciclos por segundo).” (Lorca Mella & Recabarren Herrera, 1994, pág. 15)

#### **1.2.2.4. Onda S**

“También llamada onda secundaria. Deriva su nombre de su menor velocidad de propagación en referencia a la onda P. El tipo de movimiento de las partículas es exactamente, perpendicular a la dirección de propagación de la onda (comúnmente también se denomina onda de cizalle). Debido a su naturaleza de propagación, estas ondas no son capaces de transmitirse en medios fluidos, tales como medios líquidos o gaseosos, de tal modo que se encontrarán sólo en medios sólidos.

La velocidad de las ondas P y S depende de las propiedades de la roca y suelo por la que ellas atraviesan. Velocidades típicas para la onda P en una roca llamada granito y en el agua son 5,5 km/s y 1,5 km/s, respectivamente; mientras que la velocidad de la onda S en los mismos medios sería del orden de 3,0 km/s y 0 km/s, este último valor debido a que los medios líquidos no tienen rigidez.” (Lorca Mella & Recabarren Herrera, 1994, pág. 15)

#### **1.2.2.5. Ondas superficiales**

“Un tercer tipo de onda es denominada onda superficial, porque su propagación y movimiento está restringido a las cercanías de, la superficie de, la Tierra. Tal como las olas que atraviesan un lago, la mayor parte del movimiento de la onda está situado en la superficie.

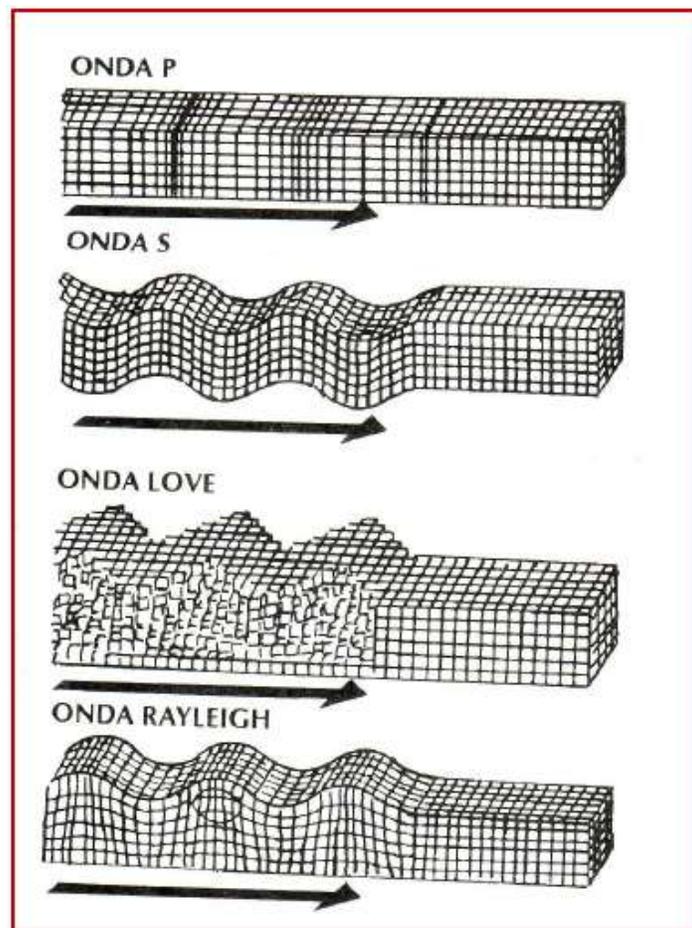
Las ondas superficiales pueden ser divididas en ondas *Love* y ondas *Rayleigh*.

El movimiento de la onda Love es esencialmente el mismo que, la onda S, que no tiene componente vertical, ella mueve el suelo de un lado a otro en un plano horizontal paralelo a la superficie de la Tierra.

El movimiento en una onda Rayleigh está confinado en el plano vertical que contiene la dirección de la propagación de las ondas. La velocidad de las ondas superficiales es menor que la velocidad de las ondas P y S y las ondas Love, generalmente, poseen mayor rapidez que las ondas Rayleigh.

Las figuras que se presentan a continuación, grafican la propagación de estos cuatro tipos distintos de ondas sísmicas.” (Lorca Mella & Recabarren Herrera, pág. 16)

*Figura 1.4 - Movimiento y propagación de ondas sísmicas.*



*Fuente: Lorca Mella, E., & Recabarren Herrera, M. (1994). Terremotos y Tsunamis o Maremotos, pág. 16.*

### **1.2.2.6. Tremor volcánico**

“El tremor volcánico es una señal sísmica caracterizada por mantener la amplitud constante durante un largo período de tiempo que puede oscilar entre los varios minutos y las horas, con contenido espectral centrado en bandas de frecuencia relativamente estrechas.

Atendiendo al contenido espectral del tremor éste se ha dividido muchas veces en varios subgrupos:

- **TH:** Tremor volcánico con contenido en altas frecuencias (> 6 Hz). A veces aparecen después de explosiones y pueden durar horas. Un tipo de señal parecida a ésta es la que aparece asociada a los géiseres. Esta señal posee un contenido espectral con frecuencias incluso superiores a 25 Hz.
- **TI:** Tremor volcánico a frecuencias intermedias (1-6 Hz). Pueden ser vibraciones de diversa duración, entre minutos a varias horas. En algunos volcanes el contenido espectral de este tipo de tremor está relacionado con el tipo de actividad. Así, en fases tranquilas el tremor puede estar centrado a bajas frecuencias (1 Hz) y en fases preeruptivas la frecuencia variar hasta el orden de los 3 Hz. Es evidente que estos cambios están relacionados con la fuente.
- **TL:** Tremor a bajas frecuencias (alrededor de 0.5 Hz o menos) que requiere sensores de banda ancha para poder ser registrados.” (Ibáñez & Carmona, 1997, pág. 6)

### **1.2.2.7. Explosiones**

“Quizás las señales provenientes de explosiones en los edificios volcánicos corresponden, junto al tremor, a las señales sísmicas más características cuando se encuentra en marcha un proceso eruptivo. En este caso, las explosiones aparecen superpuestas a una señal de tremor de fondo, y son identificables por el

aumento brusco de amplitud de la señal y un incremento en las frecuencias de la misma. Por regla general, esta señal tiene al menos dos llegadas diferentes y claras.

La primera asociada con la propagación en forma de ondas internas o superficiales de la explosión. La segunda es la llegada de lo que se conoce como ondas de aire, ondas de choque y ondas sonoras, con una velocidad de propagación clara de 340 m/s. Esta velocidad de propagación tan lenta y clara es la forma más fácil de poder identificar este tipo de eventos sobre los sismogramas cuando no ha sido posible distinguirlos en el momento de su ocurrencia.” (Ibáñez & Carmona, 1997, pág. 9)

#### **1.2.2.8. Sismos y fallas**

“Un sismo es un estremecimiento o sacudida de la Tierra. ¿Qué causa un sismo? Explosiones gigantescas pueden sacudir la Tierra, o el magma que asciende por el interior de un volcán pueden causar un sismo. Sin embargo, la mayoría de los sismos ocurren porque las rocas se mueven a lo largo de una falla.

Imaginemos lo que sucede al doblar una regla plástica. Esta tiene cierto grado de flexibilidad, pero si la doblas demasiado, la regla se quiebra y ambos trozos vuelven a una posición recta. Las rocas en la corteza de la Tierra que están bajo presión también se doblan, se quiebran y vuelven a su posición original. Una falla es una ruptura en las rocas a lo largo de la cual las rocas, se han movido. Cuando se produce el quiebre, se libera energía en forma de ondas sísmicas. Esta energía hace, que la Tierra se estremezca, y nosotros sentimos un sismo.

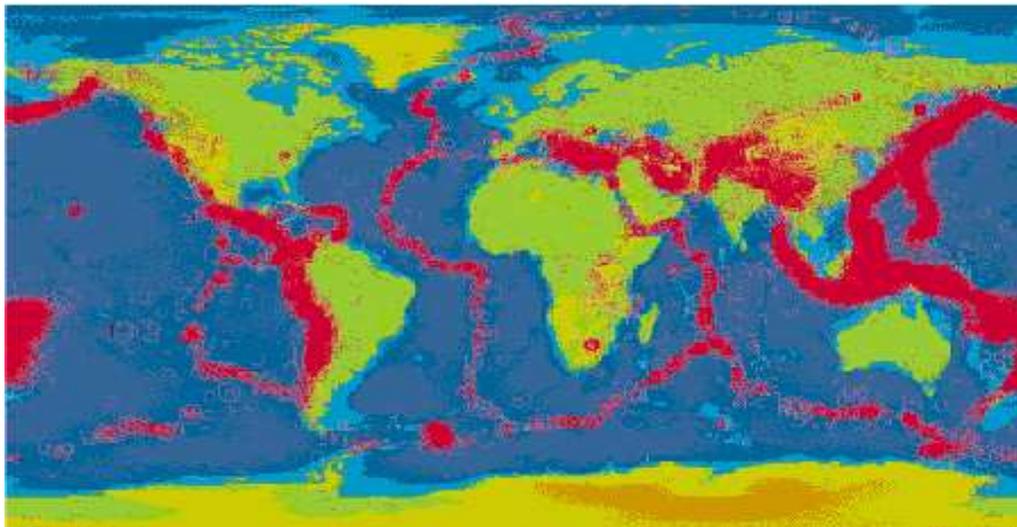
Con el advenimiento de sismógrafos de suficiente sensibilidad, distribuidos alrededor de todo el mundo, es relativamente fácil captar las perturbaciones sísmicas, aun cuando éstas no sean sensibles al hombre. Una vez que las ondas sísmicas son detectadas y registradas en varias estaciones sismológicas, es posible determinar su lugar de, origen y el momento en que se produjo. Actualmente, existen varias instituciones que se dedican a determinar los parámetros de los

sismos a nivel mundial, con lo cual se puede establecer de manera adecuada las zonas sísmicamente más activas o aquellas de baja sismicidad.

Fuera de estas zonas, gran parte del piso oceánico es asísmico. Las mesodorsales más importantes son: la Central del Atlántico, la del Océano Índico que se divide hacia el Sur en dos ramas y la cordillera del Pacífico Oriental, que nace en el Golfo de California extendiéndose hacia el Sur y, posteriormente, se divide en dos ramas a la altura de Isla de Pascua (Chile), una que prosigue hacia el Sur-Oeste y la otra que llega hasta la Península de Taitao, en Chile continental.” (Lorca Mella & Recabarren Herrera, 1994, págs. 13-50)

El mapa siguiente muestra la distribución mundial de los eventos sísmicos.

*Figura 1.5 - Distribución de sismicidad.*



Fuente: Lorca Mella, E., & Recabarren Herrera, M. (1994). *Terremotos y Tsunamis o Maremotos*, pág. 49.

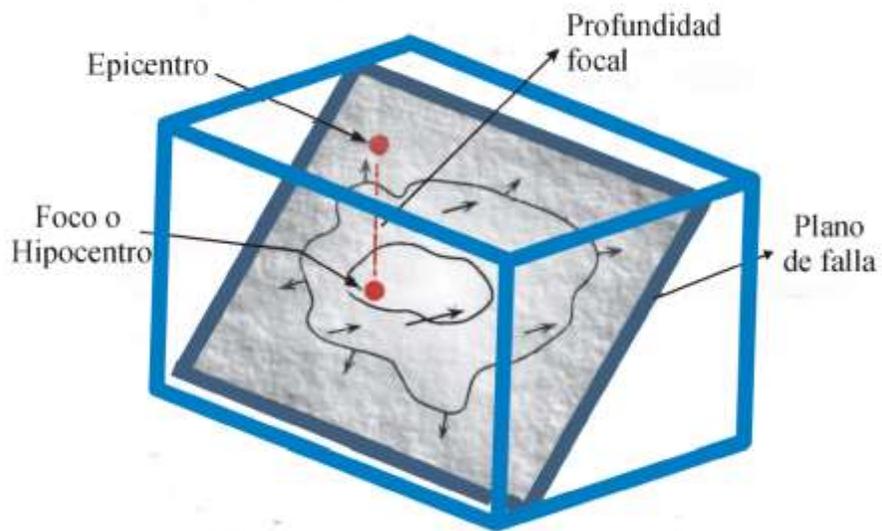
#### **1.2.2.9. Foco sísmico**

“Foco sísmico es el lugar en tiempo y espacio donde se produce la concentración de energía y a partir del cual esta se propaga en forma de ondas sísmicas. Con la creación del sismómetro y la instalación de las primeras redes sismológicas, empezó la determinación instrumental de los parámetros del foco sísmico. Estos

pueden ser determinados a partir de los registros en una o varias estaciones de las ondas de cuerpo producidas por el sismo.

En la siguiente figura, se muestra una representación tridimensional de la ubicación del foco sísmico.

*Figura 1.6 - Representación de Foco Sísmico.*



*Fuente: Vera Lizcano, B. E. (2003). Evaluación de Errores de Datos Sismológicos: Conceptos, Instrumentación y Observación Sismológica, pág. 11.*

Los parámetros que determinan el foco puntual de un sismo son:

- Las coordenadas geográficas (latitud y longitud) relacionadas a un punto en la superficie (epicentro).
- La profundidad, es decir la distancia hacia el interior de la tierra a partir del epicentro. La profundidad más el epicentro, determinan el hipocentro.
- El tiempo de origen, es decir el momento a partir del cual se inició la liberación de energía en forma de ondas sísmicas.

La determinación de los parámetros de un sismo es importante desde el punto de vista teórico, puesto que a partir de valores hipo centrales y tiempos de origen, se determinan tiempos de recorrido de las ondas sísmicas, y a partir del estudio de la propagación de las mismas se pueden conocer las propiedades físicas de la Tierra; y desde el punto de vista práctico, porque a partir de esta información se determinan zonas de riesgo, códigos de construcción, entre otros. Estas estimaciones presentan dificultades técnicas tales como la fiabilidad de tiempos absolutos en los sismogramas; dificultades observacionales, tales como la identificación de tiempos de arribo; y dificultades teóricas, tales como la estimación del error usando teoría no lineal, modelos teóricos de propagación, entre otros” (Vera Lizcano, 2003, págs. 9-13)

#### ***1.2.2.10. Clasificación de los sismos***

“Los sismos se pueden caracterizar como un proceso de ruptura y deformación elástica del material de la litósfera, y bajo esas condiciones todos los sismos son iguales, sin embargo, se ha visto que, dependiendo del tipo de falla o mecanismo causal, así como del medio de propagación, los sismos pueden tener consecuencias diferentes en la superficie.

Es por eso que se pueden clasificar a los sismos según su zona de generación, y su profundidad. De acuerdo a lo anterior tenemos la siguiente clasificación:

- A. Sismos de subducción someros:** Aquellos que se generan en las fronteras de este tipo y que ocurren a profundidades que no exceden los 40 km.
- B. Sismos de subducción profundos:** Aquellos que ocurren debido a la interacción de subducción y en la zona de fricción (interplaca), pero a profundidades mayores a los 40 km.
- C. Sismos intraplaca de profundidad intermedia:** Sismos que se presentan en la placa subducida, pero no ocasionados por la fricción entre las placas sino por fractura de la placa que ha penetrado, sus profundidades son mayores a los 80 km.

**D. Sismos de zonas de acreción:** Sismos que se presentan en este tipo de fronteras, por lo general con profundidades que no exceden los 20 km.

**E. Sismos de fallas de transurrencia:** Los que se presentan en este tipo de frontera, cuyas profundidades nos exceden los 30 km por lo común.

**F. Sismos corticales intracontinentales:** Sismos que se presentan en fallas no directamente relacionadas con los procesos de interacción entre las placas, sino al interior de una placa. Sus profundidades no exceden el grosor de la placa.

Adicionalmente, se pueden considerar los sismos volcánicos, los cuales tienen una clasificación propia. Entre estos, los llamados sismos volcano-tectónicos serían equivalentes a los sismos corticales. Otros sismos de este tipo pueden deberse al transporte de fluidos (magma o agua) en cavidades y fracturas, ocasionando la emisión de bajas frecuencias por lo que se llaman temblores.

Se ha visto que los mayores sismos son, por lo general, del tipo A. Sin embargo, los sismos de tipo B, C, E o F pueden llegar a tener consecuencias graves.” (Dávila Madrid, 2011, pág. 7)

#### **1.2.2.11. RSAM - Medida de amplitud sísmica en tiempo real**

“Durante una crisis volcánica, la sismicidad alcanza niveles a los cuales es difícil distinguir eventos sísmicos individualmente. Registros sísmicos analógicos brindan información relevante, pero no es posible hacer un análisis en un corto tiempo sin perturbar la continuidad de los registros; es difícil conseguir un análisis que tengan alta relevancia con datos que se generan continuamente en un tiempo corto determinado.

Existen varios sistemas de detección de terremotos en tiempo real, pero estos no proporcionan información cuantitativa en periodos de alta sismicidad en la cual se pueden observar en erupciones volcánicas donde la generación de información es crucial para describir los eventos volcánicos y posibles predicciones. Es en ese entonces cuando surge la necesidad de procesar información sísmica cuantitativa.

La problemática surge sobre vigilar continuamente la amplitud de las señales sísmicas durante condiciones de alta actividad volcánica, cuando los eventos individuales son difíciles de reconocer.

Tomas L. Murray y Elliot T. Endo consiguieron desarrollar un sistema de medida de Amplitud Sísmica en Tiempo Real (RSAM, por sus siglas en inglés Real Time Seismic Amplitude Measurement) en el cual usa un convertidor analógico-digital de ocho bits que es controlado por una computadora portátil para proporcionar información sobre amplitud absoluta promediada cada minuto para ocho estaciones sísmicas cerca de un volcán. El nivel absoluto para cada estación es digitalizado a una frecuencia de 60 muestras/segundo promediado y directamente transmitido a una computadora central para su posterior análisis.

El RSAM proporciona un historial continuo de actividad sísmica que es fácil para acceder, el cual ha sido una herramienta útil para la predicción de actividad eruptiva en volcanes.

#### ***1.2.2.12. Funcionamiento del RSAM***

El RSAM mide la amplitud máxima de hasta ocho señales sísmicas con una frecuencia de muestreo de aproximadamente sesenta muestras/segundo. Estos datos se utilizan para calcular dos medidas para cada señal que ingresa.

1. La amplitud promediada por un minuto para cada señal es calculada sumando las medidas hechas en el minuto y dividiendo por el número de medidas. Se encontró que almacenando y analizando datos de un minuto para periodos de mayores a unos pocos días puede ser muy pesado considerando el volumen de datos generados. Si se calcula el promedio de cada diez minutos usando los promedios de un minuto los datos son analizados y procesados con mayor facilidad. Aparentemente, no hay ninguna pérdida de información usando tales promedios de diez minutos en comparación a un minuto.

2. La cantidad de eventos para cada señal entrante ocurriendo en un periodo de diez minutos, es calculado comparando las amplitudes promedias sucesivas cada dos segundos. El contador de eventos es incrementado si se cumplen las siguientes condiciones:
  - Si  $A_{(n)}$  es mayor al THRESHOLD
  - Si  $A_{(n)}$  es mayor a RATIO por  $A_{(n-2)}$

Donde  $A_{(n)}$  son los promedios sucesivos cada dos segundos; THRESHOLD se define como la amplitud mínima para un evento y RATIO define que tanto mayor que el nivel de fondo debe ser la amplitud para ser considerada como un evento potencial. El valor estándar para THRESHOLD de cinco unidades RSAM y para RATIO de dos.

Se tiene en cuenta que el RSAM no tiene énfasis para discriminar eventos sísmicos como deslizamientos de tierra, caídas de rocas y otros factores que provoquen sismicidad baja considerada como ruido en factor a los eventos de sismicidad considerable; la medida se conoce como “Evento RSAM” y no como un evento sísmico.

La liberación de energía puede ser vigilada en tiempo casi real. Cuadrando la amplitud promedio se calcula un valor que es proporcional a la energía eléctrica generada por el movimiento del geófono. Además, se requieren estudios adicionales para determinar la relación exacta entre la energía eléctrica y la energía sísmica.

La sencillez del RSAM puede ayudar a comunicar los niveles de sismicidad a diferentes entidades que pueden desconocer la terminología usada por los sismólogos. Un gráfico de RSAM puede ilustrar la actividad relativa más efectivamente que gráficos que demuestran las profundidades y magnitudes de terremotos.

El propósito de RSAM no es reemplazar sistemas sísmicos; sino, complementar un sistema convencional, proporcionando información en tiempo real de los niveles de amplitudes de tremores, mientras que otros calculan localizaciones y magnitudes de terremotos. Durante épocas de tremores o actividad sísmica intensa, el RSAM puede mantener el ritmo de la actividad sísmica y puede ser el monitor principal de sismicidad. (Murray, 1992)

### ***1.2.3. Gases volcánicos***

“Gases volcánicos son los gases emanados por un volcán y constituyen la mayor parte del volumen de las erupciones volcánicas. Los campos fumarólicos emiten gases de origen magmático como vapor de agua, que es el componente mayoritario con aproximadamente el 95%, el otro 4% son gases como SO<sub>2</sub> (Dióxido de Azufre), H<sub>2</sub>S (Sulfuro de Hidrógeno), CO<sub>2</sub> (Dióxido de Carbono), y menos del 1% son gases inertes como Helio, Argón y otros. Los manantiales calientes se deben a un calentamiento de las aguas subterráneas (freáticas) por focos de calor atrapados en la tierra.

La presencia de ciertas especies químicas en el sistema puede indicarnos el inicio, la evolución y finalización de una crisis eruptiva; se sabe que su presencia está condicionada por los tipos de gases que se encuentran en las emanaciones de un volcán y algunas de ellas, por sus propiedades fisicoquímicas, son capaces de disolverse y permanecer en el agua por tiempos mayores, como es el caso de los gases CO<sub>2</sub> y SO<sub>2</sub>, que pueden producirnos iones carbonato y sulfato respectivamente.

A lo largo de los años, los volcanes activos han presentado emisiones gaseosas a la atmósfera, estas emisiones, resultan beneficiosas ya que de todos los gases emitidos el CO<sub>2</sub> y el vapor de agua, son gases de efecto invernadero (que ayudan a mantener la temperatura del planeta) sin embargo, algunos de ellos tienen efectos

no beneficiosos en la salud, ya que causan enfermedades debido a su exposición prolongada. Cabe mencionar que además de lo anterior expuesto, debemos recordar la capacidad oxidante de ellos (esto debido a la elevada atracción hacia los electrones) haciéndolos muy corrosivos y con la capacidad de dañar al medio ambiente y la salud de las personas y animales.

Durante el ascenso magmático (propio de una reactivación volcánica) se liberan muchos gases entre ellos el CO<sub>2</sub>, SO<sub>2</sub>, H<sub>2</sub>S, HCl (Ácido Clorhídrico) y HF (Ácido Fluorhídrico), los cuales se encuentran disueltos en el magma, y por una caída de la presión comienzan a liberarse, debido a esta propiedad podemos decir que un incremento en los valores de estos gases, estaría indicando una intrusión o ascenso de magma.

Por este motivo se vuelve importante cuantificar las cantidades de estos gases que pueden ser emitidos por los volcanes (creando una línea base) así como tomar en cuenta el tipo de reacciones que estos gases presentan en su ascenso y liberación, ya que en su camino pueden cambiar su estado de oxidación y quedar atrapados en los acuíferos del sistema interno del volcán.” (Sandoval, Montalvo, & Rosa Lúe, 2013, págs. 3-6)

A continuación, veremos la descripción y sus principales propiedades, de los gases volcánicos más comunes.

### ***1.2.3.1. Vapor de agua (H<sub>2</sub>O)***

“El vapor de agua es el gas más abundante que expulsa un cráter debido a la existencia de aguas subterráneas. Los volcanes emiten gases durante las erupciones e incluso cuando no estallan en erupción. Las grietas u orificios en el terreno permiten que los gases y el vapor de agua, presurizados en el interior del volcán, alcancen la superficie formando lo que se conoce como fumarolas.

Una fumarola, es una mezcla de gases y vapores que surgen por las grietas exteriores de un volcán (o sea en la superficie volcánica) a temperaturas altas.

También se desprenden de las coladas de lava. Su composición varía según la temperatura a que son emitidas, de tal manera que este va cambiando a lo largo del "ciclo de vida" de una fumarola. Se distinguen los siguientes grupos:

- ***Fumarolas secas (o anhidras):*** son las que emite la lava en estado de fusión, en las proximidades del cráter. Su temperatura es superior a 500 °C y están compuestas principalmente por cloruros de sodio, potasio y anhídrido sulfuroso y carbónico, careciendo por completo de vapor de agua. También contienen, aunque en pequeñas proporciones, otros cloruros (de potasio, hierro, cobre, etc.), algunos fluoruros y a veces hidrocarburos que producen llamaradas.
- ***Fumarolas ácidas (o clorhidrosulfurosas):*** no son tan calientes: se encuentran a temperaturas entre 300 °C y 400 °C. Esto porque emanan de la capa superficial de las coladas de lava. Contienen gran cantidad de vapor de agua, y proporciones menores de ácido clorhídrico y anhídrido sulfuroso.
- ***Fumarolas alcalinas (o amoniacaes):*** son relativamente más frías, alcanzando aproximadamente 100 °C. Constan sobre todo de vapor de agua con ácido sulfhídrico y cloruro amónico.
- ***Fumarolas frías (o sulfhídricas):*** sólo alcanzan unas cuantas decenas de grados, consistiendo esencialmente de vapor de agua con un pequeño porcentaje de anhídrido carbónico y sulfuroso.” (Universidad Católica de Valparaíso, 2016)

### ***1.2.3.2. Dióxido de azufre (SO<sub>2</sub>)***

“El dióxido de azufre es un gas incoloro y no inflamable, de olor fuerte e irritante. Su vida media en la atmósfera es corta de unos 2 a 4 días, y casi la mitad de las emisiones vuelven a depositarse en la superficie, mientras que el resto se

transforma en iones de sulfato. Se trata de una sustancia reductora, que con el tiempo y en contacto con el aire y la humedad, se convierte en trióxido de azufre. Es soluble en agua, formando una disolución ácida, y aun siendo inestable en estas condiciones, es capaz de formar sales como los sulfitos y bisulfitos.

El dióxido de azufre es uno de los gases más comúnmente liberados durante erupciones volcánicas (después del vapor de agua y el dióxido de carbono), y es preocupante a escala global, debido a su potencial influencia en el clima. El SO<sub>2</sub> es peligroso para los humanos en su forma gaseosa y también porque se oxida formando aerosol.

El SO<sub>2</sub> es un gas incoloro con un olor irritante característico, este olor es perceptible a diferentes niveles, dependiendo de la sensibilidad individual, pero generalmente se percibe entre 0.3-1.4 ppm y es fácilmente notable a 3 ppm. SO<sub>2</sub> no es inflamable, no es explosivo y es relativamente estable. Su densidad es más del doble que la del aire ambiental (2.62 g L a 25°C y 1 atm), y es altamente soluble en agua. En contacto con membranas húmedas SO<sub>2</sub> forma ácido sulfúrico (H<sub>2</sub>SO<sub>4</sub>), que es responsable de fuertes irritaciones en los ojos, membranas mucosas y piel.

Típicamente, la concentración de SO<sub>2</sub> en fumarolas volcánicas diluidas es < 10 ppm, tan poco como 10 km con viento a favor desde su origen, comparado con el antecedente troposférico de 0.00001-0.07 ppm. Suponiendo que el gas tuviese media vida de 6 a 24 horas, entonces solo el 5% del gas emitido está presente en la atmósfera baja después de 1 a 4 días.” (International Volcanic Health Hazard Network, 2016, págs. 4-15)

### ***1.2.3.3. Sulfuro de hidrógeno (H<sub>2</sub>S)***

“El hidrógeno de sulfuro (H<sub>2</sub>S), también conocido como sulfuro de hidrógeno o ácido sulfhídrico; es un gas incoloro con un olor distintivo a huevo podrido. La percepción del olor del H<sub>2</sub>S varía dentro de la población humana. Este gas es inflamable en el aire cuando se encuentra en concentraciones entre 4–46% y enciende con una llama color azul pálido. El ácido sulfhídrico se encuentra

naturalmente en petróleo (procesado), gas natural, gases volcánicos y manantiales de aguas termales. También puede existir en aguas pantanosas, lagunas o aguas estancadas, desagües, estanques de harina o de aceite de pescado, barcos pesqueros y alcantarillados.” (International Volcanic Health Hazard Network, 2016, pág. 16)

#### ***1.2.3.4. Dióxido de carbono (CO<sub>2</sub>)***

“El dióxido de carbono es un gas incoloro e inodoro. Es no inflamable y químicamente no reactivo. La abundancia de este gas en el aire es de aproximadamente 355 ppmV y en el agua pura saturada en contacto con el aire es de 3900 ppmV. El CO<sub>2</sub> es, junto con el vapor de agua, la principal especie volátil de los sistemas volcánicos-hidrotermales. Su reducida solubilidad en fundidos silicatados hace que sea de los primeros volátiles en escapar del magma a presiones moderadas y su alta solubilidad en el agua (759 ppmV de agua en condiciones estándar) permite que una fracción considerable del CO<sub>2</sub> emitido quede retenido en el acuífero. Debido a su elevada reactividad, la concentración del CO<sub>2</sub> disuelto en las aguas subterráneas está controlada por la temperatura y por el pH de las mismas.

El principal proceso responsable de la presencia de CO<sub>2</sub> disuelto en los acuíferos volcánicos es el aporte de origen magmático-mantélico, aunque la descomposición de la materia orgánica, el metamorfismo térmico de rocas carbonatadas, la oxidación de CO y CH<sub>4</sub>, y la aportación atmosférica, también pueden ser fuentes importantes de CO<sub>2</sub>.” (Sandoval, Montalvo, & Rosa Lúe, 2013, pág. 8)

#### ***1.2.3.5. Cloruro de hidrógeno (HCl)***

“El hidrocioruro (HCl), también conocido como ácido clorhídrico, ácido muriático, ácido marino, ácido de sal o todavía ocasionalmente llamado, ácido hidroclórico es un gas incoloro con un irritante olor acre detectable. Es muy corrosivo y ácido. Se emplea comúnmente como reactivo químico y se trata de un ácido fuerte que se disocia completamente en disolución acuosa. A temperatura ambiente, el cloruro de

hidrógeno es un gas ligeramente amarillo, corrosivo, no inflamable, más pesado que el aire, de olor fuertemente irritante. Cuando se expone al aire, el cloruro de hidrógeno forma vapores corrosivos densos de color blanco.” (International Volcanic Health Hazard Network, 2016, pág. 30)

#### **1.2.3.6. Fluoruro de hidrógeno (HF)**

“El hidrógeno de fluoruro (HF) es un gas incoloro con un olor fuertemente irritante. Es soluble al agua en todas las proporciones y no es inflamable. Tiene un sabor ácido y reacciona a la humedad del aire formando una neblina. Las concentraciones típicas de HF en fumarolas volcánicas diluidas son de < 1 ppm, y en nivel de la base troposférica es muy bajo.” (International Volcanic Health Hazard Network, 2016, pág. 33)

#### **1.2.3.7. Metano (CH<sub>4</sub>)**

“El CH<sub>4</sub> es un gas incoloro, inodoro, inflamable, no tóxico, estable a temperaturas superiores a los 700 °C y su concentración en la atmósfera es de 1,7 ppmV. El CH<sub>4</sub> puede tener origen orgánico, por reducción del CO<sub>2</sub>, o magmático, siendo esta última la menor contribución.

En el ambiente de la corteza terrestre es común la formación de CH<sub>4</sub> a partir del CO<sub>2</sub> producto de la actividad biológica.” (Sandoval, Montalvo, & Rosa Lúe, 2013, pág. 9)

#### **1.2.3.8. Hidrógeno (H<sub>2</sub>)**

El H<sub>2</sub> está presente en la mayoría de las emisiones volcánicas y ha sido usado desde hace décadas en la vigilancia volcánica. Debido a sus características físico-químicas, el H<sub>2</sub> escapa rápidamente hacia el espacio exterior, encontrándose en concentraciones bajas tanto en la atmosfera como en el agua pura saturada en aire (aproximadamente de 0,5 ppmV). El H<sub>2</sub> puede formarse en el interior de la tierra por reacción de descomposición del agua con óxidos de hierro que están presentes en el magma y/o en las rocas. (Sandoval, Montalvo, & Rosa Lúe, 2013, pág. 9)

### **1.2.3.9. Helio (He)**

Los gases nobles como el helio son muy útiles como trazadores de la contribución endógena ya que los fluidos profundos suelen estar relativamente enriquecidos en el a la vez que suele estar en una concentración reducida en la atmósfera, alrededor de 5,2 ppmV. En este sentido, el helio es un gas que posee unas propiedades muy interesantes, ya que presenta una alta movilidad, es inerte químicamente, físicamente estable, no absorbible y no es de origen biogénico, aunque presenta una importante limitación dada su reducida solubilidad en el agua en condiciones normales (aproximadamente 10 ppmV). (Sandoval, Montalvo, & Rosa Lúe, 2013, pág. 10)

### **1.2.3.10. Nitrógeno (N<sub>2</sub>)**

El nitrógeno puede ser de origen atmosférico meteórico, pero también puede ser de origen magmático, en el caso del CH<sub>4</sub> puede derivarse de rocas sedimentarias cercanas a la superficie conteniendo materia orgánica, el Ar y He pueden formarse por lixiviación de la roca o por entrada de fluido magmático y para el resto de gases inertes son casi en su totalidad de origen meteórico. (Sandoval, Montalvo, & Rosa Lúe, 2013, pág. 10)

### **1.2.3.11. Importancia de los flujos de SO<sub>2</sub> en el monitoreo volcánico**

“Durante las erupciones, se generan productos sólidos, líquidos y gaseosos. La principal fuerza impulsora detrás de las erupciones es la emanación de gas del magma durante la descompresión, que impulsa la subida a través de la corteza terrestre, como el SO<sub>2</sub>; que es uno de los compuestos más abundantes entre los gases volcánicos. Puesto que este gas es muy soluble en agua y es termodinámicamente inestable a baja temperatura, la presencia de SO<sub>2</sub> en penachos volcánicos es característica de una temperatura de alta emisión. La principal conclusión es que la altura a la que se inyectan los gases magmáticos no es arbitraria sino fuertemente dependientes en el régimen de la erupción. La altura a la que está presente en la atmósfera el SO<sub>2</sub> también tiene una influencia grande

en las técnicas para detectar las cantidades presentes, por lo tanto, la altura se convierte en una cuestión central para la caracterización de plumas.” (Atmos. Chem. Phys, 2013, pág. 1)

#### ***1.2.3.12. Mediciones de flujos de SO<sub>2</sub> volcánicos***

“Las mediciones remotas de gases volátiles magmáticos se han centrado mucho en SO<sub>2</sub> porque es sin duda el más fácilmente medible por espectroscopia de absorción, debido a su concentración de bajo fondo en el ambiente, estructuras distintivas en su espectro de absorción en el ultravioleta (UV) y longitudes de onda del infrarrojo térmico (TIR). Mediciones de flujos de SO<sub>2</sub> volcánicos son ampliamente llevadas a cabo desde la superficie a los mediados de los setenta con el instrumento espectrómetro de correlación (COSPEC). Más recientemente, consolidado las mediciones de la emisión total flujos de SO<sub>2</sub> han hecho posibles que una red de volcanes activos de desgasificación (red de observaciones volcánicas y el cambio atmosférico – NOVAC), con el advenimiento de bajo costo y alta calidad análisis instrumentos espectroscopia de absorción óptica diferencial miniatura (Mini-DOAS).

Para volcanes o erupciones volcánicas explosivas no monitorizadas, en el espacio de las mediciones de SO<sub>2</sub> son más adecuadas. Desde 1978, el espectrómetro de mapeo Total de ozono (TOMS) y los instrumentos de seguimiento han estado midiendo SO<sub>2</sub> en el rango espectral UV, aunque con un límite de detección bastante alto, ayudó a establecer la entrada volcánica a largo plazo de SO<sub>2</sub> en la atmósfera alta. En el infrarrojo, que suena en el espacio de SO<sub>2</sub> con los datos que se remonta a 1978. Las mediciones de SO<sub>2</sub> de los sensores de nadir han sufrido una evolución considerable, debido a la mayor resolución espectral, cobertura y resolución espacial.” (Atmos. Chem. Phys, 2013, pág. 2)

#### ***1.2.3.13. Emisión de gases en el volcán de Santa Ana***

“El volcán de Santa Ana, también denominado Ilimatepec, es el volcán principal del Complejo Volcánico de Santa Ana, también considerado el más alto del país y

el de mayor volumen, con 2,382 metros sobre el nivel de mar, posee un cráter circular con un diámetro aproximado de 1.5 kilómetros. Su última erupción fue el 1 de octubre de 2005. Sus erupciones anteriores fueron en los años 1621, 1874, 1904, 1920 y 1937. En su falda sur y sureste se encuentra el volcán de Izalco, el Cerro Verde y una serie de conos adventicios, tales como los Conos de Escorias, El Conejo, El Astillero y San Marcelino (Cerro Chino).” (Sol, 2016)

La siguiente figura muestra la foto del cráter del volcán Ilimatepec durante una expedición del diario El Universitario.

*Figura 1.7 - Cráter del volcán Ilimatepec de Santa Ana*



*Fuente: Sol, M. (02 de febrero de 2016). El Salvador: Un volcán Activo. El Universitario*

“Los gases volcánicos son principalmente el  $H_2O$ ,  $CO_2$  y el  $SO_2$ , además de otros como el ácido sulfhídrico o el flúor. Los gases son liberados por el volcán a través de fumarolas o en forma difusa a través del suelo.

El volcán de Santa Ana tiene un campo de fumarolas en el sector occidental del interior del cráter que emite gases de forma permanente. Aunque mayoritariamente se trata de vapor de agua, a menudo emite SO<sub>2</sub> con volúmenes que fluctúan entre 130 y 390 toneladas/día. Además, en los últimos 25 años se han reportado tres períodos de degasificación intensa (1992, 2000 y 2004), donde los volúmenes de SO<sub>2</sub> alcanzaron las 600 toneladas/día.

Estas cantidades de gases volcánicos no son dañinas para la vida de las personas, aunque pueden provocar molestias para respirar. No es conveniente que las personas con problemas respiratorios, de asma o de corazón se expongan a las emanaciones de gases. Por otra parte, el contacto permanente con este ambiente puede provocar irritaciones en la piel y partes más sensibles del cuerpo. Diferentes personas de las comunidades más cercanas al cono volcánico, como el caserío San Blas, han manifestado haber sentido molestias, especialmente en periodos de degasificación más fuerte.

Además, la interacción de los gases volcánicos con el agua precipitada durante la estación húmeda, genera lluvia ácida que daña la vegetación y los cultivos, principalmente el bosque y el cafetal del sector sur y oeste del cono volcánico.” (Servicio Nacional de Estudios Territoriales - SNET, 2016)

## ***1.2.4. Herramientas estadísticas y matemáticas***

### ***1.2.4.1. Análisis de series temporales***

“La mejora continua en la instrumentación científica, en especial en los instrumentos portátiles o de campo, en los sistemas de adquisición de datos (conversión analógico-digital más rápida y eficiente) y en los sistemas de telecomunicación (telemetría) está permitiendo la observación y recogida de una ingente cantidad de información geofísica, bien sea desde la superficie terrestre o desde el espacio.

En muchos casos, dicha información se presenta con el aspecto de una serie de datos en los que la ordenación temporal es relevante. En este caso, hablamos de series temporales.

Ejemplos de series temporales de interés para la Sismología y la Volcanología son:

- El registro sísmico
- La deformación de la corteza terrestre en una zona sísmicamente activa o de un edificio volcánico activo
- Las mareas oceánicas y terrestres
- El nivel de agua en el interior de un pozo
- la variación del campo magnético interno terrestre
- La emisión de gases a través de los suelos
- La composición química de los gases fumarólicos
- Las anomalías de gravedad, entre otros.

Algunas de las contribuciones más evidentes en el análisis de series temporales son las siguientes:

- **Filtrado.** Se emplea para reducir o eliminar el ruido inherente al registro instrumental de series temporales o para separar las distintas componentes periódicas de las componentes irregulares. También puede emplearse para reducir el tamaño de los archivos digitales en los que se almacena la información.
- **Descomposición y reconstrucción.** La descomposición de una serie temporal en componentes deterministas y no deterministas puede facilitar su análisis e interpretación. Asimismo, es posible eliminar información redundante de los registros y así facilitar la transmisión de información a distancia. Para ello se puede emplear algunas de las técnicas del análisis estadístico multivariante clásico como son la extracción de factores

(componentes principales), análisis de conglomerados (clusters), acumulación de histogramas (método HiCUM), estudio de los momentos de orden superior, entre otros.

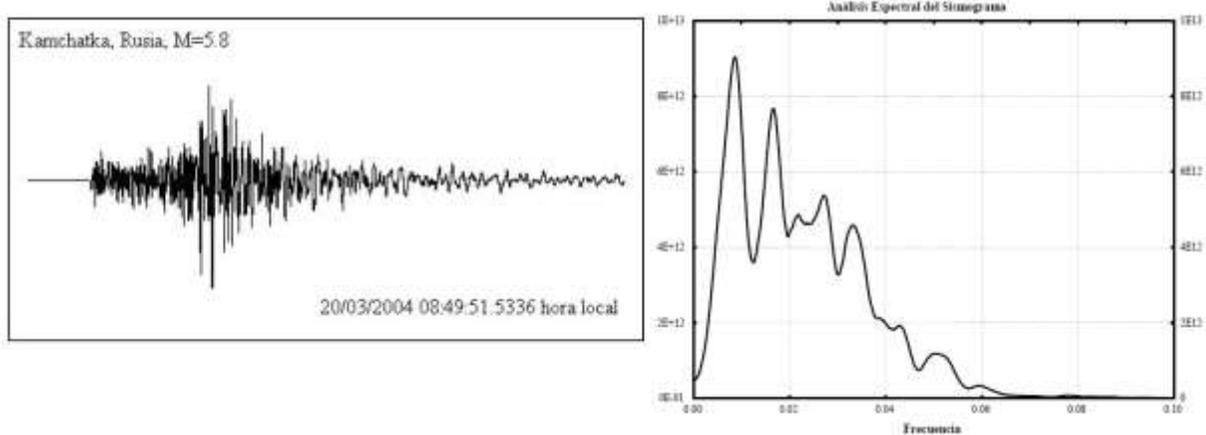
- **Modelado.** El establecimiento de modelos matemáticos, una vez validados con datos obtenidos empíricamente, permite a los científicos predecir el comportamiento del sistema en estudio.
- **Detección de anomalías y predicción.** El estudio de determinadas propiedades de una serie temporal permite establecer criterios de alarma, es decir, valores umbral por encima o por debajo de los cuales los valores observados pueden considerarse como anómalos o no normales. Entre ellas podemos citar la varianza (o covarianza), la dimensión de Hausdorff, la dimensión fractal, entre otros.

Una serie temporal puede ser analizada, al menos, en dos dominios distintos: en el dominio temporal, se muestra el registro sísmico en la escala de Richter, y en el dominio de frecuencias, se muestra el análisis espectral o análisis de Fourier obtenido utilizando una ventana de tiempo de promediado muy ancha, con objeto de suavizar el aspecto de la función de densidad espectral resultante.

La figura 1.8, muestra dos ejemplos de graficas que se realizan durante el análisis de una serie temporal y un espectral aplicado.

El análisis espectral de la señal sísmica revela la presencia de distintas componentes de baja frecuencia y la ausencia de componentes periódicas de frecuencias altas.” (Lorenzo Salazar, 2015, págs. 6-7)

Figura 1.8 - Dominio temporal, y Análisis espectral aplicado.



Fuente: Lorenzo Salazar, J. M. (2015). *Volcanes, Terremotos y Matemáticas*, pág. 7.

### 1.2.4.2. Transformada rápida de Fourier

“El análisis espectral tiene su origen en las investigaciones desarrolladas por Jean Baptiste Joseph Fourier (1768-1830) sobre las propiedades de la transferencia de calor en sólidos conductores. Sus trabajos a comienzos del siglo XIX (“La Teoría Analítica del Calor”) sentaron las bases para la disciplina matemática del Análisis de Señales. Fourier introdujo un nuevo concepto en la matemática de su época: cualquier función arbitraria, incluso aquellas que muestran discontinuidades, puede ser aproximada mediante una expresión analítica sencilla. Fourier estableció las ecuaciones en derivadas parciales que gobiernan la difusión del calor y encontró una solución empleando series infinitas de funciones trigonométricas de senos y cosenos (base ortogonal).

Sea una función continua,  $\theta(t)$ , de periodo  $T$ . Dicha función puede expresarse como serie de Fourier en los siguientes términos

$$\theta(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left\{ a_n \cos\left(\frac{2n\pi}{T}t\right) + b_n \operatorname{sen}\left(\frac{2n\pi}{T}t\right) \right\}$$

Donde los coeficientes  $a_n$  y  $b_n$  pueden ser calculados a partir de:

$$a_n = \frac{2}{T} \int_c^{c+T} \theta(t) \cos\left(\frac{2n\pi}{T} t\right) dt$$

$$b_n = \frac{2}{T} \int_c^{c+T} \theta(t) \operatorname{sen}\left(\frac{2n\pi}{T} t\right) dt$$

Los coeficientes de estas funciones base ortonormales representan la contribución de las componentes cosenoidales y sinusoidales de la señal en todas las frecuencias. De esta manera, una señal geofísica cualquiera puede ser descompuesta en las distintas componentes periódicas presentes.

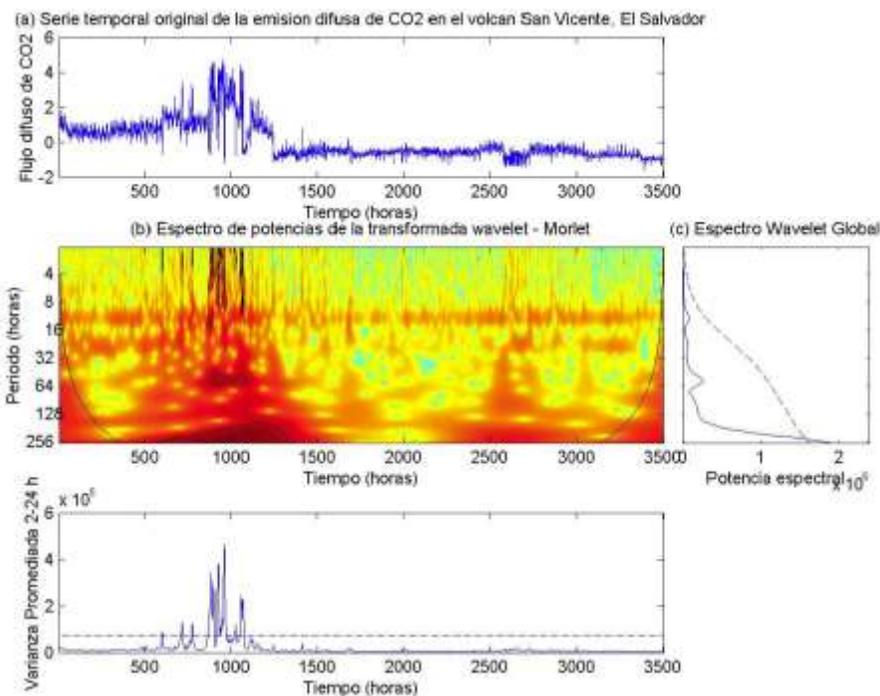
Existen sismógrafos que registran la actividad sísmica en un amplio rango de frecuencias, de manera que es factible adquirir la señal sísmica y procesarla en tiempo real para estimar su espectro de frecuencias. El espectro de frecuencias nos proporciona información sobre la naturaleza de la señal sísmica. Las señales sísmicas conocidas como tremor volcánico se asocian a la resonancia de estructuras rellenas de fluidos, bien sean estáticos o estén en movimiento, localizadas bajo un volcán activo.

El tremor volcánico se caracteriza por formas de onda persistente o sostenida en el tiempo. El tremor refleja una vibración continua del suelo o pequeños sismos muy frecuentes cuyas ondas se solapan. Si la señal sísmica presenta una frecuencia constante, hablamos de tremor armónico. Si la señal sísmica varía ostensiblemente en frecuencia o amplitud, hablamos de tremor espasmódico.

Los sismos de largo período o sismos LP pueden atribuirse a la resonancia provocada por cambios en la presión en los fluidos alojados en grietas, cavidades y conductos. En este tipo de eventos predominan las bajas frecuencias. En un volcán activo o en un sistema volcánico donde comienza a alojarse magma poco desgasificado es frecuente observar enjambres de terremotos de tipo LP con eventos individuales de frecuencias dominantes bien definidas.

A pesar de la versatilidad de la transformada de Fourier, existen una serie de inconvenientes. En primer lugar, la transformada de Fourier falla cuando se trata de representar con exactitud funciones de componentes no periódicas bien localizadas en el tiempo (o en el espacio), como pueden ser los transitorios o impulsos (ejemplo, la llegada de un tren de ondas P o S a un sismógrafo). En segundo lugar, la transformada de Fourier no proporciona información sobre la dependencia temporal de la señal ya que la transformada refleja el promediado a través de toda la duración de la serie temporal.

Figura 1.9 - Análisis tiempo-frecuencia mediante



Fuente: Lorenzo Salazar, J. M. (2015). Volcanes, Terremotos y Matemáticas, pág. 11.

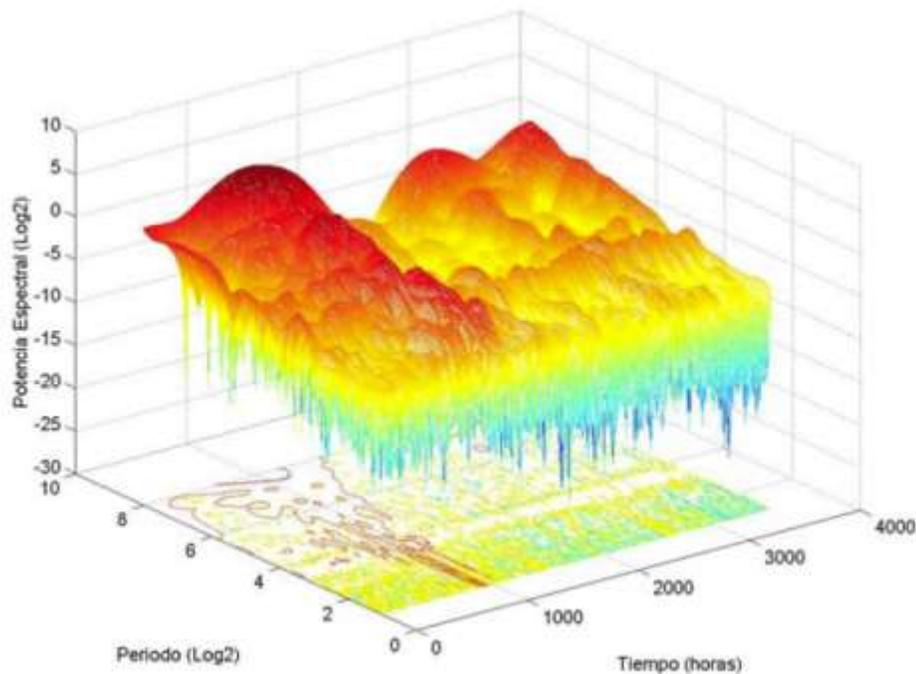
Para mitigar las limitaciones del análisis de Fourier la Matemática proporciona nuevos métodos: el análisis tiempo-frecuencia. El primer método recibe el nombre de transformada corta o de periodo corto de Fourier. El segundo método recibe el

nombre de transformada de ondícula o transformada wavelet.” (Lorenzo Salazar, 2015, págs. 8-12)

Un análisis completo de la serie temporal, acerca de la variación temporal de la emisión difusa de dióxido de carbono en el flanco del Volcán de San Vicente. Junto con la serie temporal original se puede observar un gráfico bidimensional tiempo-frecuencia (escalograma), obtenido aplicando la transformada de wavelet.

En la figura 1.10, se muestra una representación tridimensional del escalograma de la figura 1.9, modificando los ejes de frecuencia y potencia espectral en relación a logaritmos de base 2.

*Figura 1.10 - Escalograma 3D de la serie temporal de la emisión difusa de dióxido de carbono el volcán San Vicente, en 2002.*



*Fuente: Lorenzo Salazar, J. M. (2015). Volcanes, Terremotos y Matemáticas, pág. 12.*

## **1.3. PLANTEAMIENTO DEL PROBLEMA**

### ***1.3.1. Situación problema***

En la actualidad, en nuestro país se lleva a cabo un constante monitoreo sobre la actividad sísmica y volcánica, debido a la abundancia de volcanes activos en territorio salvadoreño. El Salvador se encuentra ubicado en el llamado “Cinturón de Fuego del Pacífico” donde se concentra el 75% de volcanes activos e inactivos del mundo. Uno de ellos es el volcán Ilamatepec de Santa Ana. El estudio sobre sus actividades conlleva un proceso de investigación y análisis de información, que es realizado por personal especialista del Grupo de Investigación Vulcanológica de la Universidad Nacional de El Salvador.

La obtención de información se realiza a través de sensores y mecanismos especiales, instalados en las cercanías de los volcanes, además se auxilian de datos publicados por el SNET y otras fuentes de información externas. Dentro de los parámetros monitoreados se encuentran la temperatura del aire y suelo, humedad, concentración de gases y actividad sísmica. Algunos de estos datos son obtenidos de manera remota y otros son descargados directamente en el sitio donde se encuentran los equipos.

En el volcán Ilamatepec existen 4 estaciones de monitoreo del SNET, las cuales brindan información de la actividad sísmica y algunas emisiones de gases que genera el volcán, la medición de las emisiones de dióxido de azufre de la cual el SNET no cuenta con información, la realiza el Grupo de Investigación Vulcanológica de la Universidad Nacional de El Salvador, ellos cuentan con el equipo necesario para monitorearla, sin embargo, para realizar un análisis de calidad se debe de consolidar toda la información en un solo repositorio de datos.

Para el cálculo de los indicadores, es necesario descargar toda la información sísmica de las 4 estaciones de ubicadas en el volcán desde la página web oficial del SNET, esta información en su mayoría viene en forma de imágenes porque lo

que la tabulación de datos se va ingresando de forma manual a un procesador de hojas de cálculo en este caso Microsoft Excel, esta información se actualiza de forma irregular algunas veces puede existir solo uno o un par de días de retraso y en algunos casos varias semanas.

La información de los gases se encuentra almacenada en estaciones de monitoreo próximas al volcán, estos equipos no cuentan con transferencia de información remota o acceso a internet por lo que para extraer las mediciones es necesario transportarse hasta el lugar y copiar la información directo de la placa base del sensor que generan un tipo de archivo con extensión “.pak” existen diferentes softwares capaces de descomprimir esta información en un formato de texto manipulable como por ejemplo QDOAS.

Este proceso de análisis y obtención de información genera un gasto excesivo de tiempo y recursos, por esto el equipo del Grupo de Investigación Vulcanológica de la Universidad Nacional de El Salvador no es capaz de realizar un monitoreo periódico, actualmente la medición y el cálculo de los indicadores se realiza de forma mensual, este análisis sería una herramienta útil si se pudiera calcular de forma diaria o en “tiempo real”, pudiendo ayudar a la previsión de eventos con magnitudes elevadas en el volcán. Es aquí donde es recomendable y necesario un sistema que centralice la información y permita el control diario de la actividad sísmica y emisiones de gases volcánicos, ya que a nivel nacional no se cuenta con una red de monitoreo con tales características.

### ***1.3.2. Formulación del problema***

Como se describió en la situación problema, el proceso de análisis y monitoreo volcánico del volcán Ilimatepec presenta varias deficiencias. Dentro del proceso actual puede observarse que carece de un mecanismo automatizado de transferencia, almacenamiento centralizado, procesamiento, y publicación de las mediciones. Esto genera que existan problemas de los cuales se pueden mencionar:

- a) Dificultad en la recopilación datos requeridos.
- b) Descentralización de la información.
- c) Procesamiento lento y sujeto a errores humanos.
- d) Cálculo y análisis de los indicadores limitados.
- e) Información sin posibilidad de ser compartida.
- f) Complicaciones para consultar datos históricos.

#### ***1.3.2.1. Dificultad en la recopilación de los datos requeridos***

Como no se cuenta con un mecanismo automatizado de transferencia, para la información recopilada sobre la concentración del dióxido de azufre, hace imposible realizar un análisis de calidad ya que cuando los datos son utilizados en el procesamiento ya ha transcurrido un tiempo importante desde que fueron medidos.

También obliga a los responsables a viajar a las estaciones próximas al volcán para poder realizar la extracción, ya que no se cuenta con personal en las cercanías que pueda realizar la descargar y transferencia de datos.

#### ***1.3.2.2. Descentralización de la información***

Actualmente es necesario recopilar datos de distintas fuentes dentro de las externas están EarthWorm y la más importante el Servicio Nacional de Estudios Territoriales (SNET) de El Salvador, y dentro de las mediciones internas se obtienen las emisiones de gases volcánicos. Es necesario un procesamiento extra para asegurar la calidad de la información durante la recopilación y el almacenamiento de los datos en el cual se pierde tiempo significativo.

#### ***1.3.2.3. Procesamiento lento y sujeto a errores humanos***

Los datos son recolectados y procesados para obtener información de forma tabular, graficas, o se generan reportes escritos sobre el análisis de la información, toda esta tarea se debe realizar de forma manual y por una sola persona, si consideramos el volumen de los datos, dicho proceso esta propenso a errores. Y

aun auxiliándose de herramientas de cálculo como Microsoft Excel y Matlab procesar la información mensual puede requerir varias semanas de trabajo.

#### ***1.3.2.4. Cálculo y análisis de los indicadores limitados***

Uno de los objetivos principales de la vulcanología es la prever posibles desastres ocasionados durante una erupción, para esto es necesario un monitoreo constante de la actividad sísmica y emisiones volcánicas, por consecuencia hacer un análisis de las condiciones mensuales sirve meramente para fines académicos e investigativos, pero este no es el fin que busca el Grupo de Investigación Vulcanológica de la Universidad Nacional de El Salvador; se necesita montar una red de monitoreo que pueda ayudar no solamente al campo de las ciencias investigativas sino también a la población sobre posibles riesgos ambientales.

#### ***1.3.2.5. Información sin posibilidad de ser compartida***

Es de interés de la población en general estar informada sobre las anomalías que pueden suceder en los volcanes que se encuentran en sus inmediaciones actualmente no se cuenta con una herramienta de publicación o consulta de información de actividad volcánica. Una herramienta de este tipo traería también beneficios para los equipos de investigación en el extranjero, tendrían acceso a una fuente de información centralizada y escalable para sus mediciones.

#### ***1.3.2.6. Complicaciones para consultar datos históricos***

No se posee una herramienta o software que posibilite la consulta de datos históricos, la creación de una base de datos sentaría el origen para posibles investigaciones en el futuro sobre cómo se lleva a cabo la actividad del volcán a través de una línea de tiempo abriendo un nuevo espacio para análisis predictivos

## 1.4. JUSTIFICACIÓN

La actividad volcánica se puede medir en función de la actividad sísmica que presenta, y la emisión de gases que se provocan. Estas dos variables, se utilizan como marco de referencia para identificar si existe alguna irregularidad. El análisis nos ayudará comprender, el comportamiento del volcán, así como la previsión de cualquier tipo de eventos de alta magnitud.

Existió una alta correlatividad entre el índice de amplitud sísmica RSAM y la emisión de Dióxido de Azufre que emitió el volcán Ilamatepec en la erupción de 2005, mientras se incrementaba la actividad volcánica, se podía observar un aumento acelerado de los indicadores de medición de la amplitud sísmica RSAM y la emisión de Dióxido de Azufre. Las medidas mostraron un severo incremento desde dos meses antes de la erupción, en comparación con las mediciones normales que el volcán presentaba con anterioridad. Fue tal el incremento de dióxido de azufre en las zonas con mayor carga sísmica, que se empezaron a general fumarolas de ceniza que provocaron la evacuación de miles de familias que habitaban la zona.

A pesar que instituciones como el SNET o La Geo cuentan con redes de monitoreo volcánico, no existe ninguna herramienta informática, que muestre al público en general, la relación que existe entre estas dos variables. Las herramientas de consultas a las que tiene acceso la población en general, no se encuentran actualizadas, o presentan retrasos significantes en la información publicada.

El desarrollo de un sistema de monitoreo de actividad sísmica y emisiones de dióxido de azufre en el volcán de Santa Ana, tendrá como beneficiario directo a miembros del Grupo de Investigación Vulcanológica de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador, y de igual forma, al público en general que desee conocer la manera en que varía la actividad del volcán.

Un sistema de esta índole, ayudaría en el seguimiento del comportamiento normal y anormal de la actividad volcánica, proporcionando información continua con el cálculo de indicadores, los cuales serían sometidos a análisis por parte de cualquier equipo de investigación vulcanológico dentro y fuera de nuestro país. Todo esto, gracias a la portabilidad que presentará el sistema, al ser una aplicación web, adaptable a cualquier dispositivo y en cualquier parte donde se tenga acceso a Internet.

Un sistema con tales características optimizaría el tiempo de la obtención de datos y posterior análisis, ya que actualmente se requieren visitas técnicas para la extracción de la información de los sensores; a su vez los reportes se generan de forma mensual y este hecho restringe el estudio de la actividad volcánica debido a diversos indicadores que no se obtienen de manera inmediata.

Se espera, además, sentar un punto de partida para futuros análisis sobre otros volcanes, pues de momento, no existe un sistema con las características propuestas a nivel Centroamericano. Este sistema será pionero sobre próximos estudios informáticos que se presenten bajo el área de las ciencias vulcanológicas.

## **1.5. OBJETIVOS**

### ***1.5.1. Objetivo general***

- Implementar un sistema informático para el monitoreo volcánico de variables multiparamétricas sísmicas y emisiones de dióxido de azufre en el volcán de Santa Ana, mediante el cual se automatice y centralice el análisis estadístico con la generación de gráficos que permitan prever futuras actividades volcánicas.

### ***1.5.2. Objetivos específicos***

- Comprender la relación existente entre los valores de las variables de movimiento sísmico y emisiones de dióxido de carbono, para predecir el comportamiento de un volcán.
- Diseñar un sistema capaz de relacionar las variables de movimiento sísmico y emisiones de dióxido de azufre, que pueda ser consultado desde cualquier dispositivo con acceso a Internet, responsivo y amigable al usuario.
- Desarrollar un sistema de medición de variables multiparamétricas estable, exacto, con actualización automatizada de datos en tiempo real.
- Almacenar bajo un Sistema Gestor de Base de Datos, toda la información necesaria para consultar, procesar y representar los datos de movimiento sísmico y emisiones de dióxido de azufre.
- Dotar a la Facultad Multidisciplinaria de Occidente, de la Universidad de El Salvador, con un sistema de monitoreo volcánico escalable, de fácil adaptación para análisis de otras estaciones y otros volcanes.

## 1.6. ALCANCES

La realización de nuestro proyecto, contempla los siguientes alcances:

1. **Diseño y desarrollo de sistema para monitoreo volcánico de variables multiparamétricas sísmicas y emisiones de dióxido de azufre en el volcán de Santa Ana:** Se realizará el desarrollo de un sistema que ayude a presentar y relacionar el valor de los indicadores RSAM y dióxido de azufre del Volcán de Santa Ana.
  - a. **Tipo de sistema:** El sistema a desarrollar, será un aplicativo web, el cual podrá ser consultado desde cualquier navegador. Contará con un diseño responsivo, que le permitirá adaptarse a cualquier tipo de dispositivo desde donde se consulte la información. Se limitará a automatizar la transferencia, almacenamiento y presentación de la información.
  - b. **Recopilación de datos:** Se recopilarán los datos desde dos fuentes: Los datos que obedecen a sismicidad, serán proporcionados por el SNET; el tipo de comunicación y la estructura de los datos del sistema serán definidos por esta entidad. Los datos que respectan al dióxido de azufre, serán obtenidos por medio de sensores de medición de emisión de gases, del cual se obtendrán los archivos en formato de texto. El sistema propuesto, contará con los procedimientos necesarios para recopilar y transformar estos datos; sin embargo, la generación de los mismos, y los dispositivos necesarios para su obtención, son responsabilidad de cada entidad correspondiente.
  - c. **Comunicación de datos:** Se establecerá un canal de comunicación para la transmisión de los datos ligados al dióxido de azufre, entre un proceso automático configurado en la laptop de la estación de monitoreo y el servidor. Para la comunicación de datos de actividad sísmica compartidos por el SNET, se utilizará el aplicativo Earthworm.

- d. Almacenamiento de datos:** Se diseñará y creará una base de datos que contenga información de variables sísmicas, datos relacionados a la concentración de dióxido de azufre. En esta base de datos se consolidará y centralizará toda la información obtenida de las distintas fuentes.
- 2. Pruebas de calidad del sistema:** Se realizarán pruebas que garanticen el correcto funcionamiento del sistema; desde el punto de vista técnico y funcional.
- 3. Implementación del sistema:** Una vez realizadas las pruebas de calidad, se procederá a implementar el sistema en un ambiente productivo, libre al público en general por medio de Internet. Cabe destacar, que la implementación del sistema, se realizará en un servidor privado, por un tiempo de un año calendario. Será responsabilidad del Departamento de Física de la Facultad Multidisciplinaria de Occidente, mover el sistema a servidores internos de la facultad.

## 1.7. LIMITACIONES

- 1. No se realizarán mediciones:** El sistema depende de mediciones de terceros para su funcionamiento, si no se cuenta con la entrada de datos necesaria el sistema puede quedar deshabilitado de manera indefinida.
- 2. Periodo de actualización de datos:** El origen de los datos de sismicidad serán brindados por el SNET, por lo que la actualización de los datos que maneje el sistema depende en gran medida de la frecuencia con la cual esta institución brinde los datos.
- 3. Cobertura internet móvil:** Las redes de internet móvil en nuestro país, tienen escasa cobertura en zonas rurales y montañosas, esto puede generar que los datos no se envíen con la rapidez esperada, que se reciban de manera parcial o se pierdan en su totalidad.
- 4. Escases información sobre sistemas de monitoreo volcánico:** No fue posible encontrar proyectos de desarrollo con las mismas características que el propuesto, la base de la información teórica fue recabada sobre redes de monitoreo volcánico internacional, pero no incluyen fases de diseño y desarrollo.
- 5. Desarrollo adaptado a dispositivo de mediciones específico:** Todo el diseño y desarrollo de software, se realizará alrededor del modelo S2000 del sensor de mediciones de emisiones de gas (ScanDoas), por lo que no se garantiza el correcto funcionamiento de la aplicación utilizando un dispositivo de otra serie.
- 6. Soporte de hardware de medición:** El equipo de trabajo no se hace responsable por el soporte a cualquier hardware utilizado para realizar mediciones de datos.

## 1.8. METODOLOGÍA DE DESARROLLO

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Esta sistematización nos indica cómo dividiremos un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas, nos ayuda a definir entradas y salidas para cada una de las etapas y, sobre todo, normaliza el modo en que administraremos el proyecto.

Existen diferentes tipos de metodologías de desarrollo, la mayoría, creadas a partir de la metodología del ciclo en cascada. Éste, es un ciclo de vida de software que admite iteraciones, contrariamente a la creencia de que es un ciclo de vida secuencial como el lineal. Después de cada etapa se realiza una o varias revisiones para comprobar si se puede pasar a la siguiente. Es un modelo rígido, poco flexible, y con muchas restricciones. Aunque fue uno de los primeros, y sirvió de base para el resto de los modelos de ciclo de vida.

Para nuestro proyecto, hemos decidido implementar la Metodología de *Ciclo de Vida de Software en V*.

Este ciclo fue diseñado por Alan Davis, y contiene las mismas etapas que el ciclo de vida en cascada puro. A diferencia de aquél, a éste se le agregaron dos subetapas de retroalimentación entre las etapas de análisis y mantenimiento, y entre las de diseño y debugging.

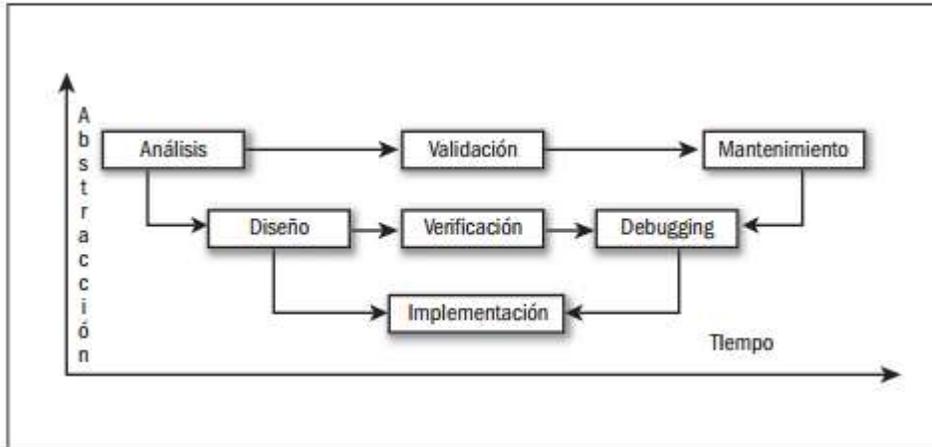
El Método-V fue desarrollado para regular el proceso de desarrollo de software por la Administración Federal Alemana. Describe las actividades y los resultados que se producen durante el desarrollo del software.

Proporciona una guía para la planificación y realización de proyectos. Los siguientes objetivos están destinados a ser alcanzados durante la ejecución del proyecto:

- **Minimización de los riesgos del proyecto:** Mejora la transparencia del proyecto y control del proyecto, especificando los enfoques estandarizados, describe los resultados correspondientes y funciones de responsabilidad. Permite una detección temprana de las desviaciones y los riesgos y mejora la gestión de procesos, reduciendo así los riesgos del proyecto.
  
- **Mejora y garantía de calidad:** Como un modelo de proceso estándar, asegura que los resultados que se proporcionan sean completos y contengan la calidad deseada. Los resultados provisionales definidos se pueden comprobar en una fase temprana. La uniformidad en el contenido del producto mejora la legibilidad, comprensibilidad y verificabilidad.
  
- **Reducción de los gastos totales durante todo el proyecto y sistema de ciclo de vida:** El esfuerzo para el desarrollo, producción, operación y mantenimiento de un sistema puede ser calculado, estimado y controlado de manera transparente mediante la aplicación de un modelo de procesos estandarizados. Reduciendo la dependencia en los proveedores y el esfuerzo para las siguientes actividades y proyectos.
  
- **Mejora de la comunicación entre todos los inversionistas:** La descripción estandarizada y uniforme de todos los elementos pertinentes y términos es la base para la comprensión mutua entre todos los inversionistas. De este modo, se reduce la pérdida por fricción entre el usuario, comprador, proveedor y desarrollador.

En la siguiente figura se muestra una representación del ciclo de vida de software en V. Resume los pasos principales de las entregas de los sistemas de validación.

Figura 1.11 - Representación de la metodología del ciclo de vida de software en V.



Fuente: Forsberg, K., Mooz, H., & Cotterman, H. (2005). *Visualizing Project Management* (Tercera ed.). New York

Nuestro proyecto se compondrá de las siguientes fases en cada etapa:

### 1. **Análisis**

- **Fase 1:** Determinación de Requerimientos.
- **Fase 2:** Estudio de Requerimientos e Historias de Usuario.
- **Fase 3:** Estudio e Identificación de Recursos.
- **Fase 4:** Determinación de Lenguajes de Programación, Sistema Gestor de Base de Datos y Protocolos de Comunicación.

### 2. **Diseño**

- **Fase 1:** Diseño General de Componentes de Sistema.
- **Fase 2:** Diseño de Base de Datos.
- **Fase 3:** Desarrollo de Core de Sistema.
- **Fase 4:** Desarrollo de Componentes Menores de Sistema.
- **Fase 5:** Diseño de Interfaz de Usuario.
- **Fase 6:** Documentación y Creación de Manuales de Desarrollador y Usuario.

### **3. Implementación**

- **Fase 1:** Implementación Previa de Sistema (Entorno de Pruebas con el Usuario).
- **Fase 2:** Implementación Productiva de Sistema.

### **4. Validación**

- **Fase 1:** Pruebas Funcionales del Sistema.
- **Fase 2:** Identificación de Bugs y Oportunidades de Mejora.

### **5. Verificación**

- **Fase 1:** Pruebas No Funcionales del Sistema.
- **Fase 2:** Identificación de Bugs y Oportunidades de Mejora.

### **6. Debugging**

- **Fase 1:** Reparación de Errores.
- **Fase 2:** Implementación de Mejoras.

### **7. Mantenimiento**

- **Fase 1:** Monitoreo de Funcionamiento de Sistema.

## 1.9. PLANIFICACIÓN DE RECURSOS A UTILIZAR

### 1.9.1. Recursos humanos

- Personal para realizar el diseño, desarrollo e implementación del proyecto.
- Personal del departamento de física para asistir en cuanto a conocimientos de las variables a medir y en realización de pruebas.

### 1.9.2. Otros recursos

- Dos planes de Internet Móvil para pruebas.
- Tres planes de servicio de Internet para investigaciones realizadas por los desarrolladores.
- Energía eléctrica.
- Vehículo para transportarse a áreas de monitoreo volcánico para pruebas.
- Datos recolectados por el SNET de mediciones sísmicas volcánicas.
- Otros recursos utilizados por los desarrolladores del proyecto.

### 1.9.3. Recursos de hardware

Tabla 1.1 - Recursos de hardware a utilizar.

ELEMENTO	CANTIDAD
Computadoras (una para cada desarrollador del proyecto)	4
Computadora portátil con requerimientos mínimos	2
Módems portátiles (USB modem)	2
Scan DOAS	2
Panel solar	2
Baterías de 12 voltios	4
Dispositivos de red varios del departamento de física	-

### **1.9.4. Recursos de software**

*Tabla 1.2 - Recursos de software a utilizar.*

<b>ELEMENTO</b>	<b>CANTIDAD</b>
Sistema Operativo para cada una de las computadoras para desarrollo	4
Sistema Operativo para el servidor	1
Servidor web	1
Gestor de base de datos	4
Herramienta de para administración del Gestor de Bases de Datos	4
Librerías extra para graficar con Java	4
Entorno de desarrollo integrado (IDE) para Java	4
Framework para soporte de programación web	4
Servidor SSH (SFTP)	4
Cliente SSH (SFTP)	7
Navegadores web para pruebas	-

---

**CAPÍTULO II:  
DETERMINACIÓN  
DE  
REQUERIMIENTOS**

---

## 2. DETERMINACIÓN DE REQUERIMIENTOS

### 2.1. REQUERIMIENTOS INFORMÁTICOS

#### 2.1.1. *Requerimientos funcionales*

*Cuadro 2.1 - Requerimientos funcionales del sistema.*

<b>CÓDIGO</b>	<b>REQUERIMIENTO FUNCIONAL</b>
RF – 01	El sistema recibirá datos de manera remota sobre las concentraciones de SO <sub>2</sub> desde las estaciones de medición.
RF – 02	El sistema recibirá datos de mediciones sísmicas, provenientes del SNET.
RF – 03	El sistema deberá unificar, transformar y almacenar los datos de emisiones de SO <sub>2</sub> y mediciones sísmicas en una sola base de datos.
RF – 04	El sistema tendrá de posibilidad de agregar múltiples estaciones de mediciones sísmicas y dióxido de azufre.
RF – 05	El sistema calculará el valor de RSAM para una determinada estación sísmica.
RF – 06	El sistema llevará el registro del número de sismos en una determinada estación.
RF – 07	El sistema relacionará las variables de RSAM y emisiones de SO <sub>2</sub> , por medio de la fecha y hora de registro de la medición.
RF – 08	La base de datos del sistema, almacenará información histórica de mediciones sísmicas y emisiones de SO <sub>2</sub> , a partir del día del primer registro de datos.
RF – 09	El sistema mostrará gráficos con las mediciones de emisiones de SO <sub>2</sub> para una estación en específico.
RF – 10	El sistema mostrará gráficos con las mediciones sísmicas para una estación en específico.
RF – 11	El sistema mostrará gráficos que relacionen las mediciones de emisiones de SO <sub>2</sub> y RSAM para una estación en específico.
RF – 12	El sistema permitirá manejar rangos de mediciones, sobre los cuales se crearán estados del comportamiento del volcán, para una estación en específico.
RF – 13	El sistema deberá permitir los roles de usuario de Administrador, Investigador y Visitante.
RF – 15	El sistema permitirá la creación de usuarios, a otros usuarios con rol de administrador
RF – 17	El sistema permitirá la modificación de rangos de fecha y hora, variables y mediciones a los usuarios con roles de Administrador e Investigador.
RF – 19	El sistema permitirá la modificación de parámetros de configuración a usuarios con rol de Administrador.
RF – 20	El sistema permitirá exportar detalles de mediciones o cálculos a usuarios con rol de Administrador o Investigador.
RF – 21	El sistema notificará, por medio de correo electrónico sobre la actividad anormal de mediciones que sobresalgan de los rangos predispuestos de estados aceptables del volcán,

	a los usuarios con rol de Investigador.
RF – 22	El sistema notificará, por medio de correo electrónico, sobre fallos en los procesos de carga de información en la base de datos, a los usuarios con rol de Administrador.
RF – 23	El sistema permitirá realizar copias de seguridad de información histórica almacenada en la base de datos.
RF – 24	El sistema mostrará en pantalla, el estado actual del volcán, en relación a los rangos de comportamiento aceptable.
RF – 25	El sistema permitirá la creación y publicación de reportes resumen de la actividad del volcán, con periodicidad diaria.

### ***2.1.2. Requerimientos no funcionales***

*Cuadro 2.2 - Requerimientos no funcionales del sistema.*

<b>CÓDIGO</b>	<b>REQUERIMIENTO NO FUNCIONAL</b>
RNF – 01	Las tecnologías de servidor necesarias para la puesta en producción del software del sistema deberán ser tecnologías libres.
RNF – 02	Los tiempos de respuesta del sistema deberán ser optimizados.
RNF – 03	La seguridad del sistema debe estar basada en la creación de perfiles de usuarios y la asignación de permisos debe ser dinámica
RNF – 04	Después de un tiempo de inactividad, el sistema deberá cerrar la sesión del usuario autenticado.
RNF – 05	El sistema deberá permitir un número limitado de intentos de ingreso al sistema, después de este deberá bloquearse.
RNF – 06	El sistema deberá presentar una Ayuda en línea.
RNF – 07	El nombre del usuario que ha iniciado sesión deberá estar visible en el sistema.
RNF – 08	El sistema deberá mostrar al usuario pantallas de alerta, información y error.
RNF – 09	El sistema deberá permitir al usuario cambiar su contraseña.
RNF – 10	El sistema deberá ser escalable, permitiendo la adición de nuevos volcanes para ser monitorizados, y nuevas estaciones
RNF – 11	Los usuarios que exporten reportes del sistema, deberán poder elegir entre formatos .XLS y .CSV

## 2.2. REQUERIMIENTOS DE DESARROLLO

### 2.2.1. *Recurso humano*

A continuación, se presenta el recurso humano necesario para el desarrollo del proyecto.

#### ***Docente Director***

Quien servirá de guía, proporcionará sus opiniones y sugerencias durante el desarrollo del proyecto, a su vez evaluará el trabajo realizado por el equipo de desarrollo al final de cada una de las etapas del proyecto.

#### ***Especialista en vulcanología***

Brindará apoyo a los desarrolladores sobre las especificaciones técnicas de los equipos de medición, análisis de variables sísmicas y temas relacionados a sismología y vulcanología en general.

#### ***Equipo de desarrollo***

Este equipo está compuesto por 4 integrantes. Cada uno tomará los roles de analista, diseñador y programador según la etapa que se esté desarrollando.

Entre los roles que cada uno de los integrantes del equipo de desarrollo deben asumir están:

*Cuadro 2.3 - Competencias de los integrantes del equipo de desarrollo.*

<b>ROL</b>	<b>CARACTERÍSTICAS</b>
Analista	<ul style="list-style-type: none"><li>▪ Habilidades para la obtención y análisis de información.</li><li>▪ Capacidad de análisis.</li><li>▪ Visión estratégica.</li><li>▪ Orientación al cliente.</li><li>▪ Comunicación efectiva.</li><li>▪ Habilidades de negociación.</li></ul>

Diseñador	<ul style="list-style-type: none"> <li>▪ Habilidades para la interpretación.</li> <li>▪ Orientación al cliente.</li> <li>▪ Conocimientos de diseño orientado a objetos.</li> <li>▪ Capacidad de establecer estándares de diseño para aplicaciones web.</li> <li>▪ Capacidad de diseñar aplicaciones clientes servidor.</li> <li>▪ Enfoque creativo.</li> </ul>
Programador	<ul style="list-style-type: none"> <li>▪ Habilidades para el análisis de información y con experiencia en programación web.</li> <li>▪ Manejo de diferentes lenguajes de programación web.</li> <li>▪ Interés por la innovación.</li> <li>▪ Habilidades lógicas.</li> <li>▪ Manejo de paradigmas de programación.</li> </ul>
Administrador de base de datos	<ul style="list-style-type: none"> <li>▪ Capacidad de diseñar bases de datos relacionales.</li> <li>▪ Conocimiento y experiencias con Gestores de Bases de Datos.</li> </ul>

En el siguiente cuadro se pueden observar las funciones que asumirán cada uno de los integrantes del equipo de desarrollo.

*Cuadro 2.4 - Funciones de los integrantes del equipo de desarrollo.*

<b>ROL</b>	<b>FUNCIONES</b>
Analista	<ul style="list-style-type: none"> <li>▪ Dirigir y coordinar la ejecución del proyecto en conjunto con el equipo de trabajo.</li> <li>▪ Interactuar con el cliente y los usuarios mediante entrevistas.</li> <li>▪ Captura, especificación y validación de requisitos.</li> <li>▪ Elaboración de las pruebas funcionales y el modelo de datos.</li> <li>▪ Trabajo de análisis.</li> <li>▪ Hacer uso de técnicas estructuradas para llevar a cabo el modelado de procesos.</li> <li>▪ Proveer las especificaciones para los programadores.</li> <li>▪ Elaborar la documentación externa.</li> </ul>
Diseñador	<ul style="list-style-type: none"> <li>▪ Se encargarán de diseñar la solución a partir de las especificaciones de los requerimientos que fueron obtenidos por los analistas.</li> <li>▪ Elaboración de las pruebas funcionales y el modelo de datos.</li> </ul>

	<ul style="list-style-type: none"> <li>▪ Representar en lenguaje gráfico la solución del problema, basándose en el documento de especificación de requerimientos.</li> <li>▪ Proveer especificaciones para los programadores.</li> <li>▪ Hacer uso de técnicas estructuradas para llevar a cabo el modelado de procesos.</li> <li>▪ Elaborar documentación externa.</li> </ul>
Programador	<ul style="list-style-type: none"> <li>▪ Construir el sistema de acuerdo a los requerimientos de diseño</li> <li>▪ Elaborar manuales informáticos y del sistema.</li> </ul>
Administrador de base de datos	<ul style="list-style-type: none"> <li>▪ Implementar, dar soporte y gestionar bases de datos corporativas.</li> <li>▪ Definir el esquema conceptual.</li> <li>▪ Crear y configurar bases de datos relacionales.</li> <li>▪ Administrar la actividad de los datos.</li> <li>▪ Establecer el diccionario de datos.</li> <li>▪ Asegurar la confiabilidad de la base de datos.</li> <li>▪ Confirmar la Seguridad de la base de datos.</li> </ul>

## ***2.2.2. Requerimientos legales***

### ***2.2.2.1. Licencias***

En el presente proyecto se respeta y se hace cumplir la ley de los derechos de autor cumpliendo con todas las prerrogativas que dicha ley establece, con la finalidad de evitar multas y demandas en el momento de implementar el sistema.

El software libre que se utilizará en el desarrollo del sistema propuesto, está bajo la Licencia Publica General de GNU, la cual, fue creada por la Free Software Foundation (FSF), y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Si se necesita soporte técnico para este tipo de software, pueden contratarse los servicios de sus fabricantes o de personas que ofrezcan dicho servicio.

### ***2.2.3. Requerimientos de software***

A continuación, veremos una breve descripción de todos los elementos de software indispensables para el diseño, desarrollo e implementación del sistema propuesto.

#### ***2.2.3.1. Lenguaje de programación***

El lenguaje de programación que se utilizara es Java, usando el framework Java Server Faces 2.2 (JSF2) con Java Persistence API (JPA). Entre sus características fundamentales se encuentran:

- Tecnología que ejecuta del lado del servidor y no del lado del cliente.
- La interfaz de usuario es tratada como un conjunto de componentes UI, Plantillas facelets para la creación de páginas XHTML.
- Define un conjunto simple de clases base de java para componentes de la interfaz de usuario. Estas clases tratan los aspectos del ciclo de la interfaz de usuario.
- Proporciona el modelo Enterprise Java Beans (EJB) Patrón Modelo-Vista-Controlador, para el envío de eventos desde los controles de la interfaz de usuario de lado del cliente al servidor.
- Define un modelo para la internacionalización de la interfaz de usuario.
- Proporciona un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de HTML, con el uso de algunos frameworks como RichFaces, PrimeFaces, MyFaces, etc.
- Conexión para la mayoría de gestores de bases de datos como: PostgreSQL, Oracle, MySQL, MS SQL, entre otros.
- Posee mucha documentación y ejemplos prácticos para el desarrollo de sistemas web.
- Programación orientada a objetos.

Dado el gran auge empresarial por el uso de tecnologías para la creación de aplicaciones web usando JAVA; JSF se considera una herramienta potencial para el desarrollo, teniendo en cuenta que existe diversa información sobre su uso, solución de errores comunes, amplia gama de ejemplos aplicados, orientado a objetos y el mantenimiento a futuro; ya que es uno de los lenguajes caracterizados por su aceptación y uso global, teniendo en cuenta que posee la característica de ser software libre, una estructura fácil de comprender, manteniendo las propiedades de Polimorfismo, Abstracción, Encapsulamiento y Herencia.

### **2.2.3.2. Entorno de desarrollo**

Se utilizará NetBeans IDE 8.2, Netbeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

Además, permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring

### **2.2.3.3. Framework**

Se utilizará el framework PrimeFaces 6, completamente compatible con JSF2.2, también es un marco de componentes para el desarrollo de interfaces gráficas, muy

liviano y de fácil instalación; además de ser intuitivo para la utilización de controles y funcionalidades. Entre sus características se pueden mencionar:

- Soporte para AJAX.
- Amplio conjunto de componentes.
- Sin dependencias, ni configuraciones además de ser muy ligero.
- Soporte para interfaces de usuario sobre dispositivos móviles.
- Múltiples temas de apariencia.
- Amplia Documentación ordenada.

Este framework es muy utilizado para el diseño de aplicaciones web por su facilidad de comprensión ayudando en gran medida con funciones preestablecidas, mejorando el tiempo invertido en el desarrollo de un proyecto, capaz de funcionar por largos periodos sin perder la esencia del sistema a desarrollar, además cuenta con soporte en foros de programación y en su sitio web oficial.

#### ***2.2.3.4. Gestor de bases de datos***

El gestor de base de datos a utilizar es MySQL 5.7.17, proporcionando una amplia gama de características importantes como:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Condición de open source de MySQL hace que la utilización sea gratuita y se puede modificar con total libertad.
- Se puede descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones.

- Es una de las herramientas más utilizadas por los programadores orientados a Internet.
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- MYSQL, es el manejador de base de datos considerado como el más rápido de Internet.
- Gran rapidez y facilidad de uso.
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- Fácil instalación y configuración.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. MySQL es la base de datos de código abierto más popular del mercado. Gracias a su rendimiento probado, a su fiabilidad y a su facilidad de uso, MySQL se ha convertido en la base de datos líder elegida para las aplicaciones basadas en web y utilizada por propiedades web de perfil alto. Además, es una elección muy popular como base de datos integrada.

#### ***2.2.3.5. Servidor de aplicaciones web***

El servidor de aplicaciones web a utilizar será Glasfish 4.1; Glassfish es un servidor de aplicaciones que implementa la plataforma JavaEE5, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java 1.0, y muchas otras tecnologías.

Glassfish además de ser un servidor de aplicaciones, es una comunidad de usuarios, que descargan y utilizan lo libremente, también existen partners que contribuyen agregándole más características importantes.

Ingenieros y beta testers que desarrollan código y prueban las versiones liberadas para eliminar todo fallo que se encuentre, y muchos otros miembros, la Comunidad Glassfish es transparente en cuanto a términos de entrega de código fuente. Glassfish es un producto de licencia de código abierto.

- Cumplimiento de estándares a nivel de servidor de aplicaciones web, JSF 2, Java Management Extension, Content Repository para Tecnologías Java.
- Incrustable, orientado a una arquitectura de servicios.
- Flexibilidad consistente.
- Soporte completo y documentación.
- Múltiples instancias de portales, internacionalización.
- Soporte para múltiples frameworks; Struts, Spring MVC, Ajax, JSF2.

#### ***2.2.3.6. Aplicación de escritorio***

Se utilizara el lenguaje de programación Java con Java Swing en el entorno de desarrollo Netbeans 8.2, permitiendo crear aplicaciones capaces de interactuar con las bases de datos y permitiendo crear aplicaciones llamadas “Robots”, con las cuales se ejecutan tareas repetitivas con el fin de alimentar la base de datos de manera simultánea a una aplicación principal o aplicación web, con motivos de recolección de datos de lugares externos o componentes electrónicos que crean datos para luego ser filtrados y agregados a la base de datos principal.

#### ***2.2.3.7. Entorno de java***

El entorno de ejecución de Java (*JRE*) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java. Básicamente se sitúa en un nivel superior al Hardware del sistema sobre el que se pretende ejecutar la aplicación, y este actúa como un puente que entiende tanto el bytecode, como el sistema sobre el que se pretende ejecutar. Este entorno es necesario para la interpretación del código

binario en el que se desarrollarán las aplicaciones de escritorio, y para el correcto funcionamiento del framework.

Para el desarrollo y ejecución del sistema, será necesaria la versión 8 de JRE (*Java SE Runtime Environment 8u121*).

### **2.2.3.8. Sistema operativo**

El Sistema operativo a utilizar es Windows 10 para la instalación de múltiples programas de desarrollo. Cabe destacar que por su popularidad es familiar a los usuarios y fácil instalación de software necesario, teniendo en cuenta que múltiples entornos de desarrollo se encuentran disponibles para las últimas versiones de Windows. Entre sus características:

- Enfoque a la armonización de experiencias de usuario y funcionalidad.
- Arquitectura multitarea.
- Instalación para computadoras de moderados recursos.
- Intuitivo al usuario.
- Instalación de programas de versiones anteriores con compatibilidad.
- Ambiente grafico que permite ejecutar programas de forma fácil y cómoda para el usuario.

Aunque también se contempla utilizar Windows 7, ya que los usuarios están más familiarizados con tal sistema operativo; mantiene los programas funcionales entre versiones posteriores y en la mayoría de casos las herramientas utilitarias existen para múltiples versiones de Windows.

### **2.2.3.9. Herramientas utilitarias y ofimática**

- **Microsoft Word 2016:** para la creación de documentos de texto, combinando un conjunto de herramientas para la edición y formato de texto y generando documentos claros y ordenados.
- **Microsoft Excel 2016:** para la creación de documentos de hojas de cálculo, en conjunto con análisis de datos y representación gráfica de la información.

- **Microsoft PowerPoint 2016:** para creación de documentos para crear presentaciones, con manipulación de imágenes y texto.
- **Microsoft Visio 2016:** herramienta para la creación de diagramas de trabajo, modelado de bases de datos, diagramas de flujo, creación de organigramas, diagramas de redes.
- **MySQL Workbench** es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.

#### ***2.2.4. Requerimientos de hardware***

Conforme a las plataformas de desarrollo y obtención de respuesta ante la ejecución de pruebas de diagnóstico y error; los resultados mostrados en la fase de desarrollo deben de ser cortos en la espera de ejecución y depuración de la aplicación. Es por ello que es necesario poseer una computadora que soporte la ejecución del entorno de programación IDE y la ejecución de carga del servidor de aplicaciones sin mostrar demoras notorias y así realizar respectivas pruebas y control de errores.

Como requerimientos mínimos se detallan:

- Sistema operativo: Windows 7 SP1.
- Procesador: Intel Core 2 Duo 3.2 GHz.
- Memoria RAM: 4 GB.
- Disco duro: 500 GB.
- Monitor: 17"

El equipo de desarrollo cuenta con los siguientes componentes:

*Cuadro 2.5 - Hardware y sus especificaciones del equipo de desarrollo.*

<b>SISTEMA OPERATIVO</b>	<b>PROCESADOR</b>	<b>ALMACENAMIENTO</b>	<b>MEMORIA RAM</b>	<b>MONITOR</b>
Windows 10	Intel Core I5 2.2GHZ	1 TB	8GB	15"
Windows 7 SP1	AMD Athlon II 2.10 GHZ	500 GB	4 GB	21"
Windows 10	Intel Core i3 2.5GHZ	500 GB	4 GB	14"
Windows 10	AMD E1-2100	500 GB	6 GB	15"

## ***2.2.5. Requerimientos de diseño de sistemas***

### ***2.2.5.1. Técnicas de análisis***

Según los estándares de programación actual, cabe destacar la existencia de entidades que no son más que tablas de información que reúnen propiedades únicas haciendo mención a un nombre de un objeto en particular, un enfoque en el cual múltiples objetos interactúan entre sí por medio de relaciones que forman propiedades dependientes de otros objetos. Con lo cual se considera la utilización de elementos para realizar diagramas que ayuden a percibir de mejor manera el ámbito general en que objetos interactúan.

Para ello es necesario la creación de diagramas UML; entre los tipos de diagrama que se usarán para el análisis de requerimientos serán:

- Diagrama de casos de usos.
- Diagrama de Secuencia.
- Diagrama de Actividad.
- Diagrama de Clases.

Cada uno proporciona un enfoque diferente, siendo una guía para entender y no desviarse de la lógica a desarrollar; siendo pilares importantes para sostener la idea principal y sus múltiples escenarios a clasificar.

### ***2.2.5.2. Técnicas de programación***

En los últimos años la frase “orientada a objetos”, se ha vuelto muy popular, escuchándose a cada momento frases como sistemas “operativos orientados a objetos”, “lenguajes orientados a objetos”, “programación orientada a objetos” (POO), etc. En esta parte se dará a conocer los principales conceptos relacionados con la POO, así como la forma de implementarlos

Se hará mención de los conceptos generales más utilizados en el modelo orientado a objetos, los cuales son: abstracción, encapsulación y modularidad.

#### ***Abstracción***

Es una descripción o especificación simplificada de un sistema que hace énfasis en algunos detalles significativos y suprime los irrelevantes.

La abstracción debe enfocarse más en qué es un objeto y qué hace antes, de pensar en la implementación. Por ejemplo, un automóvil puede abstraerse como un objeto que sirve para desplazarse a mayor velocidad, sin importar cómo lo haga.

#### ***Encapsulamiento***

Típicamente, la información acerca de un objeto está encapsulada por su comportamiento. Esto significa que un objeto mantiene datos acerca de cosas del mundo real a las que representa en su sentido verdadero.

Un objeto se le debe “pedir” o “decir” que cambie sus propios datos con un mensaje, en vez de esperar que tales datos de procesos extremos cambien la naturaleza de un objeto.

#### ***Modularidad***

La modularidad consiste en dividir un programa en partes llamadas módulos, los cuales pueden trabajarse por separado. En términos de programación, los módulos pueden compilarse por separado y la división no depende de cierto número de

líneas sino es una división en términos de integrar en un módulo un conjunto de procedimientos relacionados entre sí, junto con los datos que son manipulados por tales procedimientos. El objetivo de la modularidad es reducir el costo de elaboración de programas al poder dividir el trabajo entre varios programadores.

Con respecto a programación, son: objeto, clase, método, envío y recepción de mensajes, herencia y polimorfismo.

### ***Clases y objetos***

El estado de un objeto abarca todas las propiedades o características distintivas del mismo y los valores de cada una de estas propiedades. En términos de programación, puede decirse que las propiedades son las variables que sirven para describir tal objeto.

El comportamiento es la forma como actúa o reacciona un objeto en términos de cambio de estado, envío y recepción de mensajes. Está formado por la definición de las operaciones (funciones y procedimientos) que puede realizar este objeto. Los tipos más comunes de operaciones, o en POO métodos, son: modificar, seleccionar, iterar, construir y destruir.

### ***Herencia***

La herencia es la contribución más importante de la POO, pues mediante este mecanismo es posible lograr la principal meta de la POO que es la reutilización de código. La herencia permite proporcionar una jerarquía de clases. En tal jerarquía, algunas clases son subordinadas a otras llamadas subclases, clases derivadas. Una subclase define el comportamiento de un conjunto de objetos que heredan algunas de las características de la clase padre, pero adquieren características especiales no compartidas por el padre, en este sentido se dice que la subclase es una especialización de la clase padre.

## ***Polimorfismo***

Otro de los mecanismos aportados por la POO es el de polimorfismo, el cual es la capacidad de tener métodos con el mismo nombre pero que su implementación sea diferente. En la vida diaria se presenta el siguiente ejemplo de polimorfismo: al tratar de frenar un vehículo siempre se debe oprimir el pedal del lado izquierdo y el vehículo se detendrá sin importar si los frenos son de tambor o de disco.

### ***2.2.5.3. Técnicas de diseño***

El Diseño Orientado a los Objetos (DOO) crea una representación del problema del mundo real y la hace corresponder con el ámbito de la solución, que es el software. A diferencia con otros métodos de diseño, el DOO produce un diseño que interconecta objetos de datos y operaciones de procesamiento para esos objetos, de forma que se modulariza la información y el procesamiento, en lugar de aislar el procesamiento.

Todos los métodos de diseño intentan desarrollar software basándose en:

- Abstracción
- Ocultamiento de información
- Modularidad

El DOO proporciona un mecanismo que permite al diseñador consigue estas tres características sin dificultad. El Análisis Orientado a Objetos, el Diseño Orientado a Objetos y la Programación Orientada a Objetos comprenden un conjunto de actividades de la Ingeniería del Software para la construcción de un sistema basado en objetos.

Se debe de comprender la diferencia entre el Análisis Orientado a Objetos, que es una actividad de clasificación, y el Diseño Orientado a Objetos, que define los objetos que se derivan de cada clase, indicando las relaciones que existen entre ellos mediante una notación. El DOO tiene que comenzar con una descripción en

lenguaje natural de la estrategia de solución, y a partir de esta descripción, el diseñador identifica los objetos y operaciones.

Los pasos del Diseño Orientado a Objetos se definen en:

- Definición del problema
- Desarrollo informal de la forma de procesamiento para la realización del software
- Formalización de la forma de procesamiento, los cuales se dividen en:
  - a. Identificar los objetos y sus atributos.
  - b. Identificar las operaciones de los objetos.
  - c. Establecer las interfaces que muestren las relaciones entre los objetos y las operaciones.
  - d. Crear un diseño detallado que proporcione una descripción de la implementación de los objetos.

## 2.3. REQUERIMIENTOS OPERATIVOS

### 2.3.1. Recurso humano

Para el buen funcionamiento del sistema, es recomendable contar con un encargado de administrar los recursos de hardware y software, esta persona será la encargada de brindar soporte técnico para cualquier eventualidad que pueda surgir en el funcionamiento diario del sistema.

A continuación, se detalla el perfil requerido para el administrador del sistema.

*Cuadro 2.6 - Perfil Cargo-Puesto del Administrador del Sistema.*

<b>PERFIL DEL ADMINISTRADOR DE SISTEMAS</b>
<b>DESCRIPCIÓN DEL PUESTO</b>
El Administrador de Sistemas, es el encargado de recopilar y analizar información para proponer nuevos y mejores métodos de funcionamiento del sistema, es responsable de dar mantenimiento y soporte a todo lo relacionado con el buen funcionamiento del software y hardware, así como dar solución a los requerimientos de los clientes.
<b>FORMACIÓN ACADÉMICA</b>
Estudiante de 5° año de Ingeniería en Sistemas Informáticos o Ciencias de la Computación, técnico graduado en Ingeniería de Sistemas, Ciencias de la computación o carreras afines.
<b>HABILIDADES TÉCNICAS</b>
<ul style="list-style-type: none"><li>▪ Conocimientos en redes de computadoras.</li><li>▪ Conocimientos en mantenimiento de computadoras.</li><li>▪ Inglés básico.</li><li>▪ Conocimientos básicos de vulcanología y sismología.</li><li>▪ Conocimientos de tecnología de Java Server Faces u otras tecnologías JAVA.</li><li>▪ Manejo de bases de datos usando PostgreSQL</li><li>▪ Conocimientos sobre lenguajes de marcado HTML y hojas de estilo CSS.</li></ul>
<b>RESPONSABILIDADES</b>
<ul style="list-style-type: none"><li>▪ Mantener en buen estado los equipos en los que funciona el sistema.</li><li>▪ Administrador la base de datos.</li><li>▪ Atender los requerimientos del sistema.</li><li>▪ Monitorear el buen funcionamiento de la red.</li><li>▪ Brindar soporte técnico a los usuarios.</li><li>▪ Optimizar los recursos puestos a su disposición.</li><li>▪ Realizar pruebas de verificación periódicas al sistema.</li></ul>

## ***2.3.2. Requerimientos legales***

### ***2.3.2.1. Licencias***

Una vez desarrollado el Sistema de Monitoreo Volcánico de Variables Multiparamétricas Sísmicas y Emisiones de Dióxido de Azufre, la Universidad de El Salvador es la propietaria del sistema, la cual le entregará una copia física y escrita al Servicio Nacional de Estudios Territoriales (SNET), acuerdo llegado según las directrices que rigen a la entidad para compartir información a investigadores académicos, el sistema estará regido por la Ley de Fomento y Protección a la Propiedad Intelectual, en donde se toma en cuenta los derechos de autor que posee el sistema. En el apartado A. Disposiciones Generales literal 7 del documento de Políticas Generales de Trabajos de Graduación de la Universidad de El Salvador, se encuentra lo siguiente:

“Los productos de los trabajos de graduación son propiedad de la Universidad de El Salvador, y será ésta (a través de la Facultad de Ingeniería y Arquitectura) la única encargada de otorgar licencias de uso, a los interesados que la soliciten a través de una nota dirigida al director de la Escuela”. (Universidad de El Salvador, 2016)

## ***2.3.3. Requerimientos de software***

### ***2.3.3.1. Requerimientos generales de software del cliente***

El software necesario para el funcionamiento del sistema en una maquina cliente se detallan en el siguiente cuadro.

*Cuadro 2.7 - Requerimientos mínimos de software en máquina cliente.*

<b>CLASIFICACIÓN</b>	<b>SOFTWARE</b>
Sistema operativo	Multiplataforma
Navegadores	Mozilla Firefox 47.0, Google Chrome 49.0 e Internet Explorer 9
Paquete informático	Microsoft Office
Lector de PDF	Adobe Acrobat Reader

### ***2.3.3.2. Requerimientos generales de software del servidor***

Para la máquina que servirá como servidor es necesario cumplir con los requisitos mínimos de software que se detallan en el siguiente cuadro, para asegurar el correcto funcionamiento del sistema.

*Cuadro 2.8 - Requerimientos mínimos de software en máquina servidor.*

<b>CLASIFICACIÓN</b>	<b>SOFTWARE</b>
Sistema operativo	Multiplataforma.
Servidor Web	Glashfish 4.1
Gestor de bases de datos	MySQL Server 5.7
Tecnología del servidor	JAVA

### ***2.3.3.3. Sistema operativo***

El sistema operativo a utilizar, en el servidor de aplicaciones, es *Windows Server 2012 R2*; esto, debido a la eficiencia y robustez del mismo, fácil configuración, plena compatibilidad con el software para servidor de aplicaciones web de Java, y por el completo soporte que el proveedor ofrece. Entre sus características principales mencionamos:

- Servicio de Transferencia Inteligente en segundo plano, que mejora el rendimiento del servidor.

- Seguridad basada en hardware para proteger los datos e impedir la manipulación de disco mientras el sistema operativo está en línea.
- Puente de Centro de Datos, para mejorar las redes LAN y hacer cumplir la asignación de ancho de banda.
- Proporciona funcionalidad de agrupación que permite a varios servidores trabajar juntos para proporcionar una alta disponibilidad.
- Posee .NET Framework 4.5 que proporciona interfaces de programación de aplicaciones (API) para el desarrollo de aplicaciones y es compatible con JRE 8.
- Equilibrio de Carga de Red.
- Permite realizar copias de seguridad y restaurar el sistema operativo, el estado del sistema y los datos almacenados en un servidor.

#### ***2.3.3.4. Earthworm***

Earthworm es un software público de código abierto para la adquisición de datos, procesamiento, archivo y distribución de sismicidad. Originalmente desarrollado por el Servicio Geológico de los Estados Unidos, los binarios y los archivos fuente de Earthworm están disponibles gratuitamente para todos. El desarrollo y el mantenimiento son ahora llevados a cabo por la comunidad de usuarios de Earthworm, ISTI (Instrumental Software Technologies, Inc.) y CERI (Centro de Investigación e Información por Terremotos).

El proyecto Earthworm se inició en 1993, principalmente en respuesta a una serie de necesidades comunes identificadas por las redes sísmicas regionales: Los sistemas de procesamiento automático utilizados por la mayoría de las redes regionales envejecían y los costos de mantenimiento crecían; Los avances en la investigación sísmica requerían datos de sensores nuevos y sofisticados para proporcionar datos de investigación viables; Las crecientes demandas inmediatas de utilidad social requerían productos nuevos y altamente visibles en tiempo real; además, la reducción de los financiamientos significaba que la mayoría de las redes

individuales ya no podían apoyar los esfuerzos de desarrollo de los sistemas locales.

La mayoría de los sistemas de procesamiento en uso de ese entonces tenían una serie de inconvenientes que hacían difícil mejorarlos para satisfacer los nuevos requerimientos. Se construyeron sobre productos de proveedores específicos, los cuales los vincularon a vendedores de alto precio y les privaron de los beneficios del creciente mercado de masas.

El esfuerzo inicial de Earthworm tenía la misión específica de reemplazar a los viejos dispositivos que habían funcionado bien durante décadas y se habían convertido en el pilar para producir soluciones de eventos rápidos. Sin embargo, las piezas de repuesto ya no estaban disponibles, y había una necesidad urgente de producir un sistema de reemplazo en un tiempo mínimo. Esto significaba producir un sistema que pudiera procesar más de quinientos canales de datos de manera rápida y confiable.

Dado que el objetivo inicial de este sistema era proporcionar una notificación rápida de los eventos sísmicos, el sistema que evolucionó no tenía "memoria" de los acontecimientos pasados. Se hizo hincapié en la velocidad y fiabilidad, y un evento se completaba cuando el sistema producía una notificación. Sus características fundamentales son las siguientes:

- **Modularidad:** Cada función realizada por el sistema debe ser encapsulada en un módulo que puede funcionar independientemente de otros módulos, tanto en hardware como en software. La implicación es que se pueda garantizar que un conjunto de funciones críticas del sistema sea independiente de otras funciones del sistema. Así, nuevas funciones experimentales pueden ser añadidas sin interrumpir las operaciones críticas preexistentes

- **Independencia del sistema:** Los módulos deben operar en varias marcas de hardware de computadora y sistemas operativos, y varios tipos de tales computadoras se pueden unir para funcionar como un sistema. Esto, con la idea de modularidad, implica que el sistema puede migrar gradualmente de un tipo de computadora a otro sin interrupción.
- **Escalabilidad:** El sistema debe proporcionar una curva de coste-rendimiento suave para acomodar redes grandes y pequeñas. Esto era quizás más importante antes de la disponibilidad de computadoras baratas y poderosas producidas en serie. Sin embargo, sigue siendo pertinente con respecto a los derechos de licencia para el software comercial incluido.
- **Conectividad:** La suposición es que estos sistemas ya no están aislados, sino que tienen que interactuar rápida y confiablemente con otros sistemas automáticos en tiempo real, sistemas de análisis interactivos y varios esquemas de notificación. El objetivo es proporcionar conectividad a varios niveles de procesamiento automático e interactivo, de modo que dichos sistemas puedan configurarse para funcionar en configuraciones que van desde la operación autónoma completa hasta el funcionamiento como nodos en un sistema de área amplia distribuido.
- **Robustez:** Tradicionalmente, el papel de los sistemas de procesamiento sísmico fue principalmente de apoyo a la investigación. Como resultado, la fiabilidad del sistema era naturalmente menos importante que el coste, el tiempo hasta la finalización y las características. Sin embargo, las nuevas responsabilidades para la prensa y las agencias de respuesta a emergencias requerían niveles muy altos de confiabilidad del sistema, especialmente durante las crisis sísmicas, cuando los datos de entrada y el poder pueden ser interrumpidos y las cargas del sistema pueden

aumentar dramáticamente. Así, cuestiones como la detección y recuperación de errores, el tiempo de reparación, la degradación agraciada y el control de la carga se volvieron vitales y tuvieron que diseñarse en el sistema. Los costos de robustez a menudo no se aprecian, en términos de tiempo para diseñar, implementar y probar, así como los costos de hardware adecuado y equipo auxiliar.

Earthworm se ha convertido de manera no oficial en el software estándar para la adquisición, procesamiento, archivo y distribución de datos de sismicidad, debido a sus grandes características.

Diferentes centros de monitoreo sísmicos basan sus sistemas de medición y notificación en Earthworm y el MARN no es la excepción.

Todos los datos de sismicidad que nuestro sistema procesara serán obtenidos por medio de dos módulos (`import_ack` y `export_ack`) de Earthworm que el MARN utilizará para compartir sus datos para nuestro propio procesamiento. Por este motivo nuestro sistema tendrá una ligera integración el sistema Earthworm para la captura de datos, para luego realizar el respectivo procesamiento de datos.

#### ***2.3.3.5. Novac program***

Otro software de terceros necesario para la operatividad del sistema propuesto es Novac Program.

Novac (Network for Observation of Volcanic and Atmospheric Change), fue un proyecto fundado por la Unión Europea entre 2005 y 2010, el cual, tuvo como propósito establecer una red mundial (primeramente sólo dentro de Europa) para medidas de emisiones de gases (específicamente SO<sub>2</sub>) y aerosoles de volcanes; y utilizar esta información para implementar planes de acción ante riesgos, y fomentar la investigación vulcanológica.

A pesar que el proyecto se disolvió a finales de marzo de 2010, ha dejado de legado una serie de herramientas tanto de hardware como de software, que hoy en la actualidad siguen siendo utilizadas para el monitoreo volcánico a nivel mundial. Una de estas herramientas de software, es el Novac Program.

Novac Program, es un programa de software libre multiplataforma, creado en 2009, que se encarga de comunicar equipos de sensores de mediciones de gases con cualquier tipo de ordenador. Esto, con el objetivo de poder extraer la información escaneado por el dispositivo y poder representar los datos por medio de gráficas, y exportar resultados para futuros análisis.

Novac Program ayuda a traducir las señales capturadas por los dispositivos de medición a archivos en lenguaje máquina. Por lo general, los sensores suelen generar un archivo en formato .pak, que Novac se encarga de convertir a .txt para poder abrirlo desde cualquier editor de texto. Novac Program, utiliza el protocolo FTP para comunicarse entre los dispositivos de medición y la computadora; el dispositivo actúa como el servidor FTP, alojando los archivos que va generando en cada medición, y Novac Program se encarga de consumir esta información para procesarla y almacenarla localmente en la computadora.

A pesar de la gran utilidad que tiene la aplicación, posee las siguientes limitantes, que, se buscará erradicar con la implementación de nuestro sistema para cualquier equipo de investigación volcánica en el futuro:

- Novac únicamente procesa las mediciones que se encuentran alojadas en el servidor FTP del dispositivo de medición. No posee base de datos, por lo que no es posible realizar análisis sobre datos históricos.
- Las gráficas realizadas por la aplicación no se pueden personalizar ni ajustar a rangos o períodos definidos.
- Novac únicamente extrae los archivos alojados en el servidor FTP al momento de abrir la aplicación. No se puede hacer análisis en tiempo real.

- La aplicación exporta los archivos .pak generados por el dispositivo de medición únicamente a formato .txt.
- Su configuración es bastante complicada y tediosa.
- Debido a la separación de la Red Novac, la aplicación ya no cuenta con soporte.

Novac Program, será utilizado en nuestro sistema como el intermediario entre el dispositivo de mediciones de SO<sub>2</sub> y nuestra base de datos. Utilizaremos Novac para poder conectarnos al dispositivo físico y obtener los datos en formato .txt, que posteriormente serán ingresados en la base de datos de la aplicación.

### ***2.3.4. Requerimientos de hardware***

#### ***2.3.4.1. Requerimientos generales de hardware***

El cuadro 2.9 muestra las características mínimas y optimas que debe cumplir una maquina cliente para ejecutar el sistema.

*Cuadro 2.9 - Requerimientos mínimos de hardware en maquina cliente.*

<b>CLASIFICACIÓN</b>	<b>HARDWARE MÍNIMO</b>	<b>HARDWARE DESEABLE</b>
Velocidad del procesador	1.2 GHz.	3.2 GHz doble núcleo.
Capacidad del disco duro	5 GB.	100 GB.
Memoria RAM	256 MB.	2 GB.
Monitor	15" VGA	17" SVGA.
Periféricos	<ul style="list-style-type: none"> <li>▪ Unidad CD-ROM.</li> <li>▪ Mouse.</li> <li>▪ Teclado.</li> <li>▪ Tarjeta de red Fast Ethernet 10/100Mbps.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Unidad CD/DVD-ROM.</li> <li>▪ Mouse.</li> <li>▪ Teclado.</li> <li>▪ Altavoces</li> <li>▪ Tarjeta de red Fast Ethernet 10/100Mbps.</li> </ul>

De acuerdo a las especificaciones de software descritas anteriormente, para la maquina servidor es necesario cumplir las siguientes características de hardware para garantizar la estabilidad del sistema.

*Cuadro 2.10 - Requerimientos mínimos de hardware en maquina servidor.*

CLASIFICACIÓN	HARDWARE
Velocidad del procesador	2 GHz (dual core)
Capacidad de disco duro	320 GB
Memoria RAM	4 GB
Monitor	15" VGA
Periféricos	<ul style="list-style-type: none"> <li>▪ Unidad CD/DVD-ROM.</li> <li>▪ Mouse.</li> <li>▪ Teclado.</li> <li>▪ Tarjeta de red Fast Ethernet 10/100Mbps</li> </ul>

### **2.3.4.2. Amazon web services**

Para la implementación del sistema propuesto, se ha decidido utilizar los servicios de Cloud Computing de la suite Amazon Web Services.

Amazon Elastic Compute Cloud (*Amazon EC2*) es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para facilitar a los desarrolladores el uso de la informática en la nube a escala de la Web, mediante de la creación de máquinas virtuales.

Amazon EC2 reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en cuestión de minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, según cambien sus necesidades.

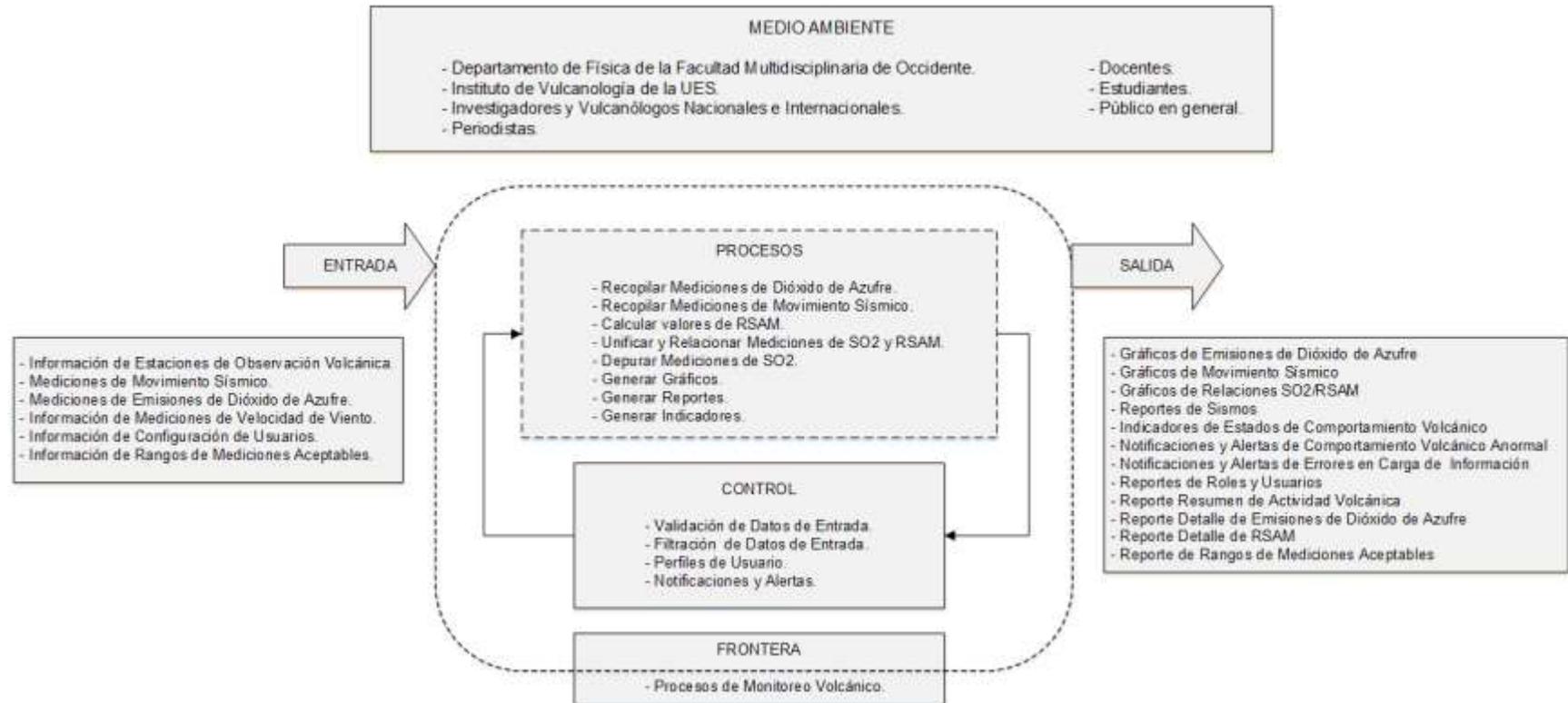
Los beneficios que ofrece Amazon EC2 son los siguientes:

- Informática a escala web elástica: Amazon EC2 permite aumentar o disminuir la capacidad en minutos, no en horas ni en días.
- Control total de las instancias: Las instancias se pueden detener, reiniciar y mantener de forma remota.
- Servicios de hospedaje en la nube flexibles: Amazon EC2 permite seleccionar una configuración de memoria, CPU y almacenamiento de la instancia, así como el sistema operativo y aplicaciones a instalar.
- Fiabilidad: Amazon EC2 ofrece un entorno de elevada confianza en el que las instancias de sustitución se pueden enviar con rapidez y anticipación.
- Seguridad: Arquitectura de red y un centro de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.
- Integridad: Amazon EC2 está integrado con la mayoría de los servicios de AWS, como Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS) y Amazon Virtual Private Cloud (Amazon VPC).

## 2.4. PROPUESTA DE SOLUCIÓN

### 2.4.1. Diagrama de enfoque del sistema propuesto

Figura 2.1 - Diagrama de enfoque del sistema propuesto.



## **2.4.2. Descripción del enfoque del sistema propuesto**

### **2.4.2.1. Medio ambiente**

- **Departamento de Física de la Facultad Multidisciplinaria de Occidente:** Serán los encargados de administrar el sistema y velar por su perfecto funcionamiento. Serán los encargados de la creación y administración de usuarios, así como del análisis completo de los datos tratados en el sistema. Recibirán alertas sobre actividad anormal del volcán en revisión y alertas sobre fallos en la aplicación.
- **Instituto de Vulcanología de la UES:** Tendrán acceso al análisis completo de los datos tratados en el sistema. Podrán definir características y parámetros de la aplicación (cómo mediciones de velocidad del viento) y definir rangos de mediciones útiles para los indicadores. Recibirán alertas sobre actividad anormal del volcán en revisión.
- **Investigadores y Vulcanólogos Nacionales e Internacionales:** Tendrán acceso al análisis completo de los datos tratados en el sistema. Podrán consumir todos los reportes y resúmenes creados por la aplicación. Recibirán alertas sobre actividad anormal del volcán en revisión.
- **Periodistas:** Tendrán acceso al análisis completo de los datos tratados en el sistema. Podrán consumir todos los reportes y resúmenes creados por la aplicación.
- **Docentes y Estudiantes:** Tendrán acceso al análisis completo de los datos tratados en el sistema. Podrán consumir todos los reportes y resúmenes creados por la aplicación. Además, podrán solicitar el código fuente de la aplicación para sugerir mejoras.

- ***Público en general:*** Cualquier persona que quiera tener acceso a los indicadores y resúmenes de la aplicación, que tenga interés sobre mediciones volcánicas.

#### ***2.4.2.2. Frontera***

- ***Procesos de Monitoreo Volcánico:*** Todos los datos tratados en la aplicación, serán para uso exclusivo de análisis del comportamiento de un volcán en específico. Análisis de cualquier otra índole quedan fuera del alcance de la aplicación.

#### ***2.4.2.3. Entrada***

- ***Información de Volcanes:*** Atributos propios de cualquier volcán que se desee monitorizar. Información propia para diferenciar análisis de diferentes volcanes, que le permite a la aplicación ser escalable.
- ***Información de Estaciones de Observación Volcánica:*** Atributos propios de las estaciones de observación, correspondientes a volcanes ingresados en la aplicación, que le permite a la aplicación ser escalable.
- ***Mediciones de Movimiento Sísmico:*** Registros de frecuencias de movimiento sísmico experimentados por un determinado volcán. Estos datos serán provistos por el SNET.
- ***Mediciones de Emisiones de Dióxido de Azufre:*** Registros de mediciones de emisiones de dióxido de azufre experimentados por un determinado volcán. Estos datos se obtendrán directamente de los dispositivos de medición alojados en las estaciones de observación.
- ***Información de Mediciones de Velocidad de Viento:*** Registros de mediciones de velocidad del viento para un momento del día determinado.

Esta información es útil para limpiar los datos de mediciones de emisiones de SO<sub>2</sub>.

- **Información de Configuración de Usuarios:** Atributos propios para la creación y administración de usuarios de la aplicación.
- **Información de Rangos de Mediciones Aceptables:** Son conjuntos de datos que definen los rangos deseables de mediciones en lo que se puede catalogar el comportamiento de un volcán. Estos rangos servirán para definir los Indicadores del Comportamiento Volcánico y el envío de alertas ante comportamientos anormales.

#### **2.4.2.4. Salida**

- **Gráficos de Emisiones de Dióxido de Azufre:** Gráfico que mostrará las medidas de SO<sub>2</sub> para un rango de fechas y horas determinado por el usuario.
- **Gráficos de Movimiento Sísmico:** Gráfico que mostrará las medidas de RSAM para un rango de fechas y horas determinado por el usuario.
- **Gráficos de Relaciones SO<sub>2</sub>-RSAM:** Gráfico que servirá para relacionar las variables de emisión de SO<sub>2</sub> y movimiento sísmico para un rango de fechas y horas determinado por el usuario.
- **Reportes de Sismos:** Reporte que mostrará la cantidad de sismos experimentados por un volcán en un rango de fechas y horas determinado por el usuario. Por defecto, mostrará un reporte diario de sismos.
- **Indicadores de Estados de Comportamiento Volcánico:** Estados del comportamiento del volcán, calculados en base a los rangos de mediciones

aceptables de un volcán determinado, que indicarán si el comportamiento del volcán es normal o presenta irregularidades.

- **Notificaciones y Alertas de Comportamiento Volcánico Anormal:** Alertas disparadas por medio de correos electrónicos, hacia los usuarios con roles de “Investigadores”. Las alertas se dispararán en base a los Rangos de Mediciones Aceptables.
  
- **Notificaciones y Alertas de Errores en Carga de Información:** Alertas disparadas por medio de correos electrónicos, hacia los usuarios con roles de “Administradores” con información de fallos técnicos de la aplicación.
  
- **Reportes de Roles y Usuarios:** Reporte resumen de los usuarios y role registrados en la aplicación, para un rango de fechas determinado por el usuario.
  
- **Reporte Resumen de Actividad Volcánica:** Reporte con el resumen de los Indicadores de Estados de Comportamiento Volcánico para un rango de fechas determinado por el usuario. Por defecto, será un reporte resumen diario.
  
- **Reporte Detalle de Emisiones de Dióxido de Azufre:** Reporte con el detalle de emisiones de SO<sub>2</sub> experimentados por un volcán determinado. El nivel de detalle y el rango de fechas y horas del reporte será determinado por el usuario.
  
- **Reporte Detalle de RSAM:** Reporte con el detalle de emisiones de SO<sub>2</sub> experimentados por un volcán determinado. El nivel de detalle y el rango de fechas y horas del reporte será determinado por el usuario.

- **Reporte de Rangos de Mediciones Aceptables:** Reporte con el detalle de los rangos ingresados como mediciones aceptables, que sirven para calcular los Indicadores de Estados de Comportamiento Volcánico.

#### **2.4.2.5. Procesos**

- **Recopilar Mediciones de Dióxido de Azufre:** Este proceso se encargará de ingresar a la Base de Datos de la aplicación, la información de mediciones de SO<sub>2</sub>. El proceso se encargará de extraer los datos directamente del dispositivo de medida de emisiones de gases, para posteriormente validarlos, tratarlos, unificarlos e ingresarlos a la base.
- **Recopilar Mediciones de Movimiento Sísmico:** Este proceso se encargará de ingresar a la Base de Datos de la aplicación, la información de mediciones de movimiento sísmico. El proceso se encargará de obtener los datos compartidos por el SNET, para luego traducirlos, validarlos, tratarlos, unificarlos e ingresarlos a la base.
- **Calcular valores de RSAM:** Este proceso, se encargará de calcular el valor de RSAM, en base a los datos extraídos de movimiento sísmico, para un rango de fechas y horas definidas en la aplicación.
- **Unificar y Relacionar Mediciones de SO<sub>2</sub> y RSAM:** Este proceso, se encargará de extraer de la Base de Datos de la aplicación, las mediciones que se relacionen entre sí, por medio de la fecha y hora de emisión de la medida.
- **Depurar Mediciones de SO<sub>2</sub>:** Este proceso se encargará de depurar las mediciones de SO<sub>2</sub>, filtrando las mediciones almacenadas por la información de la velocidad del viento, para una fecha y hora especificada. Esto, con el fin de garantizar la validez de los datos.

- **Generar Gráficos:** Este proceso se encargará de extraer datos de la Base, para posteriormente graficarlos en pantalla; en base a la información solicitada por el usuario. El proceso se encargará de establecer la comunicación entre la aplicación y la Base de Datos, la obtención y tratamiento de los datos, y la presentación gráfica de los mismos obedeciendo los diseños de la aplicación.
  
- **Generar Reportes:** Este proceso se encargará de extraer datos de la Base, para posteriormente generar reportes que podrán ser exportados o impresos; en base a la información solicitada por el usuario. El proceso se encargará de establecer la comunicación entre la aplicación y la Base de Datos, la obtención y tratamiento de los datos, la tabulación de los datos en pantalla, y la exportación de los datos a documentos físicos.
  
- **Generar Indicadores:** Este proceso se encargará de generar los Indicadores de Estados de Comportamiento Volcánico, en base a los Rangos de Mediciones Aceptables definidos para un volcán en específico. Esto, en base a la información que va siendo ingresada a la base de forma periódica.

#### **2.4.2.6. Control**

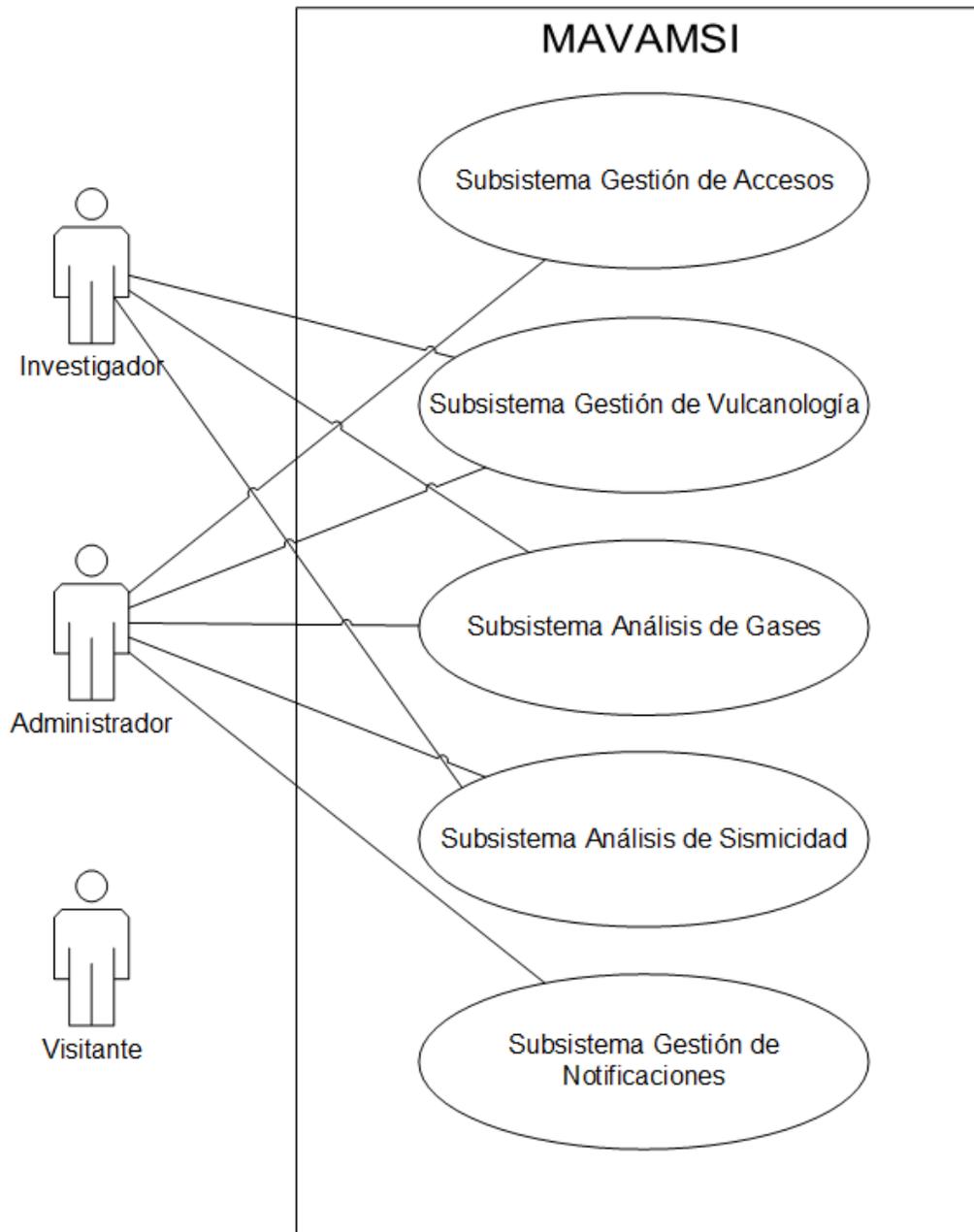
- **Validación de Datos de Entrada:** Los datos serán validados antes de ser ingresados a la Base de Datos de la aplicación, para garantizar que poseen el formato correcto que se espera obtener de cada medición.
  
- **Filtración de Resultados:** Al momento de generar un gráfico o un reporte, el usuario podrá definir los filtros de la información que considere necesarios para hacer el análisis correcto de los datos. Estos filtros incluyen rangos de fechas y horas, mediciones de velocidad y ángulo de viento y rangos de mediciones.

- **Perfiles de Usuario:** La aplicación poseerá perfiles que garanticen la seguridad de la configuración de la aplicación y que regulen el acceso a la información que poseerá la misma.
- **Notificaciones y Alertas:** El sistema será capaz de alertar a los administradores e investigadores sobre comportamientos anormales de un volcán en específico o de fallos técnicos en cualquier componente de la aplicación.

### 2.4.3. Diagramas de casos de uso (MAVAMSI)

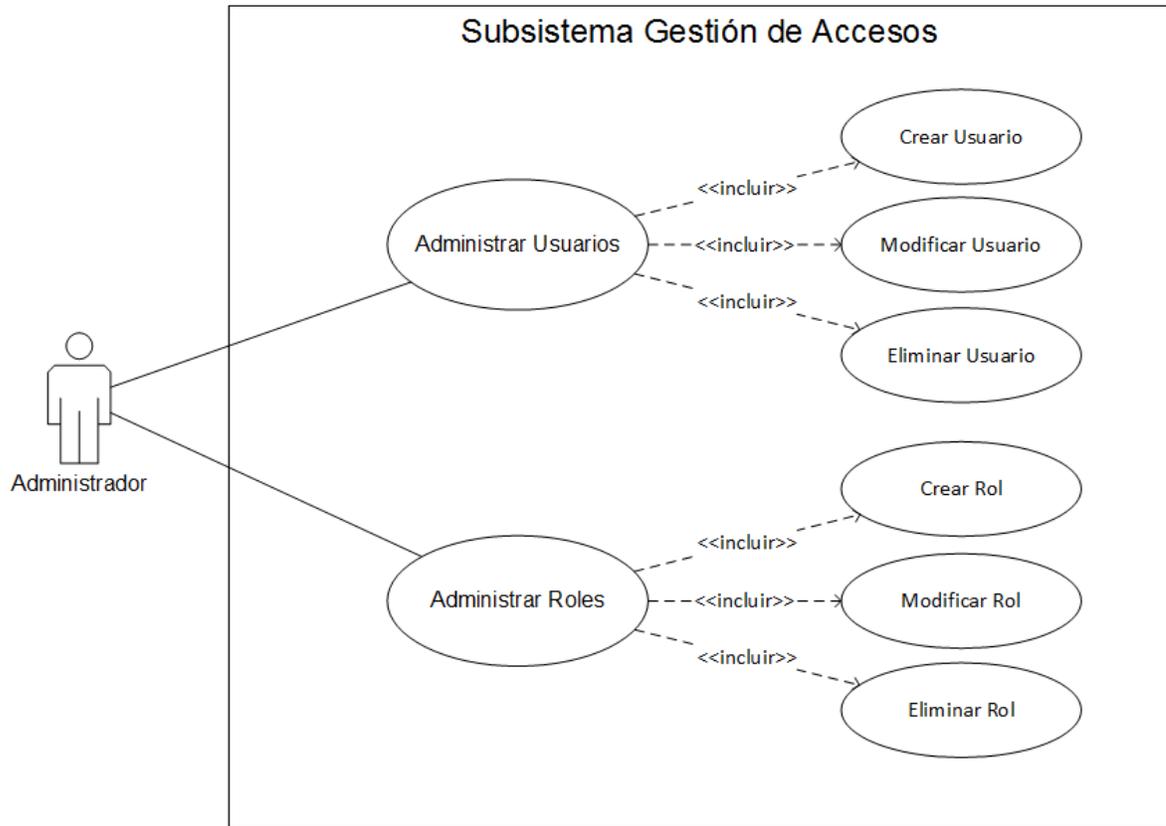
#### 2.4.3.1. Diagrama general (MAVAMSI)

Figura 2.2 - Diagrama de caso de uso "General (MAVAMSI)".



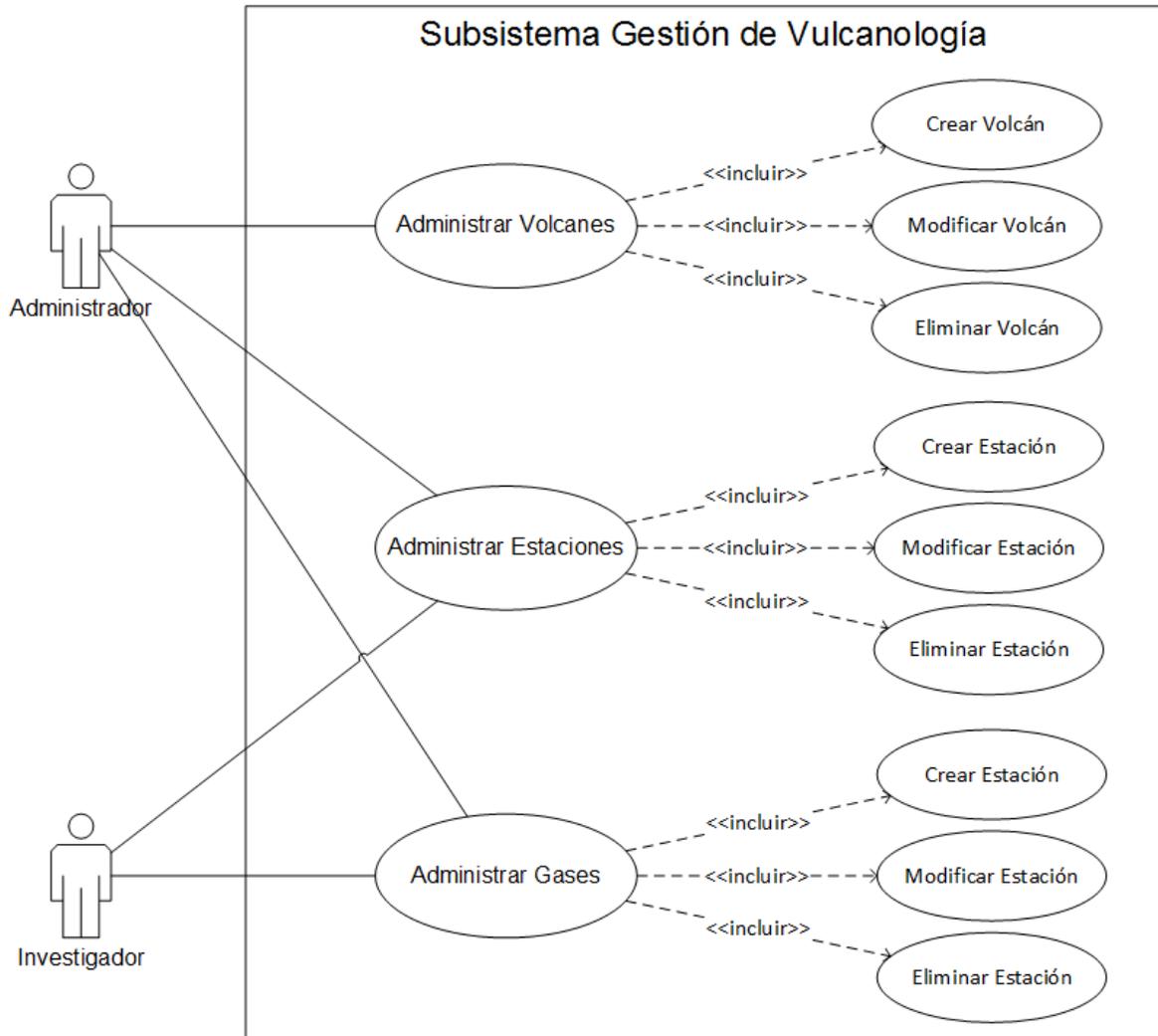
### 2.4.3.2. Subsistema gestión de accesos

Figura 2.3 - Diagrama de caso de uso "Gestión de Accesos".



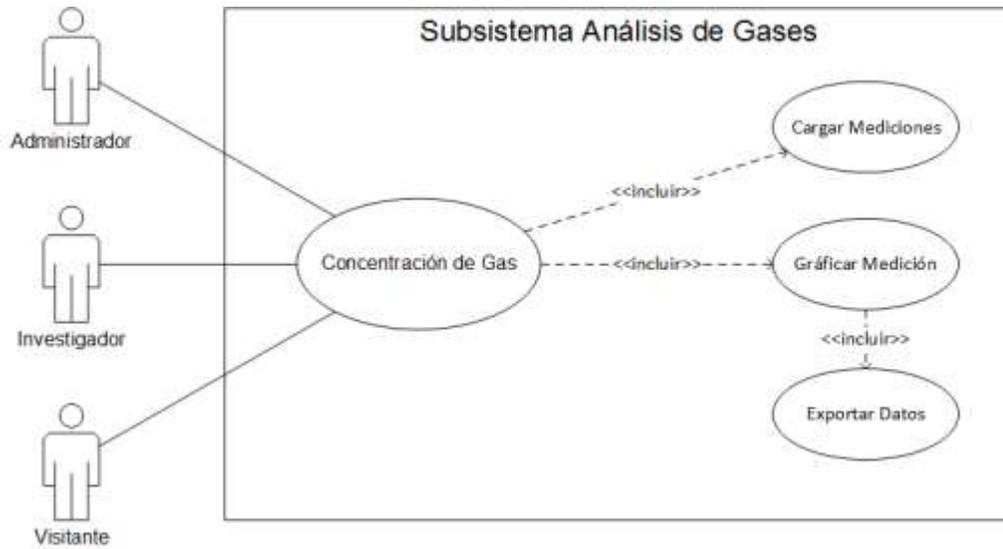
### 2.4.3.3. Subsistema gestión de vulcanología

Figura 2.4 - Diagrama de caso de uso "Gestión de Vulcanología".



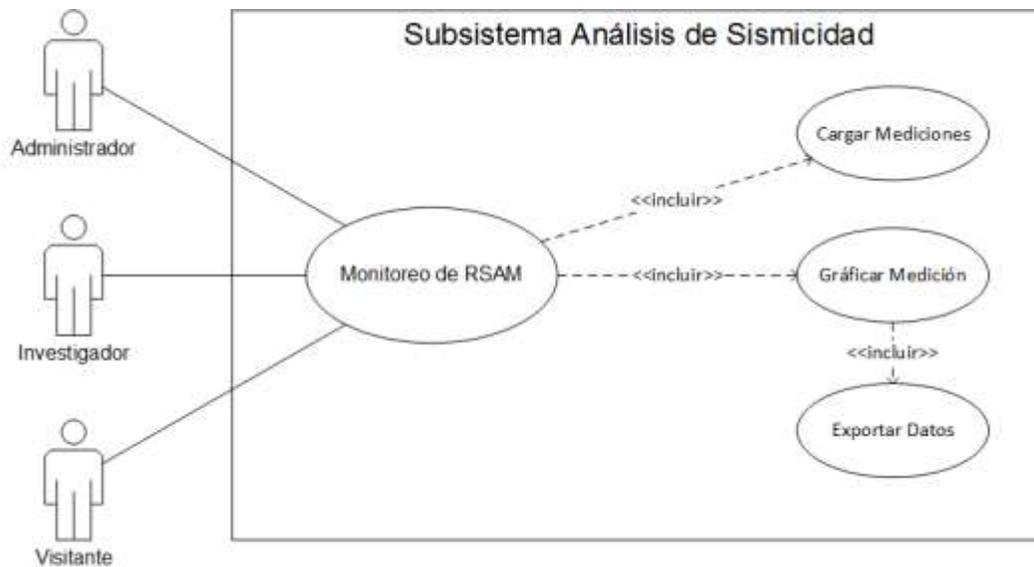
#### 2.4.3.4. Subsistema análisis de gases

Figura 2.5 - Diagrama de caso de uso "Análisis de Gases".



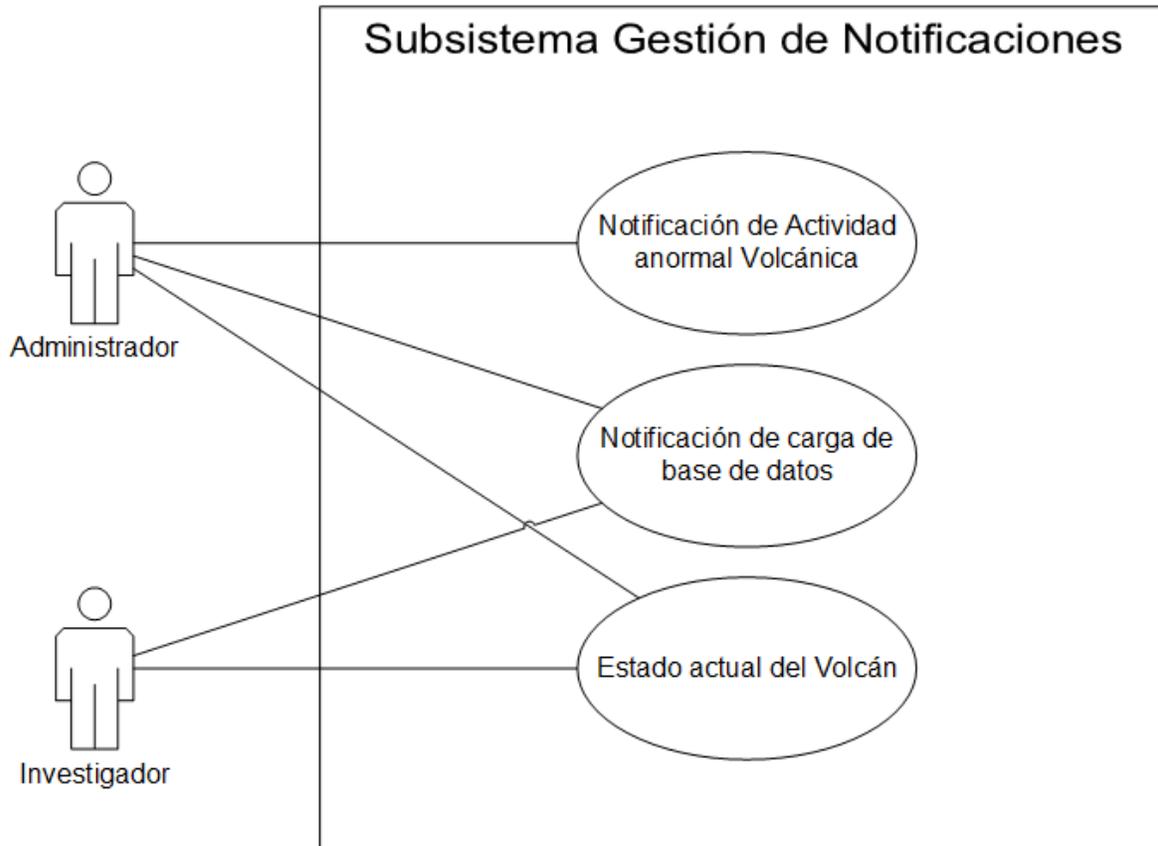
#### 2.4.3.5. Subsistema análisis de sismicidad

Figura 2.6 - Diagrama de caso de uso "Análisis de Sismicidad".



### 2.4.3.6. Subsistema gestión de notificaciones

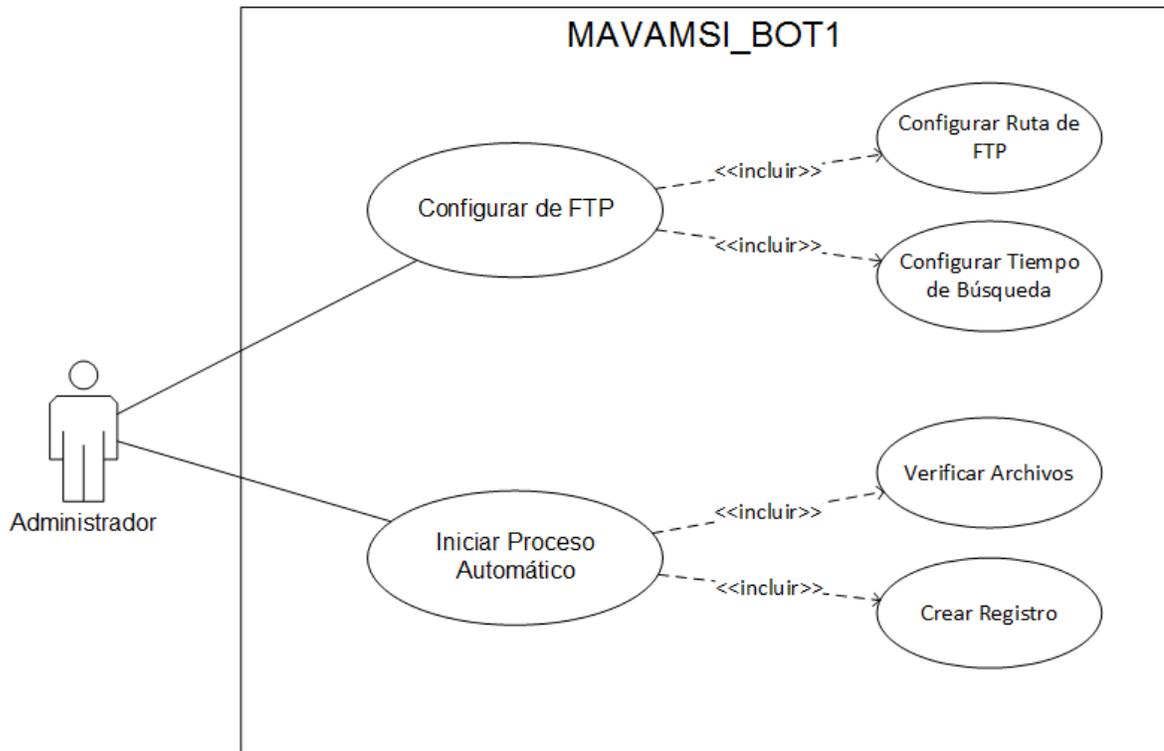
Figura 2.7 - Diagrama de caso de uso "Gestión de Notificaciones"



## 2.4.1. Diagramas de casos de uso (MAVAMSI\_BOT1)

### 2.4.1.1. Diagrama general (MAVAMSI\_BOT1)

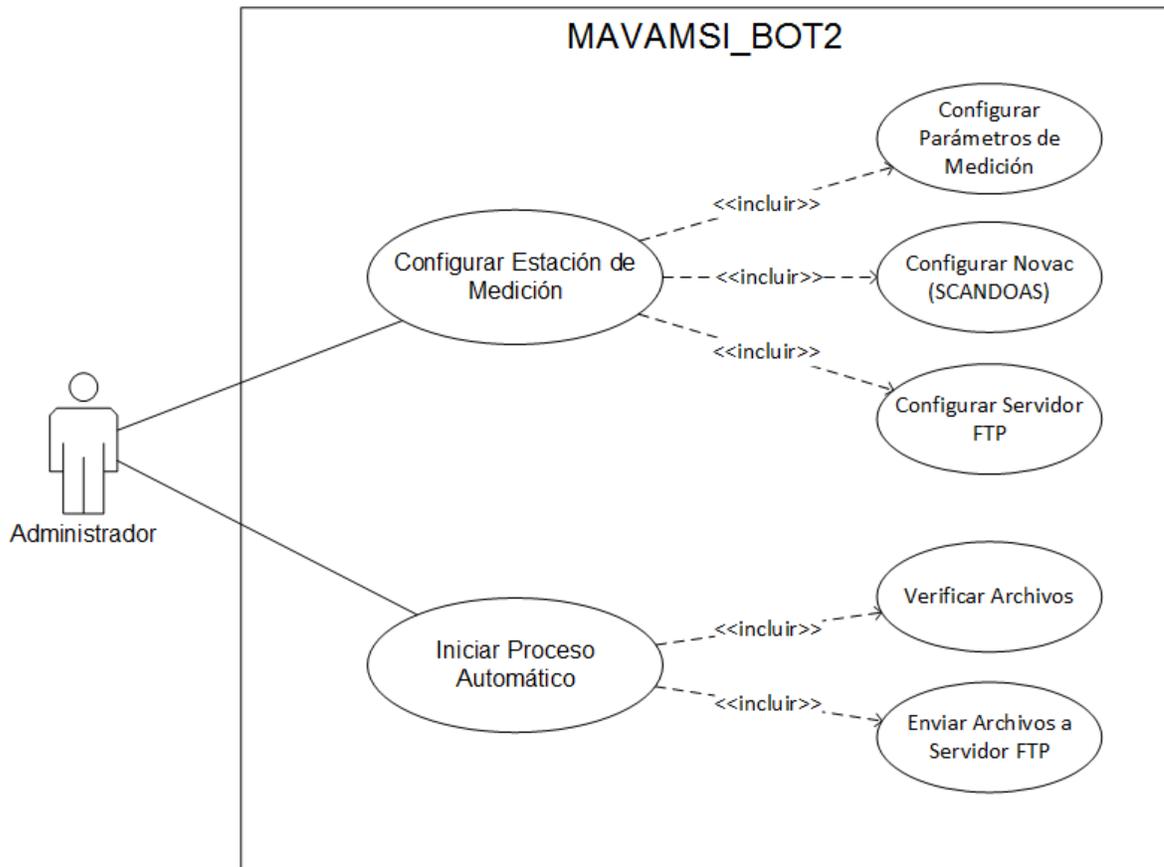
Figura 2.8 - Diagrama de caso de uso "General (MAVAMSI\_BOT1)".



## 2.4.2. Diagramas de casos de uso (MAVAMSI\_BOT2)

### 2.4.2.1. Diagrama general (MAVAMSI\_BOT2)

Figura 2.9 - Diagrama de caso de uso "General (MAVAMSI\_BOT2)".



---

# **CAPÍTULO III: DISEÑO DE LA SOLUCIÓN**

---

### **3. DISEÑO DE LA SOLUCIÓN**

En los capítulos anteriores, dimos a conocer los fundamentos teóricos y los puntos funcionales que busca solucionar nuestro sistema. El presente capítulo, presentará un enfoque meramente técnico, que guíe el desarrollo de una solución que se acople a los requerimientos recopilados en el capítulo anterior. A continuación, se presentarán, enteramente, aspectos relacionados al diseño del sistema dividido en sus diferentes componentes y módulos. Este diseño se ha basado en técnicas y recursos ligados a metodologías del diseño y desarrollo de software.

Iniciaremos, explorando los diferentes componentes de software que compondrán la Arquitectura del Sistema. Debido a que nuestro sistema será el resultado de la extracción y transformación de datos provenientes de diferentes fuentes, será necesario conocer todos y cada uno de los componentes que formarán parte del sistema.

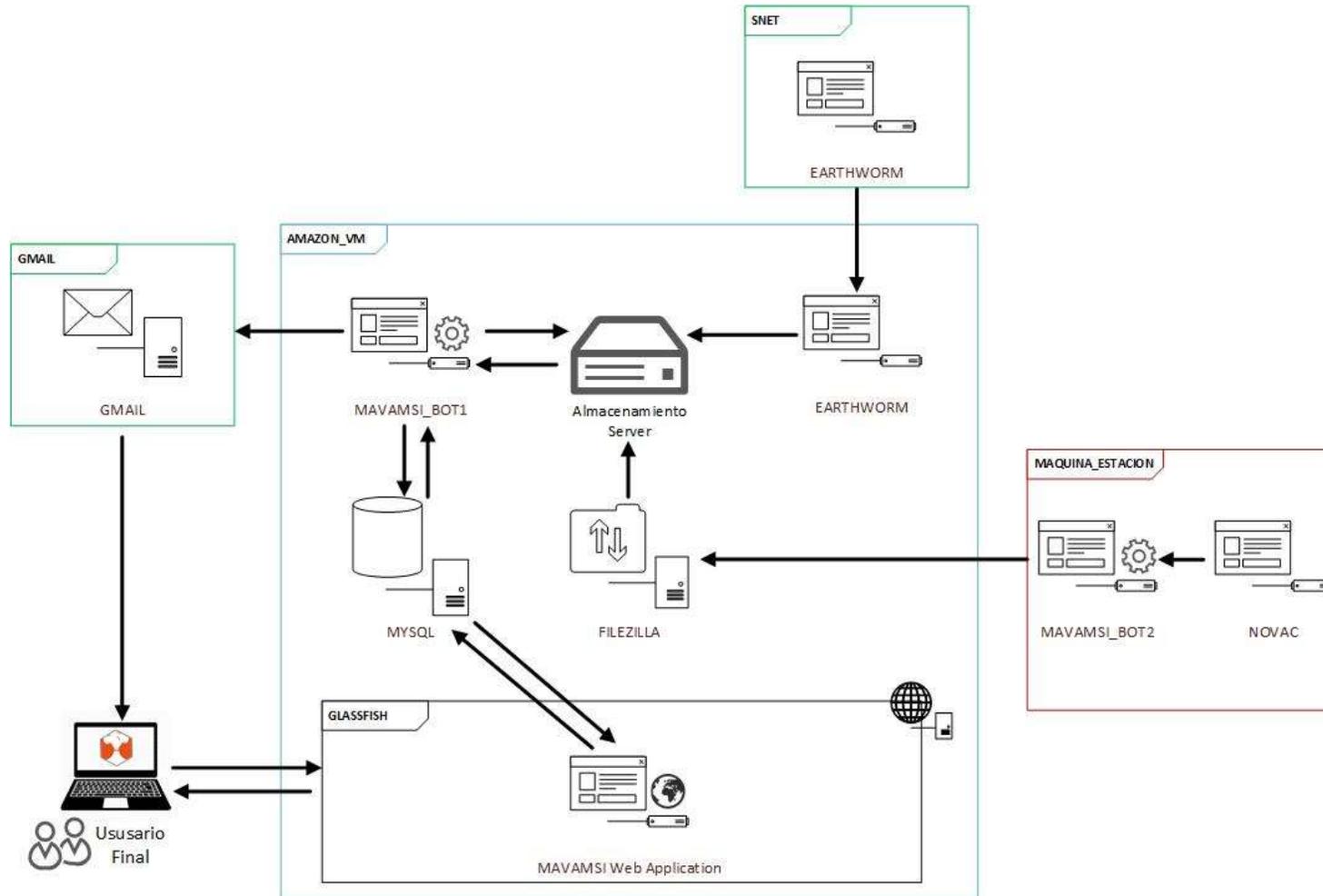
Luego, expondremos la propuesta de diseño base de datos, que tras analizar y comprender las estructuras de los datos que alimentarán nuestra aplicación, se adapta de la forma más eficiente al desarrollo de la solución.

Teniendo claros estos puntos, daremos a conocer los diferentes módulos que contendrá nuestro sistema; los cuales serán posteriormente desarrollados con el fin de obtener datos y mostrar resultados bajo los que se pueda hacer todo tipo de análisis. Se explorarán los módulos a desarrollar, necesarios para que el sistema cumpla con las expectativas decretadas en el capítulo anterior.

Finalmente, mostraremos y explicaremos de forma general, el diseño de interfaces y navegación que poseerá la aplicación; así como los mecanismos de seguridad que implementaremos para garantizar la fidelidad y resguardo de la información.

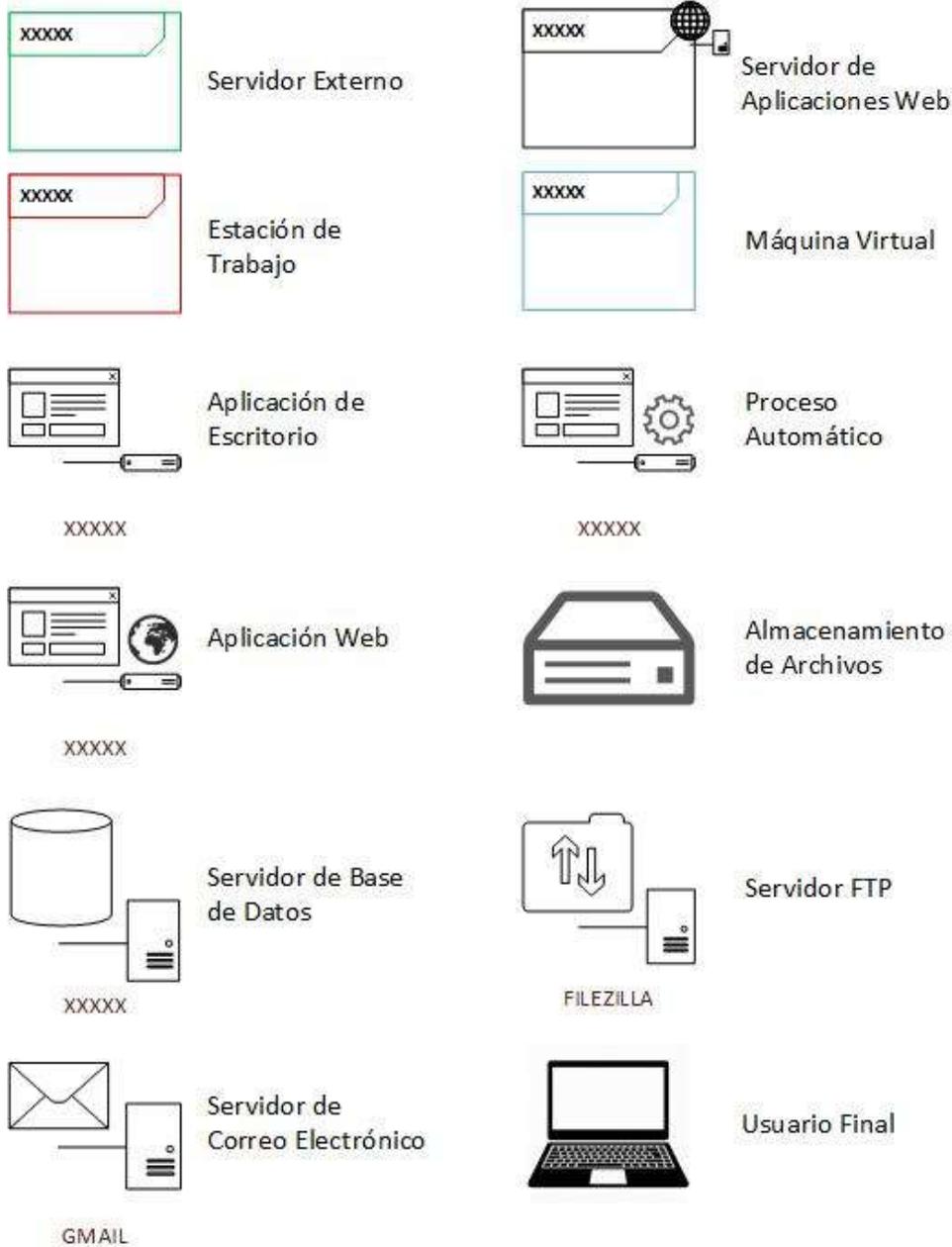
### 3.1. Diseño de Arquitectura de Software

Figura 3.1 - Diagrama de componentes de software



La figura 3.1, muestra una vista completa de todos los componentes que forman parte del sistema propuesto. Como se puede apreciar, el objetivo final de toda la solución, es la creación de una base de datos unificada, que se alimenta de diferentes fuentes de datos; la cual será consumida por cualquier usuario final, desde una aplicación web.

Figura 3.2- Símbolos de diagrama de componentes



La figura 3.2, se detalla la nomenclatura de cada símbolo utilizado en el sistema, y posteriormente la forma de categorizar los diferentes componentes que forman parte del sistema.

El diagrama tiene como objetivo, simplificar la visión general de la aplicación final. El flujo de datos comienza en la máquina destinada a capturar datos de emisión de dióxido de azufre en cualquier estación de monitoreo volcánico. Estos datos, viajan por medio de protocolos FTP hacia la máquina virtual, que alberga los componentes propios de nuestra solución. A su vez, se están capturando datos de movimiento sísmico desde los servidores del SNET, por medio de software de terceros. Al tener toda la información en el almacenamiento físico de la máquina virtual, hay un proceso automático que se encarga de trasladar toda la información a una base de datos centralizada; de la cual, se alimenta la aplicación web visible hacia el usuario final. Para complementar la funcionalidad de la aplicación, existe un servidor de correo electrónico, listo para enviar alertas a usuarios predestinados, en caso de registrar condiciones disfuncionales en los datos obtenidos de movimiento sísmico y/o dióxido de azufre.

Antes de entrar en detalle en cada componente, es necesario aclarar, que podemos catalogar cada pieza de software en 2 categorías distintas:

- Por su desarrollo: Identificando software *in house* de la solución y software de terceros
- Por su ubicación: Identificando software alojado en la máquina virtual de la solución o fuera de ésta.

A continuación, conoceremos a detalle los componentes de software que componen la aplicación. Posteriormente, describiremos el flujo que conlleva la extracción de la información, hasta la presentación de datos. Para entender a detalle el comportamiento de la aplicación, se recomienda no perder de vista el diagrama expuesto en la figura 3.1.

### ***3.1.1. Software In House de la Solución***

A partir de este punto, entenderemos por software in House, a todas aquellas herramientas de software creadas por el equipo de trabajo del presente trabajo de grado; o, cualquier servidor, framework o estación de trabajo que haya sido configurado por el equipo para el correcto funcionamiento de la aplicación como tal.

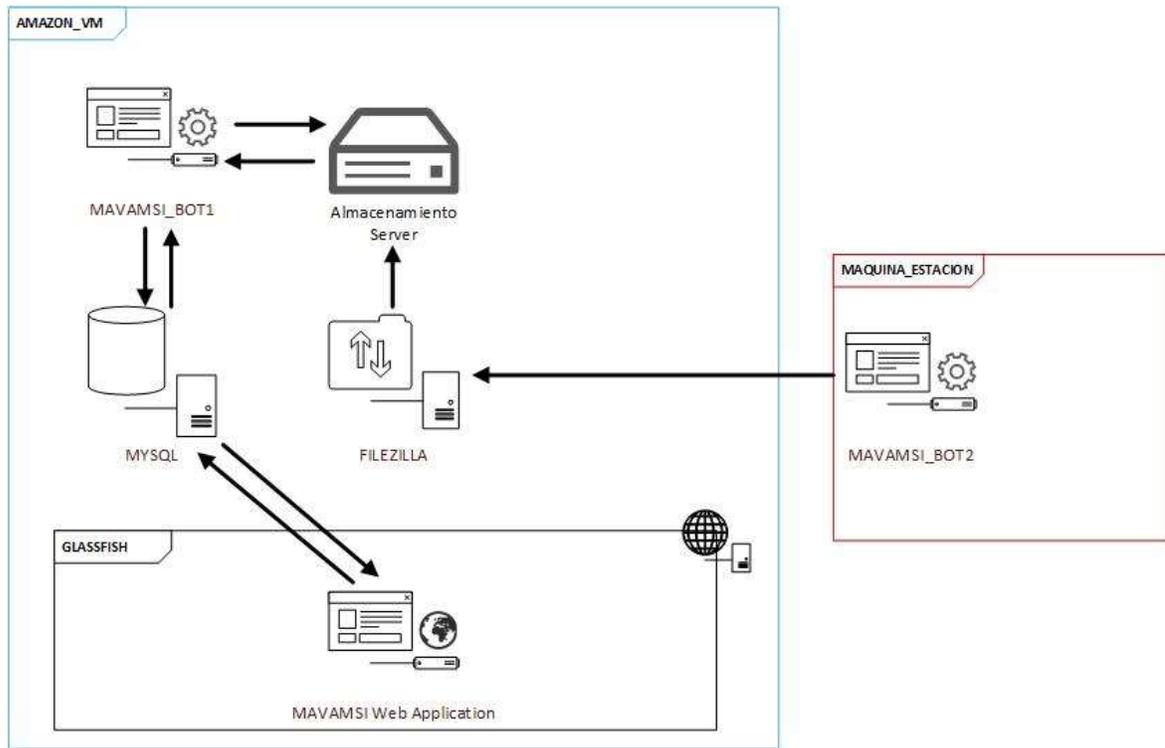
Todos los procesos y aplicaciones diseñados, desarrollados, implementados y configurados por el equipo, entran dentro de esta categoría; y es necesario saber identificarlos, ya que nuestro sistema contará con la implementación de software de terceros, de los cuales, se han tomado aplicaciones ejecutables previamente desarrollados por otras personas.

La figura 3.3, muestra todos los componentes de software que serán desarrollados por el equipo de trabajo de grado.

Quizás el producto de software más representativo de toda la solución sea la aplicación web (MAVAMSI Web Application), ya que será la cara de todo el proyecto ejecutado; sin embargo, detrás de esta aplicación web, hay un proceso de software inmerso que comprende la extracción de datos de las diferentes fuentes de información, la transformación de los mismos, y la carga final a la base de la aplicación.

Antes de entrar en detalle sobre cada componente, es necesario recalcar, que la mayoría de herramientas a utilizar estarán alojadas en una Máquina Virtual. A pesar que, obviamente, el desarrollo de la máquina virtual no corrió por parte del equipo, es considerada como In House, debido a que, para alojar todos los componentes de la aplicación, fue necesario realizar una extensa configuración previa. La máquina virtual, no sólo albergará procesos o aplicaciones, sino también servirá como un medio de almacenamiento de datos, que alimentará el resto de componentes de la aplicación; además dentro de ésta máquina, se instalaran otros servidores, necesarios para el funcionamiento de toda la solución.

Figura 3.3 - Diagrama de componentes de software In House



A diferencia de la figura 3.1, donde teníamos el entorno global de todos los componentes de la solución, en la figura 3.3, vemos plasmados los componentes creados y configurados por el equipo. Todos estos componentes han sido desarrollados con el fin de ajustar el proyecto a los alcances definidos en el capítulo 1 de este proyecto, y de las necesidades plasmadas en el capítulo 2. A continuación entraremos en más detalle de cada componente in house.

### **3.1.1.1. Máquina Virtual (AMAZON\_VM)**

La implementación de nuestro proyecto, requiere de la configuración de diferentes aplicaciones y procesos que se interrelacionen entre sí, que almacenarán archivos que, posteriormente serán compartidos por todos los componentes que forman parte la aplicación, ya sea para transformar, cargar o mostrar los datos al usuario.

Si a todas estas ideas, le sumamos el hecho que todos los componentes in house de nuestra aplicación, se deben comunicar con aplicaciones de terceros, que estarán funcionando en servidores externos; concluimos que un simple web host con alojamiento no será suficiente para albergar todos los componentes de la aplicación.

Por lo general, la implementación de un sitio web convencional, no es más difícil que un par de configuraciones sobre un servidor ya establecido, donde se albergan los archivos necesarios para el funcionamiento del sitio, la base de datos del mismo, y se dispone de un espacio de almacenamiento disponible para la escritura y modificación de archivos, necesarios para el mantenimiento y funcionalidad del sitio. Este es el concepto más tradicional de un Hosting Web. Como podemos notar, este concepto se queda corto para toda la infraestructura necesaria para contener los diferentes elementos que forman parte de nuestra aplicación.

Para implementar nuestra solución, se necesita más que un servidor de aplicaciones web; necesitamos el servidor completo, para poder configurar el resto de aplicaciones que harán funcionar a todo el sistema en conjunto. Necesitamos una computadora que funcione de servidor, que posea disponibilidad y flexibilidad todo el tiempo y en todo momento para realizar las configuraciones necesarias para que todos los componentes del sistema coexistan. Necesitamos un ordenador propio para albergar todos los servidores y aplicaciones necesarios para que funcione nuestro sistema.

De esta forma, tenemos 2 opciones para implementar todos los componentes: Disponemos de una máquina que funcione como servidor, que esté disponible 24/7 y que sea flexible a todas las configuraciones necesarias para la instalación de los diferentes componentes del sistema; o, disponemos de utilizar un servicio de máquina virtual en la nube.

“Una máquina virtual es un software que crea una capa independiente donde se emula el funcionamiento de un ordenador real con todos los componentes de

hardware que necesita para funcionar (disco duro, memoria RAM, tarjetas de red, tarjeta gráfica, etc.) y que puede ejecutar cualquier sistema operativo o programa, tal y como lo haría un ordenador real. Toda esta emulación se encapsula en una serie de archivos que actúan como contendor desde el que se ejecuta la máquina virtual en una ventana de tu ordenador como si de un programa más se tratara y sin que nada de lo que suceda en el interior de esa ventana afecte al ordenador que la ejecuta.

A grandes rasgos, la única diferencia entre tu ordenador real y una máquina virtual que ejecutes en ese mismo ordenador, es que tu ordenador sí cuenta con un hardware real, mientras que la máquina virtual emula todos sus componentes de forma que no tiene por qué corresponderse con el hardware físico que tienes instalado en tu ordenador real.

De ese modo, se pueden ajustar las características del hardware para hacerlo compatible con el sistema operativo que vayas a usar en esa máquina virtual. Es decir, aunque tu hardware real no sea compatible con un determinado sistema operativo, el de una máquina virtual que se ejecute en ese ordenador sí puede serlo.” (Andrés, 2017)

“En informática virtualización es la creación a través de software de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento etc.

Para poder disponer de una máquina virtual se necesita: Un ordenador físico, que puede albergar varias máquinas virtuales; y un programa de software específico llamado con frecuencia hiper visor, Estos programas permiten la creación y la gestión de máquinas virtuales dentro del ordenador físico.” (Microven, 2016)

Debido al grado de disponibilidad y flexibilidad que se necesita de la máquina a utilizar como servidor para la implementación de nuestra aplicación, se ha decidido optar con la contratación de una máquina virtual en la nube.

Utilizar una máquina física para realizar toda la implementación, no se valora como opción, debido a la falta de un equipo designado para este proyecto. Al solicitar un equipo, los tiempos de entrega del proyecto podrían demorar mucho tiempo, además que no garantiza que el equipo cuente con los requerimientos mínimos expuestos en el capítulo anterior del presente trabajo; o que no se permita al equipo de trabajo, realizar las configuraciones e instalaciones necesarias de los diferentes componentes de software del sistema.

No obstante, de contarse con la infraestructura de hardware de un servidor designado para este sistema, y con toda la flexibilidad de configuración del ordenador, el proceso de implementación del sistema será el mismo establecido en el próximo capítulo del presente documento.

Para la implementación de nuestro sistema, se han seleccionado 3 proveedores candidatos de servicios de máquina virtual en la nube: Microsoft Azure, Google Cloud Plataform y Amazon Web Services. Conozcamos un poco a cerca de ellas.

“**Microsoft Azure** es una creciente colección de servicios en la nube integrados que los desarrolladores y los profesionales de TI utilizan para crear, implementar y administrar aplicaciones a través de nuestra red global de centros de datos. Con Azure, obtiene la libertad de crear e implementar donde quiera, utilizando las herramientas, las aplicaciones y los marcos que prefiera.

Implemente en cualquier lugar con su elección de herramientas: Elija cómo implementar Azure (conectando recursos en la nube y locales con funcionalidad de nube híbrida coherente y tecnologías de código abierto) para maximizar la portabilidad y el valor de sus inversiones existentes.

Cree sus aplicaciones, a su manera: Use las herramientas y tecnologías de código abierto que ya conoce y en las que confía, ya que Azure admite una amplia gama de sistemas operativos, lenguajes de programación, marcos, bases de datos y dispositivos.

Conecte datos y aplicaciones locales: Azure ofrece coherencia híbrida en todas partes: en desarrollo de aplicaciones, administración y seguridad, administración de identidades, así como en toda la plataforma de datos.” (Microsoft Azure, 2017)

“**Google Cloud** se refiere al espacio virtual a través del cual se puede realizar una serie de tareas que antes requerían de hardware o software y que ahora utilizan la nube de Google como única forma de acceso, almacenamiento y gestión de datos.

Google ofrece una variedad de servicios basados en la nube. Google Cloud Print permite imprimir desde la web, el escritorio o dispositivo móvil sin la necesidad de un sistema operativo en particular o controladores. En su lugar, envías el documento a cualquier impresora conectada a la nube. Google también ofrece espacio en la nube para desarrolladores de bases de datos SQL para crear aplicaciones, así como para los usuarios de Microsoft Office que deseen editar colaborativamente documentos de Word, PowerPoint y Excel, sin necesidad de software.

Google Cloud Platform te permite centrarte en el futuro de tu empresa. Olvídate de tareas tan engorrosas y costosas como la administración de la infraestructura, el aprovisionamiento de servidores y la configuración de redes. ¿Y para qué? Pues para que los innovadores innoven y los programadores programen.” (Google Cloud, 2017)

“**Amazon Web Services (AWS)** es una plataforma de servicios de nube que ofrece potencia de cómputo, almacenamiento de bases de datos, entrega de contenido y otra funcionalidad para ayudar a las empresas a escalar y crecer. Explore cómo millones de clientes aprovechan los productos y soluciones de la nube de AWS para crear aplicaciones sofisticadas y cada vez más flexibles, escalables y fiables.

La nube de AWS proporciona un amplio conjunto de servicios de infraestructura, como potencia de cómputo, opciones de almacenamiento, redes y bases de datos,

ofertados como una utilidad: bajo demanda, disponibles en cuestión de segundos y pagando solo por lo que utiliza.

La nube de AWS funciona en 44 zonas de disponibilidad dentro de 16 regiones geográficas del mundo, con planes anunciados para crear 14 zonas más y cinco regiones adicionales en China, Francia, Hong Kong, Suecia y una segunda región AWS GovCloud en los EE.UU.” (Amazon Web Services, 2017)

“Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para facilitar a los desarrolladores el uso de la informática en la nube a escala de la Web.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar la capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en cuestión de minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, según cambien sus necesidades. Amazon EC2 cambia el modelo económico de la informática, ya que solo tendrá que pagar por la capacidad que realmente utilice. Amazon EC2 les brinda a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarlas de los casos de error comunes.” (Amazon Web Services, 2017)

Luego de exponer los puntos básicos de cada opción, veamos una comparativa entre los puntos a tomar en cuenta para la decisión de elegir un proveedor de servicios en la nube.

Tabla 3.1 - Comparación entre proveedores de servicios en la nube

Aspectos a considerar	Microsoft Azure	Google Cloud	Amazon Web Services
Precio mensual de máquina requerimientos mínimos	\$118.30	\$188.69	\$79.43
Prueba gratuita	SI	SI	SI
Tiempo de prueba gratuita	1 mes	1 año	1 año
Recursos de Prueba gratuita	2.4Ghz dual core, 4GB Ram, 20 GB SDD	2.0Ghz dual core, 7.5GB Ram, 375 GB SDD	2.4Ghz dual core, 1GB RAM, 100GB SDD
Disponibilidad 24/7	SI	NO	SI
Sistema Operativo	Windows Server 2016	Windows Server 2012	Windows Server 2012 R2
Flexibilidad de configuraciones de red de máquina virtual	NO	NO	SI
Documentación de utilidades	SI	SI	SI
Soporte de servicios, de parte de proveedor	SI	SI	SI
Costo de soporte de servicios	\$0	\$0	\$29.99 (opcional)

La Tabla 3-1, muestra un resumen de los puntos a tener en cuenta, que como equipo dispusimos de mayor importancia para elegir entre un servicio de computación en la nube.

La versión de prueba gratuita, es un punto muy importante, ya que para implementar la aplicación, necesitamos de un servicio que genere la menos cantidad de costos al equipo, puesto que de parte del Departamento de Física de la Universidad de El Salvador FMO, no cuenta con presupuesto disponible para el desarrollo de este proyecto.

Bajo este orden de ideas, no es difícil imaginar que Windows Azure, no será considerado para utilizarse como proveedor. A pesar que presenta grandes cualidades, su versión de prueba de un mes, no compite contra Google o Amazon

que ofrecen hasta un año. Esta decisión, ayuda a acotar las posibilidades a sólo dos opciones;

Tanto Google como Amazon ofrecen un período gratuito de 1 año. La diferencia radica, en que las máquinas virtuales de Google, no ofrecen la flexibilidad necesaria al momento de realizar configuraciones de red. Esto es un punto muy importante, ya que nuestra aplicación necesita de ciertas composiciones en la configuración de red de la máquina virtual, puesto que estará en constante contacto con aplicaciones alojadas en servidores externos. Una mala configuración, o un acceso restringido a modificar las condiciones de red, podrían evitar el correcto funcionamiento de toda la aplicación en conjunto.

Si a lo antes expuesto, le sumamos el hecho que la máquina virtual de Google, sólo permite 700 horas de uso al mes en su versión gratuita, esto significaría que la máquina no contará con disponibilidad 24/7, a diferencia de la máquina de Amazon, que ofrece 750 horas de uso al mes.

Uniéndole a este punto, la versión de sistema operativo más reciente en la máquina de Amazon, consideramos los servicios de Elastic Cloud (EC2) los más beneficiosos para nuestro proyecto, como proveedor de computación en la nube, y que nos permitirá crear la máquina virtual, que almacenará todos los componentes in house de nuestra aplicación.

Ahora que hemos comprendido donde estará alojada nuestra aplicación, cómo funciona la tecnología de computación en la nube, porque elegimos a Amazon como proveedor de máquina virtual y sobre todo, cuál será el papel de ésta máquina en la aplicación, podemos pasar a conocer a detalle cada uno de los componentes que integran el sistema informático propuesto.

### **3.1.1.2. Procesos automáticos**

La figura 3.3, nos mostraba la distribución de todos los elementos in house, que componen el sistema que se presenta como solución. En dicho diagrama, se puede apreciar la existencia de diversos componentes de software que ayudan con el completo funcionamiento de la aplicación. Entre esos componentes, tenemos un par de procesos automáticos.

Los procesos automáticos, no será más que aplicaciones de escritorio que nos ayudarán a automatizar los procesos de extracción, transformación y carga de datos en nuestra base de datos.

Una aplicación de escritorio “es la aplicación creada para ejecutarse en un ordenador de escritorio, sobre un sistema operativo de interfaz visual como Windows o Linux. Será un programa el encargado de realizar la funcionalidad del software implementado que instalaremos en cada puesto de trabajo y se conectará a través de Internet con la base de datos. La principal ventaja de este sistema será la rapidez de uso ya que podremos incorporar todos los controles de escritorio y todos los eventos asociados a ellos.” (Desarrollo Web, 2017)

Nuestro sistema, de cara al usuario es una aplicación web, que puede ser desplegada desde cualquier navegador; sin embargo, constará de dos aplicaciones de escritorio que fungirán como procesos de transformación, extracción y carga de datos (ETL) hacia la base.

“ETL son las siglas en inglés de Extraer, Transformar y Cargar (Extract, Transform and Load). Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

Los procesos ETL también se pueden utilizar para la integración con sistemas heredados (aplicaciones antiguas existentes en las organizaciones que se han de

integrar con los nuevos aplicativos, por ejemplo, ERP's. La tecnología utilizada en dichas aplicaciones puede hacer difícil la integración con los nuevos programas).

Las herramientas ETL, deberían de proporcionar, de forma general, las siguientes funcionalidades:

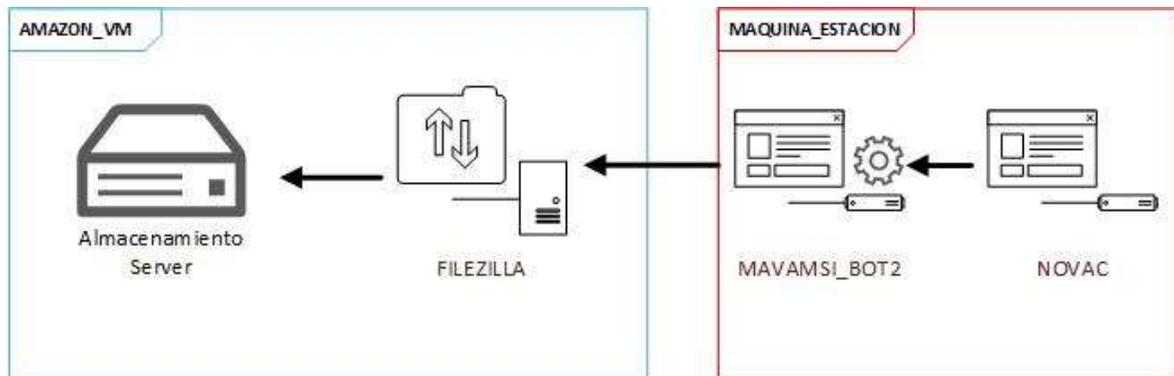
- Control de la extracción de los datos y su automatización, disminuyendo el tiempo empleado en el descubrimiento de procesos no documentados, minimizando el margen de error y permitiendo mayor flexibilidad.
- Acceso a diferentes tecnologías, haciendo un uso efectivo del hardware, software, datos y recursos humanos existentes.
- Proporcionar la gestión integrada de la base existente.
- Uso de la arquitectura de metadatos, facilitando la definición de los objetos de negocio y las reglas de consolidación.
- Acceso a una gran variedad de fuentes de datos diferentes.
- Manejo de excepciones.
- Planificación, logs, interfaces a schedulers de terceros, que nos permitirán llevar una gestión de la planificación de todos los procesos necesarios para la carga de la base.
- Interfaz independiente de hardware." (Dataprix, 2010)

En base a lo narrado anteriormente, podemos tener una idea clara de lo que hará cada uno de los procesos automático que necesita nuestro sistema. Su función se limitará a extraer los datos de las fuentes de datos, los cuales no serán in house, sino aplicaciones de terceros que nos proveerán información; luego transformar estos datos, a un formato estándar, que nos permita tener un control de lo que estamos recibiendo; y finalmente, la carga de la información en nuestra base de datos.

Debido a las necesidades de procesamiento de información, será necesaria la creación de dos procesos automáticos que fungirán como ETL de nuestro sistema.

El primero de ellos, será un proceso básico, que tendrá la responsabilidad de trasladar la información obtenida por el sensor de dióxido de azufre, hacia el almacenamiento interno de la máquina virtual que alojará la aplicación. Hemos tenido a bien nombrar este proceso como MAVAMSI\_BOT2. La siguiente figura, muestra cómo se trasladará la información por medio de este proceso.

*Figura 3.4 - Diagrama de funcionamiento de MAVAMSI\_BOT2*



La fuente de datos para este proceso, será el resultado arrojado de las mediciones de dióxido de azufre, las cuales, son extraídas del sensor por medio de la aplicación NOVAC. Ésta aplicación, arroja resultados en formato de texto, los cuales quedan alojados en el almacenamiento local de la máquina de la estación de mediciones. El proceso MAVAMSI\_BOT2, se encargará de transportar los datos arrojados por NOVAC hacia la máquina virtual que almacenará el resto de nuestra aplicación.

El traslado de datos, será realizado por medio del protocolo FTP. FTP “es un protocolo estándar de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. FTP está pensado para ofrecer la máxima velocidad en la conexión, la cual se protege mediante el uso de usuarios y contraseñas que administren el acceso a la información que se aloja en el servidor.” (Internet Engineering Task Force, 2014)

Una vez los datos llegan a la máquina virtual, es tarea del servidor FTP, recibir los archivos enviados por MAVAMSI\_BOT2, y finalmente, mantener disponibles estos archivos en el almacenamiento interno de la máquina virtual, para que entre en acción el otro proceso automático.

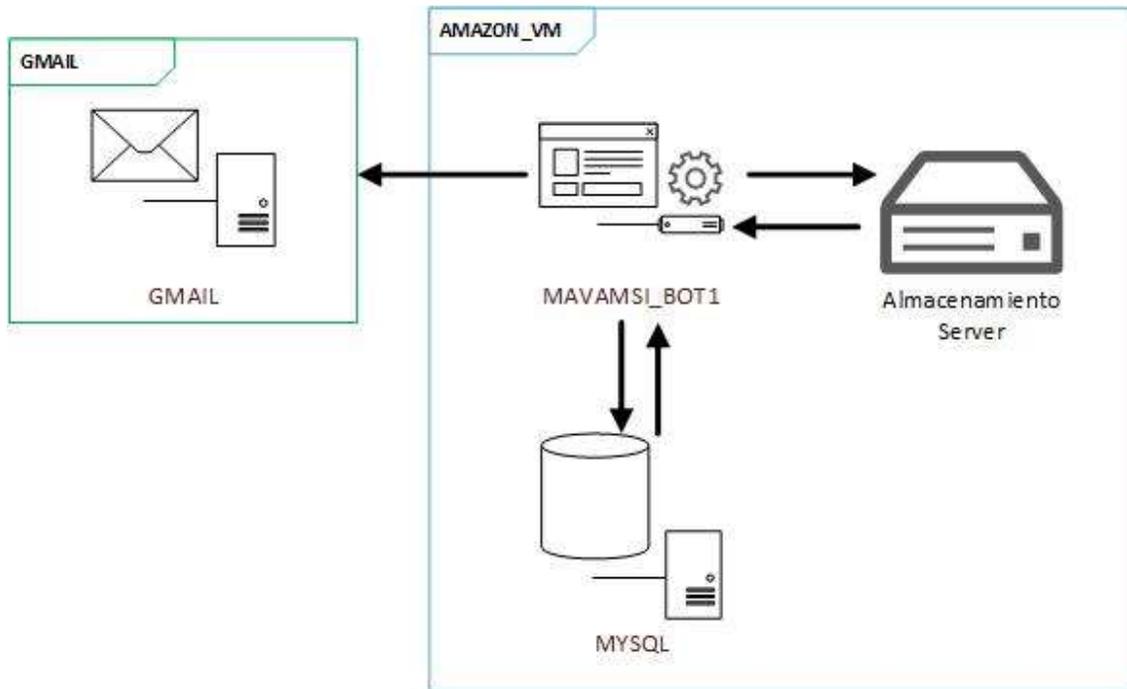
La principal labor de extracción, transformación y carga de datos, será tarea del proceso MAVAMSI\_BOT1. Este robot, será un ETL como tal, que se encargará de extraer los datos tanto de dióxido de azufre, como de RSAM, provenientes de las fuentes de datos.

La figura 3.5, muestra el flujo de información para este robot. MAVAMSI\_BOT1, extraerá toda la información proveniente del almacenamiento interno de la máquina virtual, para posteriormente, transformarlo a un formato y estructura de datos que sea útil para representar la información y que sea capa de cargar la data hacia la base del sistema.

Las funciones de MAVAMSI\_BOT1 serán las siguientes:

- Extracción, transformación y carga en la base de datos de la aplicación, de datos de dióxido de azufre.
- Extracción, transformación y carga en la base de datos de la aplicación, de datos de movimiento sísmico (RSAM).
- Envío de alertas por correo electrónico.

Figura 3.5 - Diagrama de funcionamiento de MAVAMSI\_BOT1



La fase de transformación de este ETL, juega un papel muy importante en todo el funcionamiento de la aplicación. En esta fase, se realiza todo tipo de validaciones sobre la data proveniente de las fuentes de datos, para evitar albergar en base, cualquier información incorrecta o que ensucie los análisis. Estas transformaciones, ayudan a poder tener una idea de los datos que vienen de las fuentes, lo cual hace correcto dejar la parte de envío de alertas en este proceso.

Ya que MAVAMSI\_BOT1 es el encargado de transformar la data, previo a la inserción en la base, en este punto es fácil identificar si existe alguna anomalía en el comportamiento habitual de un volcán. Si en la fase de validación de los datos, el robot identifica medidas fuera de los rangos aceptables, tanto de dióxido de azufre como de movimiento sísmico, este robot será el encargado de lanzar la señal al servidor de correos electrónicos, para poder enviar alertas a los usuarios de la aplicación.

Generalizando, la tarea primordial y más importante de MAVAMSI\_BOT1, será poblar la base de datos de nuestra aplicación, con información validada fidedigna y en formatos fáciles de representar al usuario final.

### ***3.1.1.3. Aplicación Web***

En el apartado anterior, conocimos sobre la importancia que juegan los procesos automáticos (aplicaciones de escritorio) en la extracción, transformación y carga de datos hacia la base del sistema. Estos procesos se encargan de poblar la base de la aplicación, con datos que los usuarios puedan considerar de calidad. Pero una vez cargados los datos, ¿Cómo podremos consumirlos?

Para nuestro sistema, utilizaremos una aplicación web, que se encargue de consultar la información a la base y poder representarla por medio de gráficos y tablas que muestren a un usuario en específico cómo se está comportando un volcán determinado, analizando sus condiciones de dióxido de azufre y movimiento sísmico.

“Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Una aplicación web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos

diversos y acceder a gestores de base de datos de todo tipo.” (Universidad de Alicante, 2006)

Utilizar aplicaciones web, presenta verdaderas ventajas sobre utilizar una aplicación de escritorio para la presentación de datos, las cuales nos incentivaron como equipo a optar por desarrollar el sistema en este entorno. Dichas ventajas son:

- “Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.

- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas (RIAs).” (Universidad de Alicante, 2006)

Existen numerosos lenguajes de programación empleados para el desarrollo de aplicaciones web en el servidor, entre los que destacan:

- “PHP
- Java, con sus tecnologías Java Servlets y JavaServer Pages (JSP)
- Javascript
- Perl
- Ruby
- Python
- HTML
- XML
- ASP/ASP.NET, aunque no es un lenguaje de programación en sí mismo, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET o VBScript/JScript para ASP).

Se utilizan para servir los datos adecuados a las necesidades del usuario, en función de cómo hayan sido definidos por el dueño de la aplicación.” (Universidad de Alicante, 2006)

Al elegir una aplicación de escritorio para representar los datos, estamos cumpliendo parte de los requerimientos no funcionales expuestos en el capítulo anterior del presente trabajo; entre los que destacan que la aplicación sea

responsiva, que tenga disponibilidad todo el tiempo y que los tiempos de consulta sean rápidos de cara al usuario.

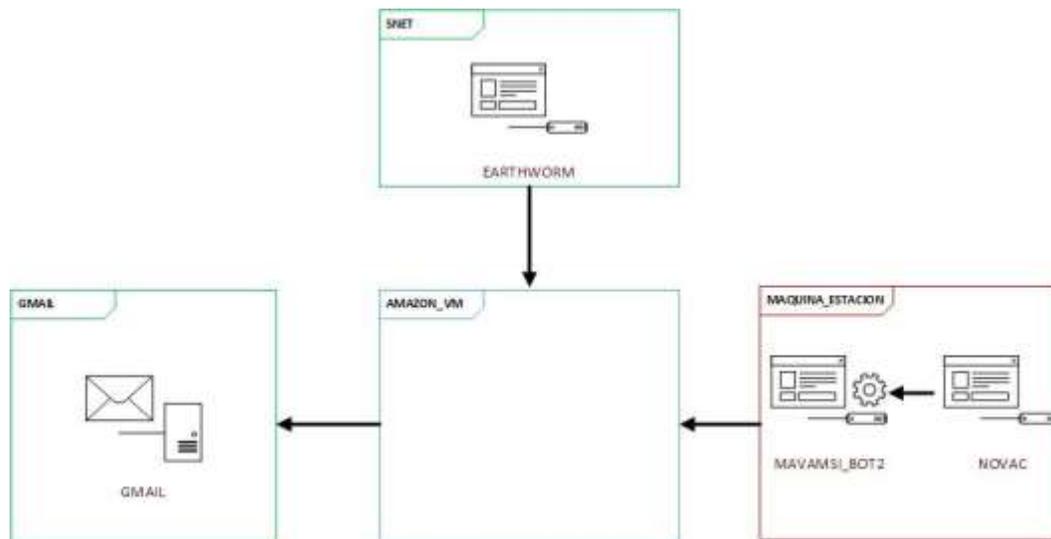
Por las ventajas expuestas, y por lo útil que será para suplir los requerimientos no funcionales de nuestro proyecto, desarrollar una aplicación web para consumir los datos de la base, es la mejor opción para nuestro proyecto.

### 3.1.2. *Software de Terceros*

#### 3.1.2.1. *Servidores de terceros*

La figura 3.1, nos mostraba la pintura general de todos los componentes que conformarán nuestra aplicación. Como podremos ver, la mayor parte del procesamiento de datos surge efecto en el servidor in house de la solución, donde se recopilan los datos de las diferentes fuentes; sin embargo la generación de datos como tal, surge efecto en servidores externos. Por lo tanto, conocerlos es de suma importancia para entender desde donde fluye la información.

Figura 3.6 - Diagrama de servidores de terceros



La Figura 3.6, nos muestra de forma general, los servidores implicados en nuestro sistema. Tal como lo indicaba la Figura 3.2, las cajas en color verde, representan servidores externos a la máquina virtual del sistema; y la caja en color

rojo, es una estación de trabajo, también externa, que será una computadora personal encargada de recibir datos procedentes del sensor de dióxido de azufre y enviarlos a la máquina virtual de la solución. Los tres elementos que rodean la máquina virtual en la figura, serán considerados como servidores de terceros.

La caja marcada en rojo, hace referencia a una estación de trabajo cualquiera, la cual, puede ser una computadora personal, una laptop o una desktop, que tenga un sistema operativo capaz de ejecutar NOVAC Program y que tenga acceso a internet de forma permanente.

Esta máquina será la responsable de comunicar el sensor de dióxido de azufre con el resto del sistema. La función de esta máquina, será obtener los datos arrojados por el sensor y leerlos por medio de NOVAC Program, luego, por medio del proceso MAVAMSI\_BOT2, enviará los datos por medio de archivos hacia la máquina virtual del sistema por medio del protocolo FTP; es por esto que necesitamos que la máquina cuente con acceso a internet de forma permanente.

La máquina designada como MAQUINA\_ESTACION, se considera un servidor, porque debe estar disponible en todo el tiempo de funcionamiento del sensor. El sensor de dióxido de azufre, tiene un tiempo efectivo de mediciones entre 6AM y 6PM, por lo que en este rango de horas, la máquina debe presentar disponibilidad y conexión estable a internet.

Volviendo a la figura 3.6, en la parte superior de la máquina virtual, se conecta con el servidor nombrado SNET. Como tal, este servidor se encuentra alojado dentro de la infraestructura informática del SNET, y nuestra relación hacia él, será por medio del software Earthworm. Este software, funciona como un almacenador de eventos de movimiento sísmico, a nivel mundial, en el cual, se pueden compartir mediciones de RSAM a lo largo del mundo. El software funciona bajo la arquitectura de cliente-servidor, por lo que para poder comunicarnos con el servidor del SNET, es necesario instalar una instancia de Earthworm en nuestra máquina virtual.

La comunicación entre nuestra instancia de Earthworm y la del SNET, será de forma unidireccional: nuestra instancia funcionará como un cliente que estará a la espera de los datos enviados por la instancia servidor del SNET. Nuestra instancia del software, se encargará de guardar los archivos recibidos en el almacenamiento interno del servidor de la solución, para que posteriormente, la data sea ingresada a la base de datos.

La tercera caja de la figura 3.6, muestra el servidor externo de Gmail. Este servidor tendrá una función clara y sencilla: Enviar alertas por medio de correo electrónico a los usuarios de determinado volcán, cuando éste presente un comportamiento anormal.

El disparador del evento que se comunique con este servidor será el proceso automático MAVAMSI\_BOT1, el cual, verifica la data antes de ser ingresada a la base. El robot, se encarga de enviar la lista de distribución de alertas, así como el mensaje del cuerpo del correo, y ya es el servidor de Gmail el que se encarga de distribuir el correo.

Se eligió este servidor para ser utilizado en el proyecto, porque Gmail es el líder de gestor de correos electrónico en la actualidad. Además, presta este tipo de herramientas para desarrolladores de forma gratuita permitiendo el envío de mensaje por correo electrónico a cualquier dirección de correo; incluso, cuando las direcciones de destinatarios se encuentran fuera del dominio de Gmail. Todo esto, por medio de su conjunto de servicios llamados G Suite.

“G Suite es un conjunto de aplicaciones inteligentes – Gmail, Docs, Drive, Calendario, Hangouts, y más – diseñado para unir a las personas, permitiendo la colaboración en tiempo real desde el principio. Pero eso no es todo, ya que hay mucho más en camino. Y es que desde las oficinas de Google tienen muy claro que cuando las personas se conectan y pueden trabajar juntos, se obtienen unos mejores y más rápidos resultados para competir en el complicado y saturado mercado actual.

Con el cambio de nombre, desde Google han aprovechado también para introducir ciertas novedades bastante interesante en sus aplicaciones de ofimática más conocidas. De este modo, Google Drive, por ejemplo, cuenta con un nuevo apartado que utiliza la inteligencia artificial para mostrarnos documentos que podemos necesitar editar, mientras que la función de las metas de Google Calendar ha empezado a desplegarse también en el sistema operativo móvil iOS.

De este modo, con el espíritu de trabajo en equipo, Google se ha asociado con algunos de los mejores ilustradores de todo el mundo para llevar sus ideas a la vida con los mejores resultados y la mayor creatividad posible. Y es que gracias a G Suite, la información podrá fluir libremente entre dispositivos, aplicaciones, personas y equipos, por lo que las grandes ideas nunca se quedarán en el olvido.” (Google Cloud, 2016)

“Puedes configurar una impresora, un escáner, un fax o una aplicación de tu empresa para enviar correos electrónicos a través de G Suite. En función de cuáles sean tus requisitos, tendrás estas opciones:

- Servicio de retransmisión SMTP de G Suite: te permite enviar correos desde tu organización mediante la autenticación de las direcciones IP. Puedes enviar mensajes a cualquier persona, tanto dentro como fuera de tu dominio.
- Servidor SMTP de Gmail: puedes enviar correos a cualquier usuario, tanto dentro como fuera de tu dominio. Para utilizar esta opción deberás autenticarte con tu cuenta de Gmail o G Suite y tu contraseña.
- Servidor SMTP de Gmail restringido: puedes enviar mensajes solo a usuarios de Gmail o G Suite. Para utilizar esta opción no es necesario autenticarse.” (Google Support, 2017)

“Un usuario de G Suite registrado puede retransmitir mensajes a un máximo de 10.000 destinatarios al día. Para ver todos los límites de retransmisión SMTP,

consulta el artículo Límites de envío del servicio de retransmisión SMTP.” (Google Support, 2017)

### **3.1.2.2. Aplicaciones de terceros**

Volviendo a analizar el diagrama de componentes de la solución en la Figura 3.1, podemos ver un par de aplicaciones que forman parte del flujo de datos, las cuales han sido desarrolladas por terceros. Este apartado busca aclarar, que estas aplicaciones han sido creadas por entidades que no tienen relación con el presente proyecto, por lo tanto, el desarrollo de estas aplicaciones queda fuera de los alcances de nuestro proyecto, y únicamente estará a nuestro cargo la adaptación y configuración de los mismos para apegarlo al flujo de datos de la solución.

En este punto, solo aclararemos que estas aplicaciones fueron desarrolladas por terceros y no por el actual equipo de trabajo. La forma en la que se usarán como fuentes de datos, será descrita en el siguiente apartado “Fuentes de Datos”; ya que cada uno de los aplicativos se apega a una fuente de datos para nuestro sistema.

El primer aplicativo es NOVAC Program. Novac Program fue desarrollado a manos de la Red de Observación de Cambio Volcánico y Atmosférico, NOVAC (por sus siglas en inglés Network for Observation of Volcanic and Atmospheric Change). El software fue desarrollado en 2009 con el objetivo de tener una aplicación que se encargara de decodificar los datos obtenidos por los sensores ScanDoas, utilizados por excelencia como instrumentos de medida de emisiones de dióxido de azufre.

A pesar que el sistema quedó desactualizado en 2010, sigue siendo el software utilizado por defecto, para las estaciones alrededor del mundo que utilizan sensores de tipo ScanDoas para realizar mediciones de gases.

La segunda aplicación desarrollada por terceros, utilizada en nuestro sistema es Earthworm. Earthworm es un sistema de software de código libre, que inició sus desarrollos a principios de los años 90 por parte de la Corporación de Tecnologías Instrumentales de Software, ISTI (por sus siglas en inglés de Instrumental Software

Tecnologies, Inc.). Esta aplicación nació como una inyección de una red, a nivel mundial, de sismología; que pretendía intercomunicar estaciones de observación de volcanes, para crear estándares de medición que definen comportamientos de volcanes.

Earthworm cuenta con el cálculo de RSAM más aceptado a nivel mundial por organizaciones de vulcanología y sismología, que se basa en el espectro de frecuencias (decibeles) en función de la profundidad y masa del material magmático del volcán. Actualmente, la relación frecuencia-tiempo utilizada por Earthworm, se ha convertido en el estándar mundial para el cálculo de RSAM por las organizaciones de observación volcánica en Europa, Asia y Oceanía.

Earthworm es el sistema implementado en el SNET, para la distribución y recopilación de datos de movimiento sísmico, a lo largo de todas las estaciones de observación volcánica de nuestro país. Por medio de Earthworm, se obtienen los datos de los sensores y dispositivos de percepción sísmica desplegados en los volcanes (tanto activos como inactivos) del país.

En base a esto, recaemos en la necesidad de desplegar una instancia de Earthworm en nuestra máquina virtual, para poder comunicarnos con la instancia servidor instalada en los servidores del SNET.

Además, el SNET no comparte información de movimiento sísmico, fuera de los formatos establecidos por Earthworm, ya que es un estándar de archivo utilizado por la entidad para el control de movimiento sísmico. Por lo que solicitar la información fuera de la aplicación, no era una opción para el proyecto.

Al ser Earthworm un sistema de código libre, y tener documentación oficial publicada en internet, hace factible su despliegue en nuestro sistema, para apegarnos al estándar de software utilizado a nivel mundial.

### ***3.1.3. Fuentes de Datos***

Ahora que conocemos todos los elementos que formarán parte del sistema propuesto, ya tenemos una idea del flujo de datos que recorrerá a lo largo de los elementos, desde su extracción hasta su presentación en aplicación web, pasando por su alojamiento en la base; sin embargo, hasta este punto no hemos aprendido cómo se originan los datos.

Antes de entrar en detalle, cabe mencionar que nuestro sistema, no es ninguna aplicación transaccional, es decir, que los datos no se generan directamente en nuestro sistema; a diferencia de lo que sería un sistema de transacciones bancarias, o un sistema de registro de ventas para una tienda, o un sistema de registro de notas para una entidad educativa. En todos estos ejemplos, los datos se originan directamente en el sistema, y es por medio de un conjunto de vistas que la información se almacena en la base de datos, para posteriormente ser consultada siempre dentro de la misma aplicación por medio de reportes, gráficos o resúmenes.

La naturaleza de nuestro sistema, se basa en la recopilación de datos de diferentes fuentes: una para emisiones de dióxido de azufre y otra para movimiento sísmico; transformarla bajo criterios necesarios para su correcta representación, y cargarla a la fuente; haciendo ver que nuestra aplicación no funcionará como un sistema transaccional, sino, como un Data Warehouse. En el siguiente apartado de “Diseño de Base de Datos” ahondaremos este término.

Dicho esto, nuestra aplicación no generará datos, sino que se limitará a recopilarlos.

En vista de esto, a continuación describiremos a los generadores de datos, cómo es el formato de salida de los datos de cada fuente, y qué información será indispensable almacenar en la base de nuestra aplicación.

### ***3.1.3.1. NOVAC Program y Emisiones de Dióxido de Azufre***

Las mediciones de emisiones de dióxido de azufre, representan un reto importante para el equipo de trabajo de este proyecto. Las mediciones de este gas, serán hechas por medio de un dispositivo llamado ScanDoas. Este dispositivo es un fonículo con un lente sensor de rayos ultravioleta, el cual, revisa el espectro en un radio de 360 grados.

Para obtener los datos obtenidos por el escáner, el ordenador al que está conectado el dispositivo debe tener instalado el software Novac Program. Éste se encarga de decodificar los resultados obtenidos por el escáner a un archivo de texto plano (formto .txt).

Novac (Network for Observation of Volcanic and Atmospheric Change), fue un proyecto fundado por la Unión Europea entre 2005 y 2010, el cual, tuvo como propósito establecer una red mundial (primeramente sólo dentro de Europa) para medidas de emisiones de gases (específicamente SO<sub>2</sub>) y aerosoles de volcanes; y utilizar esta información para implementar planes de acción ante riesgos, y fomentar la investigación vulcanológica.

A pesar que el proyecto se disolvió a finales de marzo de 2010, ha dejado de legado una serie de herramientas tanto de hardware como de software, que hoy en la actualidad siguen siendo utilizadas para el monitoreo volcánico a nivel mundial. Una de estas herramientas de software, es el Novac Program.

“El software desarrollado incluye aplicaciones para ejecutar los instrumentos y almacenar datos localmente en los instrumentos, software ubicado en los observatorios de volcanes para descargar datos de los instrumentos y realizar una evaluación de flujo de dióxido de azufre en tiempo real, software para post-procesamiento de datos y software para archivar los datos.

Cada instrumento es controlado por un equipo incorporado que ejecuta software Linux. El ordenador ejecuta las mediciones, almacena los datos sin procesar en forma comprimida en una tarjeta SD de 1 Gb (suficiente para 3 meses de funcionamiento autónomo) y tiene un ftp-server para la comunicación con el observatorio de volcanes vía LAN.

Para evaluar los datos recogidos por los instrumentos de exploración, se ha desarrollado un paquete de software denominado NovacProgram. El software está diseñado para supervisar varios instrumentos conectados simultáneamente y descargar datos de ellos, uno a la vez. Los datos brutos de los instrumentos se evalúan en tiempo real y los resultados se presentan y almacenan en el observatorio, así como se transmiten a través de Internet a un servidor ubicado en Chalmers, en Gotemburgo. El NovacProgram también tiene una unidad de post-procesamiento para una inspección más detallada y la reevaluación de los datos fuera de línea.” (NOVAC, 2011)

Novac Program, es un programa de software libre multiplataforma, creado en 2009, que se encarga de comunicar equipos de sensores de mediciones de gases con cualquier tipo de ordenador. Esto, con el objetivo de poder extraer la información escaneado por el dispositivo y poder representar los datos por medio de gráficas, y exportar resultados para futuros análisis.

Novac Program ayuda a traducir las señales capturadas por los dispositivos de medición a archivos en lenguaje máquina. Por lo general, los sensores suelen generar un archivo en formato .pak, que Novac se encarga de convertir a .txt para poder abrirlo desde cualquier editor de texto. Novac Program, utiliza el protocolo FTP para comunicarse entre los dispositivos de medición y la computadora; el dispositivo actúa como el servidor FTP, alojando los archivos que va generando en cada medición, y Novac Program se encarga de consumir esta información para procesarla y almacenarla localmente en la computadora.

Figura 3.7 - Pantalla principal de Novac Program



Novac Program ayuda a traducir las señales capturadas por los dispositivos de medición a archivos en lenguaje máquina. Por lo general, los sensores suelen generar un archivo en formato .pak, que Novac se encarga de convertir a .txt para poder abrirlo desde cualquier editor de texto. Novac Program, utiliza el protocolo FTP para comunicarse entre los dispositivos de medición y la computadora; el dispositivo actúa como el servidor FTP, alojando los archivos que va generando en cada medición, y Novac Program se encarga de consumir esta información para procesarla y almacenarla localmente en la computadora.

A continuación, mostraremos el contenido de un archivo generado por NOVAC para mediciones de dióxido de azufre.

```

<scaninformation>
  date=13.05.2017
  starttime=15:09:39
  compass=100.0
  tilt=0.0
  lat=13.959912
  long=-89.574813
  alt=1.000
  volcano=telica
  site=i2j7212
  observatory=chalmers
  serial=I2J7212
  spectrometer=S2000
  channel=0
  coneangle=60.0
  interlacesteps=1
  startchannel=0
  spectrumlength=2048
  flux=0.29
  battery=11.38
  temperature=29.48
  mode=plume
  instrumenttype=gothenburg
  version=2.2
  softwareversion=1.82
  compiledate=Feb 19 2009
</scaninformation>
<fluxinfo>
  flux=0.2917
  windspeed=10.0000
  winddirection=0.0000
  plumeheight=1000.00
  windspeedsource=default
  winddirectionsource=default
  plumeheightsource=default
  compasssource=user
  plumecompleteness=-999.00
  plumecentre=-999.00
  plumeedge1=-999.00
  plumeedge2=-999.00
</fluxinfo>
#scanangle starttime      stoptime      name  specsaturation      fitsaturation
  counts_ms  delta  chisquare  exposuretime  numspec      column(SO2)
  columnerror(SO2)  shift(SO2)  shifterror(SO2)  squeeze(SO2)
  squeezeerror(SO2)  isgoodpoint  offset flag
<spectraldata>
0  15:09:39      15:09:48      sky    0.75  0.17  83.18  0.00e+000
  0.00e+000  501  15  0.0  0.0  0.0  0.0  0.0  1
  4248  0
180 15:10:19      15:10:28      dark  0.11  0.08  1.#J  0.00e+000
  0.00e+000  0  15  0.0  0.0  0.0  0.0  0.0  1
  4232  0

```

-90	15:10:43	15:10:52	scan	0.11	0.08	1.#J	2.49e+000		
	3.23e+001	0 15	-89.78	260.66	0.00	0.00	1.00	0.00	0
	4222 0								
-86	15:10:53	15:11:01	scan	0.11	0.08	1.#J	3.70e+000		
	3.61e+001	0 15	965.44	275.34	0.00	0.00	1.00	0.00	0
	4219 0								
-82	15:11:02	15:11:10	scan	0.11	0.08	1.#J	4.03e+000		
	3.23e+001	0 15	662.54	260.46	0.00	0.00	1.00	0.00	0
	4217 0								
-79	15:11:11	15:11:20	scan	0.12	0.08	1.#J	4.50e+000		
	1.88e+001	0 15	595.89	198.74	0.00	0.00	1.00	0.00	0
	4212 0								
-75	15:11:20	15:11:29	scan	0.13	0.08	1.#J	9.17e-001		
	2.24e+000	0 15	10.79	68.59	0.00	0.00	1.00	0.00	0
	4208 0								
-72	15:11:30	15:11:38	scan	0.17	0.08	1.#J	5.90e-001	7.29e-	
001	0 15	48.12 39.13	0.00	0.00	1.00	0.00	0	4204	0
-68	15:11:39	15:11:47	scan	0.19	0.08	1.#J	4.14e-001	5.84e-	
001	0 15	59.35 35.03	0.00	0.00	1.00	0.00	0	4203	0
-64	15:11:48	15:11:57	scan	0.21	0.09	1.#J	3.64e-001	3.18e-	
001	0 15	37.85 25.85	0.00	0.00	1.00	0.00	0	4202	0
-61	15:11:57	15:12:06	scan	0.20	0.08	1.#J	4.15e-001	3.64e-	
001	0 15	111.49 27.66	0.00	0.00	1.00	0.00	0	4200	0
-57	15:12:06	15:12:15	scan	0.21	0.09	1.#J	4.55e-001	4.01e-	
001	0 15	29.58 29.02	0.00	0.00	1.00	0.00	0	4198	0
-54	15:12:16	15:12:24	scan	0.35	0.10	1.#J	2.12e-001	1.14e-	
001	0 15	35.62 15.45	0.00	0.00	1.00	0.00	1	4196	0
-50	15:12:25	15:12:33	scan	0.45	0.11	1.#J	1.15e-001	4.86e-	
002	0 15	31.47 10.11	0.00	0.00	1.00	0.00	1	4188	0
-46	15:12:34	15:12:43	scan	0.75	0.14	1.#J	8.54e-002	2.19e-	
002	0 15	14.25 6.78	0.00	0.00	1.00	0.00	1	4189	0
-43	15:12:43	15:12:52	scan	0.84	0.15	1.#J	9.00e-002	2.38e-	
002	0 15	12.95 7.07	0.00	0.00	1.00	0.00	1	4187	0
-39	15:12:53	15:13:01	scan	1.00	0.19	1.#J	7.22e-002	1.71e-	
002	0 15	5.28 5.99	0.00	0.00	1.00	0.00	1	4187	0
-36	15:13:02	15:13:10	scan	1.00	0.24	1.#J	5.72e-002	1.24e-	
002	0 15	2.95 5.11	0.00	0.00	1.00	0.00	1	4182	0
-32	15:13:11	15:13:20	scan	1.00	0.26	1.#J	7.05e-002	1.51e-	
002	0 15	0.50 5.64	0.00	0.00	1.00	0.00	1	4175	0
-28	15:13:20	15:13:29	scan	1.00	0.26	1.#J	5.53e-002	1.17e-	
002	0 15	-5.54 4.96	0.00	0.00	1.00	0.00	1	4172	0
-25	15:13:30	15:13:38	scan	1.00	0.26	1.#J	5.07e-002	1.22e-	
002	0 15	-2.75 5.06	0.00	0.00	1.00	0.00	1	4169	0
-21	15:13:39	15:13:47	scan	1.00	0.24	1.#J	7.33e-002	1.36e-	
002	0 15	3.23 5.35	0.00	0.00	1.00	0.00	1	4167	0
-18	15:13:48	15:13:57	scan	1.00	0.23	1.#J	5.71e-002	1.32e-	
002	0 15	1.25 5.26	0.00	0.00	1.00	0.00	1	4166	0
-14	15:13:57	15:14:06	scan	1.00	0.21	1.#J	5.31e-002	1.34e-	
002	0 15	-0.76 5.31	0.00	0.00	1.00	0.00	1	4161	0
-10	15:14:06	15:14:15	scan	1.00	0.21	1.#J	6.25e-002	1.34e-	
002	0 15	-3.09 5.30	0.00	0.00	1.00	0.00	1	4159	0
-7	15:14:16	15:14:24	scan	0.97	0.19	1.#J	5.17e-002	1.36e-	
002	0 15	8.45 5.35	0.00	0.00	1.00	0.00	1	4157	0

-3	15:14:25	15:14:33	scan	0.86	0.18	1.#J	6.80e-002	1.41e-
002	0 15	-11.44 5.45	0.00	0.00	1.00	0.00	1 4153	0
0	15:14:34	15:14:43	scan	0.77	0.17	1.#J	6.01e-002	1.48e-
002	0 15	6.72 5.58	0.00	0.00	1.00	0.00	1 4148	0
3	15:14:43	15:14:52	scan	0.72	0.17	1.#J	5.85e-002	1.63e-
002	0 15	14.06 5.86	0.00	0.00	1.00	0.00	1 4146	0
7	15:14:53	15:15:01	scan	0.67	0.17	1.#J	6.38e-002	2.13e-
002	0 15	0.15 6.69	0.00	0.00	1.00	0.00	1 4145	0
10	15:15:02	15:15:10	scan	0.62	0.16	1.#J	6.63e-002	2.02e-
002	0 15	4.38 6.52	0.00	0.00	1.00	0.00	1 4140	0
14	15:15:11	15:15:20	scan	0.59	0.16	1.#J	7.40e-002	1.90e-
002	0 15	12.36 6.31	0.00	0.00	1.00	0.00	1 4145	0
18	15:15:20	15:15:29	scan	0.57	0.16	1.#J	7.50e-002	1.76e-
002	0 15	17.55 6.08	0.00	0.00	1.00	0.00	1 4140	0
21	15:15:30	15:15:38	scan	0.55	0.15	1.#J	6.30e-002	1.76e-
002	0 15	14.18 6.09	0.00	0.00	1.00	0.00	1 4140	0
25	15:15:39	15:15:47	scan	0.54	0.15	1.#J	6.52e-002	1.62e-
002	0 15	8.97 5.84	0.00	0.00	1.00	0.00	1 4128	0
28	15:15:48	15:15:57	scan	0.53	0.15	1.#J	7.75e-002	1.75e-
002	0 15	15.02 6.06	0.00	0.00	1.00	0.00	1 4124	0
32	15:15:57	15:16:06	scan	0.54	0.15	1.#J	6.44e-002	2.26e-
002	0 15	17.65 6.89	0.00	0.00	1.00	0.00	1 4126	0
36	15:16:06	15:16:15	scan	0.55	0.15	1.#J	9.04e-002	2.78e-
002	0 15	18.34 7.64	0.00	0.00	1.00	0.00	1 4118	0
39	15:16:16	15:16:24	scan	0.57	0.15	1.#J	1.01e-001	2.84e-
002	0 15	22.53 7.72	0.00	0.00	1.00	0.00	1 4116	0
43	15:16:25	15:16:33	scan	0.58	0.15	1.#J	7.67e-002	2.65e-
002	0 15	14.95 7.47	0.00	0.00	1.00	0.00	1 4116	0
46	15:16:34	15:16:43	scan	0.58	0.15	1.#J	8.13e-002	2.79e-
002	0 15	20.07 7.65	0.00	0.00	1.00	0.00	1 4107	0
50	15:16:43	15:16:52	scan	0.58	0.15	1.#J	8.68e-002	2.80e-
002	0 15	23.06 7.67	0.00	0.00	1.00	0.00	1 4106	0
54	15:16:53	15:17:01	scan	0.56	0.15	1.#J	8.48e-002	2.73e-
002	0 15	27.42 7.57	0.00	0.00	1.00	0.00	1 4104	0
57	15:17:02	15:17:10	scan	0.53	0.14	1.#J	8.92e-002	3.50e-
002	0 15	35.21 8.57	0.00	0.00	1.00	0.00	1 4100	0
61	15:17:11	15:17:20	scan	0.43	0.12	1.#J	1.16e-001	5.60e-
002	0 15	22.66 10.85	0.00	0.00	1.00	0.00	1 4097	0
64	15:17:20	15:17:29	scan	0.23	0.10	1.#J	1.97e-001	1.23e-
001	0 15	40.91 16.10	0.00	0.00	1.00	0.00	1 4094	0
68	15:17:30	15:17:38	scan	0.16	0.08	1.#J	4.00e-001	4.86e-
001	0 15	39.12 31.95	0.00	0.00	1.00	0.00	0 4090	0
72	15:17:39	15:17:47	scan	0.13	0.08	1.#J	7.41e-001	
	1.50e+000	0 15	68.72 56.08	0.00	0.00	1.00	0.00	0
	4090 0							
75	15:17:49	15:17:57	scan	0.12	0.08	1.#J	1.83e+000	
	6.54e+000	0 15	24.87 117.25	0.00	0.00	1.00	0.00	0
	4079 0							
79	15:17:58	15:18:06	scan	0.11	0.08	1.#J	2.65e+000	
	1.81e+001	0 15	368.49 194.88	0.00	0.00	1.00	0.00	0
	4077 0							
82	15:18:07	15:18:15	scan	0.11	0.08	1.#J	3.47e+000	
	2.61e+001	0 15	343.38 234.02	0.00	0.00	1.00	0.00	0
	4074 0							

```

86 15:18:16    15:18:25    scan  0.11  0.07  1.#J  5.54e+000
    6.92e+001  0      15    515.80 381.25 0.00  0.00  1.00  0.00  0
    4068  0
90 15:18:25    15:18:34    scan  0.10  0.07  1.#J  4.65e+000
    4.77e+001  0      15    528.58 316.69 0.00  0.00  1.00  0.00  0
    4069  0
</spectraldata>

```

Como podremos notar, el archivo es bastante extenso y robusto. En el podremos encontrar toda la información que respecta a las mediciones de dióxido de azufre. Muchas de las estructuras de datos representadas en el archivo, no se utilizarán para el análisis de datos contemplado en el proyecto, por lo que podemos simplificar este archivo, a mantener únicamente los campos marcados en amarillo, teniendo un archivo de la siguiente forma:

```

<scaninformation>
  date=13.05.2017
  starttime=15:09:39
</scaninformation>
#scanangle starttime    stoptime    name    column(SO2)    columnerror(SO2)
  isgoodpoint
<spectraldata>
0  15:09:39    15:09:48    sky     0     0     1
180 15:10:19    15:10:28    dark   0     0     1
-90 15:10:43    15:10:52    scan  -89.78 260.66 0
-86 15:10:53    15:11:01    scan  965.44 275.34 0
-82 15:11:02    15:11:10    scan  662.54 260.46 0
-79 15:11:11    15:11:20    scan  595.89 198.74 0
-75 15:11:20    15:11:29    scan  10.79  68.59 0
-72 15:11:30    15:11:38    scan  48.12  39.13 0
-68 15:11:39    15:11:47    scan  59.35  35.03 0
-64 15:11:48    15:11:57    scan  37.85  25.85 0
-61 15:11:57    15:12:06    scan  111.49 27.66 0
-57 15:12:06    15:12:15    scan  29.58  29.02 0
-54 15:12:16    15:12:24    scan  35.62  15.45 1
-50 15:12:25    15:12:33    scan  31.47  10.11 1
-46 15:12:34    15:12:43    scan  14.25  6.78  1
-43 15:12:43    15:12:52    scan  12.95  7.07  1
-39 15:12:53    15:13:01    scan  5.28   5.99  1
-36 15:13:02    15:13:10    scan  2.95   5.11  1
-32 15:13:11    15:13:20    scan  0.5    5.64  1
-28 15:13:20    15:13:29    scan  -5.54  4.96  1
-25 15:13:30    15:13:38    scan  -2.75  5.06  1
-21 15:13:39    15:13:47    scan  3.23   5.35  1
-18 15:13:48    15:13:57    scan  1.25   5.26  1
-14 15:13:57    15:14:06    scan  -0.76  5.31  1
-10 15:14:06    15:14:15    scan  -3.09  5.3   1

```

```

-7 15:14:16 15:14:24 scan 8.45 5.35 1
-3 15:14:25 15:14:33 scan -11.44 5.45 1
0 15:14:34 15:14:43 scan 6.72 5.58 1
3 15:14:43 15:14:52 scan 14.06 5.86 1
7 15:14:53 15:15:01 scan 0.15 6.69 1
10 15:15:02 15:15:10 scan 4.38 6.52 1
14 15:15:11 15:15:20 scan 12.36 6.31 1
18 15:15:20 15:15:29 scan 17.55 6.08 1
21 15:15:30 15:15:38 scan 14.18 6.09 1
25 15:15:39 15:15:47 scan 8.97 5.84 1
28 15:15:48 15:15:57 scan 15.02 6.06 1
32 15:15:57 15:16:06 scan 17.65 6.89 1
36 15:16:06 15:16:15 scan 18.34 7.64 1
39 15:16:16 15:16:24 scan 22.53 7.72 1
43 15:16:25 15:16:33 scan 14.95 7.47 1
46 15:16:34 15:16:43 scan 20.07 7.65 1
50 15:16:43 15:16:52 scan 23.06 7.67 1
54 15:16:53 15:17:01 scan 27.42 7.57 1
57 15:17:02 15:17:10 scan 35.21 8.57 1
61 15:17:11 15:17:20 scan 22.66 10.85 1
64 15:17:20 15:17:29 scan 40.91 16.1 1
68 15:17:30 15:17:38 scan 39.12 31.95 0
72 15:17:39 15:17:47 scan 68.72 56.08 0
75 15:17:49 15:17:57 scan 24.87 117.25 0
79 15:17:58 15:18:06 scan 368.49 194.88 0
82 15:18:07 15:18:15 scan 343.38 234.02 0
86 15:18:16 15:18:25 scan 515.8 381.25 0
90 15:18:25 15:18:34 scan 528.58 316.69 0
</spectraldata>

```

Como podemos ver, el archivo consta de una estructura XML, la cual ayuda a agrupar la información por nodos, permitiendo al equipo de desarrollo obtener la información más fácil y ágil. La tabla 3.2, resume el significado de cada valor del archivo.

Los campos del nodo `scaninformation` son extremadamente útiles, ya que acá se identifica la fecha y la hora en que se tomó la medida. En base a esta información, se podrá filtrar la información desde la aplicación web, sabiendo con certeza en qué día y/u hora se realizó una medición. Además, la información de este nodo, será la utilizada como pivote, para cruzar la información de dióxido de azufre con las mediciones de RSAM.

Tabla 3.2 - Definición de campos de archivo de mediciones de dióxido de azufre

<b>Nodo</b>	<b>Nombre del campo</b>	<b>Descripción</b>
scaninformation	date	Fecha en formato “DD.MM.YYYY” de captura de los datos. Depende de la hora y fecha del sistema de la máquina donde corre Novac Program
scaninformation	starttime	Hora en formato “HH:MI:SS” de captura de los datos. Depende de la hora y fecha del sistema de la máquina donde corre Novac Program
spectraldata	#scanangle	Ángulo de la medición. La cantidad de ángulos está predefinida (53 ángulos) y son siempre los mismos en todas las mediciones
spectraldata	starttime	Hora de inicio de la medida para el ángulo en turno
spectraldata	stoptime	Hora de finalización de la medida para el ángulo en turno
spectraldata	name	Tipo de medida Posibles valores: “sky”, “dark”, “scan”
spectraldata	column(SO2)	Medición de concentración de dióxido de azufre para el ángulo en turno Unidad de medida: PPM
spectraldata	columnerror(SO2)	Rango de error en medición de concentración de dióxido de azufre para el ángulo en turno Unidad de medida: PPM
spectraldata	isgoodpoint	Bandera que indica si el dato tomado es considerado bueno, para el ángulo en turno.

La información alojada en el nodo `spectraldata` ya es información propia de la medición. Las mediciones de SO<sub>2</sub> se hacen en base al ángulo del sensor en el momento de la toma. Los ángulos en que se realizarán las mediciones ya están definidos para el sensor por medio de un archivo de configuración de Novac Program. Para el caso de nuestro proyecto, se decidió utilizar el estándar para ScanDoas, que recomienda 53 mediciones en 53 ángulos diferentes.

La información que ingresará a la base de datos, será aquella que sea considerada buena, por lo que el campo `isgoodpoint` es de suma importancia. Novac verifica en base a la calidad de la medida si un valor puede ser considerado correcto; esto debido a cualquier interferencia que pueda existir al momento que se realiza la medición (nubes que cubren rayos del sol, lluvia, polvo, humo, objetos, etc.)

La tabla 3.3, resume las generalidades y características del archivo de mediciones de dióxido de azufre:

*Tabla 3.3 - Generalidades de archivo de dióxido de azufre*

<b>Característica</b>	<b>Valor</b>	<b>Comentarios</b>
Formato	.txt	-
Peso aproximado por archivo	7-8 Kb	-
Rango de generación de archivo	8-15 minutos	Depende de la intensidad de rayos solares
Hora de inicio de mediciones	6:00 AM	Depende de la salida del sol
Hora de finalización de mediciones	6:00 PM	Depende de la puesta del sol
Mediciones por archivo	53	Una por cada ángulo predeterminado para el escáner
Gases por archivo	Ilimitado	El archivo contendrá todos los gases predeterminados para el escáner
Archivos generados por día	65-75 aproximadamente	-

Novac Program, permite la configuración de diversos gases para un mismo escaner de mediciones, por lo que se pueden identificar diferentes gases en un mismo archivo. Esto garantiza la escalabilidad en el diseño de nuestro sistema. Novac identifica la columna `column(SO2)` sustituyendo la notación del gas en cuestión en el archivo.

### **3.1.3.2. *Earthworm y RSAM***

A diferencia de las mediciones de dióxido de azufre, que debían extraerse directamente del sensor de datos; las mediciones de movimiento sísmico se obtendrán directamente del SNET.

El SNET, será nuestro proveedor de información de RSAM. Internamente, la institución se encarga de la extracción de datos de los sensores de una estación determinada, su transformación y conversión hasta convertirse en un archivo de texto plano.

La institución proveerá la información de cada volcán que sea necesario analizar, previo a la solicitud formal de cada estación que se desee. La configuración para recibir datos de diferentes centros de observación, se realiza por medio de la aplicación Earthworm. Esto garantiza la escalabilidad en el diseño de la solución propuesta.

“Earthworm es un software público de código abierto para la adquisición de datos, procesamiento, archivo y distribución de sismicidad. Originalmente desarrollado por el Servicio Geológico de los Estados Unidos, los binarios y los archivos fuente de Earthworm están disponibles gratuitamente para todos. El desarrollo y el mantenimiento son ahora llevados a cabo por la comunidad de usuarios de Earthworm, ISTI (Instrumental Software Technologies, Inc.) y CERI (Centro de Investigación e Información por Terremotos).

El proyecto Earthworm se inició en 1993, principalmente en respuesta a una serie de necesidades comunes identificadas por las redes sísmicas regionales: Los sistemas de procesamiento automático utilizados por la mayoría de las redes regionales envejecían y los costos de mantenimiento crecían; Los avances en la investigación sísmica requerían datos de sensores nuevos y sofisticados para proporcionar datos de investigación viables; Las crecientes demandas inmediatas de utilidad social requerían productos nuevos y altamente visibles en tiempo real; además, la reducción de los financiamientos significaba que la mayoría de las redes

individuales ya no podían apoyar los esfuerzos de desarrollo de los sistemas locales.

La mayoría de los sistemas de procesamiento en uso de ese entonces tenían una serie de inconvenientes que hacían difícil mejorarlos para satisfacer los nuevos requerimientos. Se construyeron sobre productos de proveedores específicos, los cuales los vincularon a vendedores de alto precio y les privaron de los beneficios del creciente mercado de masas.

El esfuerzo inicial de Earthworm tenía la misión específica de reemplazar a los viejos dispositivos que habían funcionado bien durante décadas y se habían convertido en el pilar para producir soluciones de eventos rápidos. Sin embargo, las piezas de repuesto ya no estaban disponibles, y había una necesidad urgente de producir un sistema de reemplazo en un tiempo mínimo. Esto significaba producir un sistema que pudiera procesar más de quinientos canales de datos de manera rápida y confiable.” (Instrumental Software Technologies, 2016)

Dado que el objetivo inicial de este sistema era proporcionar una notificación rápida de los eventos sísmicos, el sistema que evolucionó no tenía "memoria" de los acontecimientos pasados. Se hizo hincapié en la velocidad y fiabilidad, y un evento se completaba cuando el sistema producía una notificación. Sus características fundamentales son las siguientes:

- **Modularidad:** Cada función realizada por el sistema debe ser encapsulada en un módulo que puede funcionar independientemente de otros módulos, tanto en hardware como en software. La implicación es que se pueda garantizar que un conjunto de funciones críticas del sistema sea independiente de otras funciones del sistema. Así, nuevas funciones experimentales pueden ser añadidas sin interrumpir las operaciones críticas preexistentes

- **Independencia del sistema:** Los módulos deben operar en varias marcas de hardware de computadora y sistemas operativos, y varios tipos de tales computadoras se pueden unir para funcionar como un sistema. Esto, con la idea de modularidad, implica que el sistema puede migrar gradualmente de un tipo de computadora a otro sin interrupción.
- **Escalabilidad:** El sistema debe proporcionar una curva de coste-rendimiento suave para acomodar redes grandes y pequeñas. Esto era quizás más importante antes de la disponibilidad de computadoras baratas y poderosas producidas en serie. Sin embargo, sigue siendo pertinente con respecto a los derechos de licencia para el software comercial incluido.
- **Conectividad:** La suposición es que estos sistemas ya no están aislados, sino que tienen que interactuar rápida y confiablemente con otros sistemas automáticos en tiempo real, sistemas de análisis interactivos y varios esquemas de notificación. El objetivo es proporcionar conectividad a varios niveles de procesamiento automático e interactivo, de modo que dichos sistemas puedan configurarse para funcionar en configuraciones que van desde la operación autónoma completa hasta el funcionamiento como nodos en un sistema de área amplia distribuido.
- **Robustez:** Tradicionalmente, el papel de los sistemas de procesamiento sísmico fue principalmente de apoyo a la investigación. Como resultado, la fiabilidad del sistema era naturalmente menos importante que el coste, el tiempo hasta la finalización y las características. Sin embargo, las nuevas responsabilidades para la prensa y las agencias de respuesta a emergencias requerían niveles muy altos de confiabilidad del sistema, especialmente durante las crisis sísmicas, cuando los datos de entrada y el poder pueden ser interrumpidos y las cargas del sistema pueden

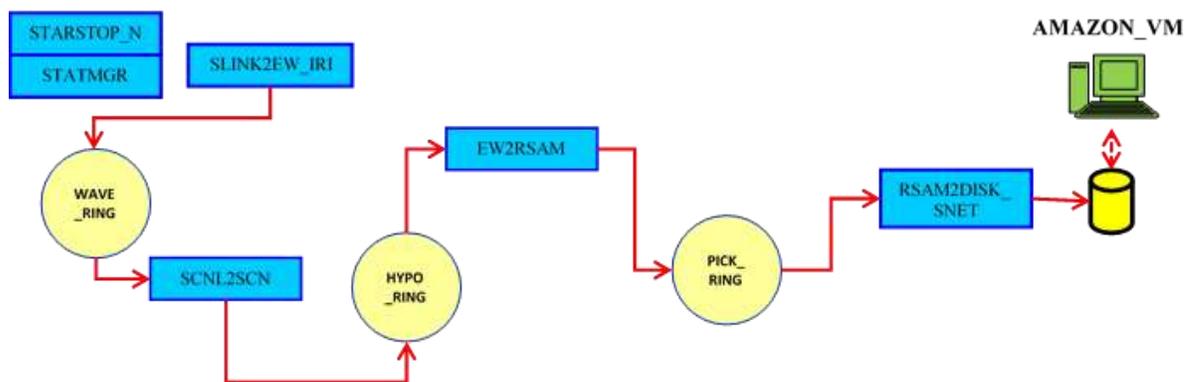
aumentar dramáticamente. Así, cuestiones como la detección y recuperación de errores, el tiempo de reparación, la degradación agraciada y el control de la carga se volvieron vitales y tuvieron que diseñarse en el sistema. Los costos de robustez a menudo no se aprecian, en términos de tiempo para diseñar, implementar y probar, así como los costos de hardware adecuado y equipo auxiliar.

“Earthworm se ha convertido de manera no oficial en el software estándar para la adquisición, procesamiento, archivo y distribución de datos de sismicidad, debido a sus grandes características.” (Instrumental Software Technologies, 2016)

Diferentes centros de monitoreo sísmicos basan sus sistemas de medición y notificación en Earthworm y el MARN no es la excepción.

Todos los datos de sismicidad que nuestro sistema procesara serán obtenidos por medio de dos módulos (import\_ack y export\_ack) de Earthworm que el MARN utilizará para compartir sus datos para nuestro propio procesamiento. Por este motivo nuestro sistema tendrá una ligera integración el sistema Earthworm para la captura de datos, para luego realizar el respectivo procesamiento de datos.

Figura 3.8 - Diagrama de módulos de Earthworm



La figura 3.8, muestra los módulos necesarios en la implementación de Earthworm en la instancia cliente a instalar en la máquina virtual de la solución. A continuación describimos cada uno de ellos:

- **STARSTOP\_N:** Inicia y detiene todos los módulos EARTHWORM en una computadora. Este módulo es el núcleo del sistema Earthworm.
- **STARMGR:** Es el modulo para monitorear la salud de todos los módulos Earthworm. Informa sobre la salud por correo electrónico y puede emitir automáticamente una solicitud de reinicio para un módulo muerto.
- **SLINK2EW\_IRI:** Es un módulo cliente de SeedLink para Earthworm. SeedLink es el componente de servidor de datos del Procesador de Comunicación Sismológica, o SeisComP, desarrollado originalmente en GEOFON. El protocolo SeedLink se puede resumir como un simple esquema de negociación seguido por la transmisión de paquetes de datos desde el servidor. Los paquetes SeedLink se componen de un encabezado pequeño seguido de un registro Mini-SEED de 512 bytes (sólo datos SEED). La fase de negociación permite al cliente solicitar sólo los tipos de datos especificados del servidor para cada flujo de datos. Un flujo de datos se define mediante un par de código de red y estación.
- **SCNL2SCN:** Es un módulo convertidor para paquetes de forma de onda). Scnl2scn es un módulo para convertir TRACEBUF2 en mensajes TRACEBUF eliminando el código de ubicación.
- **EWRSAM:** Esta modulo calcula los valores RSAM para señales SCNL o SCN en períodos especificados.
- **RSAM2DISK:** Es el modulo encargado de guardar todos los valores de RSAM almacenados temporalmente en los espacios de memoria (RING) en un directorio específico.

Earthworm se encarga de recibir la información proveniente del SNET, y la descarga al almacenamiento interno de la máquina virtual, en un archivo en formato .RSM. El contenido de este archivo es el siguiente:

08/01/2017 00:00:00 1501545600 +0039  
08/01/2017 00:10:02 1501546202 +0036  
08/01/2017 00:20:01 1501546801 +0036  
08/01/2017 00:30:02 1501547402 +0036  
08/01/2017 00:40:00 1501548000 +0036  
08/01/2017 00:50:03 1501548603 +0036  
08/01/2017 01:00:03 1501549203 +0059  
08/01/2017 01:10:01 1501549801 +0036  
08/01/2017 01:20:01 1501550401 +0036  
08/01/2017 01:30:01 1501551001 +0036  
08/01/2017 01:40:01 1501551601 +0036  
08/01/2017 01:50:03 1501552203 +0039  
08/01/2017 02:00:04 1501552804 +0037  
08/01/2017 02:10:03 1501553403 +0035  
08/01/2017 02:20:04 1501554004 +0037  
08/01/2017 02:30:00 1501554600 +0093  
08/01/2017 02:40:01 1501555201 +0037  
08/01/2017 02:50:03 1501555803 +0036  
08/01/2017 03:00:03 1501556403 +0037  
08/01/2017 03:10:01 1501557001 +0036  
08/01/2017 03:20:03 1501557603 +0037  
08/01/2017 03:30:01 1501558201 +0036  
08/01/2017 03:40:02 1501558802 +0036  
08/01/2017 03:50:04 1501559404 +0036  
08/01/2017 04:00:01 1501560001 +0036  
08/01/2017 04:10:00 1501560600 +0036  
08/01/2017 04:20:01 1501561201 +0036  
08/01/2017 04:30:04 1501561804 +0035  
08/01/2017 04:40:01 1501562401 +0036  
08/01/2017 04:50:02 1501563002 +0036  
08/01/2017 05:00:01 1501563601 +0035  
08/01/2017 05:10:01 1501564201 +0035  
08/01/2017 05:20:03 1501564803 +0035  
08/01/2017 05:30:03 1501565403 +0035  
08/01/2017 05:40:00 1501566000 +0035  
08/01/2017 05:50:01 1501566601 +0035  
08/01/2017 06:00:03 1501567203 +0036  
08/01/2017 06:10:00 1501567800 +0036  
08/01/2017 06:20:01 1501568401 +0036  
08/01/2017 06:30:03 1501569003 +0036  
08/01/2017 06:40:02 1501569602 +0035  
08/01/2017 06:50:01 1501570201 +0036  
08/01/2017 07:00:00 1501570800 +0036  
08/01/2017 07:10:00 1501571400 +0036  
08/01/2017 07:20:03 1501572003 +0036  
08/01/2017 07:30:01 1501572601 +0036  
08/01/2017 07:40:00 1501573200 +0037  
08/01/2017 07:50:02 1501573802 +0035  
08/01/2017 08:00:03 1501574403 +0035

El contenido del archivo de movimiento sísmico es muy diferente al archivo generado por Novac, para emisiones de SO<sub>2</sub>. Para empezar, el archivo no tiene ninguna estructura, ningún conjunto de nodos o ningún formato; es más, ni siquiera tiene encabezados para cada columna que contiene. Para entender un poco más este archivo, la tabla 3.4 explica su contenido:

*Tabla 3.4 - Definición de campos de archivo de mediciones de RSAM*

<b>Columna</b>	<b>Nombre del campo</b>	<b>Descripción</b>
Columna 1	fecha	Fecha en formato “DD/MM/YYYY” de captura de los datos. Depende de la hora y fecha del sistema del servidor de Earthworm que envía los datos.
Columna 2	hora	Hora en formato “HH:MI:SS” de captura de los datos. Depende de la hora y fecha del sistema del servidor de Earthworm que envía los datos.
Columna 3	id	Identificador único de cada medida.
Columna 4	rsam	Valor de medida de movimiento sísmico en unidades de medida RSAM.

La forma en la que éste archivo se genera, difiere mucho de la forma en que se genera el archivo de dióxido de azufre. Para iniciar, este archivo se genere una vez por mes, y guarda un línea por cada medición tomada cada 10 minutos. Esto quiere decir que el archivo se está sobrescribiendo conforme avanza el día; esto, representa un reto sustancial para el equipo de desarrollo, ya que el archivo estará en constante cambio.

El archivo de datos de RSAM, es generado y actualizado por medio de Earthworm, por lo que la integridad del mismo está garantizada por los entornos de software que se encargan de proveerlo; sin embargo, a pesar de esto, la calidad de la información también se puede ver afectada.

Habrán ocasiones, en las que el código identificador de una medida será “0000000000”; esto, hace referencia a una medida defectuosa, ya sea porque

Earthworm identifica que la medida posee ruino, interferencia u otra disrupción; o, porque se generó algún problema en los instrumentos de medida. Las medidas bajo este ID, no se tomarán en cuenta para ser parte de la data de nuestra aplicación, ya que no representan más que cálculos fallidos.

La tabla 3.5, resume las generalidades y características del archivo de mediciones de dióxido de azufre:

*Tabla 3.5 - Generalidades de archivo de RSAM*

<b>Característica</b>	<b>Valor</b>	<b>Comentarios</b>
Formato	.rsm	-
Peso aproximado por archivo	160-180 Kb	-
Rango de generación de archivo	1 mes	
Rango de actualización de archivo	10 minutos	Se toman medidas cada 10 minutos
Horas al día en que se realizan mediciones	24 h	Las mediciones no paran en ningún momento del día
Mediciones por archivo	4,032- 4,464	Una por cada ángulo predeterminado para el escáner
Volcanes por archivo	1	Cada volcán configurado genera un archivo diferente
Mediciones generadas al día	144	-

## **3.2. DISEÑO DE BASE DE DATOS**

La figura 3.1, nos mostraba de forma general, todos los componentes que formarán parte de nuestro sistema, y como era de esperar, la base de datos no podía faltar en el flujo. La base de datos de nuestro sistema, contendrá toda la información necesaria para mostrar los datos por medio de la aplicación web que estará de cara al usuario.

Tal y como lo revisamos en el apartado anterior de este trabajo, nuestro sistema no será encargado de generar datos; todos los datos que se alberguen en la base, serán producto de la extracción de diferentes fuentes de transformación, que pasarán por los debidos procesos de transformación de datos, hasta desencadenar la carga de datos en la base. A este tipo de procesos, los identificamos como ETL (Extract, Transform and Load).

### ***3.2.1. Data Warehouse***

Debido a que la información que ingresará a la base de dato, no se genera propiamente en la aplicación, no podríamos considerar nuestra base como una base de datos Transaccional, a la cual, se le ingresan registros por cada transacción que genera un usuario. Más bien, nuestra base se alimentará de lotes definidos de información, los cuales serán productos de los procesos ETL que se encarguen de hacer fluir la data desde las fuentes hasta la base. En este orden de ideas, podemos concluir, que nuestra base de datos hará las funciones de un Data Warehouse.

“Un data warehouse es un repositorio unificado para todos los datos que recogen los diversos sistemas de una organización. El repositorio puede ser físico o lógico y hace hincapié en la captura de datos de diversas fuentes sobre todo para fines analíticos y de acceso.

Normalmente, un data warehouse se aloja en un servidor corporativo o cada vez más, en la nube. Los datos de diferentes aplicaciones de procesamiento de transacciones Online (OLTP) y otras fuentes se extraen selectivamente para su uso por aplicaciones analíticas y de consultas por usuarios.

Data Warehouse es una arquitectura de almacenamiento de datos que permite a los ejecutivos de negocios organizar, comprender y utilizar sus datos para tomar decisiones estratégicas. Un data warehouse es una arquitectura conocida ya en muchas empresas modernas.

La arquitectura de un data warehouse puede ser dividida en tres estructuras simplificadas: básica, básica con un área de ensayo y básica con área de ensayo y data marts.

Con una estructura básica, sistemas operativos y archivos planos proporcionan datos en bruto que se almacenan junto con metadatos. Los usuarios finales pueden acceder a ellos para su análisis, generación de informes y minería.

Al añadir un área de ensayo que se puede colocar entre las fuentes de datos y el almacén, ésta proporciona un lugar donde los datos se pueden limpiar antes de entrar en el almacén. Es posible personalizar la arquitectura del almacén para diferentes grupos dentro de la organización.

Se puede hacer agregando data marts, que son sistemas diseñados para una línea de negocio en particular. Se pueden tener data marts separados para ventas, inventario y compras, por ejemplo, y los usuarios finales pueden acceder a datos de uno o de todos los data marts del departamento.

Históricamente, los data warehouses se habían formado utilizando datos repetitivos estructurados que eran filtrados antes de entrar en el data warehouse. Sin embargo, en los últimos años, el data warehouse ha evolucionado debido a información contextual que ahora se puede adjuntar a los datos no estructurados y que también puede ser almacenada.

Aquellos primeros datos relacionales estructurados no podían ser mezclados y emparejados para temas analíticos con datos textuales no estructurados. Pero con el advenimiento de la contextualización, estos tipos de análisis ahora sí pueden hacerse de formas naturales y fáciles.

En el data warehouse, datos no repetitivos, como los comentarios en una encuesta, correos electrónicos y conversaciones, se tratan de forma diferente a las ocurrencias repetitivas de datos, como el flujo de clics, mediciones o el procesamiento máquina o analógico. Los datos no repetitivos son datos basados en textos que fueron generados por la palabra escrita o hablada, leída y reformateada y, lo que es más importante, ahora puede ser contextualizada. Con el fin de extraer cualquier sentido de los datos no repetitivos para su uso en el Data Warehouse, deben tener el contexto de los datos establecidos.

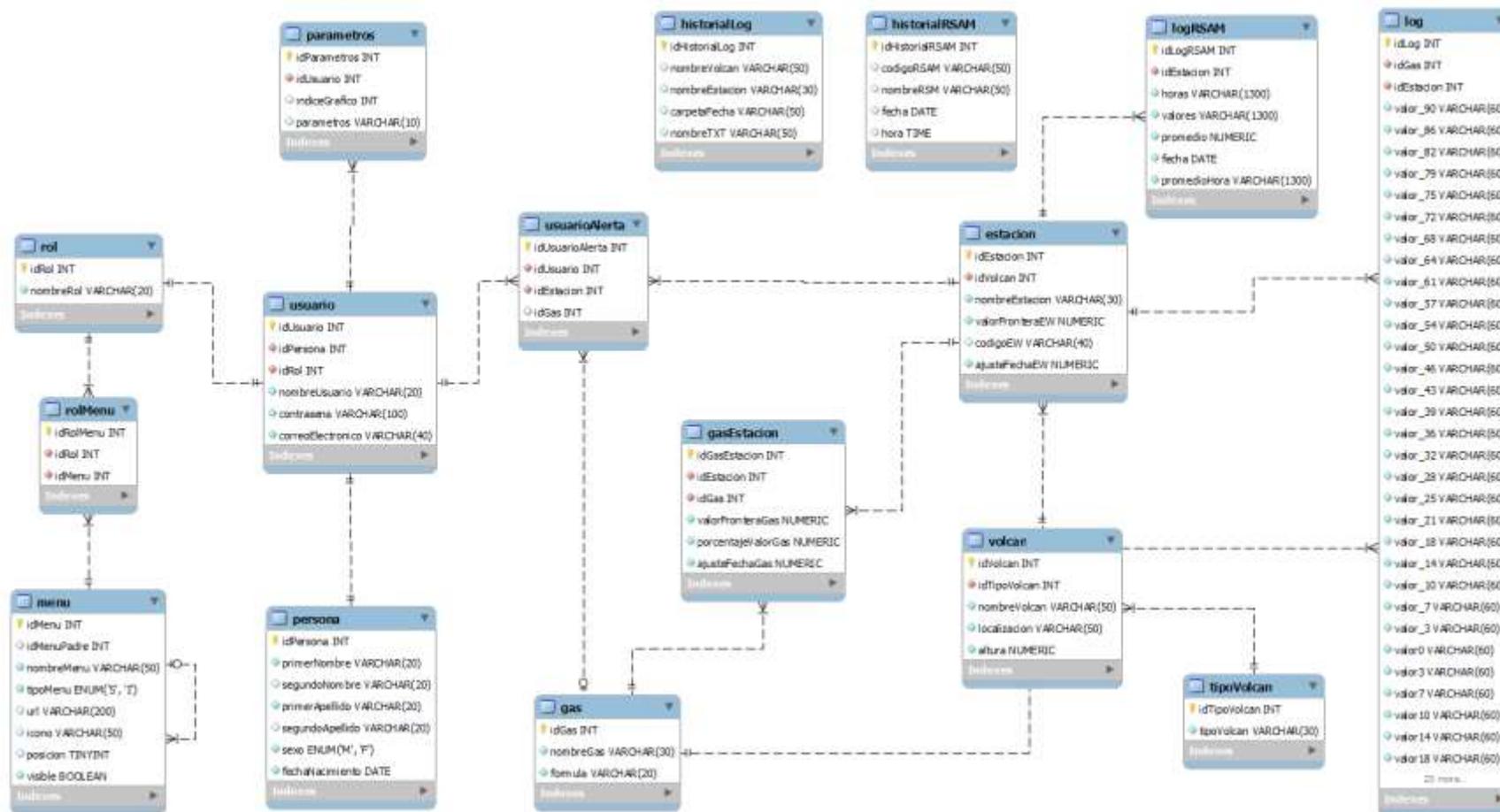
En muchos casos, el contexto de los datos no repetitivos es más importante que los datos en sí. En cualquier caso, los datos no repetitivos no pueden utilizarse para la toma de decisiones hasta que se haya establecido el contexto.” (Power Data, 2016)

La base de datos de nuestra aplicación, fungirá las veces de un repositorio de datos que tendrá un gran objetivo: recopilar la información de emisiones de dióxido de azufre y de movimiento sísmico, para posteriormente, mostrar análisis unificados al usuario, por medio de una aplicación web o reportes.

A pesar que nuestra base sea independiente a las fuentes de datos que la alimentan, no hay que perder de vista la estructura de la información que conocimos en el apartado anterior; ya que en base a los archivos recibidos por las fuentes es que se diseña la base. La figura 3.6, muestra el modelo entidad relación que regirá nuestra base de datos.

### 3.2.2. Modelo Entidad Relación

Figura 3.9 - Modelo Entidad Relación de Base de Datos de la Sistema



### 3.2.3. Tablas y Campos

La figura 3.9, muestra de forma explícita toda la definición de la base de datos a utilizar. De momento, solo conocemos la forma en que las tablas se relacionan entre sí, y hacen fluir la información hacia la aplicación web que mostrará datos al usuario.

A continuación, mostraremos la definición de cada tabla en específico que conforma la base de datos. Cabe mencionar, que el orden en el que se mostrarán las tablas no es significativo; mostraremos tabla por tabla en orden alfabético.

*Cuadro 3.1 - Definición de tabla ESTACION*

<b>TABLA:</b>	<b>ESTACION</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>NULO</b>	<b>LLAVE</b>	<b>COMENTARIO</b>
<b>IDESTACION</b>	INT(11)	NO	PRIMARY	
<b>IDVOLCAN</b>	INT(11)	NO	FOREING	Apuntando hacia tabla VOLCAN
<b>NOMBREESTACION</b>	VARCHAR(30)	NO	UNIQUE	
<b>VALORFRONTERAEW</b>	DECIMAL(10,0)	NO		Valor frontera para envío de alertas RSAM
<b>CODIGOEW</b>	VARCHAR(40)	YES	UNIQUE	Código de estación configurado en Earthworm
<b>AJUSTEFECHEAEW</b>	DECIMAL(10,0)	NO		Diferencia en horas entre hora del servidor y hora local
<b>DESCRIPCIÓN DE TABLA:</b>	Esta tabla contendrá la información de cada estación de observación y monitoreo volcánico. Se relaciona de forma directa con la tabla de volcanes, ya que un volcán, puede tener asociadas muchas estaciones			

*Cuadro 3.2 - Definición de tabla GAS*

<b>TABLA:</b>	<b>GAS</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>NULO</b>	<b>LLAVE</b>	<b>COMENTARIO</b>
<b>IDGAS</b>	INT(11)	NO	PRIMARY	
<b>NOMBREGAS</b>	VARCHAR(30)	NO	UNIQUE	
<b>FORMULA</b>	VARCHAR(20)	NO	UNIQUE	Fórmula molecular
<b>DESCRIPCIÓN DE TABLA:</b>	Tabla para almacenar los distintos gases que soportará la aplicación. Se creó con el fin de garantizar la escalabilidad de la herramienta			

Cuadro 3.3 – Definición de tabla GASESTACION

<b>TABLA:</b>	<b>GASESTACION</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>NULO</b>	<b>LLAVE</b>	<b>COMENTARIO</b>
<b>IDGASESTACION</b>	INT(11)	NO	PRIMARY	
<b>IDESTACION</b>	INT(11)	NO	FOREING	Apuntando hacia tabla ESTACION
<b>IDGAS</b>	INT(11)	NO	FOREING	Apuntando hacia tabla VOLCAN
<b>VALORFRONTERAGAS</b>	DECIMAL(10,0)	NO		Valor frontera para envío de alertas de gas
<b>PORCENTAJEVALORGAS</b>	DECIMAL(10,0)	NO		Porcentaje de valores a tomar en cuenta para envío de alertas de gas
<b>AJUSTEFECHAGAS</b>	DECIMAL(10,0)	NO		Diferencia en horas entre hora del servidor y hora local
<b>DESCRIPCIÓN DE TABLA:</b>	Tabla que rompe la relación de muchos a muchos entre tablas GAS y ESTACION. Guarda información apegada a cada gas configurado para una estación en específico.			

Cuadro 3.4 - Definición de tabla HISTORIALLOG

<b>TABLA:</b>	<b>HISTORIALLOG</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>NULO</b>	<b>LLAVE</b>	<b>COMENTARIO</b>
<b>IDHISTORIALLOG</b>	INT(11)	NO	PRIMARY	
<b>NOMBREVOLCAN</b>	VARCHAR(50)	YES		
<b>NOMBREESTACION</b>	VARCHAR(30)	YES		
<b>CARPETAFECHA</b>	VARCHAR(50)	YES		Nombre de la carpeta del sistema operativo
<b>NOMBRETXT</b>	VARCHAR(50)	YES		Nombre del archivo generado por Novac Program
<b>DESCRIPCIÓN DE TABLA:</b>	Tabla que almacena un registro por cada archivo de emisión de gases que procesa el robot MAVAMSI_BOT1.			

Cuadro 3.5 - Definición de tabla HISTORIALRSAM

TABLA:	HISTORIALRSAM			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDHISTORIALRSAM	INT(11)	NO	PRIMARY	
CODIGORSAM	VARCHAR(50)	YES		
NOMBRERSM	VARCHAR(50)	YES		Nombre del archivo generado por Earthworm
FECHA	DATE	YES		
HORA	TIME	YES		
DESCRIPCIÓN DE TABLA:	Tabla que almacena un registro por cada archivo de RSAM que procesa el robot MAVAMSI_BOT1.			

Cuadro 3.6 - Definción de tabla LOG

TABLA:	LOG			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDLOG	INT(11)	NO	PRIMARY	
IDGAS	INT(11)	NO	FOREING	Apuntando hacia tabla GAS
IDESTACION	INT(11)	NO	FOREING	Apuntando hacia tabla ESTACION
VALOR_90	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -90 grados
VALOR_86	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -86 grados
VALOR_82	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -82 grados
VALOR_79	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -79 grados
VALOR_75	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -75 grados
VALOR_72	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -72 grados
VALOR_68	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -68 grados
VALOR_64	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -64 grados
VALOR_61	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -61 grados
VALOR_57	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -57 grados
VALOR_54	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -54 grados
VALOR_50	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -50 grados
VALOR_46	VARCHAR(60)	NO		Medición de emisión de gas, escaner a -46 grados

<b>VALOR_43</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -43 grados
<b>VALOR_39</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -39 grados
<b>VALOR_36</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -36 grados
<b>VALOR_32</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -32 grados
<b>VALOR_28</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -28 grados
<b>VALOR_25</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -25 grados
<b>VALOR_21</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -21 grados
<b>VALOR_18</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -18 grados
<b>VALOR_14</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -14 grados
<b>VALOR_10</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -10 grados
<b>VALOR_7</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -7 grados
<b>VALOR_3</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a -3 grados
<b>VALOR0</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 0 grados
<b>VALOR3</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 3 grados
<b>VALOR7</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 7 grados
<b>VALOR10</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 10 grados
<b>VALOR14</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 14 grados
<b>VALOR18</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 18 grados
<b>VALOR21</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 21 grados
<b>VALOR25</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 25 grados
<b>VALOR28</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 28 grados
<b>VALOR32</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 32 grados
<b>VALOR36</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 36 grados
<b>VALOR39</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 39 grados
<b>VALOR43</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 43 grados

<b>VALOR46</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 46 grados
<b>VALOR50</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 50 grados
<b>VALOR54</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 54 grados
<b>VALOR57</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 57 grados
<b>VALOR61</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 61 grados
<b>VALOR64</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 64 grados
<b>VALOR68</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 68 grados
<b>VALOR72</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 72 grados
<b>VALOR75</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 75 grados
<b>VALOR79</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 79 grados
<b>VALOR82</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 82 grados
<b>VALOR86</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 86 grados
<b>VALOR90</b>	VARCHAR(60)	NO	Medición de emisión de gas, escaner a 90 grados
<b>AJUSTE</b>	VARCHAR(7)	NO	Mayor error de medición, de archivo generado por Novac Program
<b>FECHA</b>	DATE	NO	
<b>HORA</b>	TIME	NO	
<b>DESCRIPCIÓN DE TABLA:</b>	Tabla central de almacenamiento de emisión de gases. Contiene una línea por cada archivo generado por Novac Program.		

Cuadro 3.7 - Definición de tabla LOGRSAM

<b>TABLA:</b>	<b>LOGRSAM</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>NULO</b>	<b>LLAVE</b>	<b>COMENTARIO</b>
<b>IDLOGRSAM</b>	INT(11)	NO	PRIMARY	
<b>IDESTACION</b>	INT(11)	NO	FOREING	Apuntando hacia VOLCAN
<b>HORAS</b>	VARCHAR(1300)	NO		
<b>VALORES</b>	VARCHAR(1300)	NO		Mediciones en unidades de medida de RSAM
<b>PROMEDIO</b>	DECIMAL(10,0)	NO		
<b>FECHA</b>	DATE	NO		
<b>PROMEDIOHORA</b>	VARCHAR(1300)	NO		
<b>DESCRIPCIÓN DE TABLA:</b>	Tabla central de almacenamiento de movimiento sísmico. Contiene una línea por cada medición (cada línea) de archivo generado por Earthworm.			

Cuadro 3.8 - Definción de tabla MENU

<b>TABLA:</b>	<b>MENU</b>			
<b>CAMPO</b>	<b>TIPO</b>	<b>NULO</b>	<b>LLAVE</b>	<b>COMENTARIO</b>
<b>IDMENU</b>	INT(11)	NO	PRIMARY	
<b>IDMENUPADRE</b>	INT(11)	YES	FOREING	Apuntando hacia MENU
<b>NOMBREMENU</b>	VARCHAR(50)	NO		
<b>TIPOMENU</b>	ENUM('S','I')	NO		'S' para "Superior"; 'I' para "Interno"; según sea el tipo de menú
<b>URL</b>	VARCHAR(200)	YES		URL de la página web hacia la que apunta
<b>ICONO</b>	VARCHAR(50)	YES		
<b>POSICION</b>	TINYINT(4)	YES		'0' para valores con TIPOMENU = 'S'; números cardinales para el resto de menús internos, según orden de aparición
<b>VISIBLE</b>	TINYINT(1)	NO		'1' para menú visible, '0' para ocultar menú
<b>DESCRIPCIÓN DE TABLA:</b>	Tabla para almacenar los menús a desplegar en la aplicación, posterior al inicio de sesión. Útil para la segmentación de la aplicación por roles y usuarios.			

Cuadro 3.9 - Definición de tabla PARAMETROS

TABLA:		PARAMETROS			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO	
IDPARAMETROS	INT(11)	NO	PRIMARY		
IDUSUARIO	INT(11)	NO	FOREING	Apuntando hacia USUARIO	
INDICEGRAFICO	INT(11)	YES		'0' para gráfico de Gases, '1' para gráfico de RSAM y '2' para gráfico RSAM vs PPM	
PARAMETROS	VARCHAR(10)	YES		Valores iniciales del gráfico	
DESCRIPCIÓN DE TABLA:	Esta tabla se utiliza para configurar el gráfico inicial en la pantalla principal de la aplicación; configurado según usuario.				

Cuadro 3.10 - Definición de tabla PERSONA

TABLA:		PERSONA			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO	
IDPERSONA	INT(11)	NO	PRIMARY		
PRIMERNOMBRE	VARCHAR(20)	NO			
SEGUNDONOMBRE	VARCHAR(20)	YES			
PRIMERAPELLIDO	VARCHAR(20)	NO			
SEGUNDOAPELLIDO	VARCHAR(20)	YES			
SEXO	ENUM('M','F')	NO			
FECHANACIMIENTO	DATE	NO			
DESCRIPCIÓN DE TABLA:	Tabla para centralizar información personal de usuarios.				

Cuadro 3.11 - Definición de tabla ROL

TABLA:		ROL			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO	
IDROL	INT(11)	NO	PRIMARY		
NOMBREROL	VARCHAR(20)	NO	UNIQUE		
DESCRIPCIÓN DE TABLA:	Tabla para almacenar los roles que soportará la aplicación.				

Cuadro 3.12- Definición de tabla ROLMENU

TABLA:	ROLMENU			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDROLMENU	INT(11)	NO	PRIMARY	
IDROL	INT(11)	NO	FOREING	Apuntando hacia ROL
IDMENU	INT(11)	NO	FOREING	Apuntando hacia MENU
DESCRIPCIÓN DE TABLA:	Tabla para romper relación de muchos a muchos entre tablas ROL y MENU. Busca dejar por sentado a qué menús tendrá acceso cada rol de la aplicación.			

Cuadro 3.13 - Definición de tabla TIPOVOLCAN

TABLA:	TIPOVOLCAN			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDTIPOVOLCAN	INT(11)	NO	PRIMARY	
TIPOVOLCAN	VARCHAR(30)	NO		
DESCRIPCIÓN DE TABLA:	Tabla para centralizar información del tipo de volcán. Ésta solo representa descripción informativa para filtrar registros.			

Cuadro 3.14 - Definición de tabla USUARIO

TABLA:	USUARIO			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDUSUARIO	INT(11)	NO	PRIMARY	
IDPERSONA	INT(11)	NO	FOREING	Apuntando hacia PERSONA
IDROL	INT(11)	NO	FOREING	Apuntando hacia ROL
NOMBREUSUARIO	VARCHAR(20)	NO	UNIQUE	
CONTRASENA	VARCHAR(100)	NO		Registros almacenados con encriptamiento MD5
CORREOELECTRONICO	VARCHAR(40)	NO	UNIQUE	
DESCRIPCIÓN DE TABLA:	Esta tabla se utiliza para almacenar la información de usuarios de la aplicación. El acceso a la aplicación, depende de la información cargada en esta tabla.			

Cuadro 3.15 - Definición de tabla USUARIOALERTA

TABLA:	USUARIOALERTA			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDUSUARIOALERTA	INT(11)	NO	PRIMARY	
IDUSUARIO	INT(11)	NO	FOREING	Apuntando hacia USUARIO
IDESTACION	INT(11)	NO	FOREING	Apuntando hacia ESTACION
IDGAS	INT(11)	YES	FOREING	Apuntando hacia GAS
DESCRIPCIÓN DE TABLA:	Tabla para identificar qué usuarios recibirán notificaciones por correo electrónico, y de qué tipo de información. Si un usuario no aparece en esta tabla, significa que no recibirá notificaciones.			

Cuadro 3.16 - Definición de tabla VOLCAN

TABLA:	VOLCAN			
CAMPO	TIPO	NULO	LLAVE	COMENTARIO
IDVOLCAN	INT(11)	NO	PRIMARY	
IDTIPOVOLCAN	INT(11)	NO	FOREING	Apuntando hacia TIPOVOLCAN
NOMBREVOLCAN	VARCHAR(50)	NO		
LOCALIZACION	VARCHAR(50)	NO		
ALTURA	DECIMAL(10,0)	NO		
DESCRIPCIÓN DE TABLA:	Tabla que centraliza información descriptiva de un volcán a tener en observación y monitoreo			

### 3.2.4. Sistema de Gestor de Base de Datos

“Un Sistema Gestor de Bases de Datos (SGBD) o DGBA (Data Base Management System) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. A través de él se maneja todo acceso a la base de datos con el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

Gracias a este sistema de software invisible para el usuario final, compuesto por un lenguaje de definición de datos, un lenguaje de manipulación y de consulta, es posible gestionar los datos a distintos niveles. Tanto almacenar, modificar y acceder a la información como realizar consultas y hacer análisis para generar informes.

Básicamente, el gestor controla cualquier operación ejecutada por el usuario contra la base de datos. Para ello, se utilizan herramientas específicas, como sistemas de búsqueda y de generación de informes, además de distintas aplicaciones.” (Power Data, 2016)

Para la implementación de nuestro sistema, es necesaria la elección de un sistema gestor de base de datos que nos permita tener acceso a la información en todo momento, y que además, garantice la durabilidad, fiabilidad y persistencia de los datos. La tabla 3.6, muestra una comparativa entre los puntos importantes a tener en cuenta para la selección de un gestor, entre sistemas populares en el medio.

*Tabla 3.6 - Comparación entre sistemas gestores de base de datos*

<b>Aspectos a considerar</b>	<b>MySQL</b>	<b>Oracle</b>	<b>PostgreSQL</b>	<b>SQL Server</b>
Tipo de licencia	Código Libre	Propietario	Código Libre	Propietario
Multiplataforma	SI	SI	SI	NO
Interfaz gráfica nativa	NO	NO	NO	SI
Software Interfaz Gráfica	MySQL Workbench	SQL Developer	pgAdmin	SQL Server Management Wizard
Instalación	Wizard	Compleja	Wizard	Wizard
Lenguajes de programación	C, C#, C++, Java, Ruby, y Objective C, entre otros...	C, C#, C++, Java, Ruby, y Objective C, entre otros...	C, C#, C++, Java, Ruby, y Objective C, entre otros...	Java, Ruby, Python, VB, .Net, and PHP entre otros...
Grado de Documentación	Alto	Alto	Medio	Medio
Grado de conocimiento del entorno de parte del equipo de trabajo	Alto	Alto	Bajo	Bajo

Debido a que no existe un presupuesto designado para el desarrollo y puesta en marcha del presente proyecto, se ha optado elegir por cualquier opción de software

de código libre que nos permitan tener acceso a la base. Esto facilita nuestra opción a MySQL o PostgreSQL.

A pesar del alto poder de procesamiento, la arquitectura que optimiza las consultas, y la robusta implementación de Oracle, no podrá ser considerada debido a los costos que éste le generaría al proyecto.

Al tener que elegir entre MySQL y PostgreSQL, la decisión se acorta al hecho que el equipo de trabajo, tiene más experiencia trabajando con este gestor. Esto, sin duda alguna representa una ventaja de cara al desarrollo de la solución, ya que acortará los tiempos de autoaprendizaje o capacitación que serán necesarios para conocer la herramienta.

“MySQL es la base de datos de código abierto más popular del mercado. Gracias a su rendimiento probado, a su fiabilidad y a su facilidad de uso, MySQL se ha convertido en la base de datos líder elegida para las aplicaciones basadas en web y utilizada por propiedades web de perfil alto, como Facebook, Twitter y YouTube. Además, es una elección muy popular como base de datos integrada, distribuida por miles de proveedores de software. Sus características la convierten en líder de software gratuito para gestión de bases de datos:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Condición de open source de MySQL hace que la utilización sea gratuita y se puede modificar con total libertad.

- Se puede descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones.
- Es una de las herramientas más utilizadas por los programadores orientados a Internet.
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- MYSQL, es el manejador de base de datos considerado como el más rápido de Internet.
- Gran rapidez y facilidad de uso.
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- Fácil instalación y configuración.” (Oracle, 2017)

“Ademas ofrece:

*Escalabilidad y flexibilidad:* El servidor de bases de datos MySQL ofrece lo último en escalabilidad, siendo capaz de manejar bases de datos empotradas ocupando sólo 1MB, y hacer funcionar data warehouses que contengan terabytes de información. La flexibilidad de plataforma es una característica clásica de MySQL, soportando distintas versiones de Linux, UNIX y Windows Y, por supuesto, la naturaleza open source de MySQL permite una personalización completa para aquellos que deseen añadir características al servidor.

*Alto rendimiento:* Una arquitectura única de motores de bases de datos permite a los profesionales configurar el servidor MySQL para aplicaciones específicas, dando como resultado un rendimiento espectacular MySQL puede cumplir con las expectativas de rendimiento de cualquier sistema, ya sea un sistema de procesamiento transaccional de alta velocidad, o un sitio web de gran volumen sirviendo un billón de consultas diarias MySQL ofrece la munición adecuada para sistemas críticos mediante herramientas de carga de alta velocidad, índices full-text y otros mecanismos de mejora del rendimiento.

*Alta disponibilidad:* Solidez y disponibilidad constante son características distintivas de MySQL, con clientes confiando en ellas para garantizar el uptime en todo momento. MySQL ofrece una amplia variedad de soluciones de alta disponibilidad, desde replicación a servidores de cluster especializados, u ofertas de terceros.

*Robusto soporte transaccional:* MySQL ofrece uno de los motores de bases de datos transaccionales más potentes del mercado. Las características incluyen un soporte completo de ACID (atómica, consistente, aislada, duradera), bloqueo a nivel de filas, posibilidad de transacciones distribuidas, y soporte de transacciones con múltiples versiones donde los lectores no bloquean a los escritores y viceversa. También se asegura una integridad completa de los datos mediante integridad referencial, niveles de aislamiento de transacciones especializados, y detección de deadlocks.

*Fortalezas en Web y Data Warehouse:* MySQL es el estándar de-facto para sitios web de gran tráfico por su motor de consultas de alto rendimiento, su posibilidad de insertar datos a gran velocidad, y un buen soporte para funciones web especializadas como las búsquedas fulltext. Estas mismas fortalezas también se aplican a entornos de data warehousing, donde MySQL escala hasta el rango de los terabytes tanto para un solo servidor, como para varios. Otras características como las tablas en memoria, índices B-tree y hash, y tablas comprimidas hasta un 80% hacen de MySQL una buena opción para aplicaciones web y de business intelligence.

*Fuerte protección de datos:* Porque proteger los datos es el trabajo principal de los profesionales de bases de datos, MySQL ofrece características de seguridad que aseguran una protección absoluta de los datos. En cuanto a autenticación, MySQL ofrece potentes mecanismos para asegurar que sólo los usuarios autorizados tienen acceso al servidor. También se ofrece soporte SSH y SSL para

asegurar conexiones seguras. Existe una estructura de privilegios que permite que los usuarios sólo puedan acceder a los datos que se les permite, así como potentes funciones de cifrado y descifrado para asegurarse de que los datos están protegidos. Finalmente, se ofrecen utilidades de backup y recuperación por parte de MySQL y terceros, que permiten copias completas, tanto lógicas como físicas, así como recuperación point-in-time.

*Desarrollo de aplicaciones completo:* Uno de los motivos por los que MySQL es la base de datos open source más popular es que ofrece un soporte completo para cualquier necesidad de desarrollo. En la base de datos se puede encontrar soporte para procedimientos almacenados, triggers, funciones, vistas, cursores, SQL estándar, y mucho más. Existen librerías para dar soporte a MySQL en aplicaciones empotradas. También se ofrecen drivers (ODBC, JDCBC,...) que permiten que distintos tipos de aplicaciones puedan usar MySQL como gestor de bases de datos. No importa si es PHP, Perl, Java, Visual Basic, o .NET, MySQL ofrece a los desarrolladores todo lo que necesitan para conseguir el éxito en el desarrollo de sistemas de información basados en bases de datos.

*Facilidades de gestión:* MySQL ofrece posibilidades de instalación excepcionales, con un tiempo medio desde la descarga hasta completar la instalación de menos de quince minutos. Esto es cierto sin importar que la plataforma sea Windows, Linux, Macintosh, o UNIX. Una vez instalado, características de gestión automáticas como expansión automática del espacio, o los cambios dinámicos de configuración descargan parte del trabajo de los atareados administradores. MySQL también ofrece una completa colección de herramientas gráficas de gestión que permiten al DBA gestionar, controlar y resolver problemas en varios servidores desde una misma estación de trabajo. Además, hay multitud de herramientas de terceros que gestionan tareas como el diseño de datos y ETL, administración, gestión de tareas y monitorización.

*Open Source y soporte 24 / 7:* Muchas empresas no se atreven a adoptar software open source porque creen que no podrán encontrar el tipo de soporte o servicios profesionales en los que confían con su software propietario actual. Las preguntas sobre indemnizaciones también aparecen. Estas preocupaciones pueden desaparecer con el completo servicio de soporte e indemnización disponibles. MySQL no es un proyecto típico Open Source ya que todo el software es propiedad de MySQL AB, lo que permite un modelo de coste y soporte que ofrece una combinación única entre la libertad del open source y la confianza de un software con soporte.

*Coste Total de Propiedad menor:* Al migrar aplicaciones actuales a MySQL, o usar MySQL para nuevos desarrollos, las empresas están ahorrando costes que muchas veces llegan a las siete cifras. Las empresas están descubriendo que, gracias al servidor MySQL y las arquitecturas scale-out que utilizan hardware económico, pueden alcanzar niveles sorprendentes de escalabilidad y rendimiento, y todo a un coste bastante menor que el de los sistemas propietarios. Además, la robustez y facilidad de mantenimiento de MySQL implican que los administradores no pierden el tiempo con problemas de rendimiento o disponibilidad, sino que pueden concentrarse en tareas de mayor impacto en el negocio.

## **3.3. DISEÑO DE COMPONENTES DE SOFTWARE**

### ***3.3.1. Entornos y Lenguaje de Programación***

Antes de iniciar a mostrar el diseño del sistema a crear, y de empezar a plantear todo lo que conlleve el mismo, es necesario dejar por sentado un punto muy claro antes de dar inicio. Debemos determinar en qué lenguaje de programación serán desarrolladas las aplicaciones in house, que formarán parte de la arquitectura de nuestra aplicación.

Hay 2 puntos muy importantes a tomar en cuenta que nos ayudarán a delimitar las opciones de lenguaje de programación, y en base a esta decisión, los entornos de software a utilizar para llevar a cabo el desarrollo del proyecto.

El primer punto, es que, el área usuario de nuestra aplicación, el Departamento de Física de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente; no expone como tal, la necesidad de utilizar algún lenguaje de programación en específico. Al área como tal, únicamente le interesa, que todo el desarrollo realizado, sea de código libre, para que cualquier otro equipo de trabajo en un futuro, pueda retomarlo y mejorarlo conforme las necesidades de software cambien en el tiempo. A modo de resumen, al área no le interesa qué lenguaje de programación se escoja para la realización del proyecto.

El segundo punto, ligado a punto expuesto anteriormente, delimita a que se pueda seleccionar un lenguaje de programación con licencia libre; que se pueda utilizar algún entorno IDE de forma gratuita y que no incurra en costos para el proyecto como tal; esto, debido a que el área usuario no dispone de ningún presupuesto para el desarrollo del proyecto.

A estos tópicos, hay que agregarle el hecho que la aplicación a desarrollar debe ser multiplataforma, por lo que se necesita un lenguaje que soporte la compilación de su código en entornos de ejecución de aplicaciones multiplataforma. Además,

hay que tomar en cuenta, la creación de un sitio web para mostrar la información recopilada de cara al usuario; por lo que el lenguaje a elegir, debe contar con soporte a desarrollo web.

Tomando en cuenta los puntos considerados, la lista de lenguajes de programación a elegir, se reduce a lenguajes de programación de código libre, con entornos de desarrollo gratuitos, que sea multiplataforma y que permita el desarrollo web. Algunos lenguajes que cumplen con estas condiciones son los siguientes.

*Tabla 3.7 - Comparación entre lenguajes de programación*

<b>Aspectos a considerar</b>	<b>Java</b>	<b>Python</b>	<b>Ruby</b>
Tipo de licencia	Código Libre	Código Libre	Código Libre
Multiplataforma	SI	SI	SI
Soporte POO	SI	SI	SI
Conexión con MySQL	SI	SI	SI
IDE dedicado con compilador	Netbeans, Eclipse	IDLE	RubyMine
Grado de Documentación	Alto	Alto	Medio
Grado de conocimiento del entorno de parte del equipo de trabajo	Alto	Bajo	Bajo

La tabla 3.7, muestra un comparativo rápido. Podemos ver fácilmente, que las condiciones de cada lenguaje son similares, y que cualquiera de ellos podría ser elegido para modelar la aplicación.

Debido a que no existe como tal, una restricción en el lenguaje de programación a utilizar; el equipo, ha decidido a conveniencia utilizar Java para la creación de componentes in house. Esta elección, se realiza debido a que es el lenguaje con el que el equipo más se familiariza y que por ende, requerirá de menor tiempo en la capacitación del mismo, otorgando una curva de aprendizaje del lenguaje bastante baja.

### **3.3.1.1. Java**

“Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

La versión más reciente de Java contiene importantes mejoras para el rendimiento, estabilidad y seguridad de las aplicaciones Java que se ejecutan en su equipo. La instalación de la actualización gratuita garantiza que sus aplicaciones Java sigan ejecutándose de forma segura y eficaz.” (Java, 2017)

“La tecnología Java se usa para desarrollar aplicaciones para un amplio alcance de entornos, desde dispositivos del consumidor hasta sistemas empresariales heterogéneos. En esta sección, obtenga una vista de alto nivel de la plataforma Java y sus componentes.

Como cualquier lenguaje de programación, el lenguaje Java tiene su propia estructura, reglas de sintaxis y paradigma de programación. El paradigma de programación del lenguaje Java se basa en el concepto de programación orientada a objetos (OOP), que las funciones del lenguaje soportan.

El lenguaje Java es un derivado del lenguaje C, por lo que sus reglas de sintaxis se parecen mucho a C: por ejemplo, los bloques de códigos se modularizan en métodos y se delimitan con llaves ({ y }) y las variables se declaran antes de que se usen.

Estructuralmente, el lenguaje Java comienza con paquetes. Un paquete es el mecanismo de espacio de nombres del lenguaje Java. Dentro de los paquetes se encuentran las clases y dentro de las clases se encuentran métodos, variables,

constantes, entre otros. En este tutorial, aprenderá acerca de las partes del lenguaje Java.” (Perry, 2012)

“Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa. Con más de 9 millones de desarrolladores en todo el mundo, Java le permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios.

Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

- El 97% de los escritorios empresariales ejecutan Java
- El 89% de los escritorios (o computadoras) en Estados Unidos ejecutan Java
- 9 millones de desarrolladores de Java en todo el mundo
- La primera opción para los desarrolladores
- La primera plataforma de desarrollo
- 3 mil millones de teléfonos móviles ejecutan Java
- El 100% de los reproductores de Blu-ray incluyen Java
- 5 mil millones de Java Cards en uso
- 125 millones de dispositivos de televisión ejecutan Java
- 5 de los 5 principales fabricantes de equipos originales utilizan Java ME

Java ha sido probado, ajustado, ampliado y probado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Java está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible. Al poner a disposición de todo el mundo aplicaciones en entornos heterogéneos, las empresas pueden proporcionar más servicios y mejorar la productividad, las comunicaciones y colaboración del usuario final y reducir drásticamente el costo de

propiedad tanto para aplicaciones de usuario como de empresa. Java se ha convertido en un valor impagable para los desarrolladores, ya que les permite:

- Escribir software en una plataforma y ejecutarla virtualmente en otra
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico.” (Java, 2017)

“Cuando usted programa para la plataforma Java, escribe el código de origen en archivos .java y luego los compila. El compilador verifica su código con las reglas de sintaxis del lenguaje, luego escribe los códigos byte en archivos .class. Los códigos byte son instrucciones estándar destinadas a ejecutarse en una Java Virtual Machine (JVM). Al agregar este nivel de abstracción, el compilador Java difiere de los otros compiladores de lenguaje, que escriben instrucciones apropiadas para el chipset de la CPU en el que el programa se ejecutará.” (Perry, 2012)

### ***3.3.1.2. Java Virtual Machine***

“Java Virtual Machine (JVM) es un conjunto de programas de software que permiten la ejecución de instrucciones y que normalmente están escritos en código byte de Java. Las máquinas virtuales de Java están disponibles para las plataformas de hardware y software de uso más frecuente.” (Java, 2017)

“Java Virtual Machine es sólo un aspecto del software de Java que interviene en interacción web. Java Virtual Machine está incorporado en la descarga del software de Java y ayuda a ejecutar las aplicaciones Java.” (Java, 2017)

“Al momento de la ejecución, la JVM lee e interpreta archivos .class y ejecuta las instrucciones del programa en la plataforma de hardware nativo para la que se escribió la JVM. La JVM interpreta los códigos byte del mismo modo en que una CPU interpretaría las instrucciones del lenguaje del conjunto. La diferencia es que la JVM es un software escrito específicamente para una plataforma particular. La JVM es el corazón del principio "escrito una vez, ejecutado en cualquier lugar" del lenguaje Java. Su código se puede ejecutar en cualquier chipset para el cual una implementación apropiada de la JVM está disponible. Las JVM están disponibles para plataformas principales como Linux y Windows y se han implementado subconjuntos del lenguaje Java en las JVM para teléfonos móviles y aficionados de chips.

En lugar de forzarlo a mantenerse a la par con la asignación de memoria (o usar una biblioteca de terceros para hacer esto), la plataforma Java proporciona una gestión de memoria lista para usar. Cuando su aplicación Java crea una instancia de objeto al momento de ejecución, la JVM asigna automáticamente espacio de memoria para ese objeto desde el almacenamiento dinámico, que es una agrupación de memoria reservada para que use su programa. El recolector de basura Java se ejecuta en segundo plano y realiza un seguimiento de cuáles son los objetos que la aplicación ya no necesita y recupera la memoria que ellos ocupan. Este abordaje al manejo de la memoria se llama gestión de la memoria implícita porque no le exige que escriba cualquier código de manejo de la memoria. La recogida de basura es una de las funciones esenciales del rendimiento de la plataforma Java.” (Perry, 2012)

“Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados

- Liberar la memoria no usada (garbage collection).
- Asignar variables a registros y pilas
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java

Esta última tarea, es una de las más importantes que realiza la JVM. Además, las propias especificaciones del lenguaje Java contribuyen extraordinariamente a este objetivo:

- Las referencias a arrays son verificadas en el momento de la ejecución del programa
- No hay manera de manipular de forma directa los punteros
- La JVM gestiona automáticamente el uso de la memoria, de modo que no queden huecos.
- No se permiten realizar ciertas conversiones (casting) entre distintos tipos de datos.” (Universidad del País Vasco, 2017)

### ***3.3.1.3. Java Runtime Environment***

“El Java Runtime Environment (JRE, también conocido como el Java Runtime) incluye las bibliotecas de códigos de la JVM y los componentes que son necesarios para programas en ejecución escritos en el lenguaje Java. Está disponible para múltiples plataformas. Puede redistribuir libremente el JRE con sus aplicaciones, de acuerdo a los términos de la licencia del JRE, para darles a los usuarios de la aplicación una plataforma en la cual ejecutar su software.” (Perry, 2012)

“Java Runtime Environment (JRE) es lo que se obtiene al descargar el software de Java. JRE está formado por Java Virtual Machine (JVM), clases del núcleo de la plataforma Java y bibliotecas de la plataforma Java de soporte. JRE es la parte de

tiempo de ejecución del software de Java, que es todo lo que necesita para ejecutarlo en el explorador web.” (Java, 2017)

“La Caché de Java Runtime Environment, es el área de almacenamiento de la consola de Java que, en ocasiones, se debe borrar de forma manual para permitir la carga e instalación de la última versión de Java.” (Java, 2017)

#### **3.3.1.4. *Java Developer Kit***

“Se trata de un conjunto de herramientas (programas y librerías) que permiten desarrollar (compilar, ejecutar, generar documentación, etc.) programas en lenguaje Java.

Existen versiones del JDK para prácticamente todos los Sistemas Operativos y existen también distintos programas comerciales. Los JDK incorporan una herramienta de Debugger (detener la ejecución de un programa en la línea que se desee y poder conocer el valor de las variables en ese momento)” (Universidad de Navarra, 2003)

“Cuando usted descarga un kit de desarrollo de Java (JDK), obtiene, — además del compilador y otras herramientas, — una librería de clase completa de programas de utilidad preconstruidos que lo ayudan a cumplir cualquier tarea común al desarrollo de aplicaciones. El mejor modo para tener una idea del ámbito de los paquetes y bibliotecas JDK es verificar la documentación API JDK.

El JRE se incluye en el JDK.

El JDK incluye un conjunto de herramientas de línea de comandos para compilar y ejecutar su código Java, que incluye una copia completa del JRE. Aunque usted ciertamente puede usar estas herramientas para desarrollar sus aplicaciones, la mayoría de los desarrolladores valoran la funcionalidad adicional, la gestión de tareas y la interfaz visual de un IDE.” (Perry, 2012)

“El Kit de desarrollo de Java (JDK) contiene las herramientas y librerías necesarias para crear y ejecutar aplicaciones en Java. A continuación se listan algunas de las utilidades que se pueden encontrar en el JDK:

- *javac*: Es el compilador de Java. Se encarga de convertir el código fuente escrito en Java a bytecode.
- *java*: Es el intérprete de Java. Ejecuta el bytecode a partir de los archivos class.
- *appletviewer*: Es un visor de applets. En la mayoría de las ocasiones puede utilizarse en lugar de un Navegador Web.
- *javadoc*: Se utiliza para crear documentación en formato HTML a partir de el código fuente Java y los comentarios que contiene.
- *javap*: Es un desensamblador de Java.
- *jar*: Es una herramienta utilizada para trabajar con los archivos JAR.”  
(Universidad de Chile, 2017)

### **3.3.1.5. NetBeans IDE**

“NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Al día de hoy hay disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.” (NetBeans, 2017)

“NetBeans IDE es un entorno de desarrollo integrado (IDE), modular, de base estandar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. Posee las siguientes características:

- Soporte JavaScript:
  - Sintaxis Resaltada
  - Completación de Código y Análisis de Tipeo
  - Soluciones Rápidas (Quick Fixes) y Verificación de Sintaxis
  - Refactorización
  
- Mejoras en el Desempeño:
  - Inicio hasta 40% más rápido
  - Promociones más inteligentes, así que la completación de código es más rápida
  - Menor consumo de memoria
  
- Nuevo Soporte MySQL en Exploración de Bases de Datos:

- Registro de servidores MySQL
- Ver, crear y borrar bases de datos
- Fácil lanzamiento de la herramienta de administración para MySQL
  
- Soporte Ruby/JRuby:
  - Mejoras en el editor, incluyendo nuevas sugerencias y soluciones
  - Soporte de depuración rápida en JRuby
  - Administrador de Plataforma
  - Mejoras en la integración de servidores y bases de datos en proyectos Rails
  
- Completación de Código Javadoc:
  - Soporte de etiquetas (tags) estandares: @param, etc.
  - Completación de Código para parámetros, excepciones, etc.
  
- Soporte para los Web APIs Más Usados:
  - Fácil creación de aplicaciones remezcladas (mashup)
  - Operaciones de Arrastrar y soltar dentro del entorno POJO, Servlet, JSP y servicios web RESTful para que NetBeans IDE genere todo el código para acceder a los servicios
  - Soporte de web APIs tales como Google, Facebook, Yahoo y YouTube.” (NetBeans, 2017)

### ***3.3.2. Diseño y Desarrollo de Componentes In House***

La figura 3.3, nos daba una vista rápida y clara de los componentes de software in house a necesitar para el correcto funcionamiento del sistema que se idealiza implementar. En ese apartado, hablábamos un poco respecto a que el software in house de la solución, serían componentes a ser desarrollados por parte de equipo

de trabajo. El presente apartado, pretende delimitar los alcances de cada elemento, así como la importancia de cada uno de ellos.

La necesidad de la creación de estas aplicaciones, recae en la necesidad de recopilar y transformar los resultados de mediciones de las diferentes fuentes de datos de la aplicación, y poder unificar todo en una sola base de datos, que permita tener lectura de todo el flujo de información para, posteriormente, mostrar en pantalla los resultados al usuario.

### ***3.3.2.1. MAVAMSI\_BOT2***

La figura 3.4, nos mostraba los componentes bajo los cuales labora MAVAMSI\_BOT2. El flujo de extracción de emisiones de gases, emerge con este proceso.

Cabe aclarar, que a pesar que en la secuencia del flujo de información, MAVAMSI\_BOT2 se debe ejecutar antes que MAVAMSI\_BOT1, el nombre obedece a que éste fue el segundo robot que empezó a desarrollarse por parte del equipo, y que por orden de robustez y ejecución, el proceso MAVAMSI\_BOT1 representa un grado mayor de atención, por lo que decidió llamarlo con el sufijo 1.

MAVAMSI\_BOT2, es una pequeña aplicación de escritorio creada en Java. Es un proceso automático que tiene el fin de recoger los archivos de mediciones de emisiones de gases, directamente del ordenador configurado para comunicarse con el sensor ScanDoas; al identificar las mediciones tomadas en la fecha actual, las recopila de la carpeta de salida de la máquina y entabla comunicación con el servidor de nuestra aplicación (AMAZON\_VM) vía protocolo FTP.

El trabajo de la aplicación, a nivel de software es bastante simple: realizar una búsqueda de archivos en la carpeta especificada; entablar comunicación con servidor AMAZON\_VM vía FTP; realizar el traslado de archivo a carpeta de almacenamiento interno de servidor, cerrar comunicación; imprimir un breve log de la transacción realizada.

El desarrollo de este elemento, será bastante simple. Debido a la simpleza que presenta su diseño, no será necesaria la creación de una estructura de clases orientada a objetos, sino que bastará con un código bajo el paradigma de programación estructurada.

“La visión clásica de la programación estructurada se refiere al control de ejecución. El control de su ejecución es una de las cuestiones más importantes que hay que tener en cuenta al construir un programa en un lenguaje de alto nivel. La regla general es que las instrucciones se ejecuten sucesivamente una tras otra, pero diversas partes del programa se ejecutan o no dependiendo de que se cumpla alguna condición. Además, hay instrucciones (los bucles) que deben ejecutarse varias veces, ya sea en número fijo o hasta que se cumpla una condición determinada.

Esta forma de programar se basa en un famoso teorema, desarrollado por Edsger Dijkstra, que demuestra que todo programa puede escribirse utilizando únicamente las tres estructuras básicas de control siguientes:

- *Secuencia*: el bloque secuencial de instrucciones, instrucciones ejecutadas sucesivamente, una detrás de otra.
- *Selección*: la instrucción condicional con doble alternativa, de la forma "if condición then instrucción-1 else instrucción-2".
- *Iteración*: el bucle condicional "while condición do instrucción", que ejecuta la instrucción repetidamente mientras la condición se cumpla.

Los programas que utilizan sólo estas tres instrucciones de control básicas o sus variantes (como los bucles for, repeat o la instrucción condicional switch-case), pero no la instrucción goto, se llaman estructurados.” (Aguilar, 2017)

Cabe mencionar que, a pesar que el método de desarrollo de la funcionalidad del robot será estructurado, la aplicación como tal, no pierde el paradigma de programación orientada a objetos. Esto se instancia de forma que, para el

funcionamiento del mismo, será necesaria la implementación de una librería con clases predefinidas, que ayudarán a entablar la comunicación vía FTP con el servidor, estamos hablando de la librería Apache Commons Net.

“La librería Apache Commons Net™ implementa el lado del cliente de muchos protocolos básicos de Internet. El propósito de la biblioteca es proporcionar acceso fundamental al protocolo, no abstracciones de nivel superior. Por lo tanto, parte del diseño viola los principios de diseño orientados a objetos. Nuestra filosofía es hacer accesible la funcionalidad global de un protocolo (por ejemplo, FTP para enviar y recibir un archivo) cuando sea posible, pero también proporcionar acceso a los protocolos fundamentales donde sea aplicable para que el programador pueda construir sus propias implementaciones personalizadas.

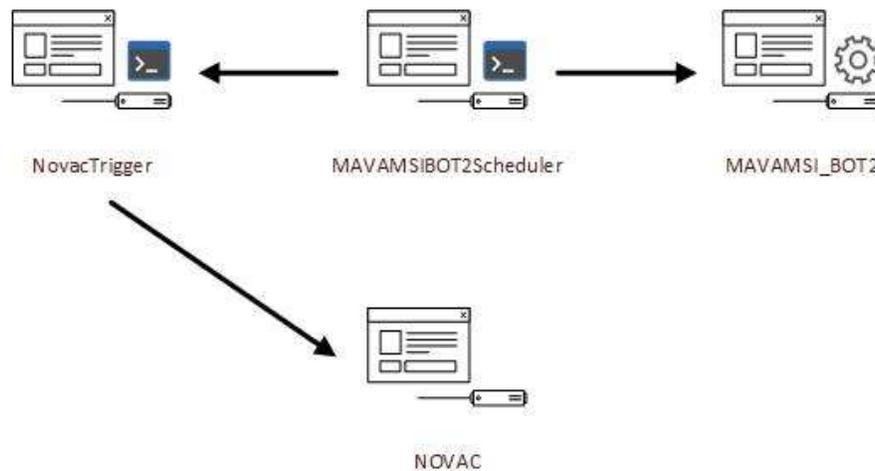
Apache Commons Net comenzó como una librería comercial de Java llamada NetComponents, originalmente desarrollada por ORO, Inc. en los primeros días de Java. Después de su versión 1.3.8 en 1998, el código fuente fue donado a la Apache Software Foundation y disponible bajo la Licencia Apache. Desde entonces, muchos programadores han contribuido al desarrollo continuo de Commons Net. El esquema de numeración de la versión actual no guarda ninguna relación con el antiguo. En otras palabras, Commons Net 1.0 tuvo éxito y suplantó NetComponents 1.3.8.” (Apache Commons, 2017)

La versión actual de la librería es la 3.5, la cual, ha sido utilizada en el desarrollo de MAVAMSI\_BOT2, para ayudar a entablar la relación entre máquina de estación y servidor vía FTP.

La ejecución de este robot, se realizará en la máquina que reciba los datos del sensor de mediciones de gases (MAQUINA\_ESTACION). El robot se ejecutará continuamente, según el tiempo que los analistas de información consideren necesario; ya que el robot, enviará al servidor los archivos conforme los vaya generando Novac Program.

La ejecución de este programa, la realizará un proceso batch, el cual, se encargará de ejecutar Novac Program para realizar lecturas de emisiones de gases. Otro proceso batch, se encargará de tener un temporizador para interrumpir la ejecución de Novac Program, y darle paso a la ejecución de MAVAMSI\_BOT2 para realizar la transferencia de archivo al servidor. La figura 3.10 ilustra a la perfección lo descrito anteriormente:

Figura 3.10 - Elementos que interactúan con MAVAMSI\_BOT2



El proceso batch MAVAMSIBOT2Scheduler, no será nada más que un programador de tareas. Se encargará de ejecutar un contador que determina cuanto tiempo deberá estar activo el proceso batch NovacTrigger.

NovacTrigger tendrá una función más que simple: ejecutar Novac Program. Se realizará de esta manera, para poder cerrar Novac Program de forma independiente, matando todos los procesos de ejecución de la aplicación. Así evitaremos que un archivo que no se halla decodificado correctamente quede corrupto o incompleto. Al volver a abrir la aplicación de Novac Program, volverá a tratar los archivos pendiente, tendiendo siempre versiones actualizadas de la información.

Finalmente, MAVAMSIBOT2Scheduler ejecutará a MAVAMSI\_BOT2, para iniciar la transferencia de los archivos al servidor FTP.

### **3.3.2.2. MAVAMSI\_BOT1**

La figura 3.5 nos mostraba cómo se relaciona MAVAMSI\_BOT1 con el resto de aplicaciones del sistema, para convertirse en el eje central del abastecimiento de datos hacia la base de nuestra aplicación.

MAVAMSI\_BOT1, que por definición, es un ETL complejo, se encarga de tomar directamente del almacenamiento en disco del servidor, todos los archivos producidos, tanto de emisiones de gases, como de movimiento sísmico; posteriormente se encarga de transformar los archivos en insumos para la base de datos de la aplicación; para finalmente, ingresar los datos a la base.

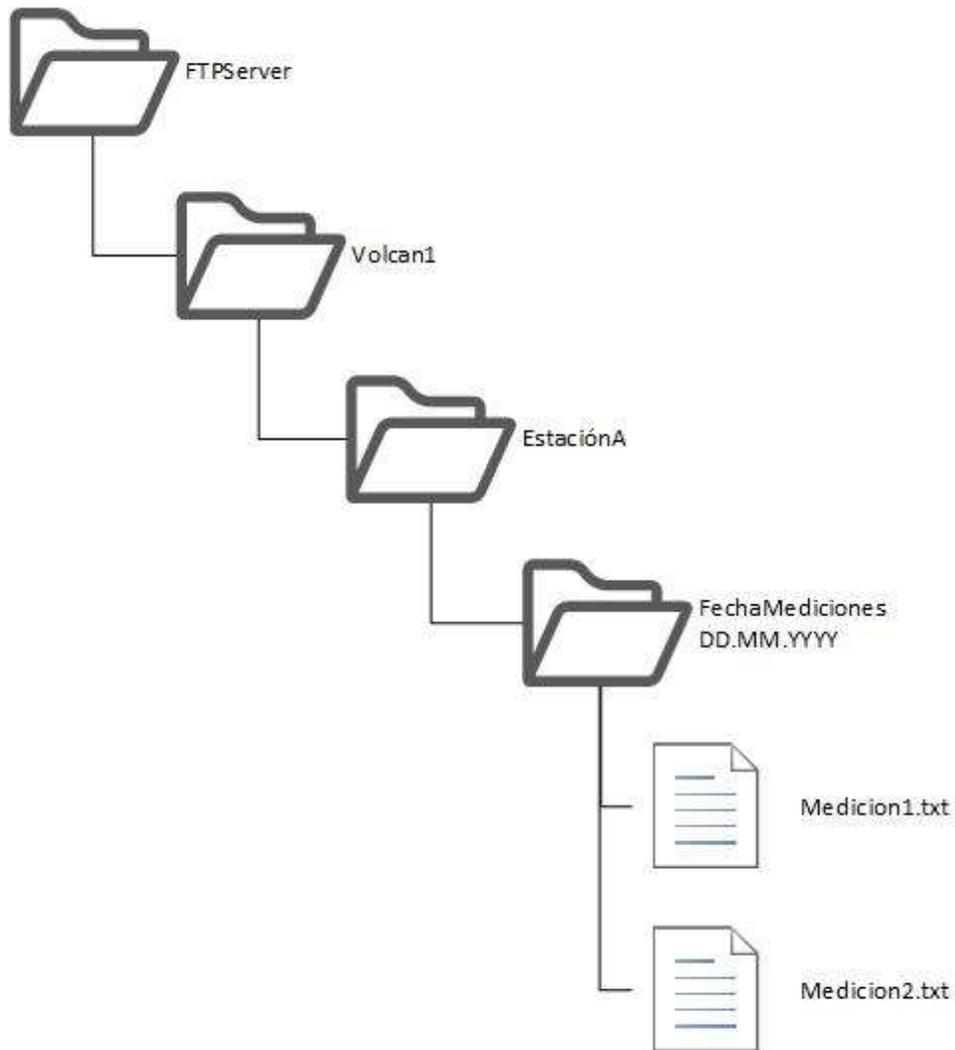
La estructura de software que MAVAMSI\_BOT1 necesita para funcionar, es más compleja que la de MAVAMSI\_BOT2. MAVAMSI\_BOT1, es por naturaleza, una aplicación orientada a objetos, con una estructura de clases definida, que hace de este robot, una aplicación robusta.

A diferencia de MAVAMSI\_BOT2, que necesitaba de un proceso batch para realizar tareas de programador de tareas, MAVAMSI\_BOT1, cuenta internamente con su propio contador. De esta forma, solo es necesario dejar ejecutando la aplicación en el servidor, y ésta se encargará de repetir las tareas de extracción bajo los rangos de tiempo definidos para realizar las tareas esperadas. Otra característica remarcada de este robot, es que su ejecución es única para el servidor de la aplicación, y no se ejecutará fuera de éste; como el caso de MAVAMSI\_BOT2, que debe ejecutarse en cada equipo de centro de observación que se conecte con un sensor de emisiones de gas.

El flujo de trabajo de MAVAMSI\_BOT1, inicia con la lectura de archivos en el almacenamiento principal de la aplicación; donde se encargará de revisar los archivos a ingresar a la base; tanto de gases, como de RSAM. Para ello, el almacenamiento interno del servidor cuenta con un sistema de carpetas, que le

ayudan al robot a identificar de donde proviene la información. Las figuras 3.11 y 3.12 muestran la estructura que necesita tener el almacenamiento interno para que MAVAMSI\_BOT1 pueda leer los archivos antes de iniciar su flujo de inserción en la base de datos.

*Figura 3.11 - Estructura de carpetas de archivos de emisión de gases en servidor*

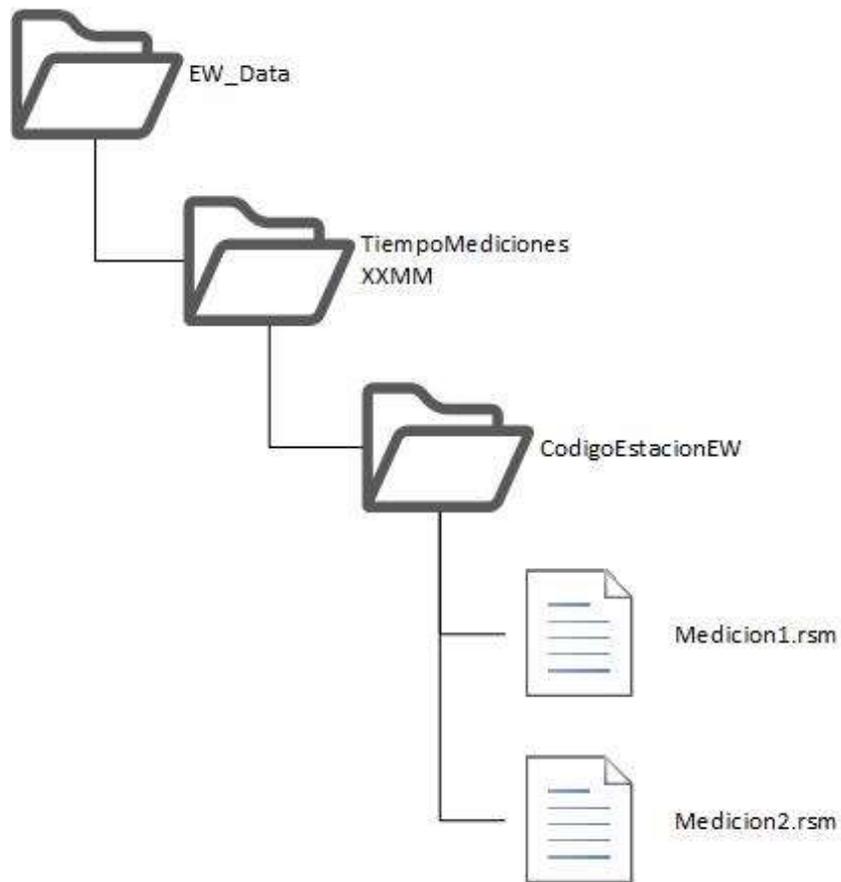


Con los archivos de gas, MAVAMSI\_BOT2 recorre las carpetas iniciando en FTPServer. El nombre de esta carpeta debe su nombre, a que esa es la carpeta padre designada para recibir archivos vía conexión FTP de parte de MAVAMSI\_BOT2.

La estructura de carpetas es creada por MAVAMSI\_BOT2; al momento de enviar los archivos, busca cada volcán y estación, según como se ha configurado en la máquina de la estación donde se capturan los datos.

MAVAMSI\_BOT1, verifica en cada carga de datos que las carpetas estén nombradas según los volcanes que están ingresados en la base de datos de la herramienta, esto con el fin de resguardar la información.

*Figura 3.12 - Estructura de carpetas de archivos de RSAM en servidor*



La estructura de carpetas para mediciones RSAM, tiene un nivel menos, ya que en Earthworm, basta con configurar la estación de donde provienen los datos. La estructura de la base de datos, que relaciona las tablas de estaciones y volcanes, ayuda a identificar a qué volcán pertenece cada estación.

Toda esta estructura de carpetas, ha sido creada con el fin de poder garantizar la escalabilidad de nuestra solución, dando la posibilidad de implementar análisis de cualquier volcán y de cualquier estación con total libertad, sin necesidad de adaptaciones a nivel código.

Una vez abordada la parte de extracción de la información, ahora hablemos un poco de la inserción de datos en la base. MAVAMSI\_BOT1, utiliza librerías propias de MySQL para poder comunicar la aplicación con la base de datos. Por medio de esta librería, podemos ingresar cada medición a la base, donde estará almacenada para su posterior consumo.

Para realizar esta comunicación entre la aplicación y la base, utilizamos las librerías de MySQL Connector/J.

“MySQL proporciona conectividad para aplicaciones de tipo cliente, desarrolladas en lenguaje de programación Java con MySQL Connector / J, un controlador que implementa la API JDBC (Java Database Connectivity).

MySQL Connector / J es un controlador JDBC tipo 4. Existen diferentes versiones compatibles con las especificaciones JDBC 3.0 y JDBC 4.x (consulte el Capítulo 2, Versiones Connector / J). La designación de Tipo 4 significa que el controlador es una implementación Java pura del protocolo MySQL y no se basa en las bibliotecas cliente de MySQL.

Para programas a gran escala que usan patrones de diseño comunes de acceso a datos, considere el uso de uno de los marcos populares de persistencia como Hibernate, las plantillas JDBC de Spring o MyBatis SQL Maps para reducir la cantidad de código JDBC para depurar, afinar, proteger y mantener.” (MySQL, 2017)

Recordando nuevamente la figura 3.5, podemos ver que MAVAMSI\_BOT1, además de registrar la información de emisiones de gas y RSAM en la base de datos, también se encarga de enviar alertas vía correo electrónico.

Esta necesidad nació como un requerimiento funcional de la aplicación, donde se hablaba de poder enviar correos en señal de alerta, a usuarios de la aplicación interesados en conocer de primera mano, un comportamiento anormal de un volcán en específico, en base a sus valores base que se pueden considerar como valores aceptables.

Para concretar este fin, MAVAMSI\_BOT1, también tendrá librerías de la API de Gmail que, permitirá enviar correos electrónicos contruidos desde la aplicación.

“La API de Gmail es una API que se puede utilizar para acceder a buzones de correo de Gmail y enviar correo. Para la mayoría de las aplicaciones web (incluidas las aplicaciones para móviles), la API de Gmail es la mejor opción para el acceso autorizado a los datos de Gmail de un usuario.

La API de Gmail brinda acceso flexible y REST en la bandeja de entrada del usuario, con una interfaz natural para Hilos, Mensajes, Etiquetas, Borradores, Historial y Configuración. Cualquier aplicación puede usar la API para agregar funciones de Gmail como:

- Leer mensajes de Gmail
- Enviar mensajes de correo electrónico
- Modificar las etiquetas aplicadas a mensajes e hilos
- Búsqueda de mensajes e hilos específicos
- Crear filtros para etiquetar, reenviar o archivar automáticamente los mensajes

Todo lo que se necesita para utilizar la API de Gmail es la biblioteca de cliente para su elección de idioma y una aplicación que puede autenticarse como usuario de Gmail.

La API de Gmail se puede utilizar en una variedad de aplicaciones diferentes, incluyendo, por lo general:

- Extracción, indexación y copia de seguridad de correo de sólo lectura
- Gestión de etiquetas (añadir / quitar etiquetas)
- Envío de mensajes automatizado o programático
- Migración de cuentas de correo electrónico de otros proveedores
- Establecer firmas de correo electrónico estandarizadas para usuarios en un dominio” (Google, 2017)

### **3.3.2.3. MAVAMSI Aplicación web**

Finalizando el análisis de componentes del sistema en la figura 3.1, tenemos una aplicación web, la cual, será el punto de vista bajo el que los usuarios realizaran sus análisis. Todo el proceso de extracción, transformación y carga de datos culmina aquí, en una aplicación web que presente todo este proceso en forma de gráficos y tablas fáciles de entender y visualizar por parte de los usuarios.

El funcionamiento de la aplicación web es simple: Leer los datos utilizados como filtros de parte del usuario, y utilizarlos para construir consultas de extracción de datos hacia la base de datos; utilizar el insumo devuelto por la base, para la construcción de tablas, gráficas y reportes que puedan mostrar el comportamiento de un volcán o estación para rangos de fechas determinados.

La aplicación web, llamada también MAVAMSI, es una aplicación creada para funcionar en entornos web, adaptables a cualquier tipo de dispositivo. Su funcionamiento, está determinado por sentencias de código Java, el cual, se encarga de conectar la aplicación a la base de datos por el mismo API de MySQL Connector /J que utiliza MAVAMSI\_BOT1.

Java siempre ha sido conocido por ofrecer a los usuarios diversos framework de programación a elegir, los cuales, unifican métodos y estándares de programación que ayudan a que el desarrollo sea óptimo y se apegue a los ideales de funcionamiento requeridos en los requerimientos no funcionales del presente

proyecto. Para el caso de MAVAMSI, se utilizará Java Server Faces como framework.

“JavaServer Faces (JSF) es un framework basado en el patrón MVC (Modelo Vista Controlador) para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java.

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente suelen incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Ofrece una clara separación entre el comportamiento y la presentación, lo que permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas.” (Universidad Carlos III de Madrid, 2008)

“La tecnología JavaServer Faces incluye:

- Un conjunto de API para representar los componentes de interfaz de usuario y administrar su estado, gestionar eventos y validación de entrada, definir la navegación de páginas y apoyar la internacionalización y la accesibilidad.
- Una biblioteca de etiquetas personalizadas de JavaServer Pages (JSP) para expresar una interfaz de JavaServer Faces dentro de una página JSP.

Diseñada para ser flexible, la tecnología JavaServer Faces aprovecha los conceptos existentes de interfaz de usuario y de nivel web sin limitar a los desarrolladores a un lenguaje de marcado, protocolo o dispositivo cliente concreto. Las clases de componente de interfaz de usuario incluidas con la tecnología JavaServer Faces encapsulan la funcionalidad del componente, no la presentación

específica del cliente, permitiendo así que los componentes de interfaz de usuario de JavaServer Faces se representen en varios dispositivos cliente. Al combinar la funcionalidad del componente de interfaz de usuario con los procesadores personalizados, que definen los atributos de representación para un componente de interfaz de usuario específico, los desarrolladores pueden crear etiquetas personalizadas para un dispositivo cliente concreto. Como comodidad, la tecnología JavaServer Faces proporciona un renderizador personalizado y una biblioteca de etiquetas personalizadas de JSP para renderizar a un cliente HTML, lo que permite a los desarrolladores de aplicaciones de Java Platform, Enterprise Edition (Java EE) utilizar la tecnología JavaServer Faces en sus aplicaciones.

La facilidad de uso que es el objetivo principal, la arquitectura JavaServer Faces claramente define una separación entre la lógica de la aplicación y la presentación al mismo tiempo que facilita la conexión de la capa de presentación al código de la aplicación. Este diseño permite a cada miembro de un equipo de desarrollo de aplicaciones web centrarse en su parte del proceso de desarrollo, y también proporciona un modelo de programación simple para vincular las piezas. Por ejemplo, los desarrolladores de páginas web sin experiencia en programación pueden utilizar las etiquetas de componentes de JavaServer Faces UI para vincularse al código de la aplicación desde una página web sin necesidad de escribir scripts.” (Oracle, 2017)

En próximos apartados, ahondaremos en qué estructura de clases se optó por seguir bajo éste framework que ha dado buenos resultados a diversas empresas e instituciones que ha optado por implementarlo.

Además de esta tecnología, la cual, se enfoca más del lado back-end de desarrollo de la aplicación; tenemos otro framework a implementar, como front-end, para que nos ayude a representar la información de forma ordenada y clara de cara al usuario; se trata de Prime Faces.

Cuando se utilizan la tecnología JSF para la vista de una aplicación Web, no podemos menos que sentir que te faltan componentes más potentes, y con más versatilidad. Para suplir estas carencias existen varias empresas que compiten con distintos frameworks, y que nos hacen la vida más fácil. PrimeFaces es una biblioteca de componentes para JSF de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2.

El punto fuerte de PrimeFaces es la sencillez de instalación y lo poco pesado que es. El mantenerlo liviano, sin complicaciones a la hora de instalarlo, es decir, sin dependencias ni configuraciones, hace que podamos estar usándolo en unos pocos segundos. Algunas de sus características principales son:

“Simplicidad y rendimiento: PrimeFaces es una biblioteca ligera, todas las decisiones tomadas se basan en mantener PrimeFaces lo más ligero posible. Por lo general, añadir una solución de terceros podría traer una sobrecarga sin embargo esto no es el caso con PrimeFaces.

Facilidad de uso: Los componentes de PrimeFaces se desarrollan con un principio de diseño que establece que "un buen componente de interfaz de usuario debe ocultar la complejidad pero mantener la flexibilidad".

Fuerte Retroalimentación Comunitaria; La comunidad PrimeFaces ayuda continuamente al desarrollo de PrimeFaces proporcionando retroalimentación, nuevas ideas, informes de errores y parches.” (PrimeFaces, 2017)

La utilización de estos framework, ayudará a que el desarrollo de MAVAMSI sea más simple, y le presente mayor riqueza al usuario al momento de consumir datos de la aplicación.

### ***3.3.3. Diagramas de Flujo***

A continuación, presentaremos diagramas de flujo mediante los cuales se explica el funcionamiento que se espera que tengan los procesos de software que se comentaba anteriormente que se necesitaba construir.

Con estos diagramas, buscamos resumir el flujo mediante el cual se guiará el funcionamiento de cada proceso determinado a construir, y que sirva de base para el desarrollo de los mismos. Cada diagrama, representa un proceso como tal en el cual, fluye información transversal a algún componente de software.

Las figuras comprendidas entre la figura 3.13 y la figura 3.21, describirán el flujo que deberá transitar la información en cada una de las fases de nuestra aplicación. La distribución de los diagramas estará dividida por aplicaciones para que su entendimiento sea más simple.

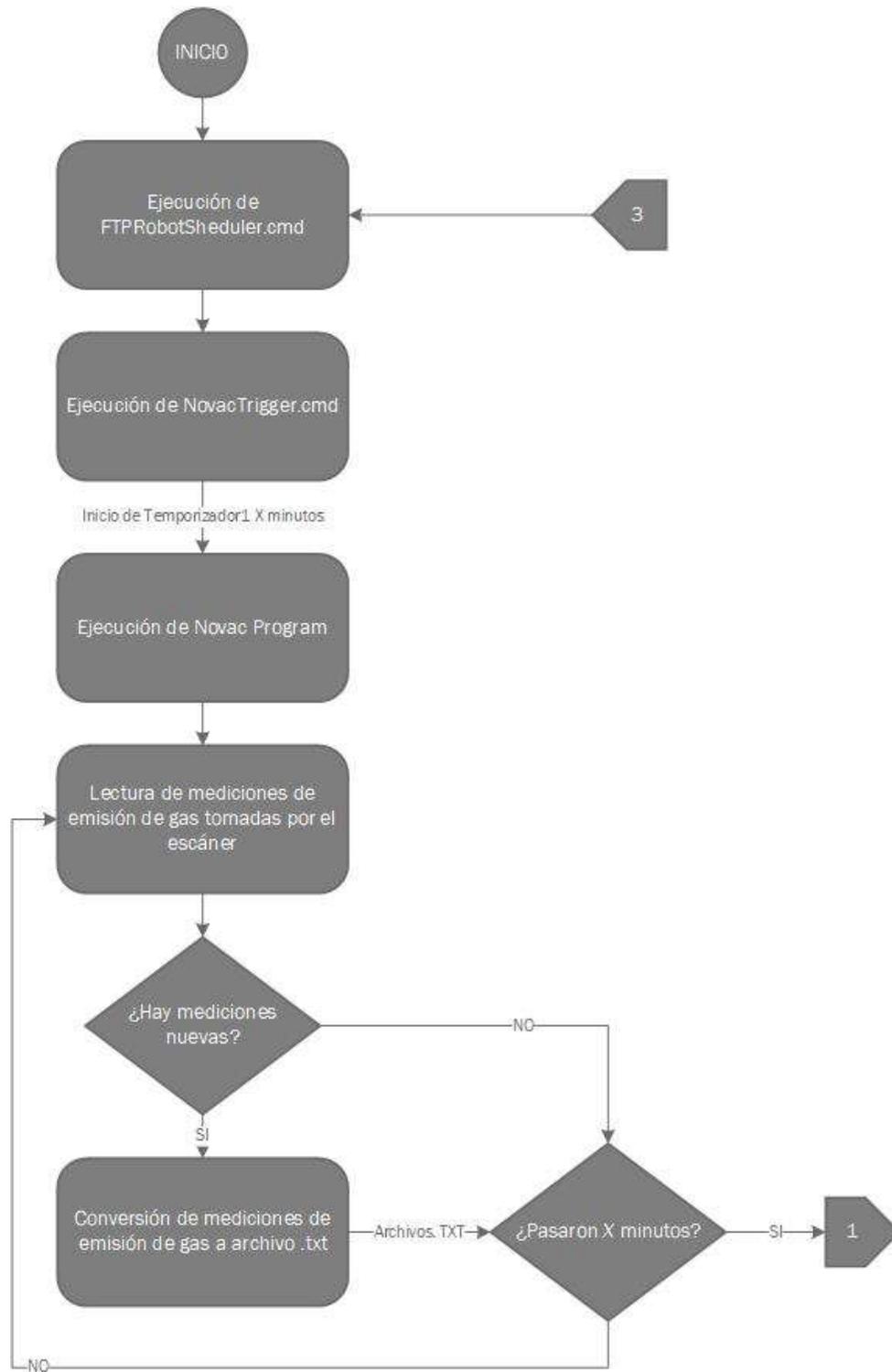
Primero, veremos el flujo a seguir de la información para MAVAMSI\_BOT2, donde se expondrá el paso a paso de ésta aplicación en cada máquina de estación de observación vulcanológica; desde la obtención de los datos de ScanDoas hasta su paso al servidor de nuestro sistema.

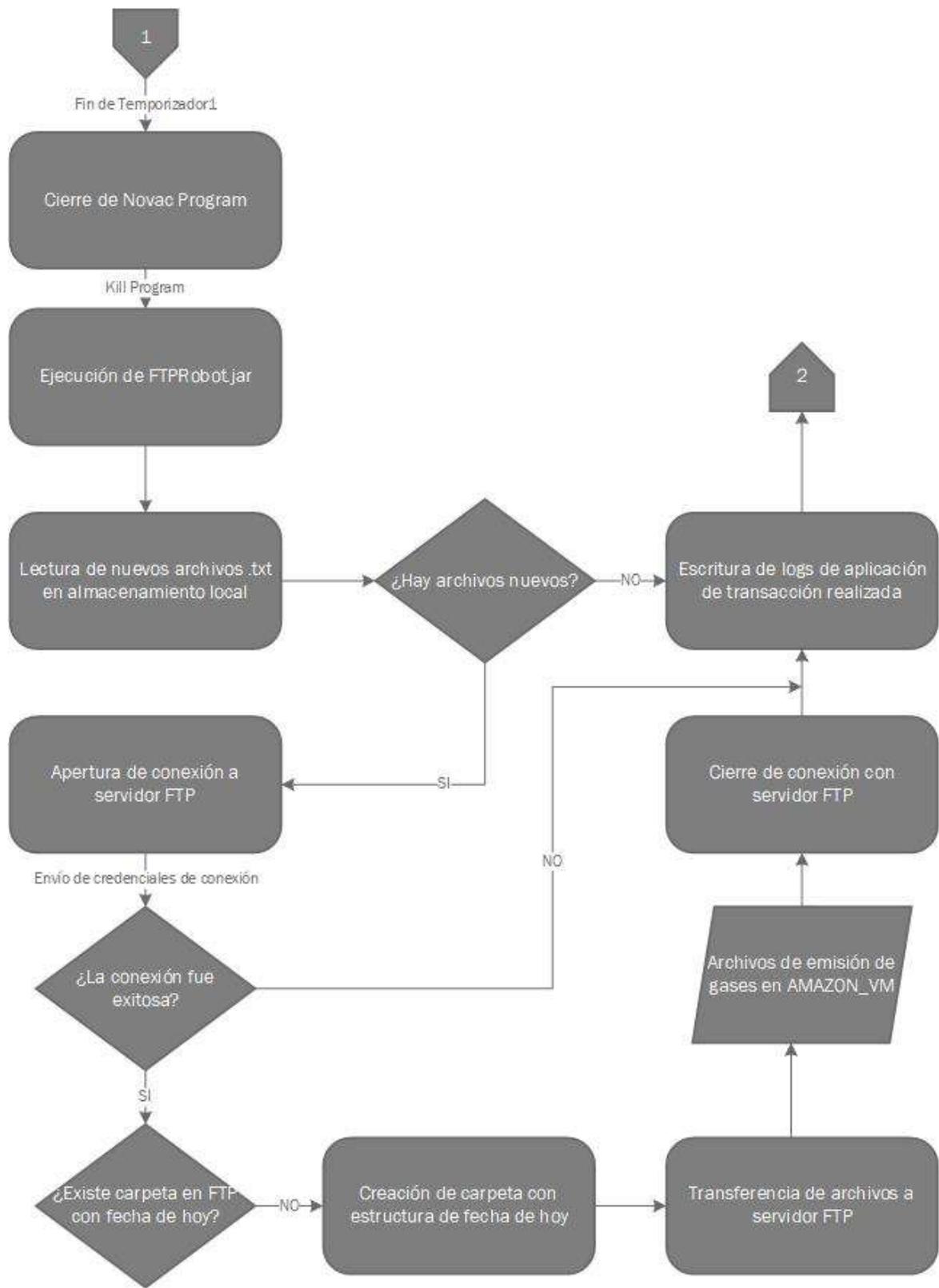
Posteriormente, veremos los flujos a seguir de MAVAMSI\_BOT1, donde se explicará el proceso de transformación de datos previo a su inserción a la base; descubriremos como se abastecerá la base de datos de nuestra aplicación y cómo funcionará el envío de correos de alerta.

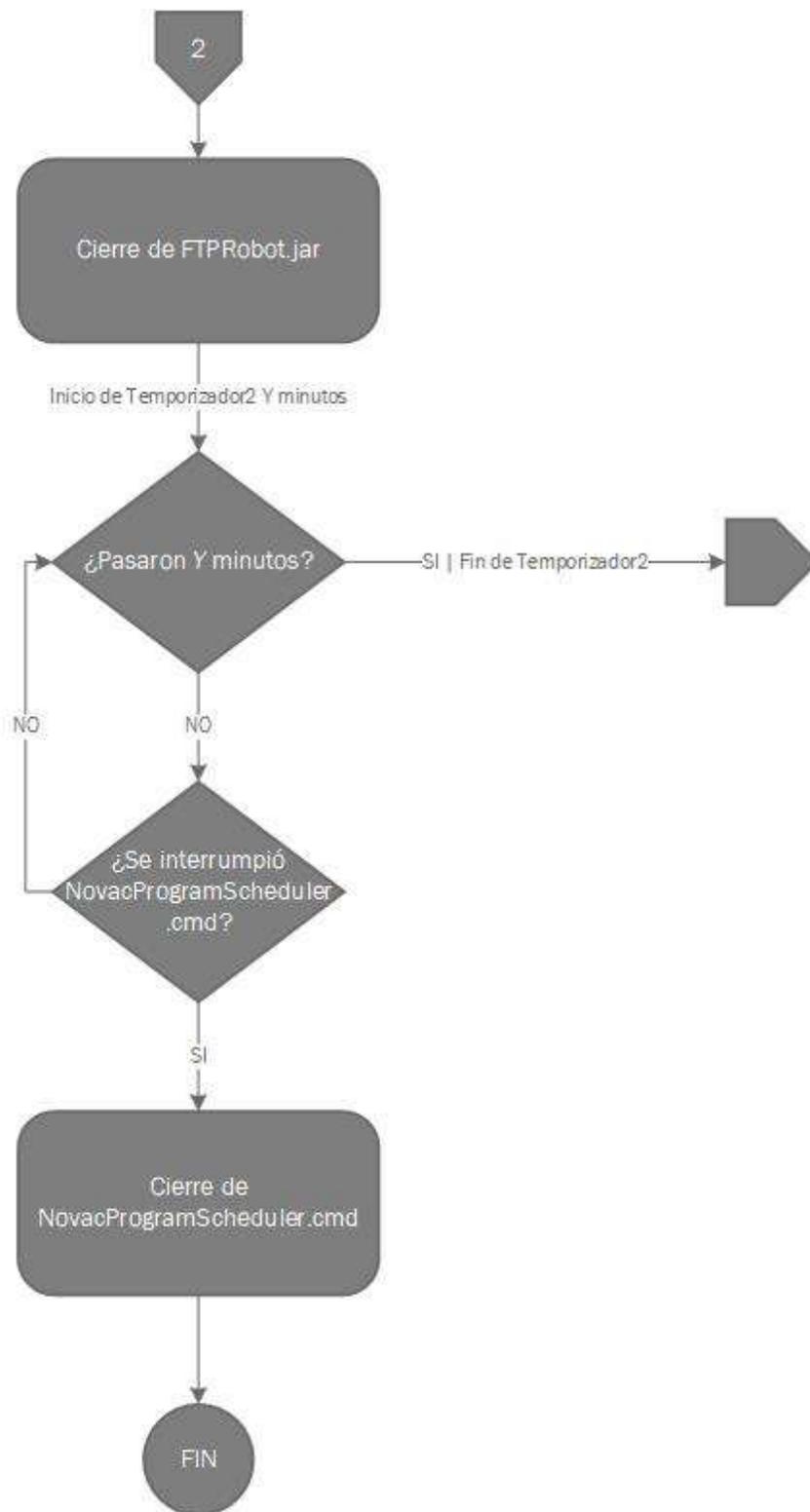
Finalmente, veremos los flujos para mostrar la información al usuario por medio de MAVAMSI, la aplicación web. Aquí se describirá la conexión a la base para el consumo de datos, y la construcción de gráficos y tablas que ayuden a representar la información.

### 3.3.3.1. Diagrama de flujo de MAVAMSI\_BOT2

Figura 3.13 - Diagrama de flujo de proceso de transmisión de archivos de mediciones de emisión de gases vía FTP







### 3.3.3.2. Diagramas de flujo de MAVAMSI\_BOT2

Figura 3.14 - Diagrama de flujo de proceso de obtención de archivos de mediciones de RSAM vía Earthworm

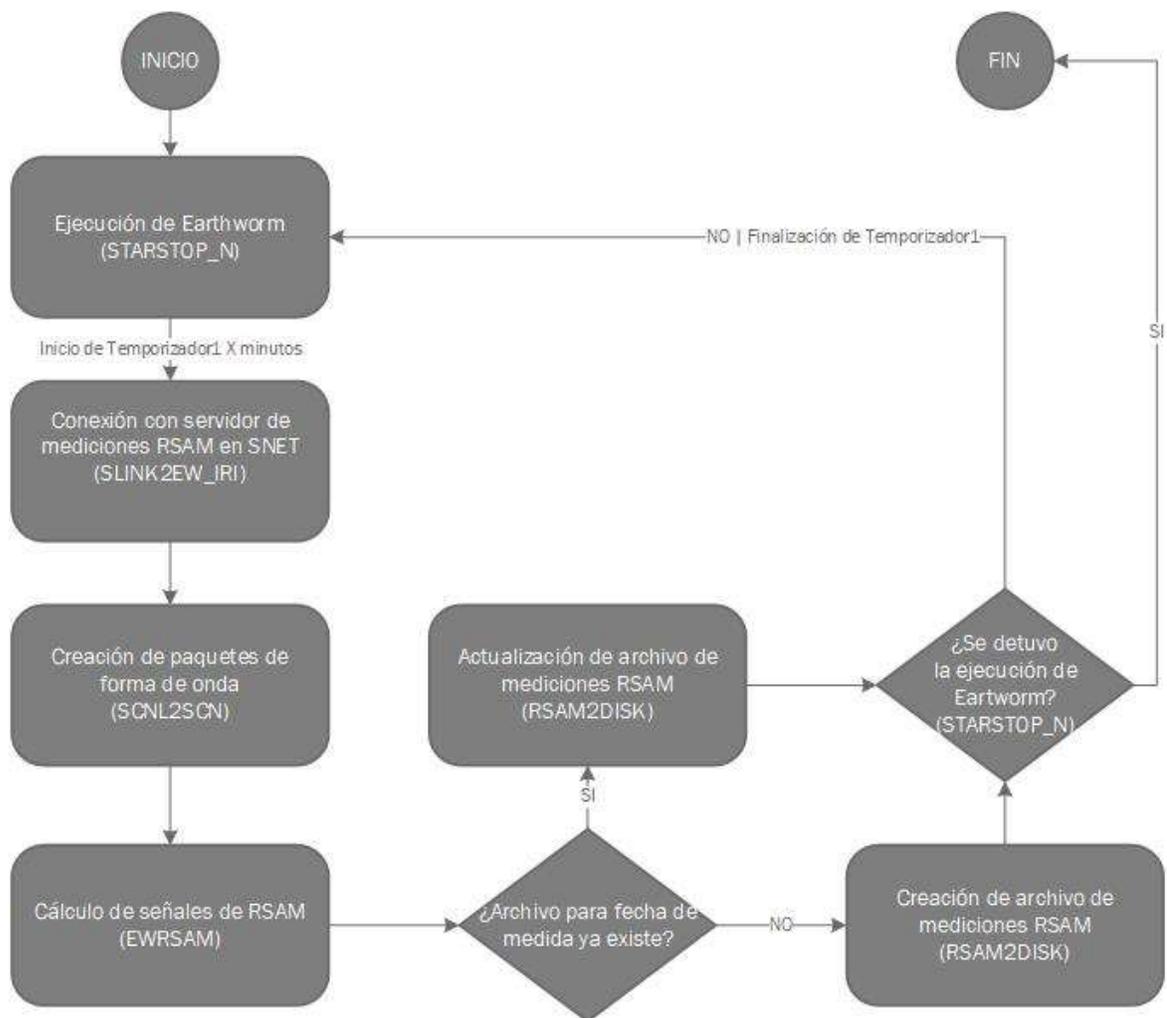
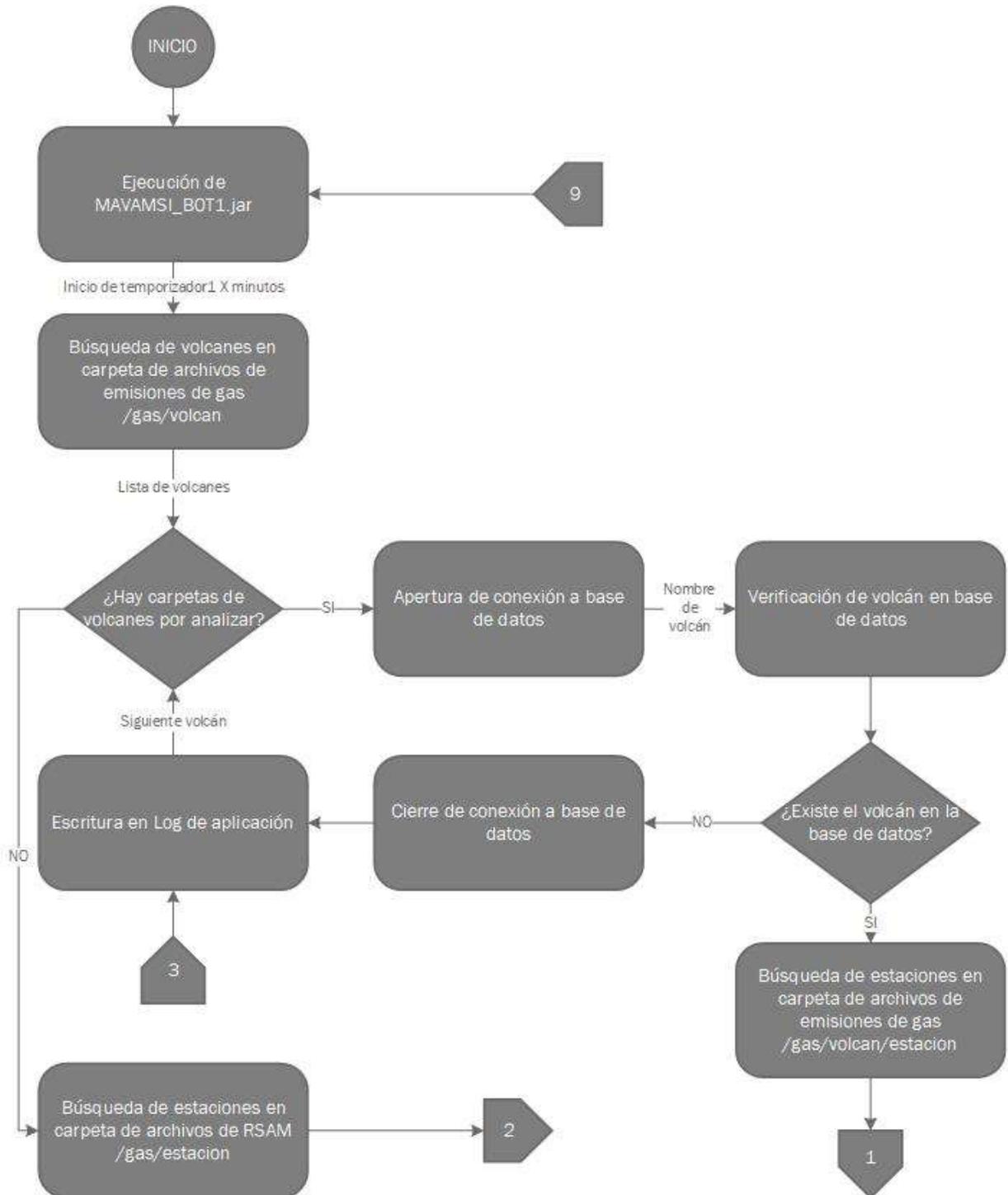
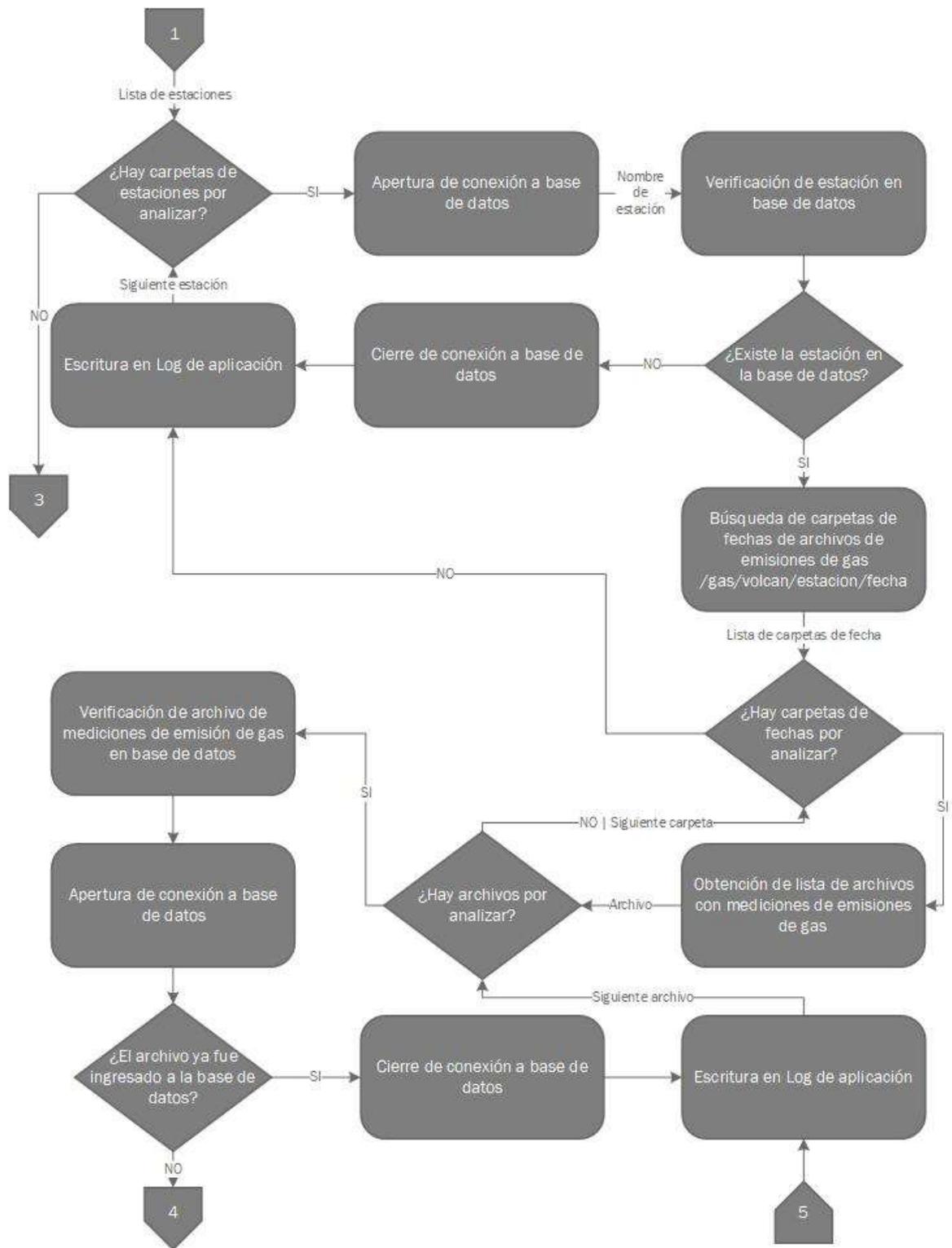
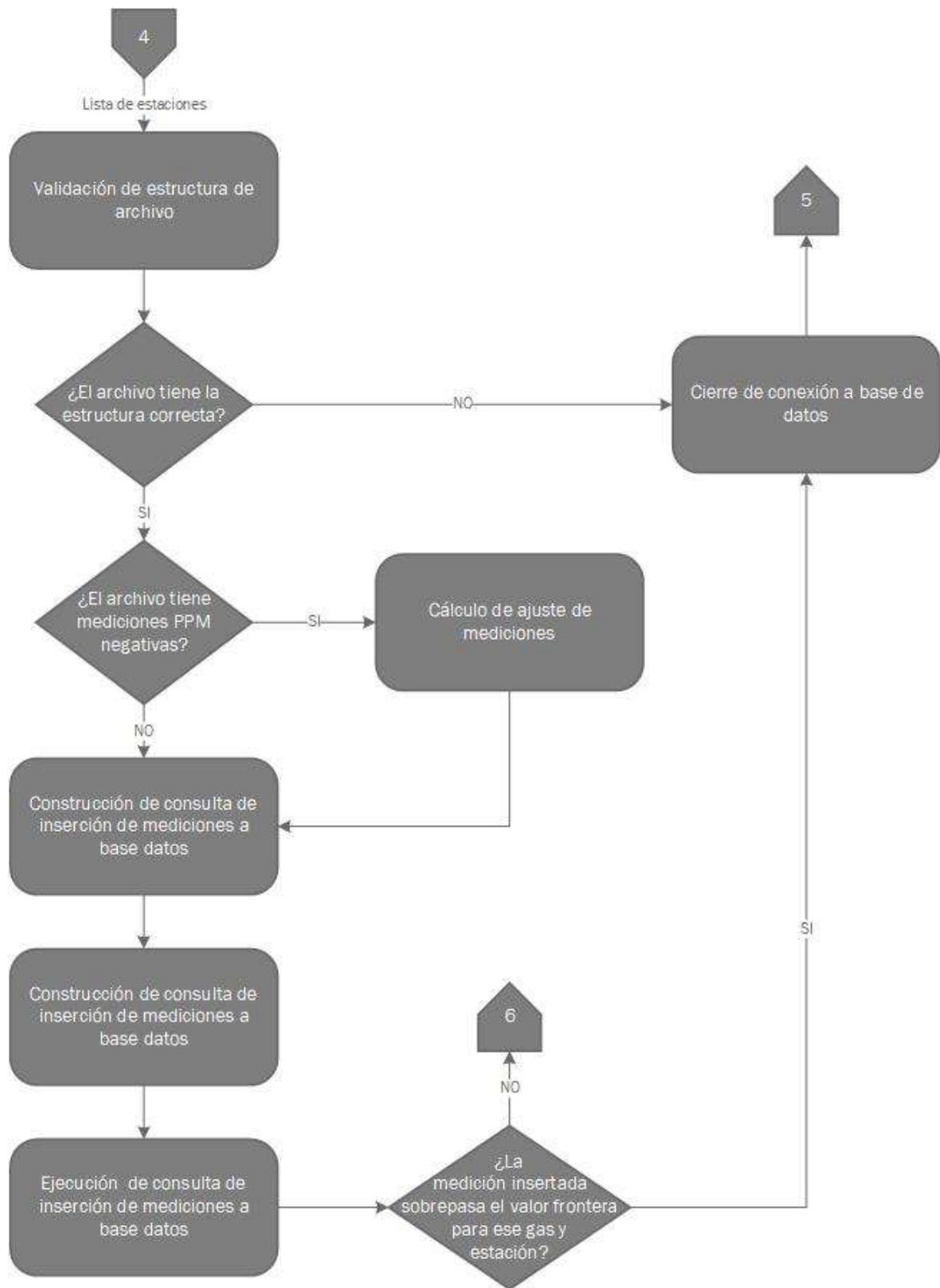
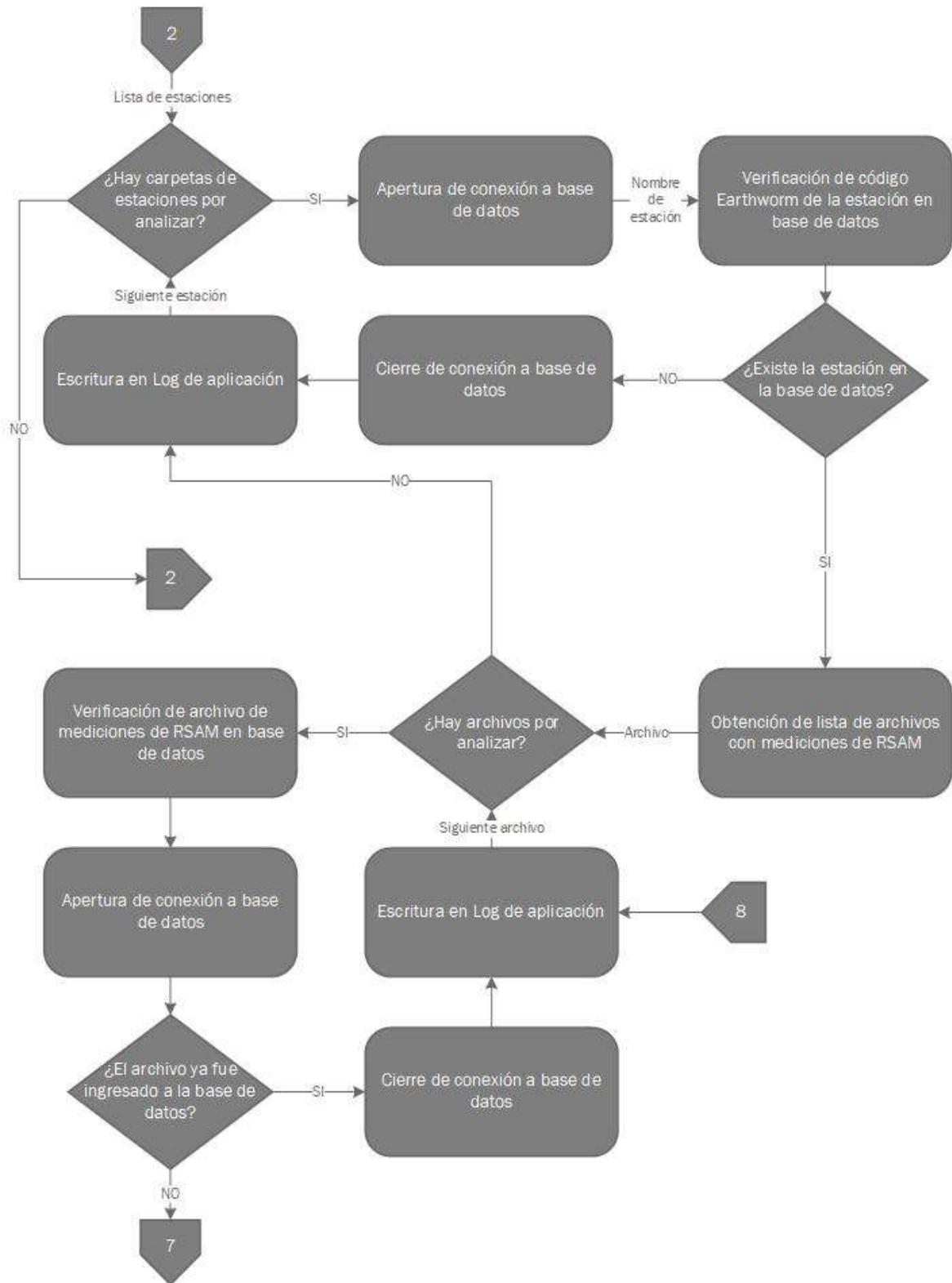


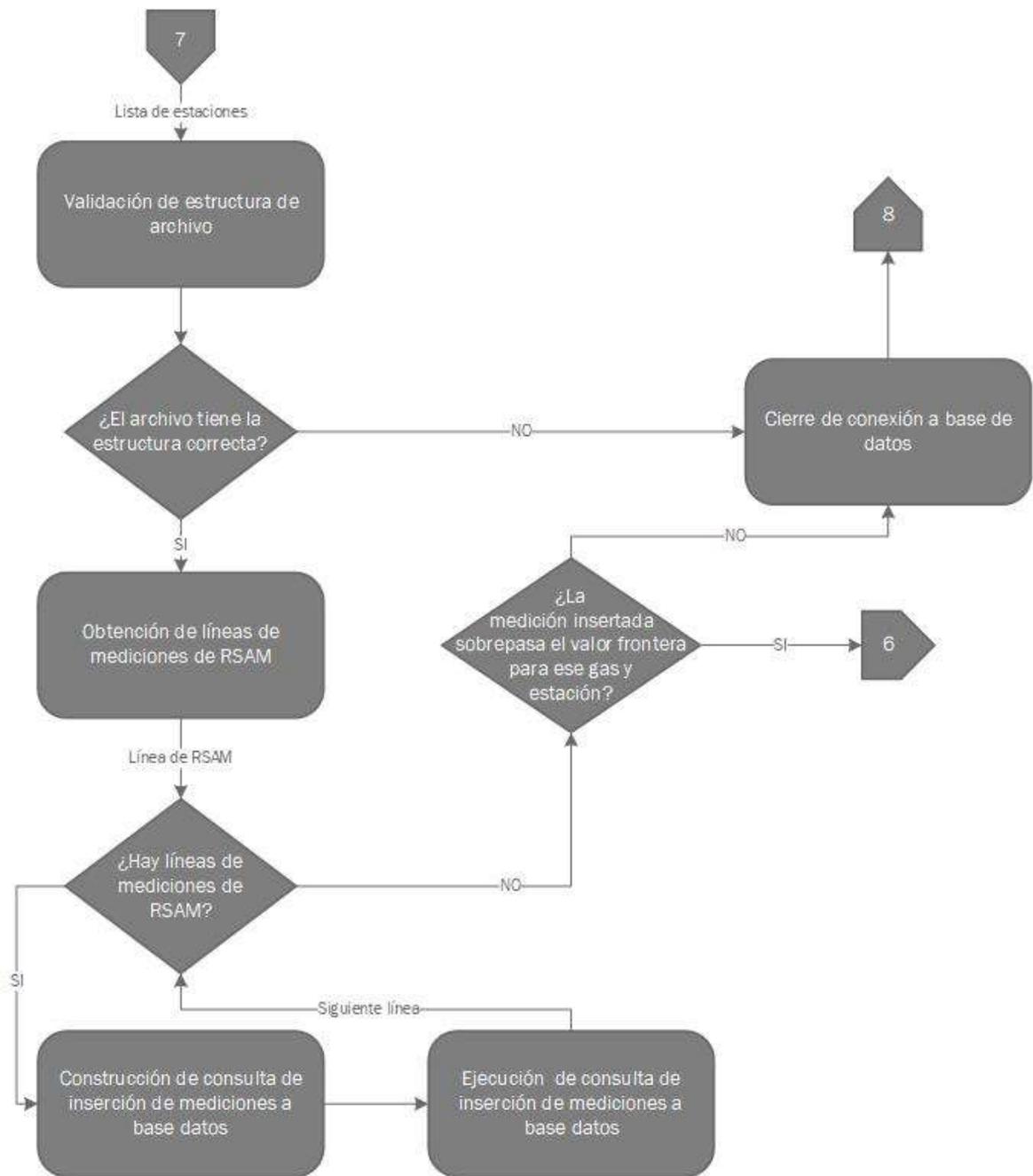
Figura 3.15 - Diagrama de flujo de proceso de inserción de mediciones en base de datos y envío de alertas por correo electrónico

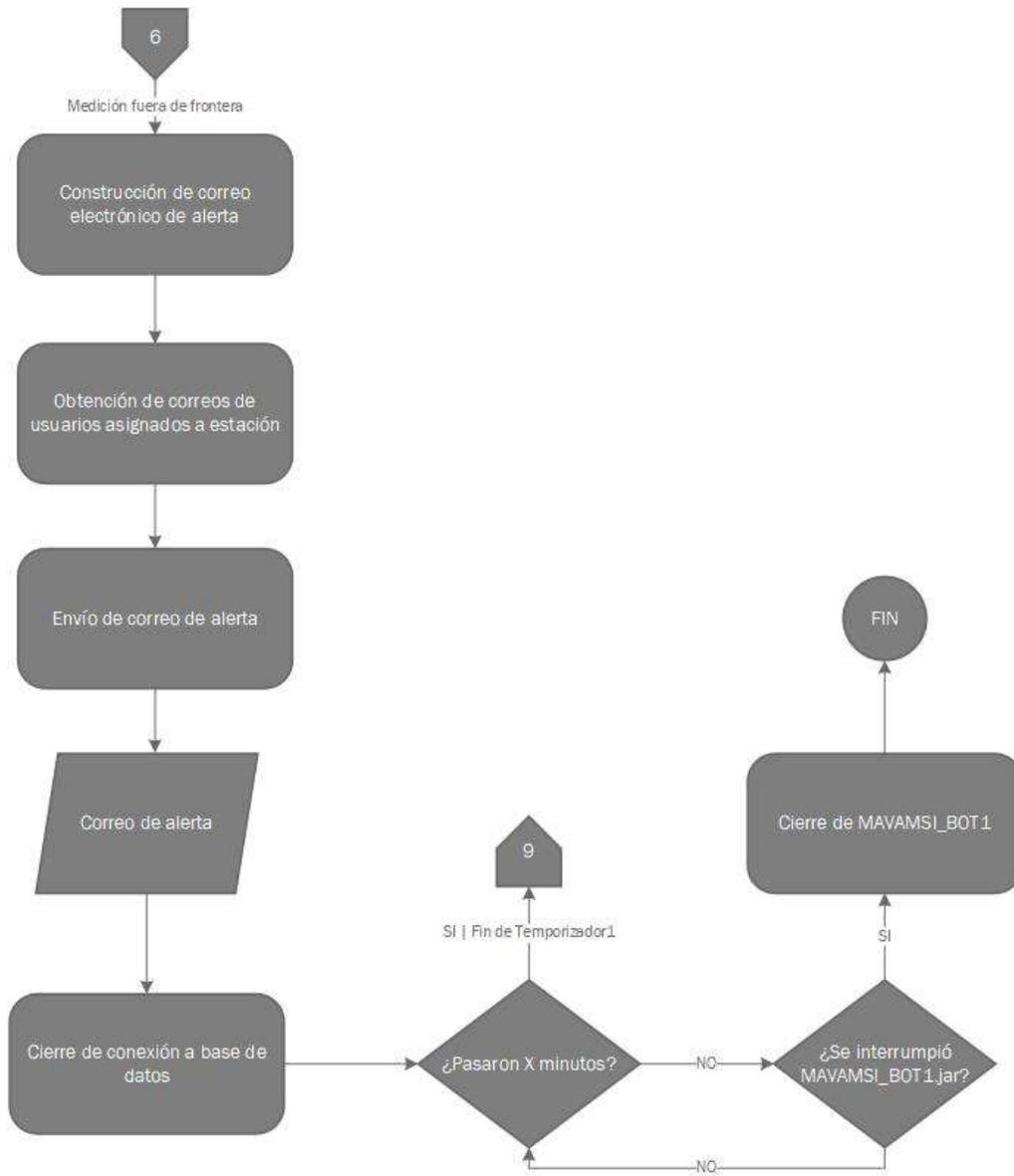












### 3.3.3.3. Diagramas de flujo de MAVAMSI

Figura 3.16 - Diagrama de flujo de proceso de inicio de sesión en MAVAMSI

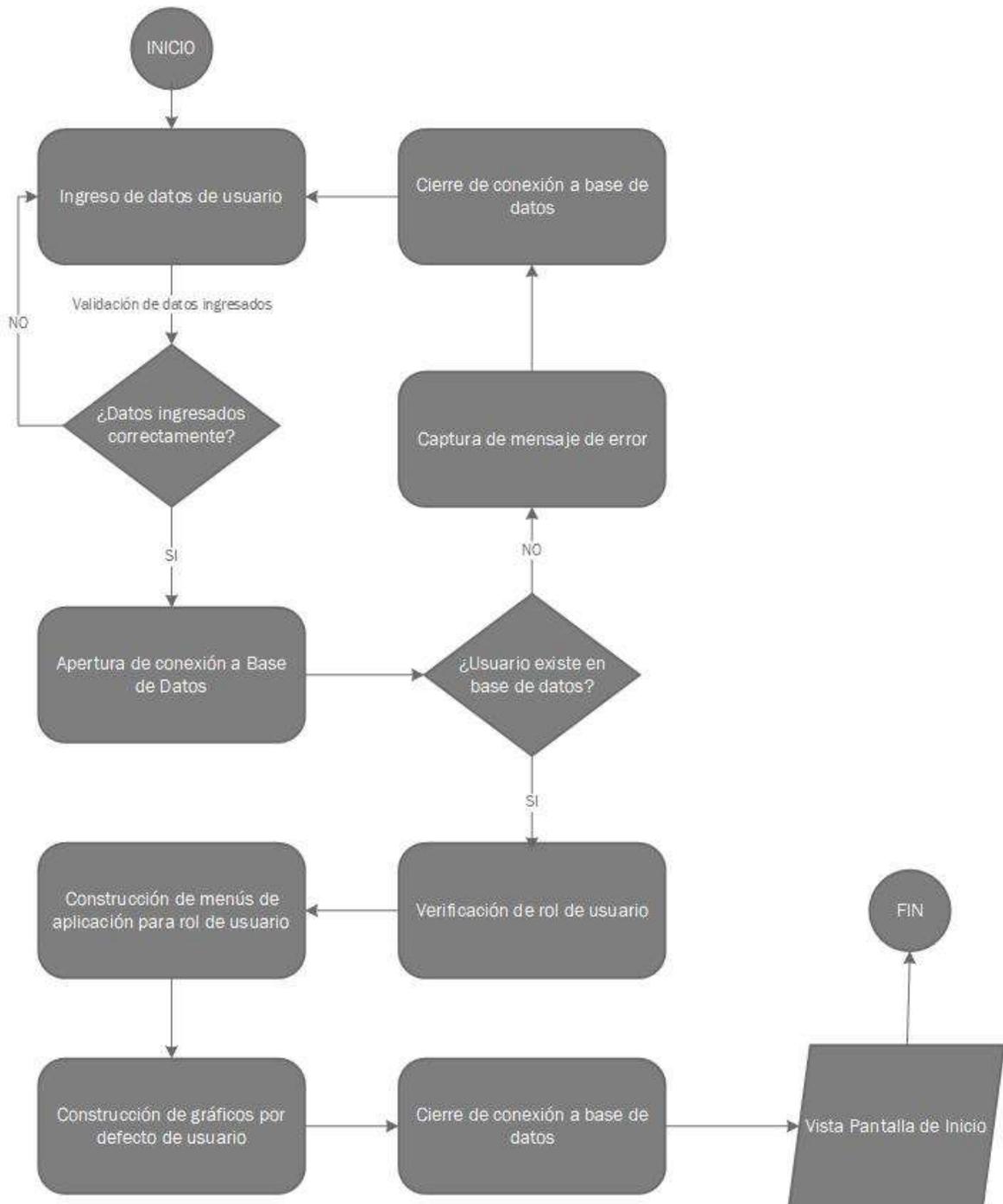
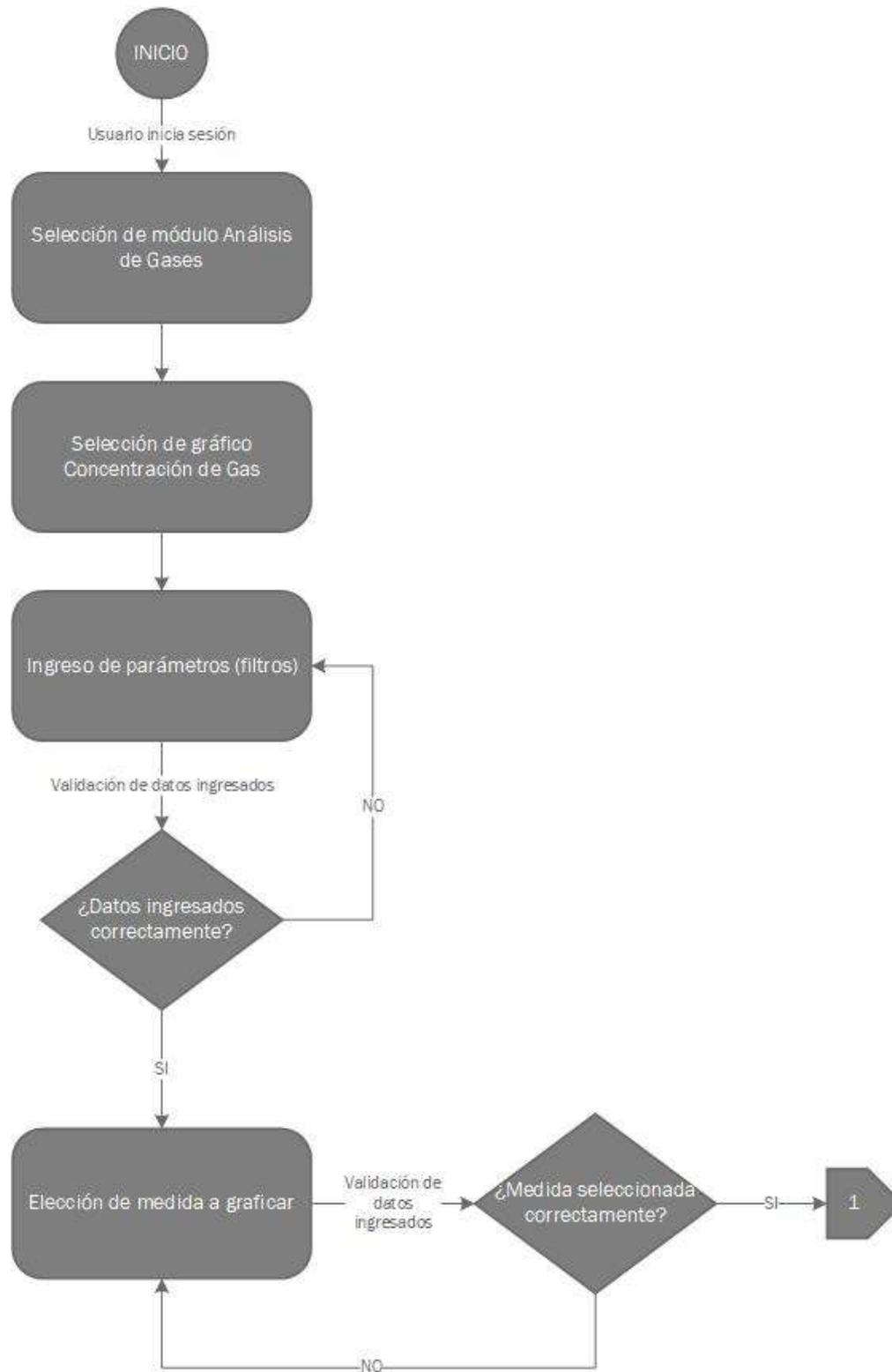
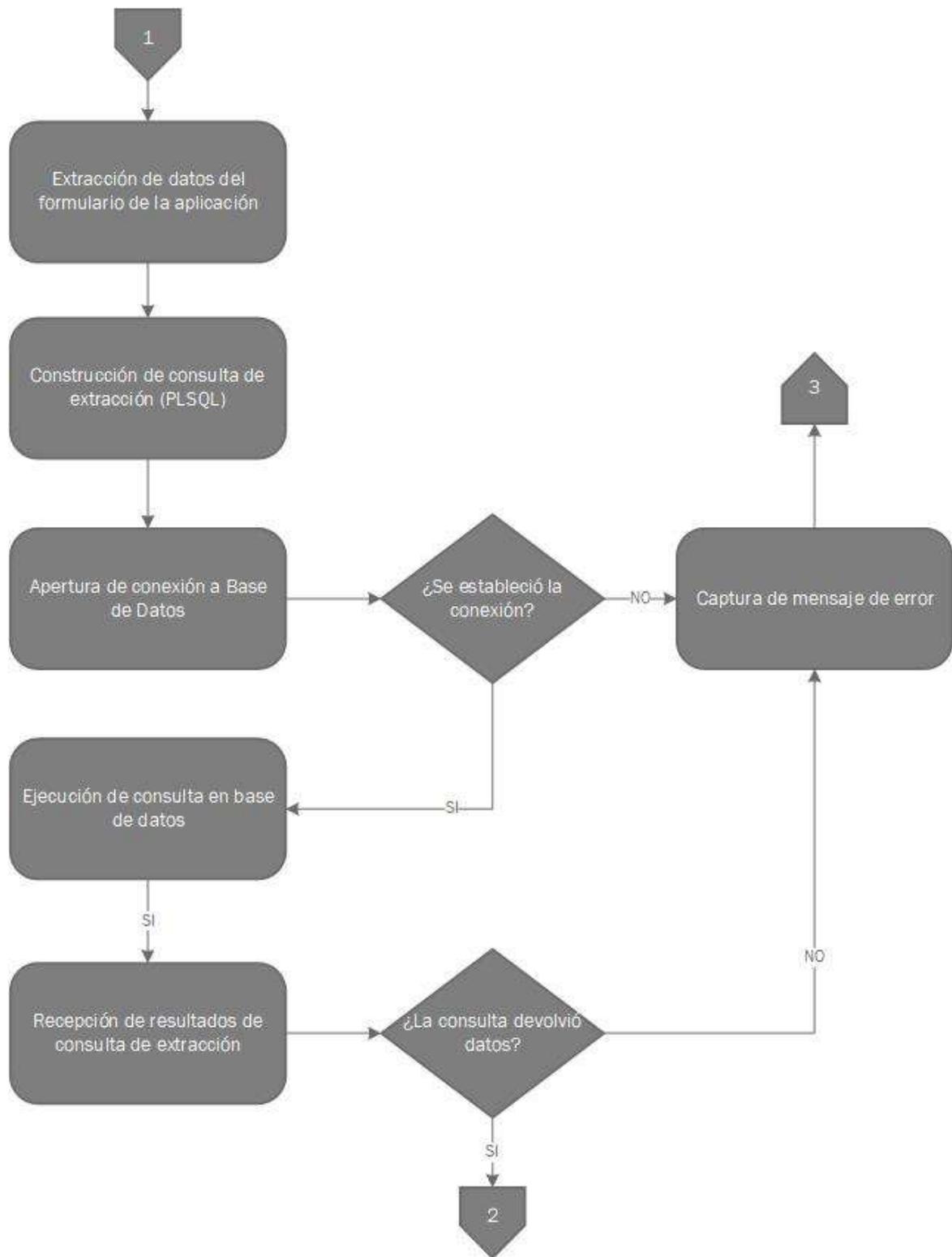


Figura 3.17 – Diagrama de flujo de proceso de despliegue de gráfico de concentración de gas





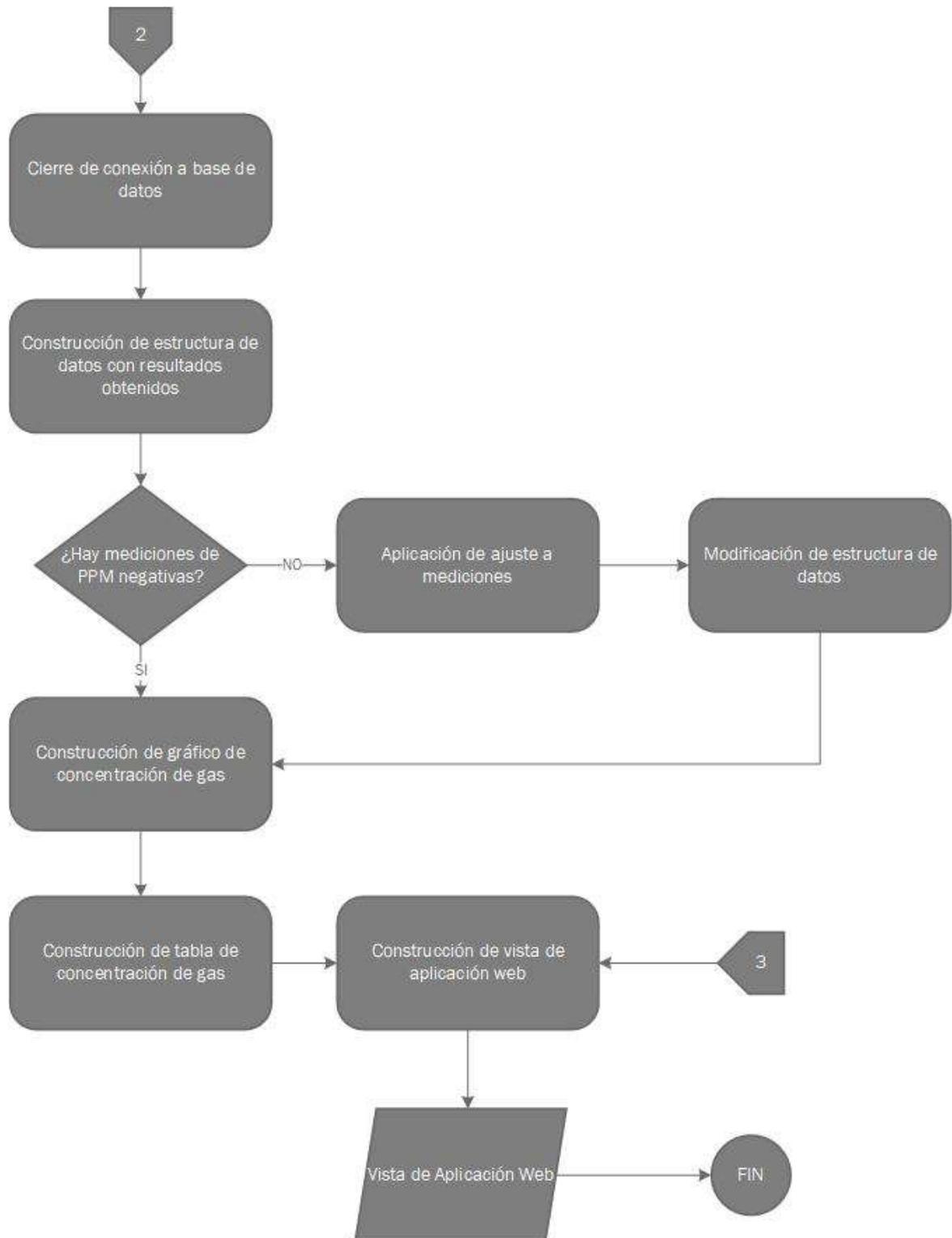
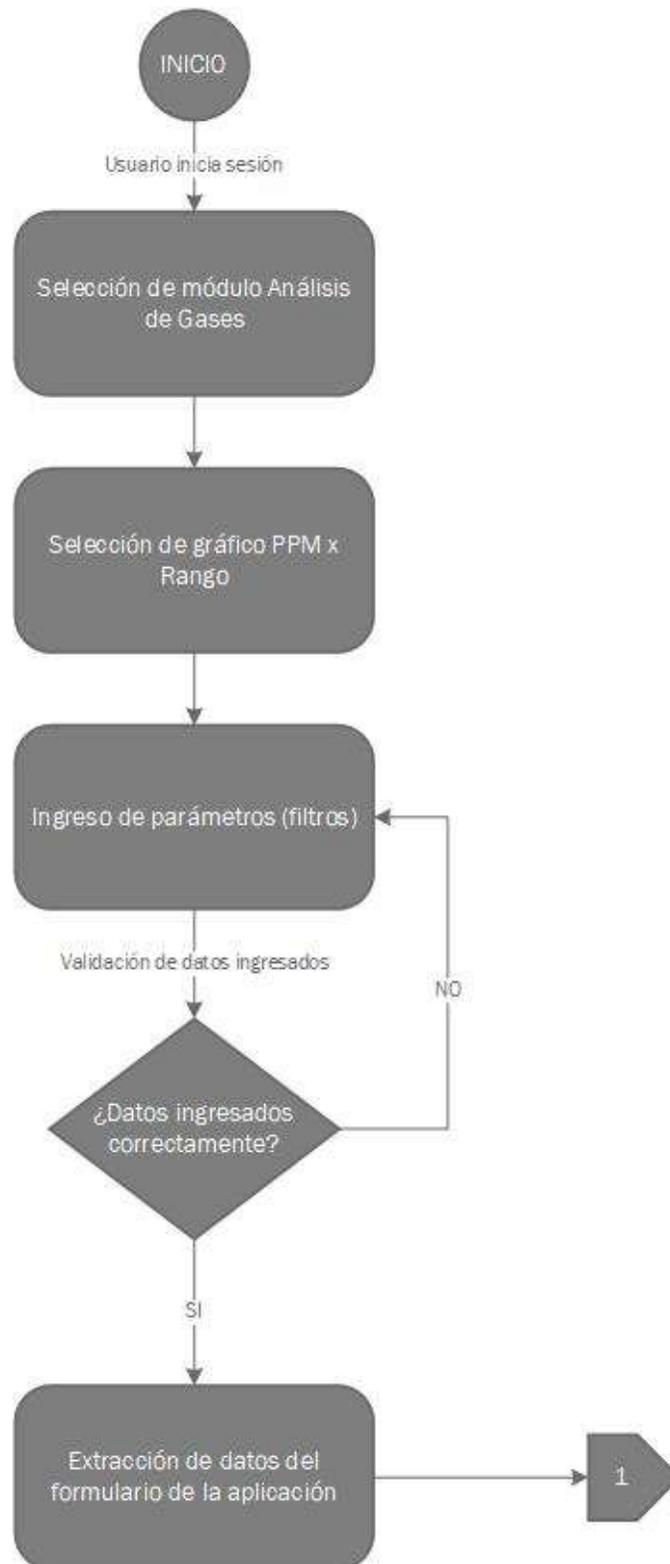
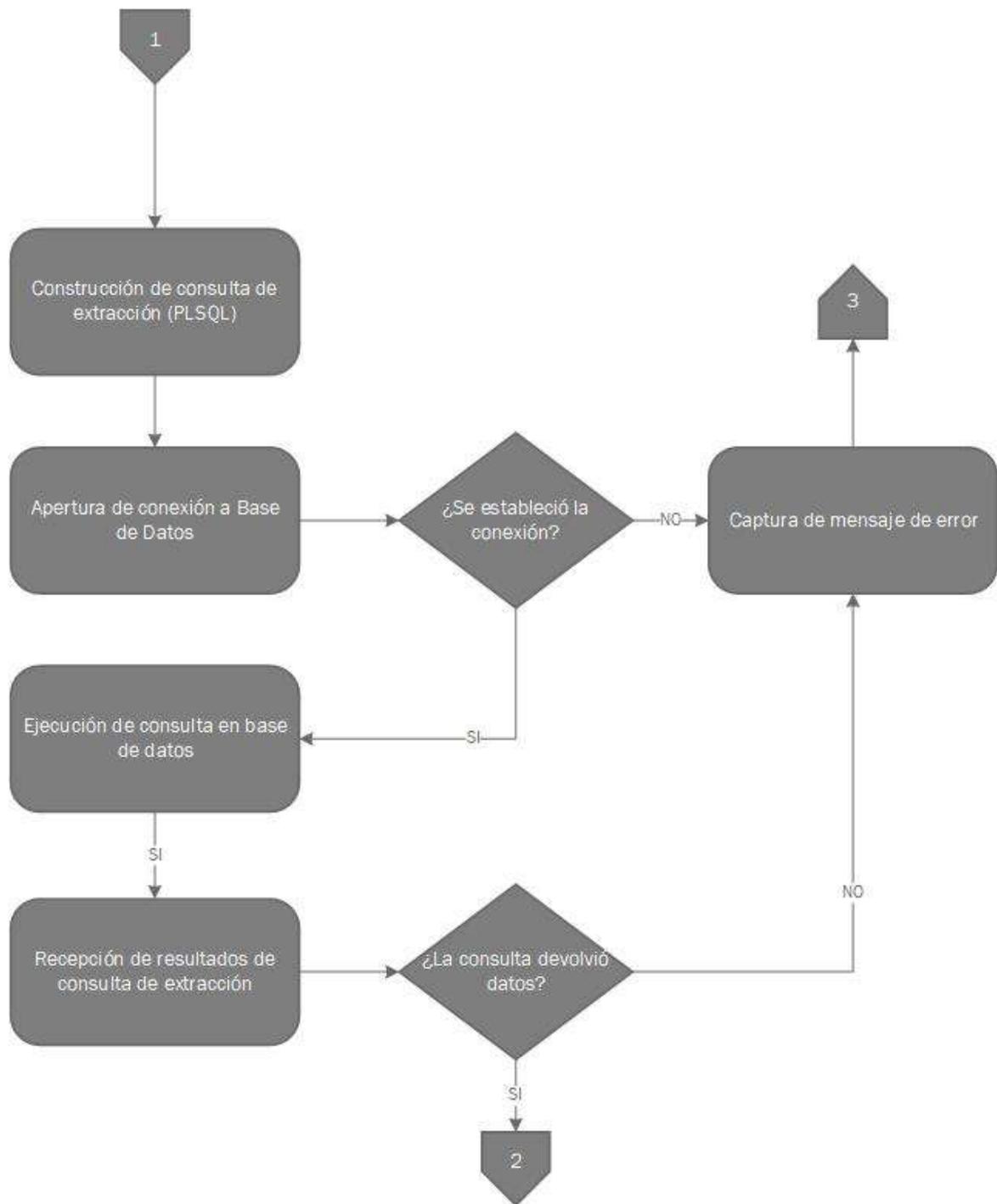


Figura 3.18 - Diagrama de flujo de proceso despliegue de gráfico de contracción de gas por rango





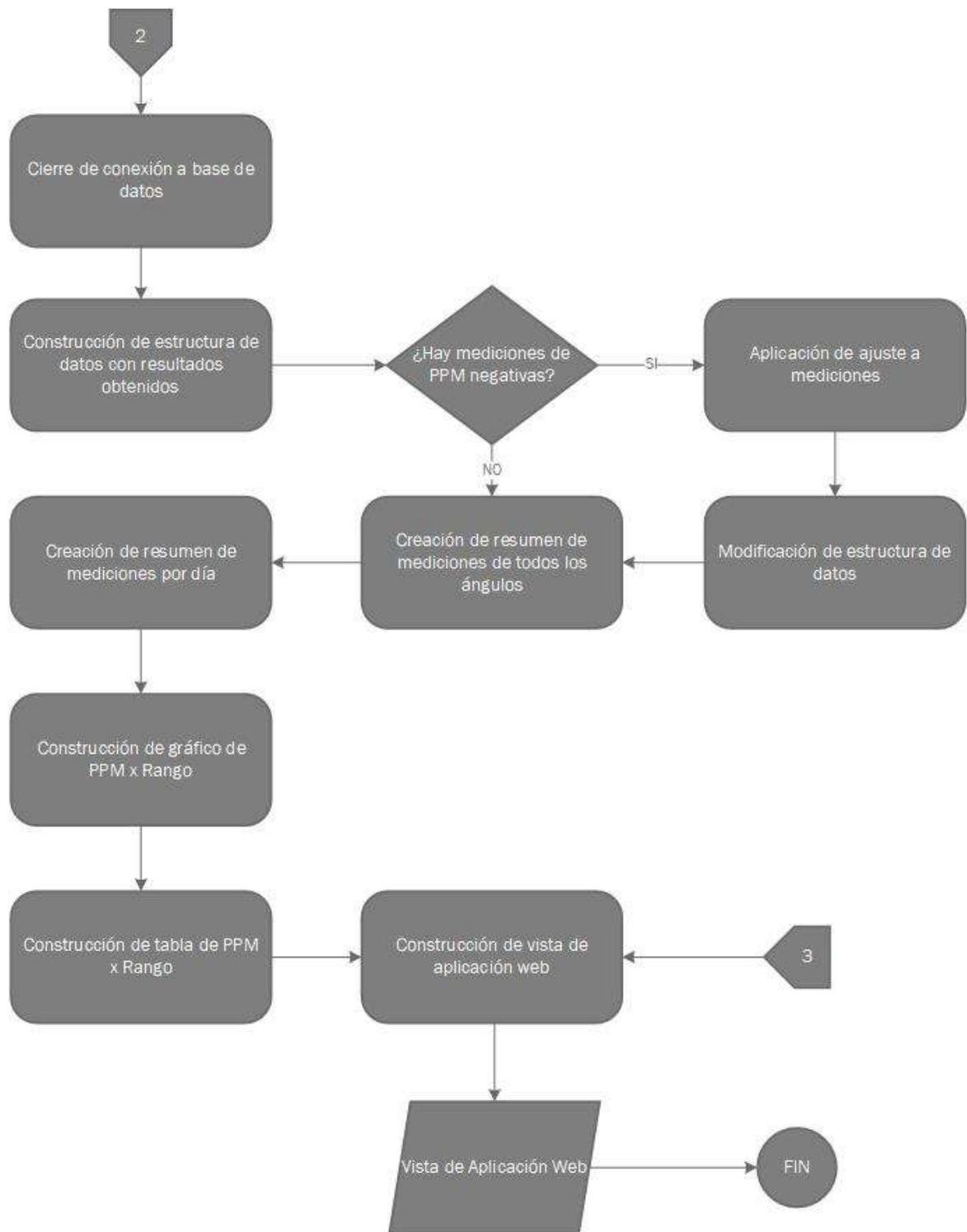
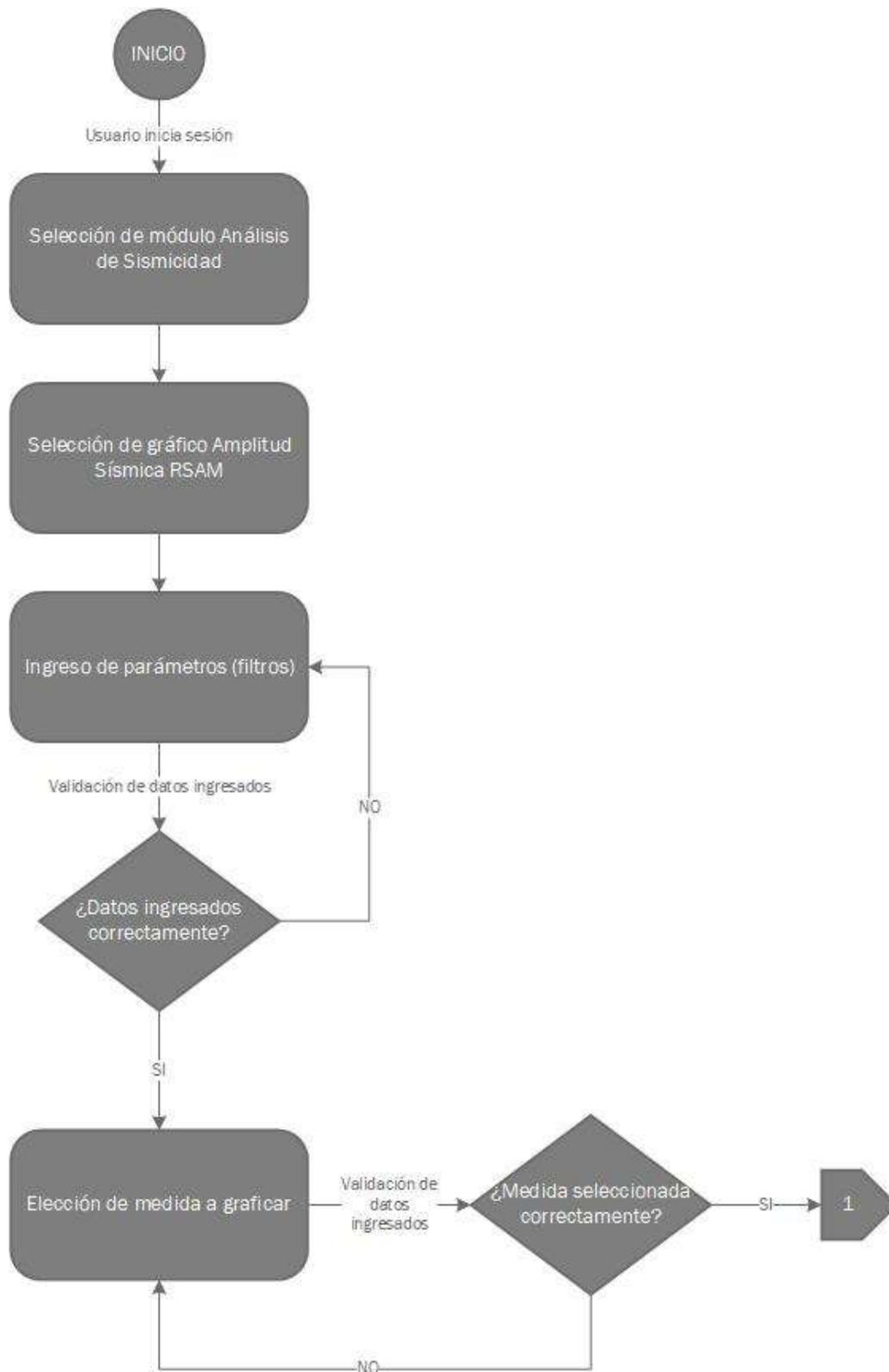


Figura 3.19 - Diagrama de flujo de proceso de despliegue de gráfico de RSAM



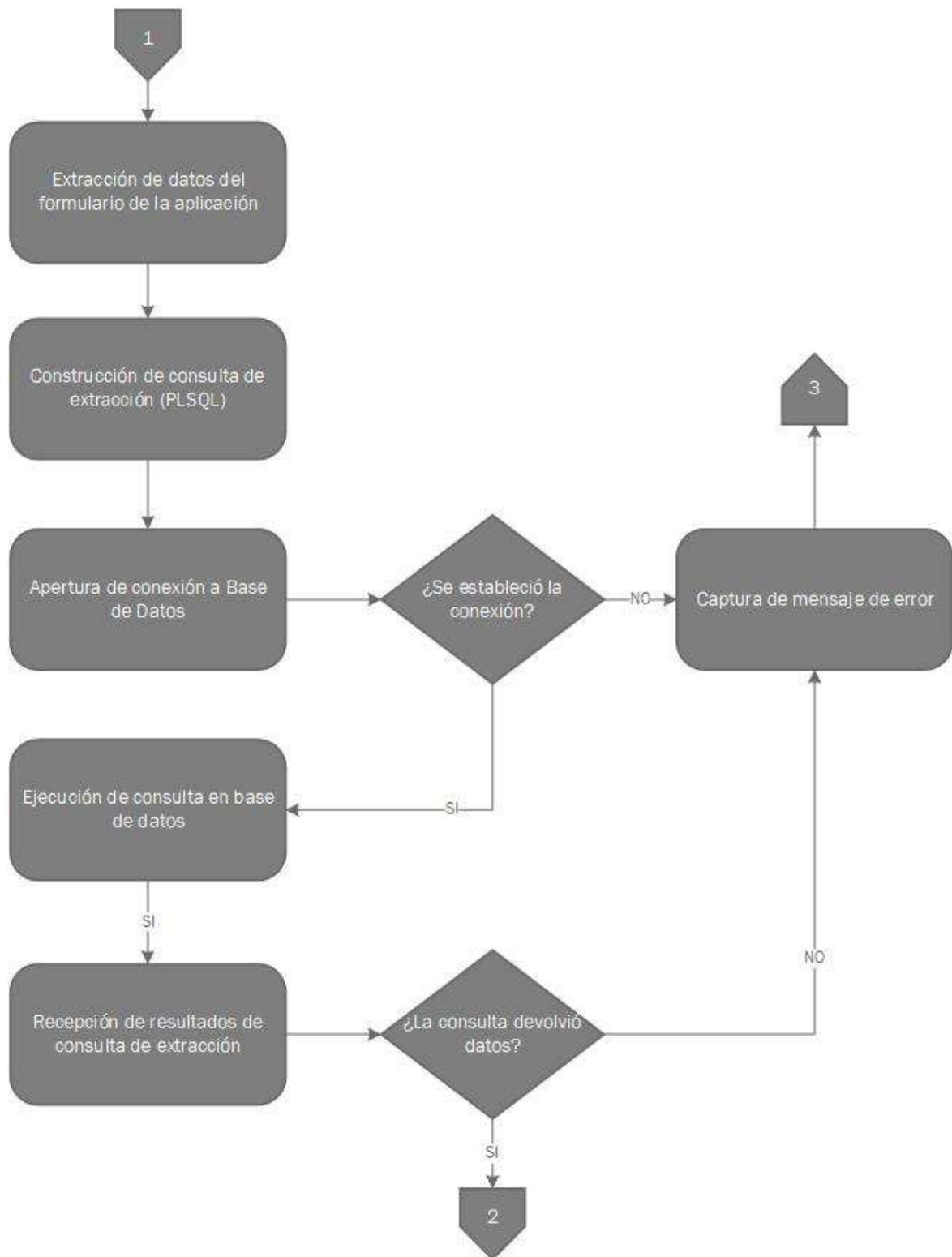
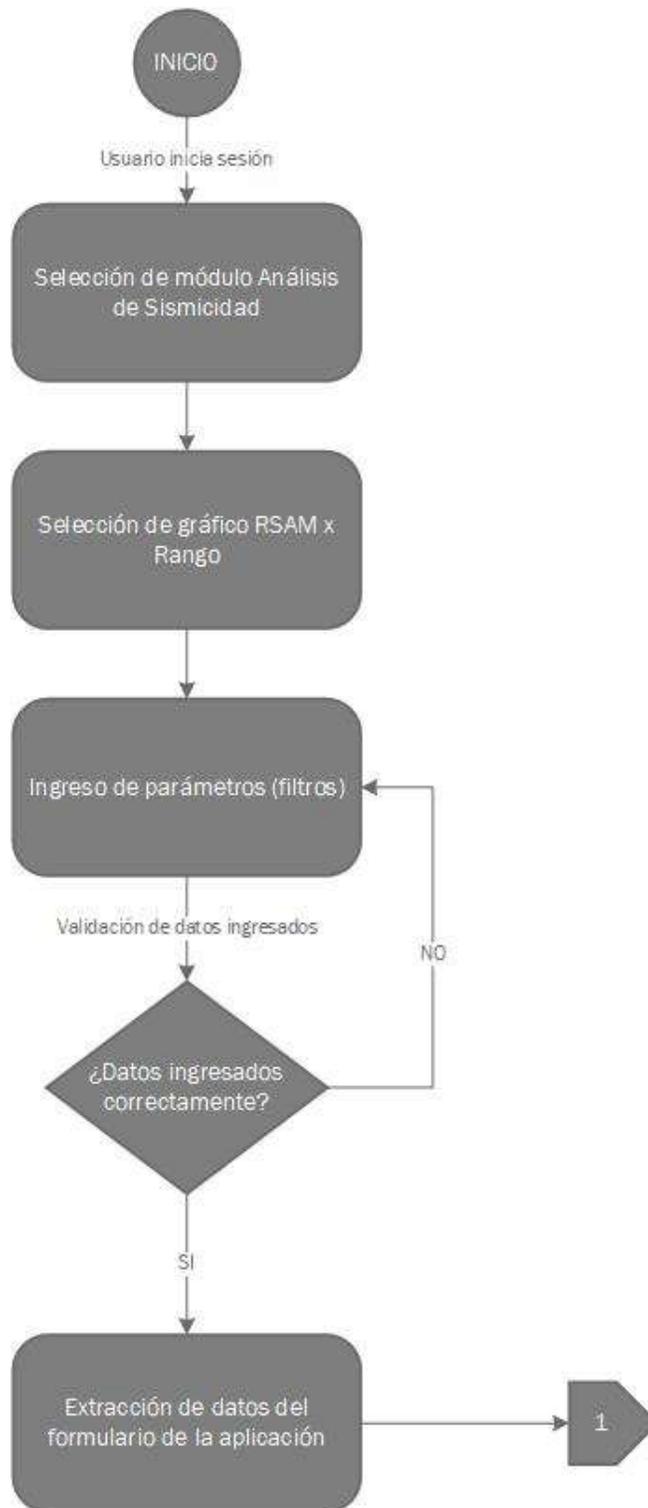
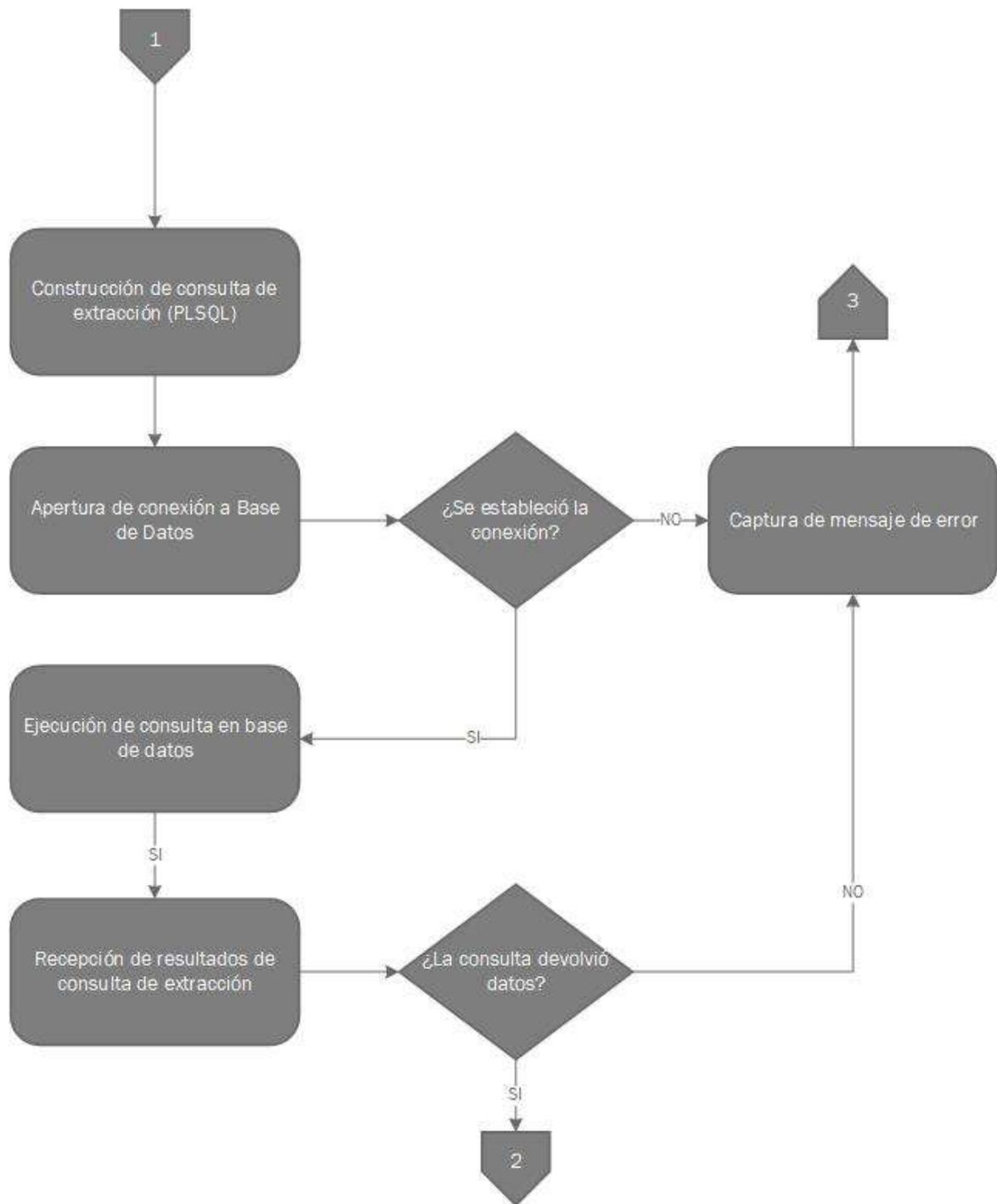




Figura 3.20 - Diagrama de flujo de proceso despliegue de gráfico de RSAM por rango





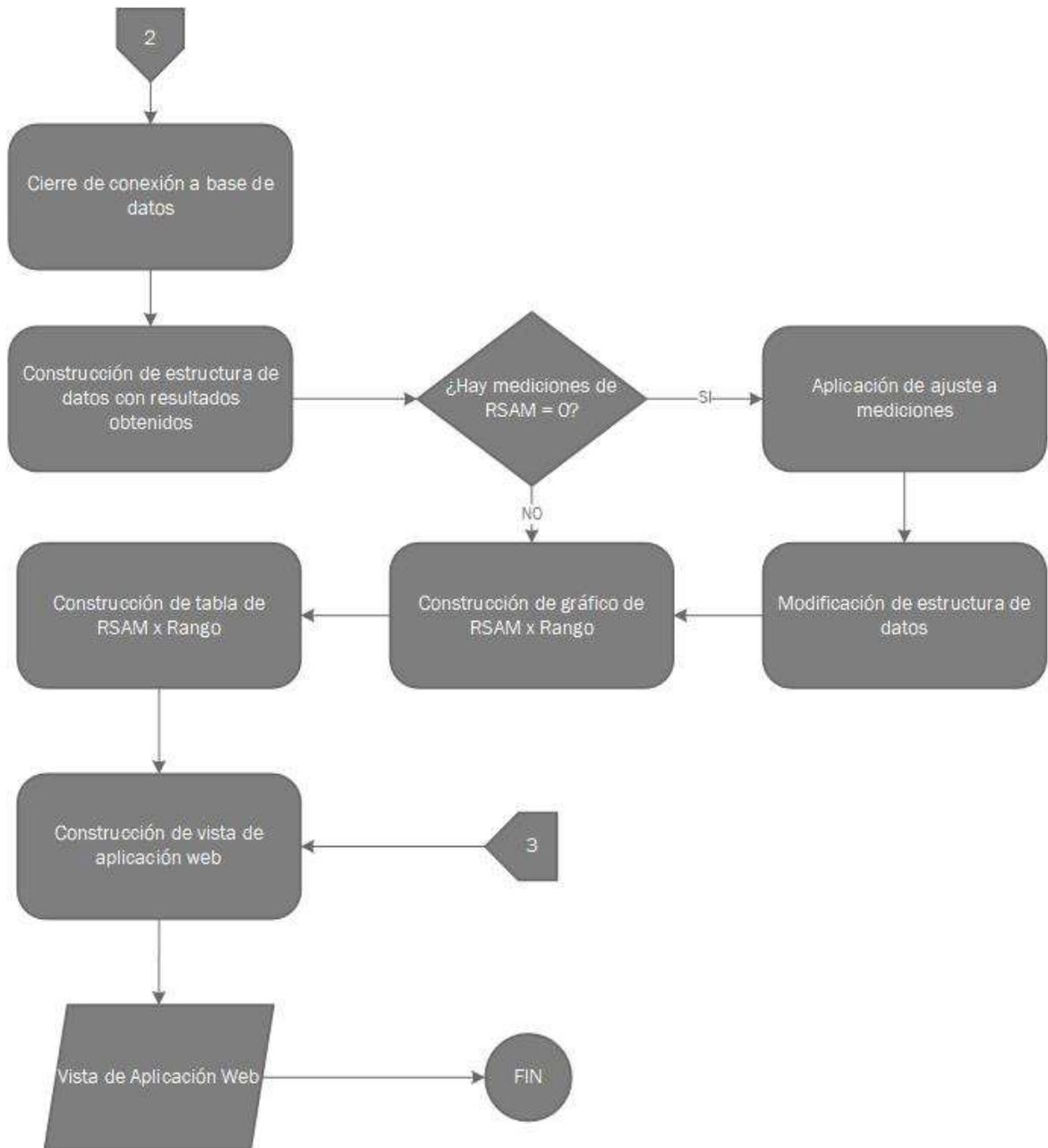
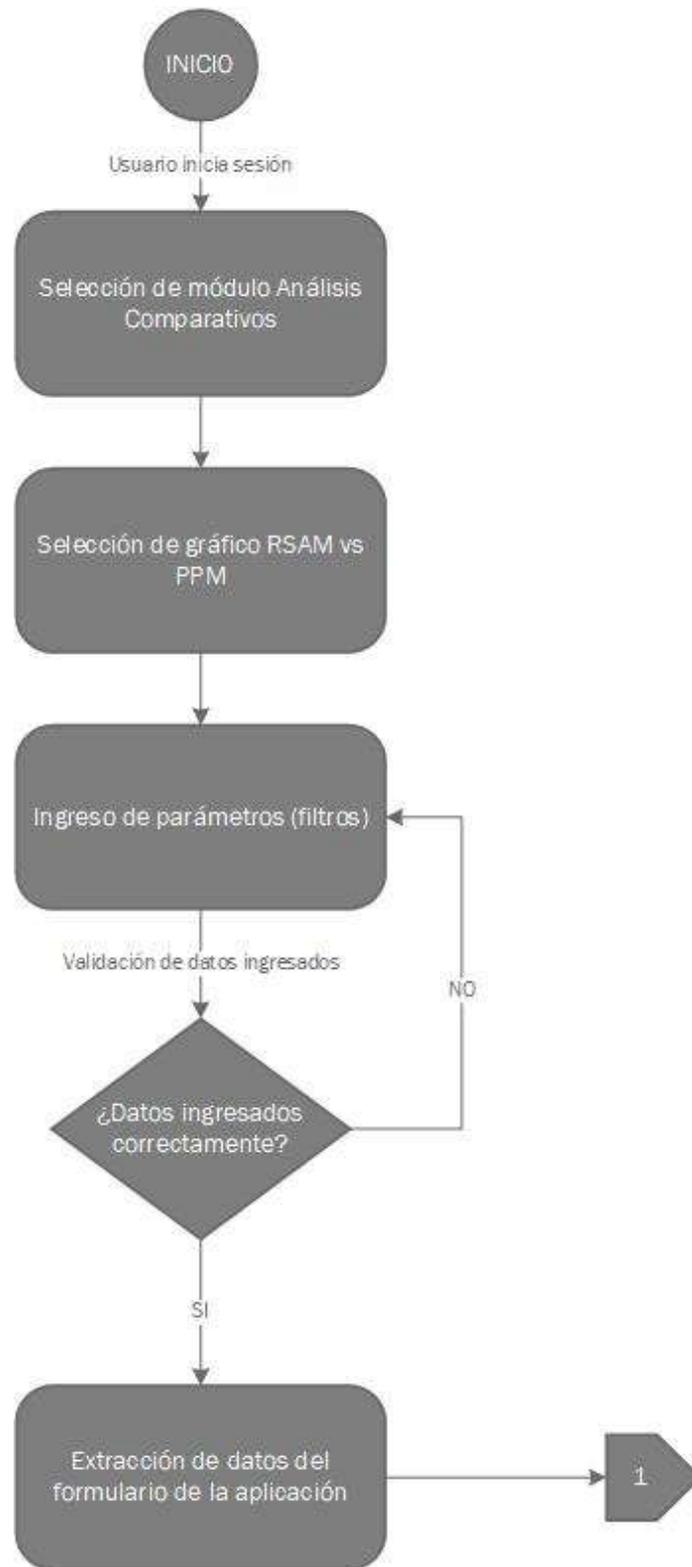
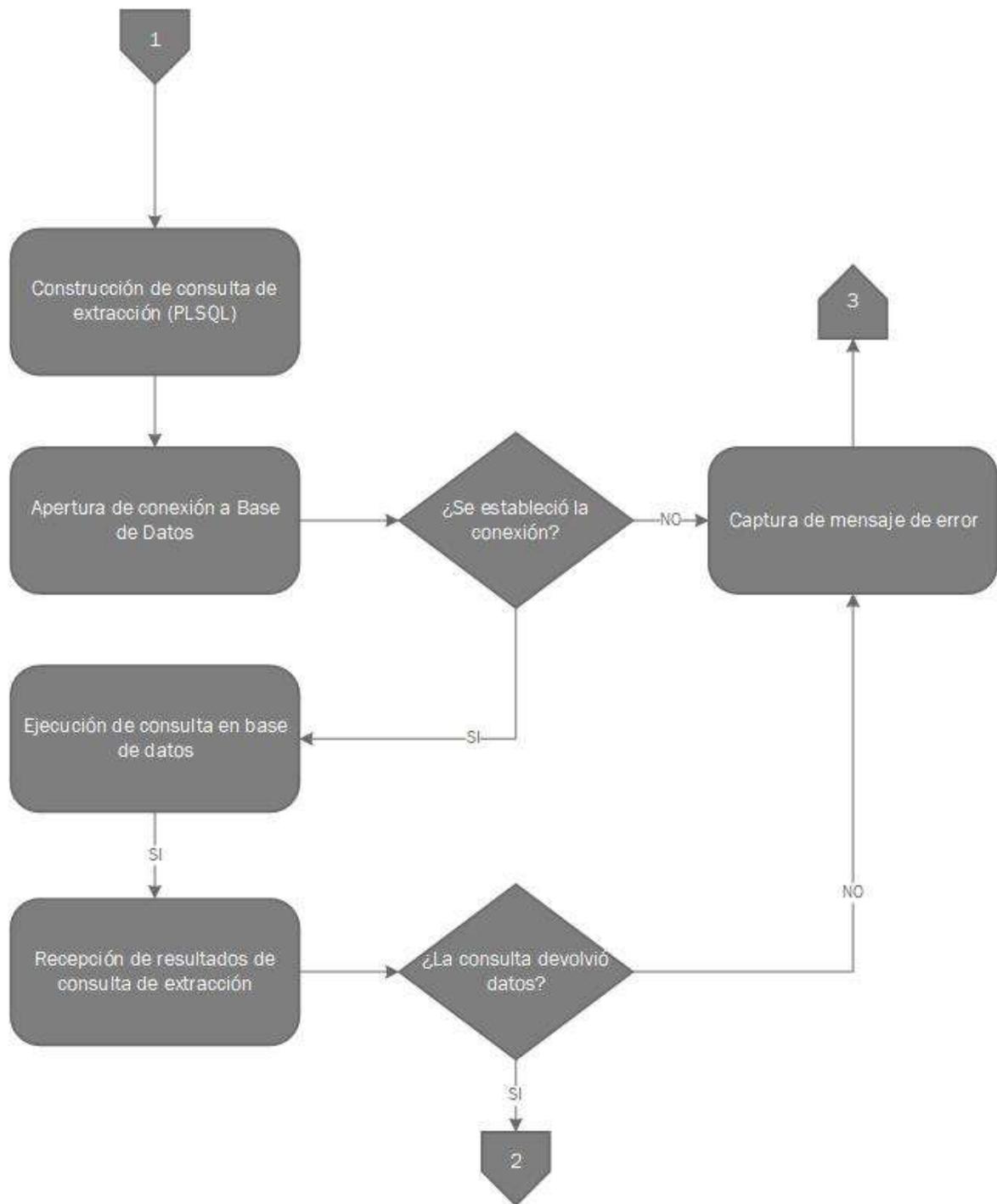
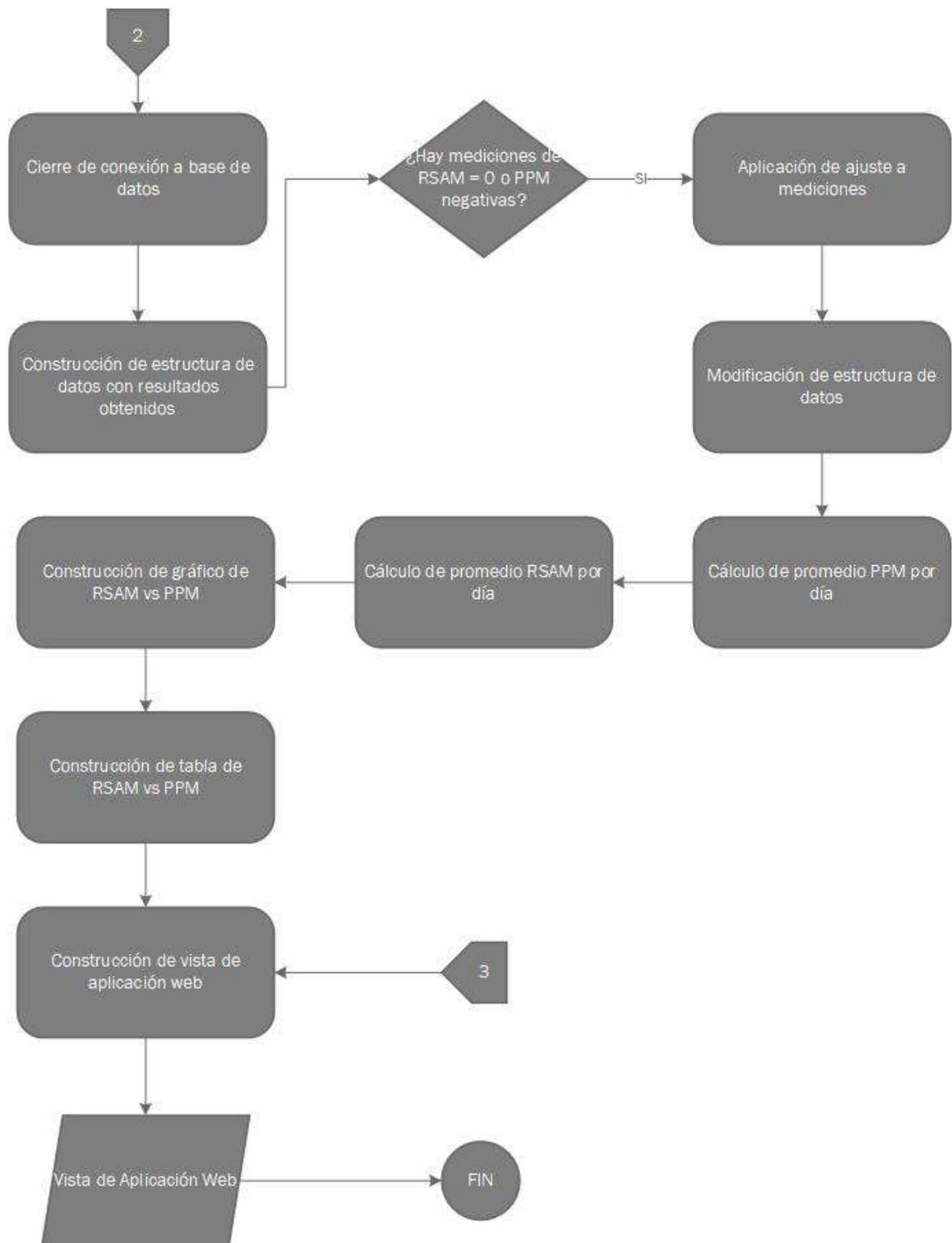


Figura 3.21 - Diagrama de flujo de proceso despliegue de gráfico de RSAM vs Emisiones de gas







### ***3.3.4. Diagramas de Clases***

El desarrollo de software, es donde se desencadena todo el diseño de nuestro proyecto. A continuación, presentaremos todas las clases que formarán parte de nuestro proyecto, en los cuales, esperamos dejar por sentado, cómo trabajará nuestro sistema; desde la fase de extracción de datos de las diferentes fuentes, hasta la representación de datos por medio de gráficos en la aplicación web.

Los diagramas que veremos en este apartado, están en formato UML. “El lenguaje de modelado unificado (UML) es un lenguaje de modelado visual de uso general que se utiliza para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Captura las decisiones y la comprensión de los sistemas que deben ser construidos. Se utiliza para entender, diseñar, examinar, configurar, mantener el control de información sobre tales sistemas. Está diseñado para ser usado con todo tipo de desarrollo, fases del ciclo de vida, dominios de aplicación y medios. El modelado lingüístico pretende unificar las experiencias pasadas sobre técnicas de modelado e incorporar las mejores prácticas de software actuales en un enfoque estándar.

UML incluye conceptos semánticos, notación y directrices. Tiene características estáticas, dinámicas, ambientales, y las partes organizativas. Se pretende que sea apoyado por herramientas interactivas de modelado visual que tienen generadores de código y escritores de informes. La especificación de UML no define un proceso estándar, sino que pretende ser útil con un proceso de desarrollo iterativo. Se pretende que soporte la mayoría de los objetos orientados a objetos existentes en el proceso de desarrollo.” (Rumbaugh, 2005)

Cabe mencionar, que para el desarrollo de MAVAMSI\_BOT2, no se ha desarrollado un diagrama de clases, debido a que, como se comentó en el apartado anterior, éste robot será desarrollado bajo el paradigma de programación estructurada.

### 3.3.4.1. Clases de MAVAMSI\_BOT1

A partir de la figura 3.22 hasta la figura 3.29, veremos el funcionamiento de MAVAMSI\_BOT1 descrito por las clases que describirán su funcionamiento. Las clases estarán implementadas en lenguaje Java bajo el paradigma de programación orientada a objetos. Ya que no existe ninguna relación de herencia entre cada clase, no hay un diagrama de clases para este modelo. Cada clase es independiente una de la otra.

Figura 3.22 - Diagrama de clase Archivo

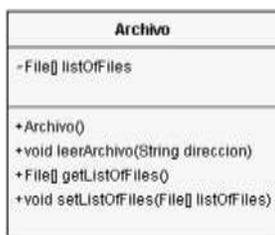


Figura 3.23 - Diagrama de clase Conexion

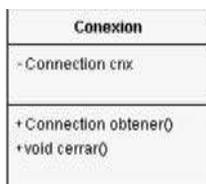


Figura 3.24 - Diagrama de clase Gas

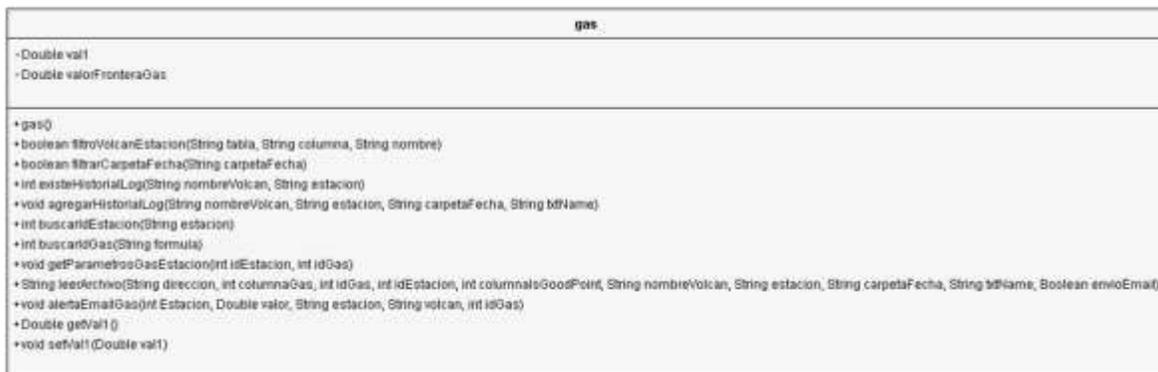


Figura 3.25 - Diagrama de clase MainView

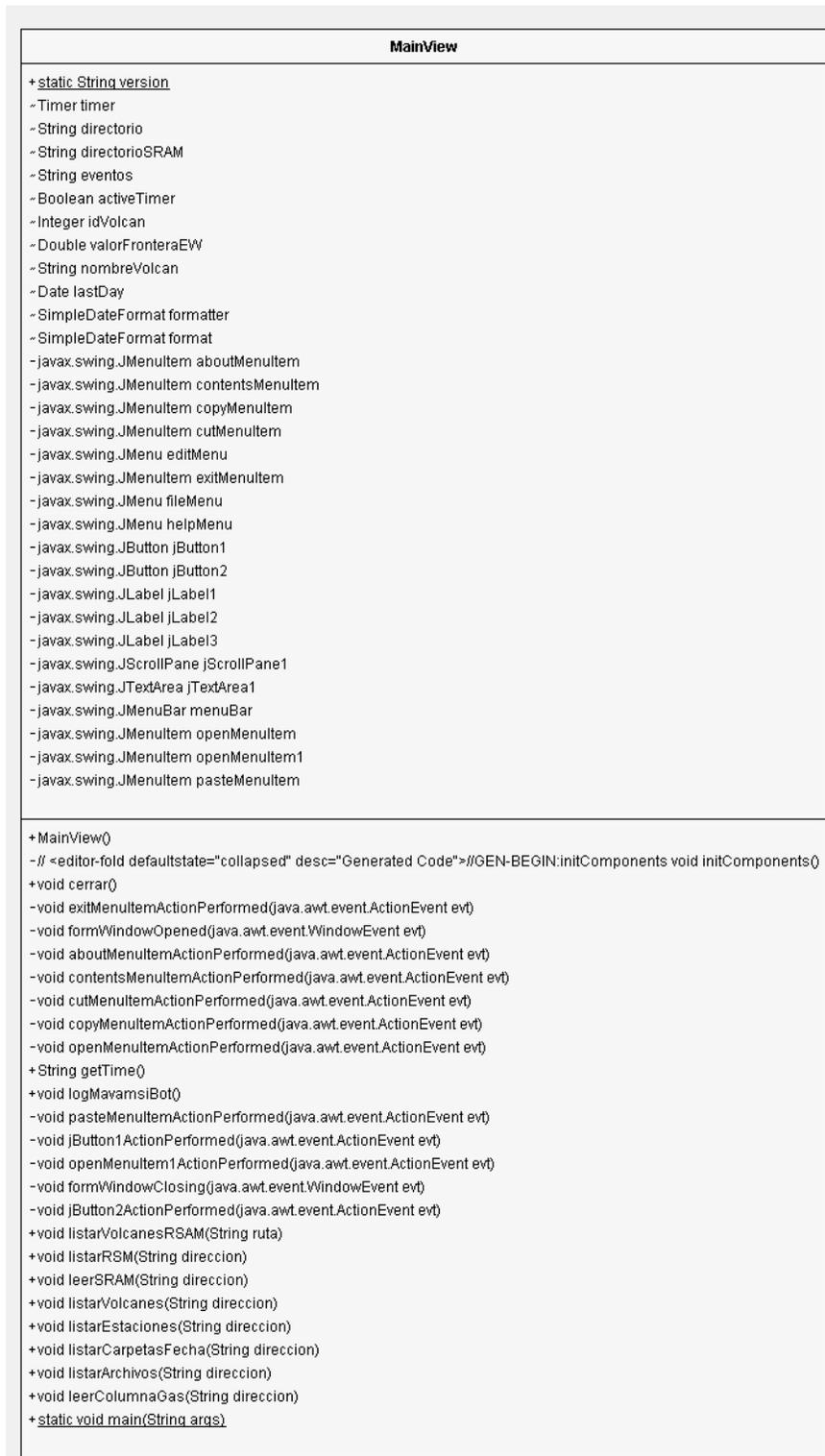


Figura 3.26 - Diagrama de clase MAVAMSIBOT

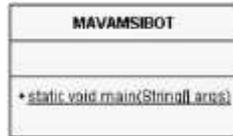


Figura 3.27 - Diagrama de clase Volcan

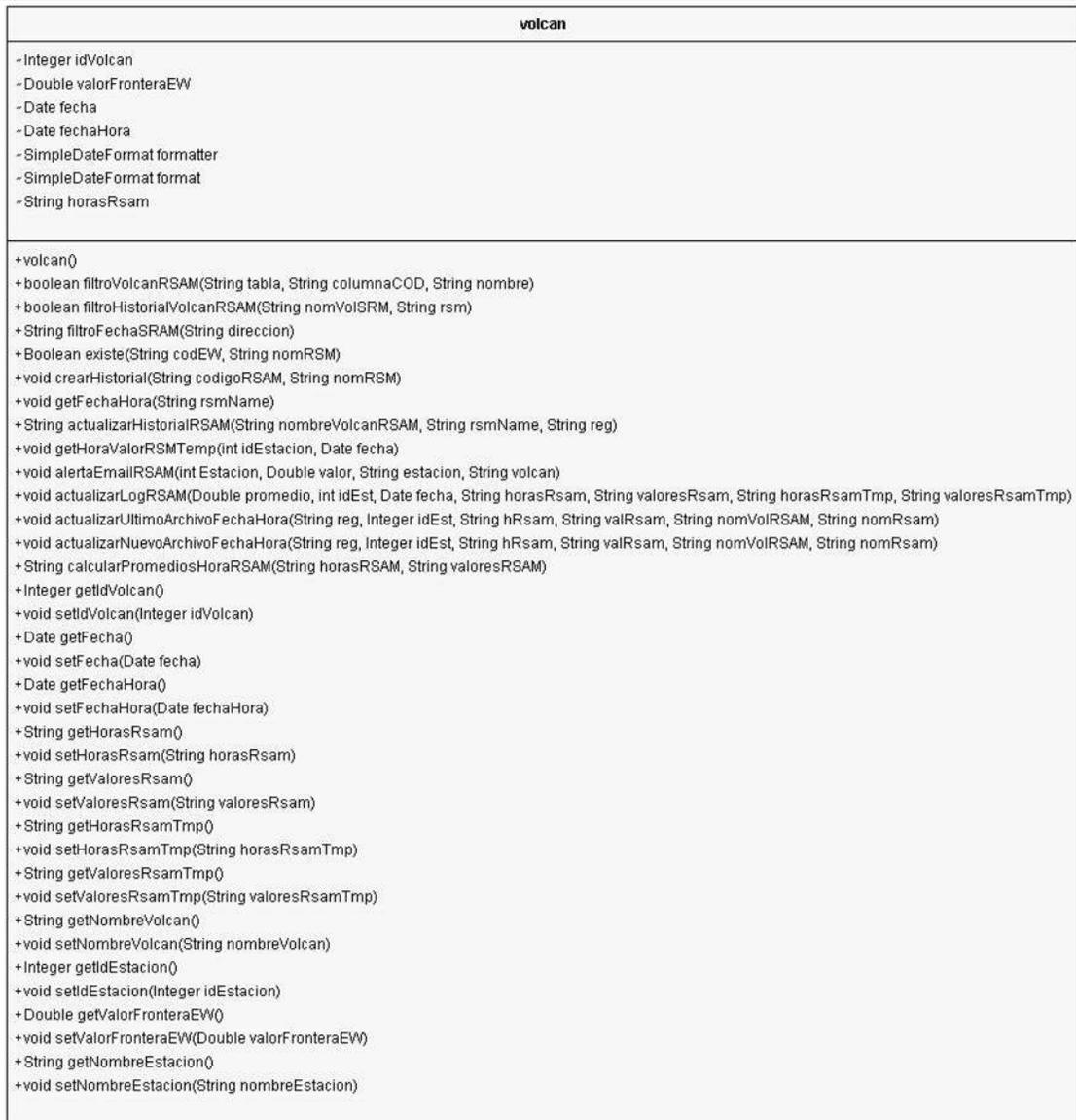


Figura 3.28 - Diagrama de clase Property

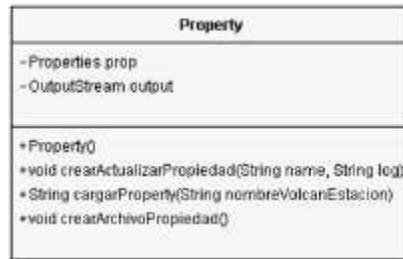
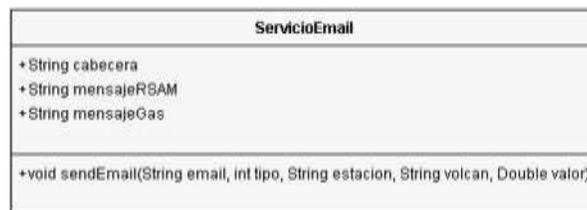


Figura 3.29 - Diagrama de clase ServicioEmail



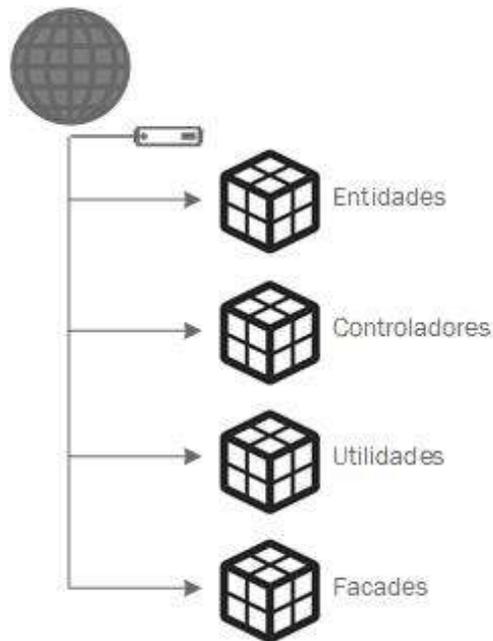
### 3.3.4.2. Clases MAVAMSI

La aplicación web, requerirá de un desarrollo más profundo y robusto; ya que además de conectarse con la base de datos, requerirá de la capacidad de realizar cálculos en base a los datos extraídos, para posteriormente, generar vistas que representen esta información.

Para realizar toda esta transaccionalidad, se ha determinado la creación de 4 paquetes de clases, que no solamente ayudarán a mantener el código más ordenado y organizado; si no que también ayudarán a aprovechar las bondades de la programación orientada a objetos utilizando la herencia y el polimorfismo.

La figura 3.30, muestra la estructura de paquetes que poseerá el código en clases que determinarán el comportamiento de la aplicación web.

Figura 3.30 - Diagrama de paquetes de clases de aplicación web MAVAMSI



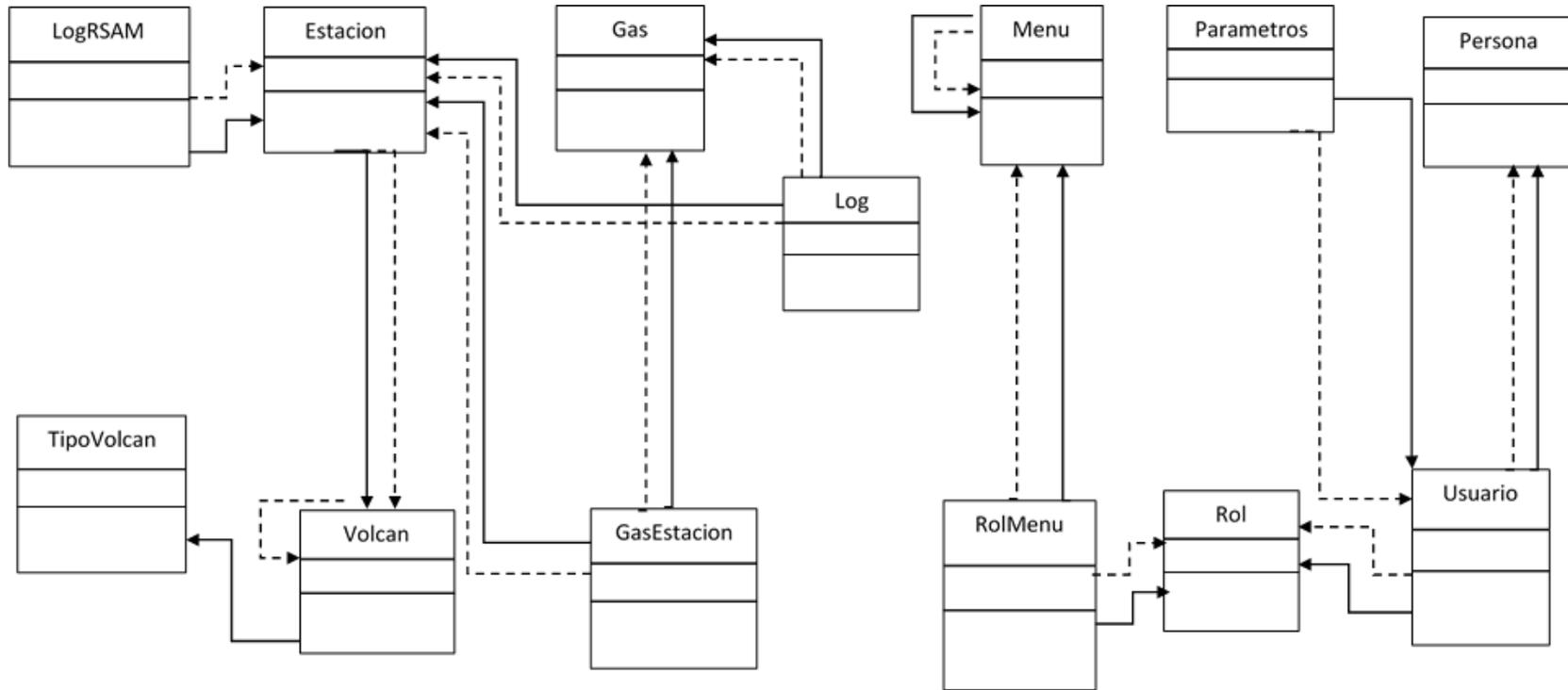
El primer paquete, contiene información propia de las entidades creadas a nivel de base de datos. Las clases de este paquete corresponden a estructuras de datos que ayudarán a mantener la integridad de la información obtenida de la base de datos para poder representarla gráficamente.

La figura 3.31, muestra todas las clases que forman parte de este paquete, así como la relación de herencia que existe entre cada una de ellas.

Las clases descritas en este paquete, instancian métodos y objetos descritos en las clases de los demás paquetes, es por eso, que podremos observar múltiples métodos en cada una de ellas.

Los atributos de cada clase, corresponden a los atributos de su tabla contraparte en la base de datos; esto con el fin de hacer que la aplicación sea consistente de inicio a fin, y de garantizar la fiabilidad de cada registro mostrado al usuario.

Figura 3.31 - Diagrama de clases de paquete Entidades, de proyecto MAVAMSI



Como se puede apreciar en la figura 3.31, solo vemos el diagrama con el nombre de cada clase. La herencia que existe entre clases, ayuda a que el código sea más simple y ayuda a que la aplicación se compile de forma más limpia; obteniendo resultados bastante aceptables en cuanto a rendimiento se refiere.

A continuación, desde la figura 3.32 hasta la figura 3.43 detallamos el contenido de cada clase.

*Figura 3.32 - Diagrama de clase LogRSAM*

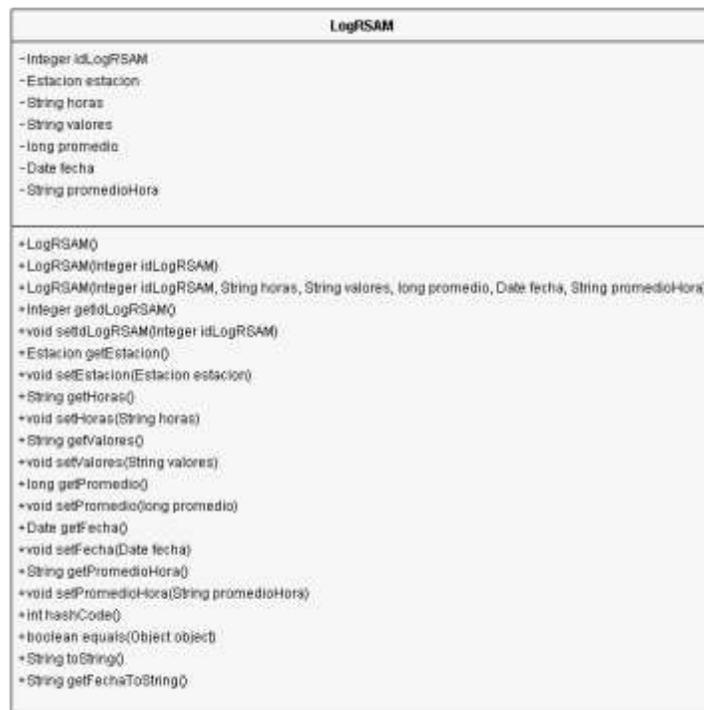


Figura 3.33 - Diagrama de clase Estacion



Figura 3.34 - Diagrama de clase Volcan



NOTA: No confundir con clase Volcan (figura 3.27), del proyecto MAVAMSI\_BOT1. A pesar que ambas clases tienen el mismo nombre, respetan a aplicaciones diferentes.

Figura 3.35 - Diagrama de clase Log

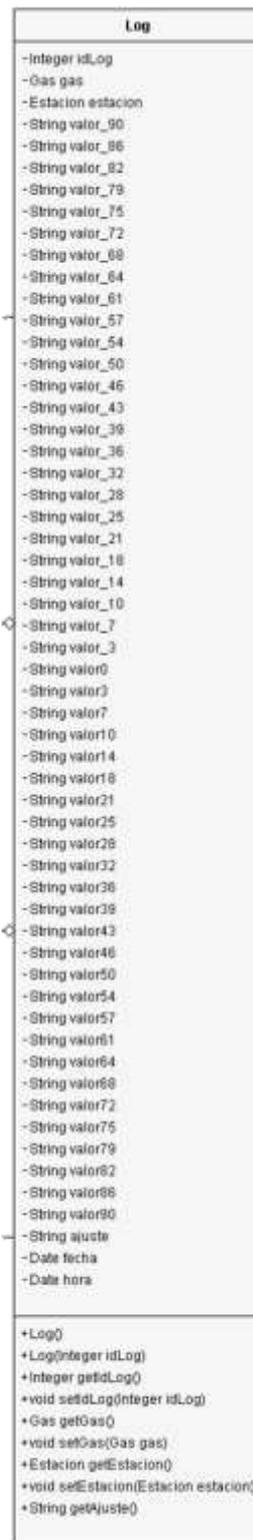


Figura 3.36 - Diagrama de clase Gas



NOTA: No confundir con clase Gas (figura 3.24), del proyecto MAVAMSI\_BOT1. A pesar que ambas clases tienen el mismo nombre, respectan a aplicaciones diferentes.

Figura 3.37 - Diagrama de clase GasEstacion

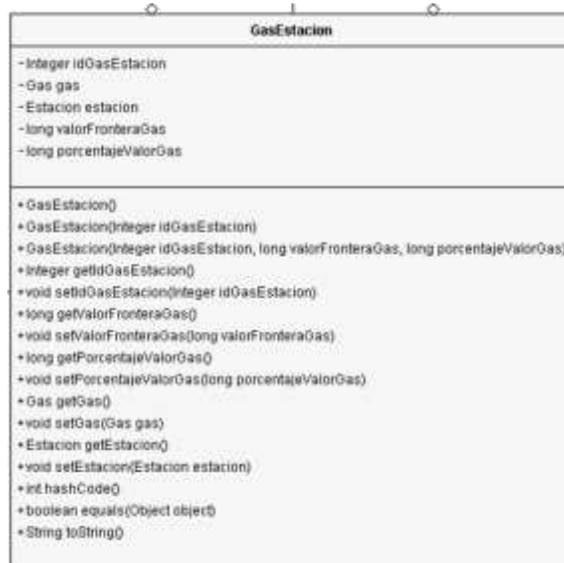


Figura 3.38 - Diagrama de clase Menu



Figura 3.39 - Diagrama de clase Parametros



Figura 3.40 - Diagrama de clase Persona

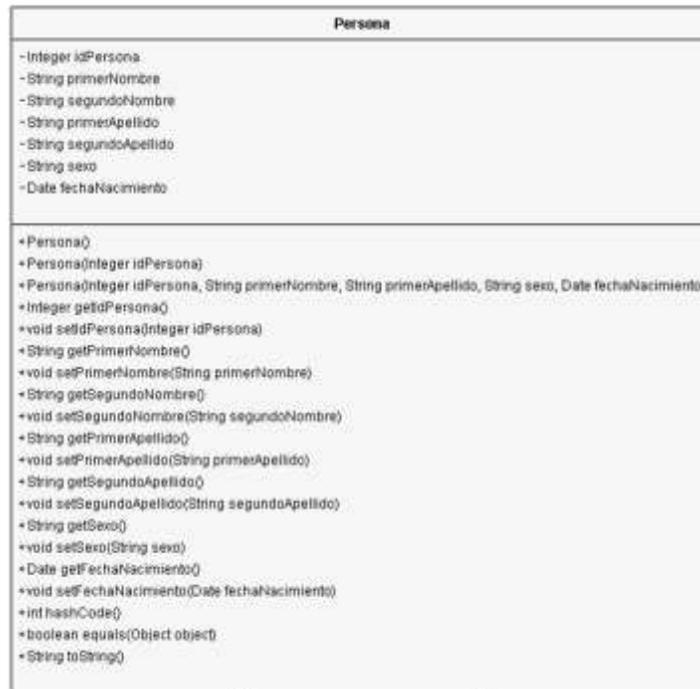


Figura 3.41 - Diagrama de clase Usuario

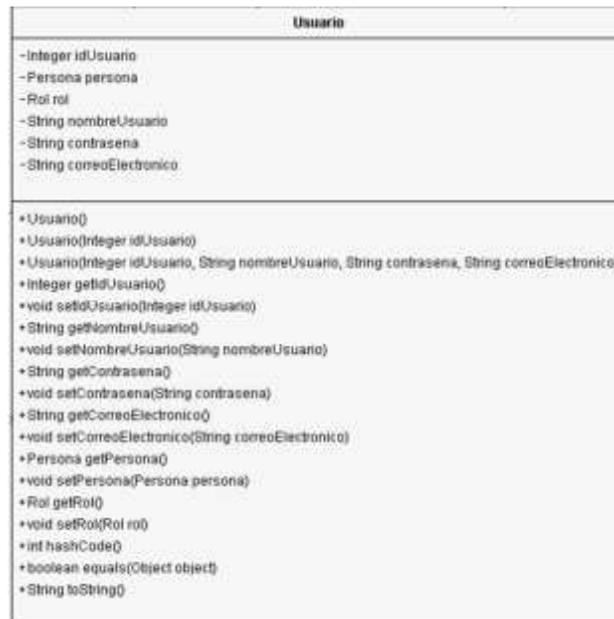


Figura 3.42 - Diagrama de clase Rol



Figura 3.43 - Diagrama de clase RolMenu



Volviendo al análisis de la figura 3.30, ahora veremos el detalle de las clases que forman parte el paquete Controladores.

En este paquete, se agrupan las clases que utiliza la aplicación para obtener todas las entradas del usuario en cada vista de la aplicación. Ya sea por medio de un formulario que agrupa valores a filtrar, inicio de sesión o cualquier otra interacción que realiza el usuario con la aplicación, ejerce el uso de alguna clase de este paquete.

Las clases del paquete Controladores, instancias clases del paquete Entidades, para posteriormente comunicar las entradas del usuario con la base de datos. A

partir de la figura 3.44 a la figura 3.52 se muestra el detalle de cada clase de este paquete.

Figura 3.44 - Diagrama de clase UsuarioController

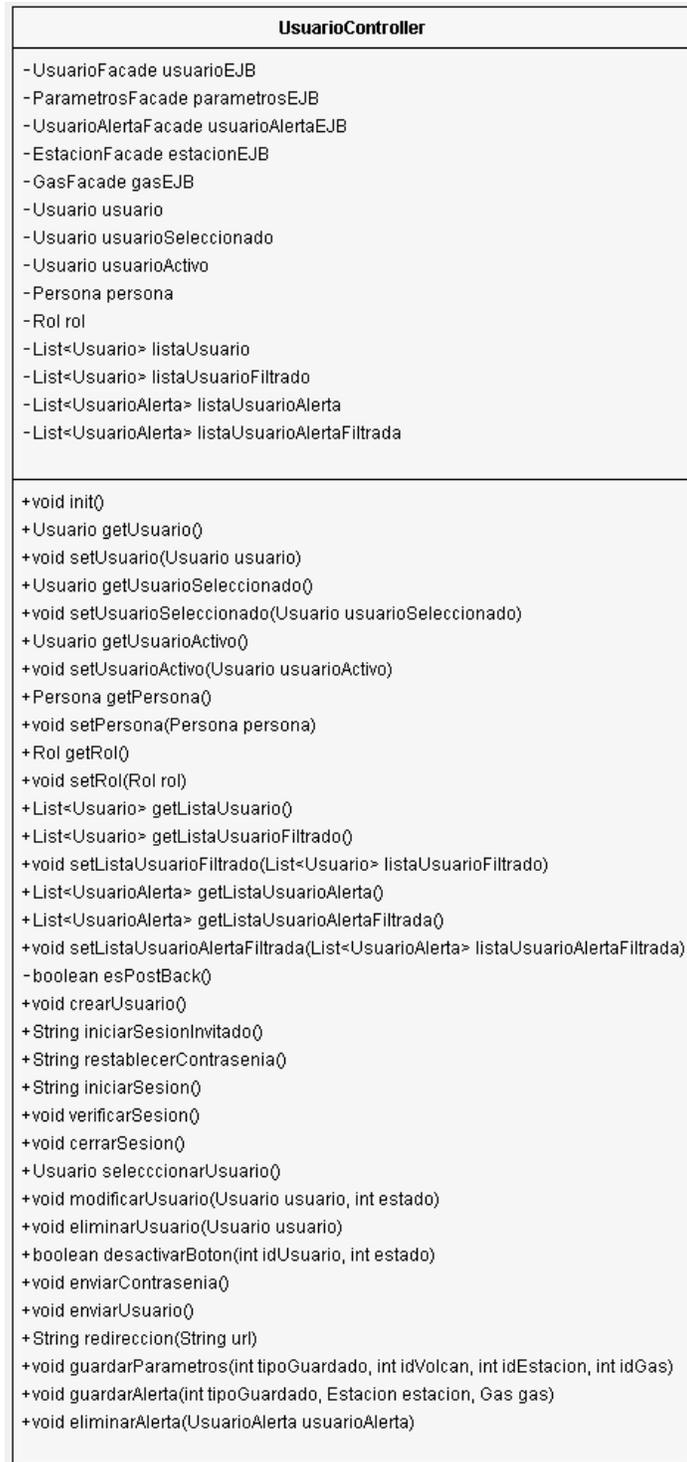


Figura 3.45 - Diagrama de clase RolMenuController

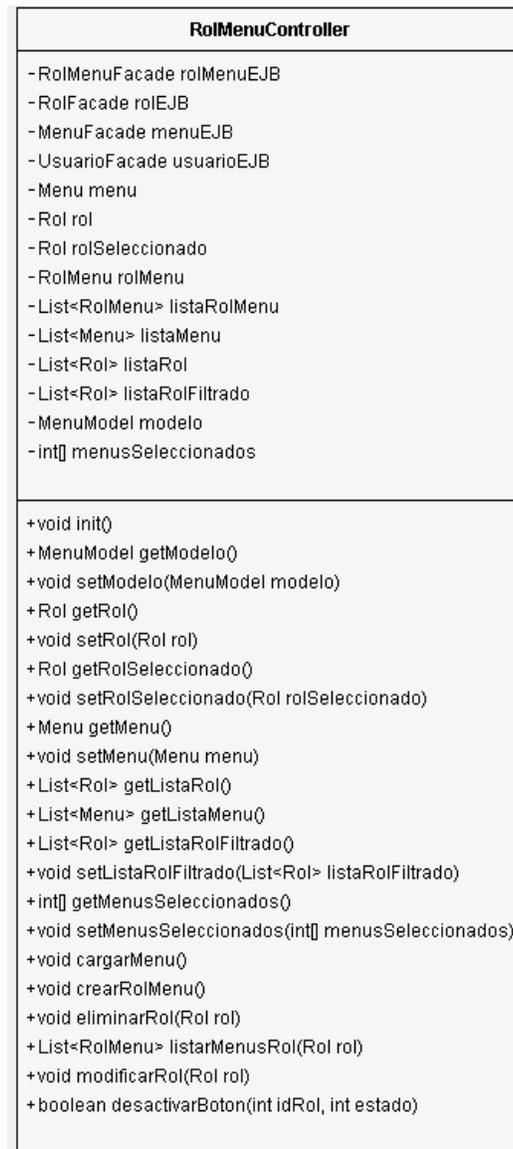


Figura 3.46 - Diagrama de clase LogController



Figura 3.47 - Diagrama de clase LogRSAMController



Figura 3.48 - Diagrama de clase GasController

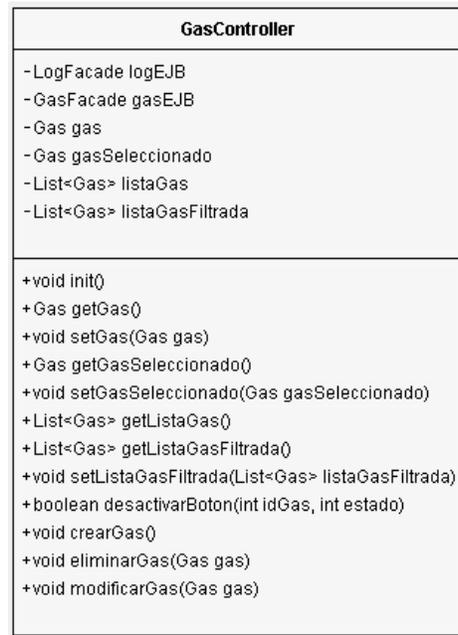


Figura 3.49 - Diagrama de clase EstacionController



Figura 3.50 - Diagrama de clase GasEstacionController

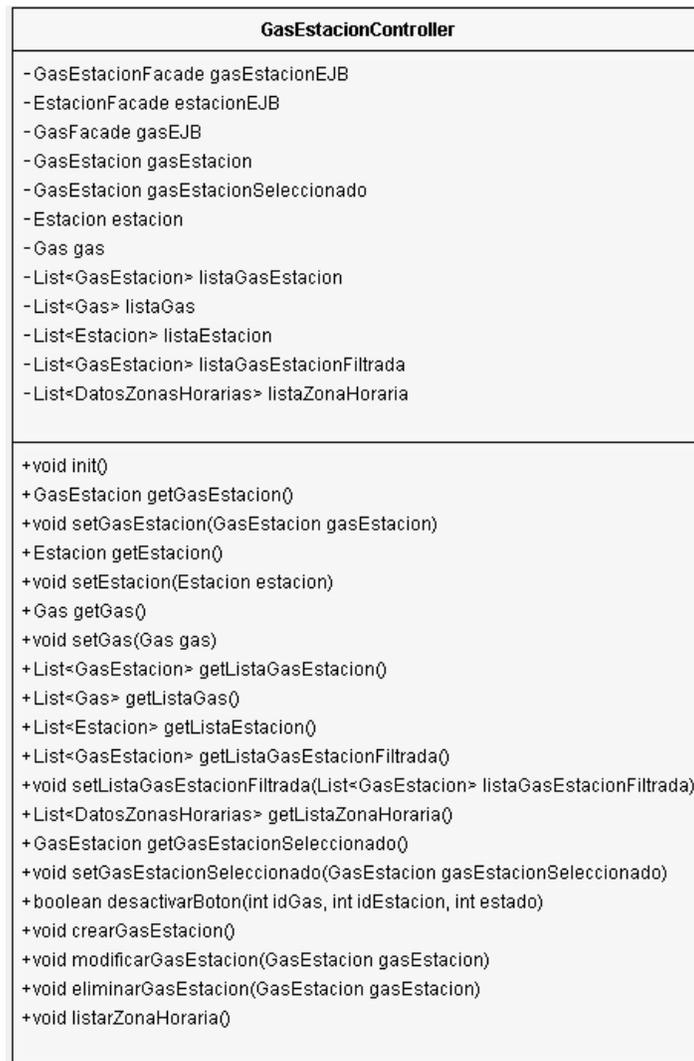


Figura 3.51 - Diagrama de clase VolcanController

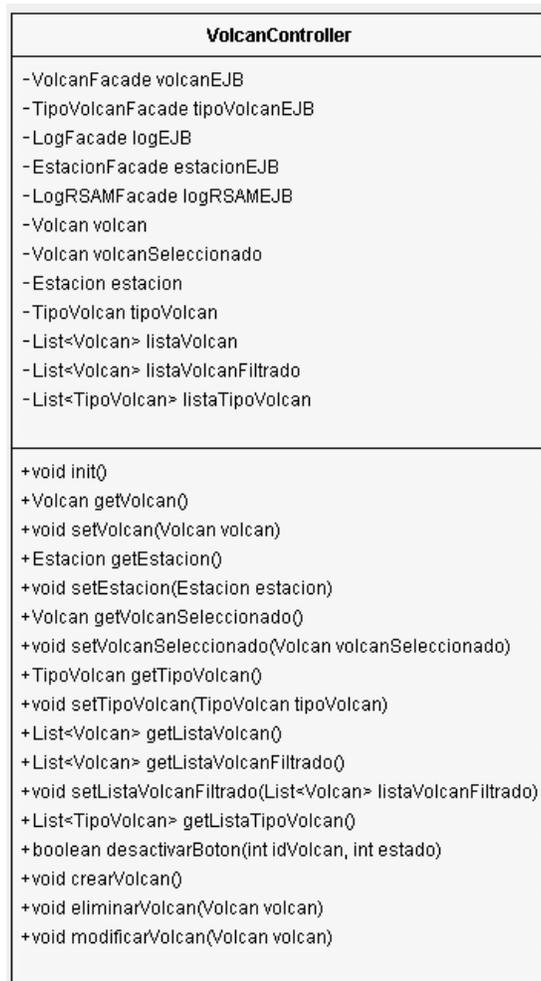
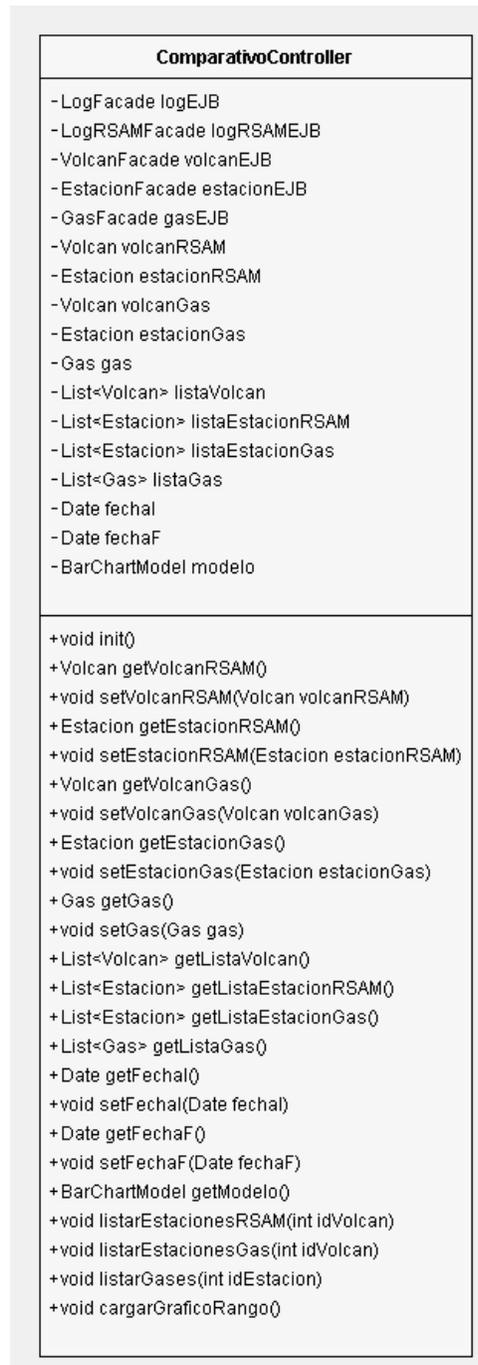


Figura 3.52 - Diagrama de clase ComparativoController



Volviendo a revisar la figura 3.30, toca revisar el contenido del paquete Utilidades.

El paquete Utilidades, contiene clases con método que ayudan a realizar cálculos, ajustes y relación de mediciones entre RSAM y emisiones de gas. Todas las clases de este paquete con clases estáticas, y todos los métodos de cada clase son métodos estáticos. Es por este motivo que veremos que todos los métodos en los diagramas serán públicos.

“El lenguaje de programación Java soporta métodos estáticos y variables estáticas. Los métodos estáticos, que tienen el modificador static en sus declaraciones, deben invocarse con el nombre de la clase, sin necesidad de crear una instancia de la clase.” (Oracle, 2015)

Desde la figura 3.53 a la figura 3.57, veremos el detalle de cada clase del paquete Utilidades.

*Figura 3.53 - Diagrama de clase DatosGas*

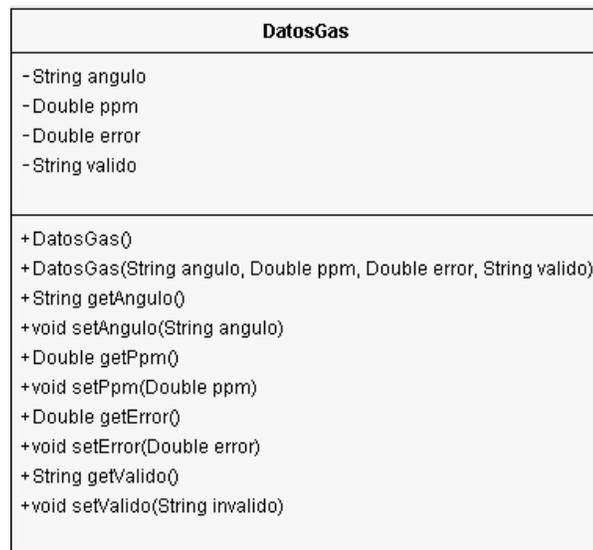


Figura 3.54 - Diagrama de clase DatosRSAM

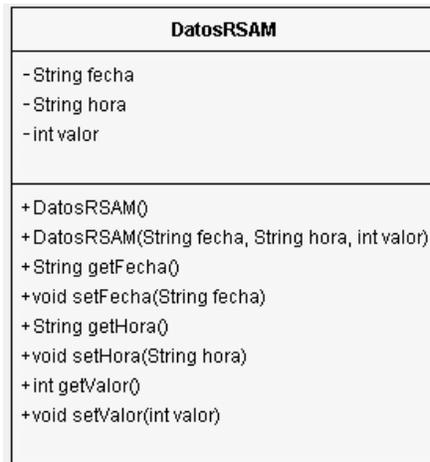


Figura 3.55 - Diagrama de clase DatosGasRango

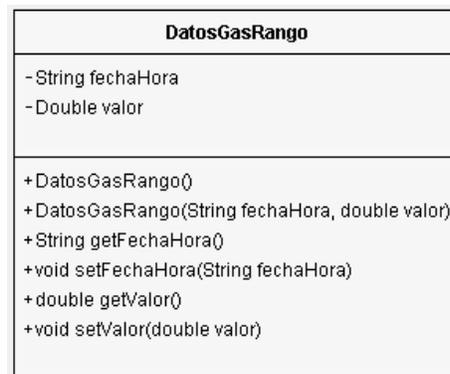


Figura 3.56 - Diagrama de clase ServicioCorreo

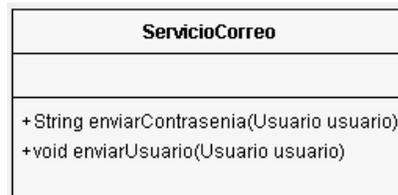
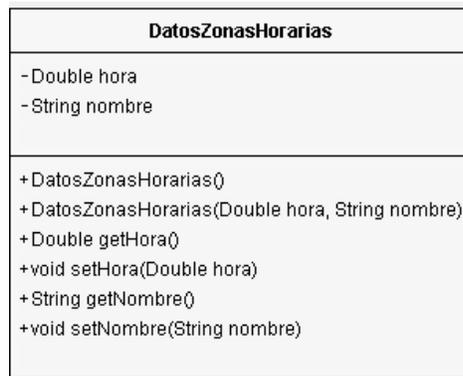


Figura 3.57 - Diagrama de clase DatosZonasHorarias



Para finalizar el estudio de la figura 3.30, veamos las clases que forman parte del paquete Facades.

Este paquete, agrupa todas las clases con métodos propios de conexión a la base de datos. En este paquete, está la definición de la construcción de todos los métodos de conexión a la base. También, se agrupan todos los constructores de consultas SQL que ingresarán a la base de datos.

Estas clases, también de encargan de devolver los datos bajo los que se construyen las instancias de las clases del paquete Entidades, propias para la construcción de las vistas que verá el usuario en la aplicación final.

La figura 3.58, muestra el diagrama de estas clases, que todas representan una herencia directa de la clase padre AbstractFacade; mediante esta clase, se construye la cadena de conexión de la aplicación a la base; y contiene los métodos necesarios para el inicio y la finalización de la conexión.

Las figuras entre la figura 3.59 y la figura 3.72 muestran el detalle de las clases que forman parte del paquete Facades.

Figura 3.58 - Diagrama de clases de paquete Facades de proyecto MAVAMSI

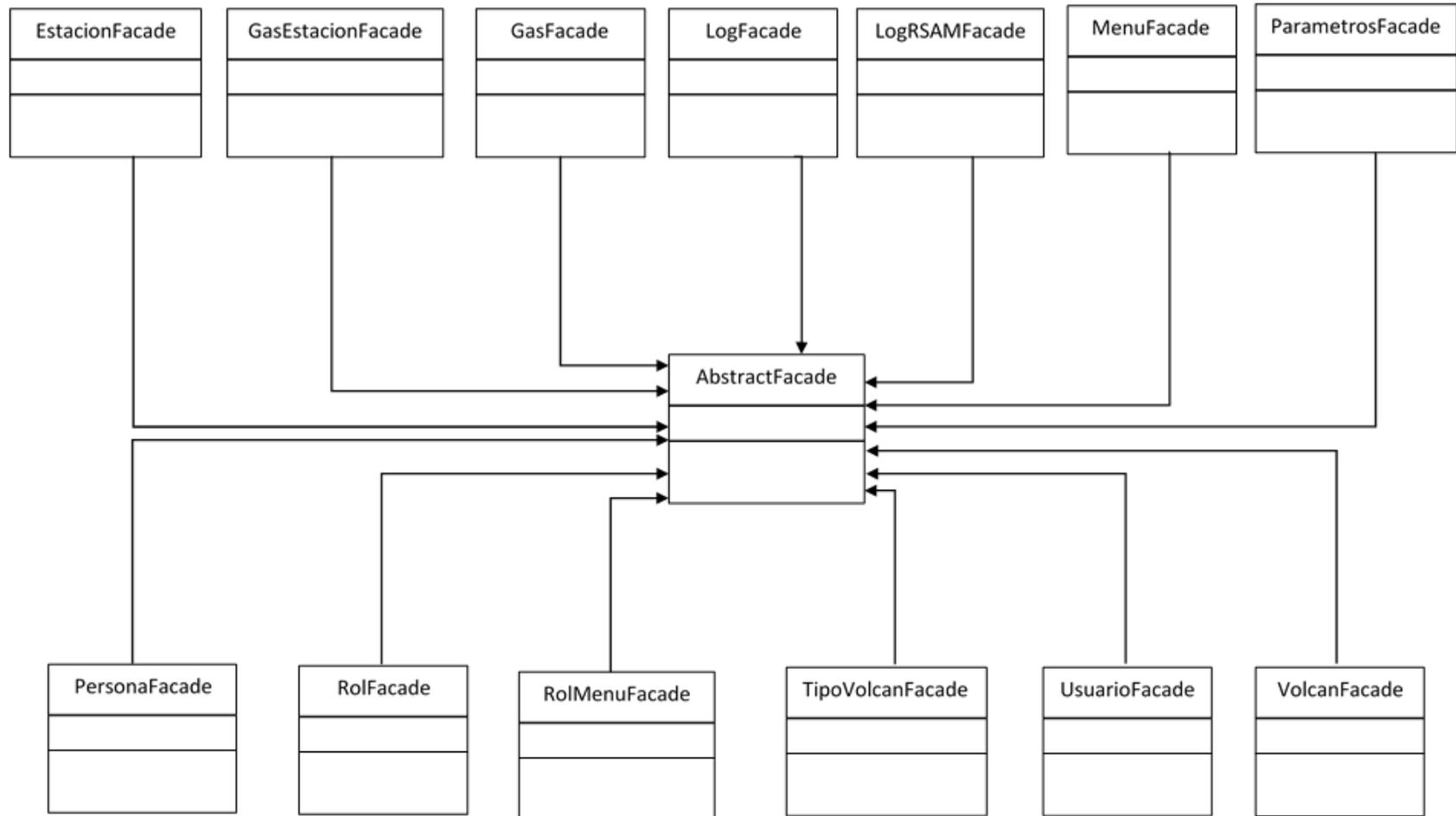


Figura 3.59 - Diagrama de clase AbstractFacade

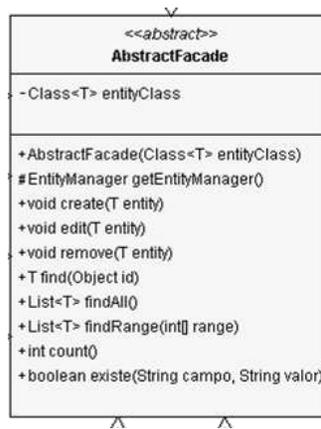


Figura 3.60 - Diagrama de clase EstacionFacade

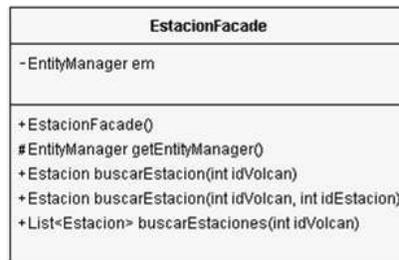


Figura 3.61 - Diagrama de clase GasEstacionFacade

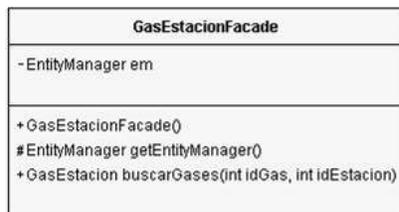


Figura 3.62 - Diagrama de clase GasFacade

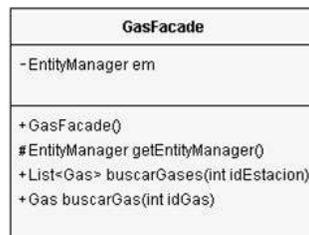


Figura 3.63 - Diagrama de clase LogFacade

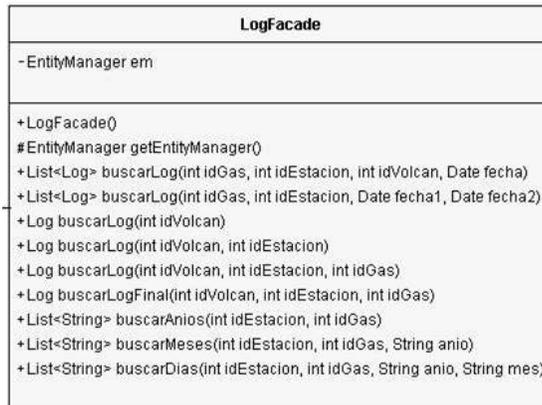


Figura 3.64 - Diagrama de clase LogRSAMFacade

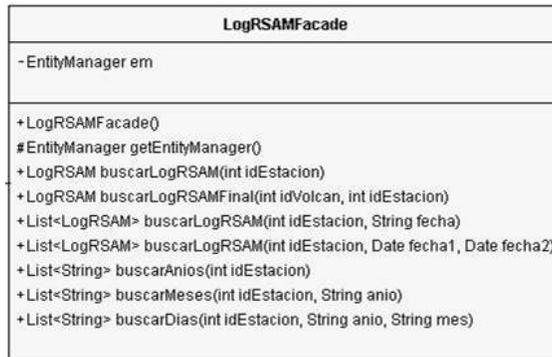


Figura 3.65 - Diagrama de clase MenuFacade

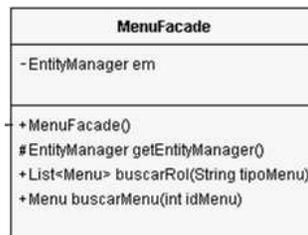


Figura 3.66 - Diagrama de clase ParametrosFacade

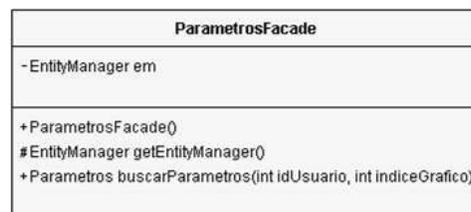


Figura 3.67 Diagrama de clase PersonaFacade

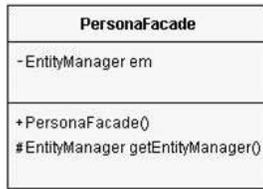


Figura 3.68 - Diagrama de clase RolFacade

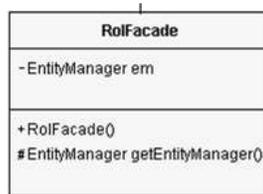


Figura 3.69 - Diagrama de clase RolMenuFacade

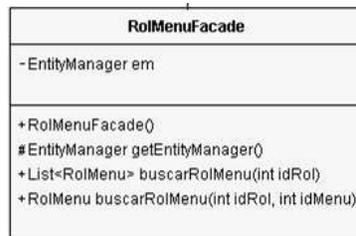


Figura 3.70 - Diagrama de clase TipoVolcanFacade

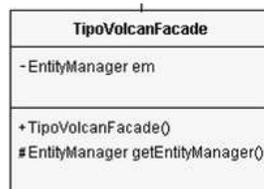


Figura 3.71 - Diagrama de clase UsuarioFacade

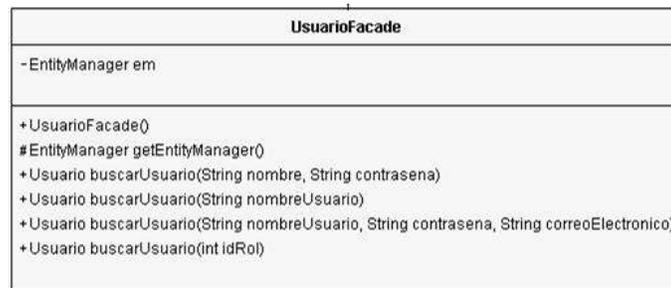
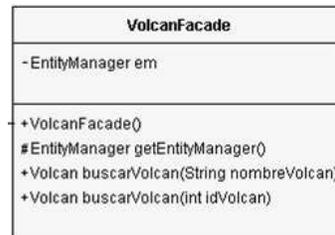


Figura 3.72 - Diagrama de clase VolcanFacade



## **3.4. DISEÑO DE INTERFACES Y NAVEGACIÓN**

En los apartados anteriores, conocimos a fondo cómo estará constituida la aplicación desde sus entrañas. Definimos el comportamiento de la misma por medio de flujogramas y clases que nos ayudaron a establecer el compartimiento de la aplicación en su conjunto. Resumiendo, se ha definido todo el comportamiento back-end de la aplicación. En el presente apartado, buscaremos definir el comportamiento front-end de MAVAMSI, la aplicación web; es decir, lo que estará de cara al usuario.

### ***3.4.1. MAVAMSI, el sitio web***

Cuando volvemos a revisar la figura 3.1, entendemos que MAVAMSI no es más que el resultado final de una extensa cadena de componentes de software. Todo el sistema en conjunto busca unificar la información en un solo alojamiento, para que al final del día, el usuario pueda acceder a ella desde cualquier dispositivo, a cualquier hora, y desde cualquier lugar.

Esta serie de ideas, nos hacen pensar que la forma más efectiva para representar la información es por medio de un sitio web; que si bien, de cara al usuario no dejará de ser un sitio más; de fondo tiene una robusta estructura que le permite ser una aplicación escalable y adaptable a cualquier análisis volcánico, de cualquier volcán.

El nombre MAVAMSI, es un conjunto de siglas que engloban el objetivo de la aplicación. MAVAMSI proviene de Monitoreo y Análisis de Variables Multiparamétricas y Sismisidad.

Las siglas ayudan a darle identidad propia a la aplicación, y ayuda a distinguirla de cualquier otro sitio web o software que pueda presentar datos similares. Para ayudar a estos propósitos, se diseñó un logo que ayude a identificar el sistema. La figura 3.73 muestra el logo de la aplicación que estará de cara al usuario.

*Figura 3.73 - Logo de MAVAMSI*



El resto del diseño del sitio, tratará de ser lo suficientemente fácil de usar por cualquier usuario, y que permita hacer el análisis de forma clara y legible.

Se han utilizado librerías de PrimeFaces para la representación de datos y gráficos, en compañía con Javascript para que ayude con la funcionalidad de cada elemento interactivo del sitio.

“JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos.

JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. El núcleo de JavaScript puede extenderse para varios propósitos, complementándolo con objetos adicionales, por ejemplo:

- Client-side JavaScript extiende el núcleo del lenguaje proporcionando objetos para controlar un navegador y su modelo de objetos (o DOM, por las iniciales de Document Object Model). Por ejemplo, las extensiones del lado del cliente permiten que una aplicación coloque elementos en un

formulario HTML y responda a eventos del usuario, tales como clicks del ratón, ingreso de datos al formulario y navegación de páginas.

- Server-side JavaScript extiende el núcleo del lenguaje proporcionando objetos relevantes a la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comunice con una base de datos, proporcionar continuidad de la información de una invocación de la aplicación a otra, o efectuar manipulación de archivos en un servidor.

JavaScript y Java son similares en algunos aspectos, pero fundamentalmente diferentes en otros. El lenguaje JavaScript se parece a Java pero no tiene el tipo estático (static) de Java, ni tiene un chequeo de tipos fuerte. JavaScript usa la mayoría de la sintaxis de expresiones de Java, convenciones de nombrado, y las construcciones básicas de control de flujo, razón por la cual se le cambió el nombre del original LiveScript a JavaScript.

En contraste con el sistema de clases construidas por declaraciones que se usa en tiempo de compilación de Java, JavaScript soporta un sistema de tiempo de ejecución basado en un pequeño número de tipos de datos que representan valores numéricos, lógicos, y de cadena de caracteres (string). JavaScript tiene un modelo de objetos basado en prototipos en lugar del modelo de objetos basado en clases, que es más común. El modelo basado en prototipo proporciona herencia dinámica; esto es, que lo que se hereda puede variar entre objetos individuales. JavaScript también soporta funciones sin ningún requerimiento declarativo especial. Las funciones pueden ser propiedades de los objetos, ejecutándose como métodos levemente tipados.

Comparado con Java, JavaScript es un lenguaje muy libre de forma. No hay que declarar todas las variables, clases, y métodos. No hay que preocuparse de si los métodos son públicos, privados, o protegidos, y no hay que implementar interfases.

Las variables, parámetros, y retornos de funciones no tienen que declararse explícitamente de un tipo dado.

Java es un lenguaje de programación basado en clases, diseñado para lograr seguridad de tipos, y ejecución rápida. Seguridad de tipos significa, por ejemplo, que no es posible hacer una conversión de tipos a un Integer de Java para obtener una referencia a un objeto, o acceder a memoria protegida mediante la alteración del bytecode de una clase. El modelo basado en clases de Java implica que los programas consisten exclusivamente en clases y sus métodos. La herencia de clases de Java y el tipado fuerte generalmente requiere que jerarquías de objetos fuertemente acopladas. Estos requerimientos hacen que la programación en Java sea más compleja que la programación en JavaScript.

En contraposición, JavaScript proviene en espíritu de una línea de lenguajes de programación más pequeños, con tipado dinámico, como HyperTalk, y dBASE. Estos lenguajes de scripting hacen accesibles las herramientas de programación a audiencias mucho más amplias. Esto se debe a su sintaxis más indulgente, funcionalidad especializada ya incluida, y mínimos requisitos para la creación de objetos.” (Mozilla, 2017)

Además, se utilizaron conjuntos de librerías CSS y JavaScript, que ayudan a que nuestra aplicación se vea bien, se adapte a cualquier tipo de pantalla y sea lo suficientemente interactiva con el usuario, para que su uso sea simple y fácil. La librería de estilos utilizada es Bootstrap.

“Bootstrap es el framework más popular del mundo para crear sitios y aplicaciones móviles responsivos. En su interior encontrará HTML, CSS y JavaScript de alta calidad para que el inicio de cualquier proyecto sea más fácil que nunca.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los

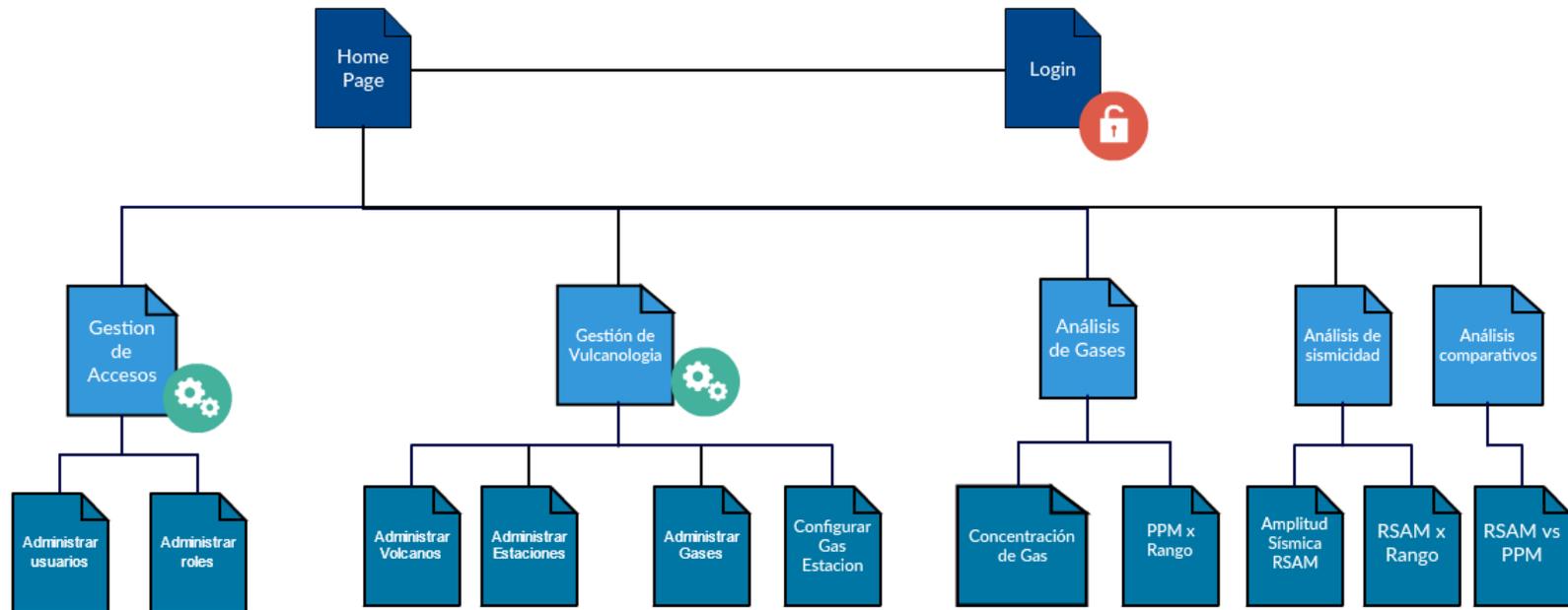
dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada bootstrap.less incluye los componentes de las hojas de estilo. Los desarrolladores pueden adaptar el mismo archivo de Bootstrap, seleccionando los componentes que deseen usar en su proyecto.” (Bootstrap, 2017)

### ***3.4.2. Mapa de Navegación***

La figura 3.74, muestra el mapa a seguir para navegar en las diferentes páginas del sitio.

Figura 3.74 - Mapa de Navegación de MAVAMSI



### 3.4.3. Diseño de Vistas

La figura 3.74, contiene una serie de vistas, que no dejan de ser páginas web. Unificar el formato de éstas vistas, es importante, ya que de ello dependerá el éxito del sistema.

Ya que lo que se busca es crear un sistema unificado, que sea fácil de manejar para el usuario, la figura 3.75 expresa de forma simple, el formato de diseño que se espera tener de cada una de las vistas que verá el usuario.

*Figura 3.75 - Diseño de plantilla de vistas de MAVAMSI*



La vista será bastante limpia, mostrando únicamente la información que le genere valor a los usuarios de la aplicación; evitando confusiones al momento de buscar en qué parte del sitio hay que acceder para ver cierto reporte, si todo el sitio les puede ser familiar al observar la vista de la pantalla de inicio.

Cada vista constará de un panel de menús, donde estará disponible todo el contenido del sitio. El panel de la izquierda en la figura 3.75 cumple con estos

propósitos. En la parte superior, veremos siempre visible el logo de la aplicación (figura 3.73), el cual, además, siempre apuntará al home page de la aplicación.

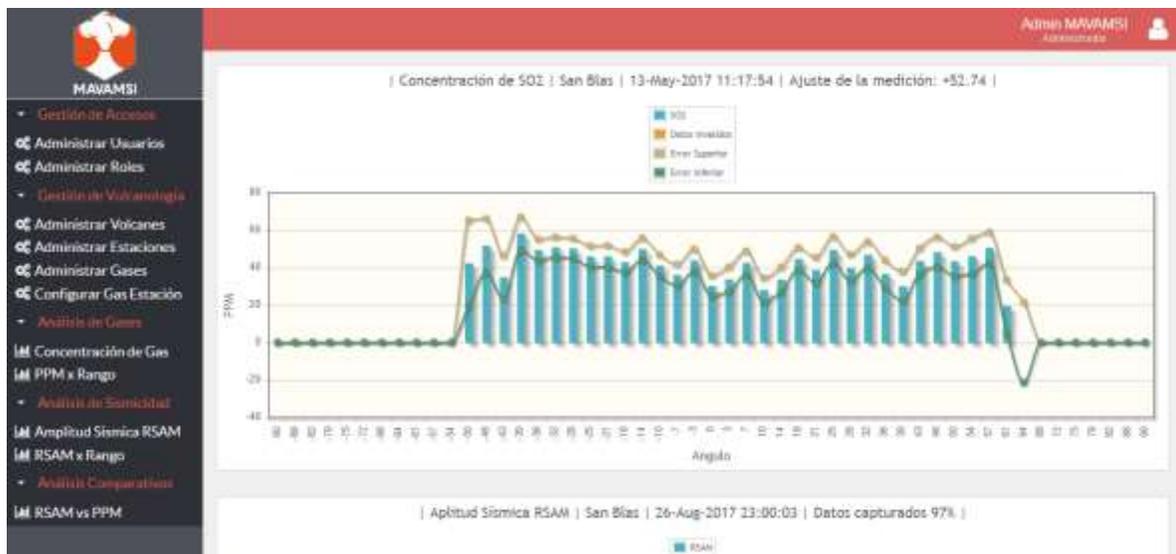
El panel de trabajo, que es el panel central de la figura anterior, será la porción de pantalla donde veremos, ya sea los formularios para ingreso de filtros de reportes a construir, así como el reporte como tal, compuesto de gráficos y tablas.

En la parte superior de la aplicación, siempre tendremos visible el nombre del usuario que se ha registrado para esa sesión, acompañado de su nombre, tal cual se ha registrado en la base de la aplicación.

Finalmente, en la parte inferior de la aplicación, se contará con un pie de página descriptivo a la versión de la implementación del sistema, y la fecha en la que fue modificado.

La figura 3.76, muestra el producto final de la vista de una pantalla de MAVAMSI.

Figura 3.76 - Pantalla de inicio de MAVAMSI



En la figura, podemos ver los menús de la aplicación desplegados, haciendo referencia al mapa de navegación expuesto en la figura 3.74.

## 3.5. DISEÑO DE SEGURIDAD

Debido a que MAVAMSI es una aplicación web, la seguridad de la información es un punto de vital importancia para su correcto uso y funcionamiento. En este apartado, se pretenden exponer, las formas en la que la aplicación resguarda la información por sí misma, evitando que caiga en manos equivocadas que quieran malversar los datos.

### 3.5.1. Inicio de Sesión

La figura 3.77, muestra la vista para iniciar sesión en la aplicación. No importa desde qué dirección del sitio se quiera ingresar directamente en el navegador, la primera página en aparecer de cara al usuario, siempre será la de inicio de sesión.

Figura 3.77 - Módulo de vista de inicio de sesión a MAVAMSI



La imagen muestra una interfaz de usuario para el inicio de sesión. El título del formulario es "Inicio de Sesión". Hay dos campos de entrada de texto: "Usuario" y "Contraseña". Debajo de ellos hay un botón rojo con un icono de flecha y el texto "Iniciar Sesión". En la parte inferior del formulario, hay un recuadro gris que contiene un cuadro de verificación con el texto "No soy un robot" y el logo de reCAPTCHA. Debajo de este recuadro hay un enlace que dice "¿Olvidó su contraseña?". En la parte inferior del formulario, hay un enlace que dice "Ingresar como invitado".

Cuando el usuario ha iniciado sesión en la aplicación, podrá navegar tranquilamente explorando el contenido del sitio; sin embargo, si la sesión se ha vencido por inactividad, el usuario estará obligado a iniciar sesión nuevamente.

En la figura anterior además, podemos ver que tiene un módulo incrustado de validación de personas; esto se ha implementado con el fin de proteger al sistema de posibles ataques cibernéticos haciendo peticiones ilimitadas al servidor que pueda bajar de línea la aplicación. Con este pequeño filtro, evitamos estas acciones, obligando al usuario a identificar que no es una máquina quien trata de acceder al portal.

Para estos fines, se ha implementado otro producto de Google, llamado reCaptcha. “ReCAPTCHA protege a los usuarios de Internet del spam y del abuso donde quiera que vayan. Como parte de la API de Google SafetyNet, mitiga los riesgos en los dispositivos móviles a través de una mayor seguridad y utiliza la última tecnología Invisible reCAPTCHA para optimizar la experiencia del usuario.

ReCAPTCHA es un servicio gratuito que protege su sitio web del spam y el abuso. ReCAPTCHA utiliza un motor de análisis de riesgo avanzado y CAPTCHAs adaptativos para evitar que el software automatizado se involucre en actividades abusivas en su sitio. Lo hace mientras deja pasar a sus usuarios válidos con facilidad.

ReCAPTCHA ofrece más que protección contra spam. Cada vez que nuestros CAPTCHAs son resueltos, ese esfuerzo humano ayuda a digitalizar texto, anotar imágenes y construir conjuntos de datos de aprendizaje de máquina. Esto, a su vez, ayuda a preservar libros, mejorar mapas y resolver problemas duros de IA.

ReCAPTCHA está construido para la seguridad. Armado con el estado del arte de la tecnología, siempre se mantiene a la vanguardia de las tendencias de lucha contra el spam y el abuso. ReCAPTCHA está en guardia para ti, así que puedes descansar tranquilo.” (Google, 20017)

### ***3.5.2. Roles y permisos***

Una vez adentro de la aplicación, los usuarios tendrán acceso únicamente a la información para la cual, fue planeada la creación de su usuario como tal; de esta

forma garantizaremos que usuarios que no necesitan realizar análisis sobre determinada información, no tengan acceso a información privilegiada.

La tabla 3.8, muestra los roles iniciales que se implementarán al momento de realizar el despliegue de la aplicación.

Cabe mencionar que este modelo es sólo una propuesta, ya que cualquier usuario con rol de Administrador, podrá modificar los permisos a los que tendrá acceso cada rol. Incluso, de considerar necesario, el administrador de la aplicación, podrá crear nuevos roles, para determinar permisos de acceso a módulos específicos a cierto grupo de usuarios.

El rol de un usuario, se especifica justo al momento de la creación del mismo; sin embargo, este puede ser modificado por un usuario Administrador, y así, cambiar el rol para obtener permisos específicos de la aplicación.

Tabla 3.8 - Roles y Permisos de MAVAMSI

ROL	DESCRIPCIÓN	PERMISOS
<b>Administrador</b>	<p>Tiene acceso a todos los módulos del sistema, puede eliminar crear y modificar usuarios, estaciones, roles, volcanes, gases además de configurar los valores predeterminados que definen cuando se enviará una alerta y visualizar cada uno de los gráficos y análisis</p>	<ul style="list-style-type: none"> <li>• Gestión de Accesos</li> <li>• Administrar Usuarios</li> <li>• Administrar Roles</li> <li>• Gestión de Vulcanología</li> <li>• Administrar Volcanes</li> <li>• Administrar Estaciones</li> <li>• Administrar Gases</li> <li>• Configurar Gas Estación</li> <li>• Análisis de Gases</li> <li>• Concentración de Gas</li> <li>• PPM x Rango</li> <li>• Análisis de Sismicidad</li> <li>• Amplitud Sísmica RSAM</li> <li>• RSAM x Rango</li> <li>• Análisis Comparativos</li> <li>• RSAM vs PPM</li> </ul>
<b>Investigador</b>	<p>Tiene los mismos accesos que el administrador a excepción de la gestión de accesos, estos quiere decir que este rol no puede eliminar usuarios cambiarles el rol ni crear ningún tipo de acceso.</p> <p>Fuera de esto si es posible que pueda configurar nuevas estaciones, volcanes y gases y puede visualizar todos los gráficos.</p>	<ul style="list-style-type: none"> <li>• Gestión de Vulcanología</li> <li>• Administrar Volcanes</li> <li>• Administrar Estaciones</li> <li>• Administrar Gases</li> <li>• Configurar Gas Estación</li> <li>• Análisis de Gases</li> <li>• Concentración de Gas</li> <li>• PPM x Rango</li> <li>• Análisis de Sismicidad</li> <li>• Amplitud Sísmica RSAM</li> <li>• RSAM x Rango</li> <li>• Análisis Comparativos</li> <li>• RSAM vs PPM</li> </ul>
<b>Invitado</b>	<p>Para el usuario invitado solo es posible acceder a la parte de los análisis y gráficos por lo que no puede realizar ninguna configuración de usuario como por ejemplo configurar alertas, cambiar los gráficos del menú principal o pedir restablecer la contraseña. No es necesario que inicie sesión</p>	<ul style="list-style-type: none"> <li>• Análisis de Gases</li> <li>• Concentración de Gas</li> <li>• PPM x Rango</li> <li>• Análisis de Sismicidad</li> <li>• Amplitud Sísmica RSAM</li> <li>• RSAM x Rango</li> <li>• Análisis Comparativos</li> <li>• RSAM vs PPM</li> </ul>

---

# **CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN**

## 4. DESARROLLO DE LA SOLUCIÓN

En el capítulo anterior, se expuso de forma detallada el paso a paso a seguir para el desarrollo, configuración y puesta en marcha de todos los componentes necesarios para el funcionamiento de nuestra aplicación.

Quizás el punto en el que se trató de hacer más énfasis, es el hecho que nuestra solución requerirá de 3 aplicaciones in house, que serán desarrolladas por parte del equipo de trabajo de este proyecto. La figura 3.1 nos mostraba de forma descriptiva todos los componentes que será necesario poner en funcionamiento para que el sistema en conjunto trabaje de forma óptima.

A pesar que son varios componentes, la mayoría son aplicaciones de terceros, desarrolladas para fines específicos, que servirán como insumos de datos para nuestro sistema. Pero al final, la creación como tal de software, desencadenará en 3 aplicativos.

El desarrollo de este proyecto, se vio dificultado por la configuración del software de terceros, el cual, tuvo que adaptarse para el funcionamiento del resto de componentes de nuestro sistema. Además, se tuvieron diversos problemas por el acceso a la información y la extracción de la misma.

El presente capítulo, busca narrar todas las actividades de desarrollo planeadas al inicio del proyecto. La planeación inicial de estas actividades tenía un fin simple: entregar herramientas de software funcionales y adaptables fácilmente al comportamiento en conjunto de todo el sistema. Sin embargo, esta planeación se fue modificando conforme avanzaba el proyecto, debido a las diferentes circunstancias que se presentaban que hacían agregar nuevas actividades o modificar tiempos de entrega de las mismas.

## 4.1. PLAN DE ACTIVIDADES DE DESARROLLO INICIAL

Figura 4.1 - Diagrama de plan de actividades de desarrollo inicial de aplicaciones in house

TAREA	RECURSOS	SEMANAS																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Repaso de Tecnologías Java	4 Programadores, 4 Máquinas	█																							
Repaso de sentencias SQL y bases de datos MySQL	4 Administradores de base de datos, 4 Máquinas		█																						
Aprendizaje de tecnologías de transferencia de archivos	1 Programador, 1 Máquina	█																							
Aprendizaje de protocolo FTP	1 Programador, 1 Máquina		█																						
Aprendizaje de software FileZilla	1 Programador, 1 Máquina		█																						
Aprendizaje de software Novac Program	4 Programadores, 4 Máquinas			█	█																				
Aprendizaje de protocolo SMTP	2 Programadores, 2 Máquinas		█																						
Aprendizaje de librerías de lectura y escritura de archivos de texto plano	1 Programador, 1 Máquina			█																					
Aprendizaje de lectura de sistema de carpetas y archivos	1 Programador, 1 Máquina			█																					
Aprendizaje de temporizadores	1 Programador, 1 Máquina				█																				

TAREA	RECURSOS	SEMANAS																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Aprendizaje de Batch	1 Programador, 1 Máquina																								
Aprendizaje de GlassFish	1 Programador, 1 Máquina																								
Aprendizaje de JSF y JSP	3 Programadores, 3 Máquinas																								
Aprendizaje de PrimeFaces	1 Programador, 1 Máquina																								
Aprendizaje de EarthWorm	4 Programadores, 4 Máquinas																								
Gestión administrativa con MARN para acceso a EarthWorm	4 Analistas																								
Estudio y comprensión de señales RSAM y emisiones de gases	4 Analistas																								
Análisis de requerimientos de usuario	5 Analistas																								
Diseño de Base de Datos	4 Administradores de base de datos, 4 Máquinas																								
Desarrollo de MAVAMSI_BOT2	1 Programador, 1 Máquina																								
Desarrollo de MAVAMSI_BOT1	1 Programador, 1 Máquina																								

TAREA	RECURSOS	SEMANAS																								
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Comunicación de MAVAMSI_BOT1 con Base de Datos	1 Programador, 1 Máquina																									
Desarrollo de vista configuración para MAVAMSI_BOT1	1 Programador, 1 Máquina																									
Desarrollo de Plantillas HTML/CSS	1 Programador, 1 Máquina																									
Desarrollo de módulos de acceso a MAVAMSI	1 Programador, 1 Máquina																									
Desarrollo de módulos de gestión de volcanes	1 Programador, 1 Máquina																									
Desarrollo de módulos de análisis de gases	1 Programador, 1 Máquina																									
Desarrollo de módulos de análisis de sismicidad	1 Programador, 1 Máquina																									
Desarrollo de módulo de análisis comparativo	1 Programador, 1 Máquina																									
Pruebas individuales de cada herramienta	4 Analistas																									
Pruebas integradas de todo el sistema	4 Analistas																									
Creación de video tutoriales de usuario	4 Analistas																									

La figura 4.1, nos muestra todas las actividades que se consideraron necesarias para el desarrollo de todos los componentes de software de la solución, al inicio del mismo.

El avance de actividades marcado en color azul, representa a las actividades previas o de capacitación necesarias para el desarrollo de la aplicación. Este tipo de actividades, se contempló necesarias, ya que había que conocer bien las tecnologías que estábamos por emplear en el desarrollo de proyecto. Además, en este apartado se contempla el aprendizaje de fundamentos sísmicos y de emisiones de gases, indispensables para la recopilación de requerimientos con el usuario.

El avance de actividades marcado en color verde, representa a las actividades propias del desarrollo de software. Cabe mencionar, que en este punto, tenemos únicamente actividades de desarrollo de componentes de software, no se han tomado en cuenta configuraciones de elementos o aplicaciones creados por terceros. La configuración de estos elementos, la veremos a fondo en el próximo capítulo del presente trabajo.

Inicialmente, se pensó en el desarrollo del sistema en 6 meses, esto, debido a la complejidad que presentaba el mismo desde la fase de inceptión. Se asignaron tiempos prudenciales para la creación de cada uno de los componentes de software que formarían parte del sistema. Además, se agregaron tareas de aprendizaje y capacitación, de los componentes de terceros que sería necesario implementar.

Sin embargo, conforme el desarrollo fue avanzando, empezaron a surgir problemas e impedimentos que ocasionaron atrasos en la planificación mostrada en la figura 4.1. A continuación, veremos las actividades de desarrollo implementadas en cada componente in house de la solución, así como los diversos problemas que se fueron atravesando en el avanzar del proyecto.

## **4.2. DESARROLLO DE MAVAMSI\_BOT2**

En capítulos anteriores, describimos la funcionalidad necesaria de MAVAMSI\_BOT2. A resumidas cuentas, su funcionamiento se resume en la extracción de archivos de emisiones de gases de la computadora que recibe los datos del sensor ScanDoas y enviarlos a la máquina virtual de nuestro sistema.

A continuación, describimos el desarrollo y los retos enfrentados para crear este robot.

### ***4.2.1. Recursos necesarios***

- 1 programador de software.
- 1 máquina para desarrollo de software.
- 1 escáner de medición de emisiones de gas (ScanDoas).

### ***4.2.2. Actividades previas y de preparación***

- Repaso de tecnologías Java, sintaxis de programación, aplicaciones de escritorio, entorno de programación NetBeans.
- Aprendizaje de tecnologías de transferencia de archivos.
- Aprendizaje de protocolo FTP y librerías en Java.
- Aprendizaje de software FileZilla.
- Aprendizaje de software Novac Program.

### ***4.2.3. Actividades de desarrollo***

- Desarrollo de aplicación de escritorio para transferencia de archivos vía FTP.
- Desarrollo de vista de mantenimiento para configuración de aplicación de escritorio.

#### ***4.2.4. Actividades de pruebas de desarrollo***

- Prueba de salida de mediciones de emisión de gas en formato de archivos de texto (.txt).
- Prueba de transmisión de archivos de gas vía FTP.

#### ***4.2.5. Retos surgidos durante desarrollo y pruebas***

- La configuración de Novac Program, es única para cada sensor que se esté utilizando, por lo que se vio la necesidad de agregar un campo para alojar el código de sensor en los parámetros de la aplicación.
- Novac Program únicamente lee los archivos almacenados en la memoria del sensor, y los convierte a formato de texto (.txt) y no se encuentra en una constante conversión de archivos. Esto ocasiona que los archivos de nuevas mediciones producidas por el sensor no se convierten en tiempo real a formato de texto; por lo tanto, nace la necesidad de construir un programador de tareas que se encargue de abrir y cerrar Novac Program cada cierto tiempo, para tener diferentes conversiones de medidas.
- La estructura de carpetas, respecta al día, mes y año en que fue tomada la medida. La primera medida de cada día, ocasionaba error, pues no existía carpeta con este formato en el servidor. Fue necesario crear una cláusula que evaluara la fecha, y si no existía una carpeta con ese formato, crearla en el servidor de la aplicación.
- La implementación de este robot, sería muy complicada, ya que había que configurar un archivo de parámetros de la aplicación para adaptar su funcionamiento. Para hacer esta tarea más fácil, se creó una vista de configuración, que tomaba los valores del archivo de parámetros, y le permite al usuario modificarlos en la primera ejecución del programa.

- En una de las pruebas, se identificó que el sensor estaba dañado, por lo que las pruebas e implementación, se retrasaron mientras se gestionaba el acceso a un nuevo sensor, o la reparación del sensor dañado.

#### ***4.2.6. Actividades de ajuste de desarrollo***

- Modificación de código fuente de aplicación, para reparar puntos de mejora identificados en el punto anterior.
- Modificación de código fuente de vista de mantenimiento.
- Creación de proceso batch, que trabaje como programador de tareas, para ejecución y cierre de Novac Program.
- Creación de proceso batch, que trabaje como programador de tareas, para ejecución de MAVAMSI\_BOT2.

#### ***4.2.7. Limitantes de desarrollo***

- No se contaba a total disposición con el sensor ScanDoas, por lo que se iban haciendo ajustes al código después de realizar pruebas de mediciones de gases. Estas pruebas se realizaban de 2 a 3 veces por mes, por lo que la accesibilidad al hardware fue una grave limitante.
- Se requiere de total acceso a internet, para el funcionamiento de la aplicación.
- Existe escasa (y en parte nula) documentación de Novac Program y del sensor ScanDoas. La mayor parte del desarrollo se realizó a base de la experiencia del personal de investigación volcanológica de la Universidad de El Salvador.

## **4.3. DESARROLLO DE MAVAMSI\_BOT1**

En el capítulo anterior, describimos la absoluta importancia de este aplicativo para el funcionamiento de nuestro sistema. A modo de recordatorio, el robot MAVAMSI\_BOT1, se encarga de tomar toda la información arrojada por los sistemas fuentes de datos (Earthworm y Novac Program) y se encarga de extraer, transformar y cargar la data a la base de la aplicación (ETL). Además, MAVAMSI\_BOT1, se encarga del envío de alertas por correo, a los diferentes usuarios de determinada estación, cuando se observa un comportamiento anormal de un volcán.

A continuación, describimos el desarrollo y los retos enfrentados para crear este robot.

### ***4.3.1. Recursos necesarios***

- 1 programador de software.
- 1 máquina para desarrollo de software.

### ***4.3.2. Actividades previas y de preparación***

- Repaso de tecnologías Java, sintaxis de programación, aplicaciones de escritorio, entorno de programación NetBeans.
- Aprendizaje de lectura y escritura de archivos de texto plano.
- Aprendizaje de protocolo SMTP, librerías en Java, conexión a bases de datos
- Aprendizaje de lectura de sistema de carpetas y archivos.
- Aprendizaje acerca del uso de “timer task” en java.

### ***4.3.3. Actividades de desarrollo***

- Desarrollo de aplicación de escritorio para lectura de archivos de texto plano
- Inserción y actualización de registros con el contenido de los archivos de texto plano.
- Desarrollo de envío de alertas por correo electrónico.
- Desarrollo de vista principal para configuración de aplicación en modo automático y manual.

### ***4.3.4. Actividades de pruebas de desarrollo***

- Prueba de lectura de archivos de texto plano como “.txt”, “.RSM”.
- Prueba de lectura y desplazamiento de estructura de carpetas.
- Prueba de conexión a base de datos, consulta, inserción y actualización de datos.
- Prueba de envío de mensajes vía email a través de un correo electrónico de tercero.
- Prueba de ejecución de procesos cada cierto tiempo utilizando un temporizador y monitoreo sobre la ejecución de procesos en automático para “N” días.
- Pruebas de integridad de la información según escenarios problemáticos, verificando si la solución fue correcta.
- Prueba de carga masiva de datos, monitoreo sobre recursos que consume la aplicación.

### ***4.3.5. Retos surgidos durante desarrollo y pruebas***

- El objetivo principal de esta aplicación, es la captura de datos en tiempo real, el cual implica que los archivos generados por los sistemas

Earthworm y MAVAMSI\_BOT2, generen la correcta información en un sistema de carpetas. Esto implicó la creación de un algoritmo de lectura de carpetas hasta llegar a obtener la carpeta que contenía múltiples archivos a cargar en la base de datos. Las estructuras de los archivos “.txt” y “.RSM” difieren uno del otro, por lo cual se crearon métodos específicos de lectura y captura de la información.

- Para la lectura de datos de Gases, contenía una estructura de carpetas de 4 niveles en el siguiente orden: “Nombre del volcán” > “Nombres de estaciones de un volcán” > “carpetas de fechas” > “grupo de archivos .txt”. Se propuso esta estructura para agrupar los archivos de gas generados por estación sobre un volcán y acorde a fechas de creación; ya que en un día se crean “N” archivos cada cierto tiempo (promedio de 15 minutos como máximo). Antes de empezar a capturar la información se realizaron filtros por cada carpeta, en el cual debe de existir el nombre de la carpeta como registro en la base de datos principal, para agrupar la información según la base de datos. Un volcán tiene “N” estaciones, una estación tiene “N” fechas de creación, una fecha contiene “N” archivos “.txt” generados por el sistema Novac Program, un archivo “.txt” contiene “N” tipos de gas en un bloque de información de un rango desde -90 grados a 90 grados segmentados. Cada nivel tiene un filtro contra parámetros que deben de ser registrados desde la aplicación web (MAVAMSI), antes de que la aplicación de escritorio proceda a la lectura.
- Conforme se avanzó con las pruebas, se descubrió que debería de existir un registro del último archivo leído para no generar duplicidad de información en el cual se crearon tablas para mantener el registro del último archivo leído por nombre de volcán, estación, fecha y archivo “.txt” ingresado a la base de datos.
- Para la lectura de datos “RSM” se creó la estructura de 2 niveles, el primer nivel contiene el nombre del código Earthworm proporcionado por el mismo sistema, segundo nivel están todos los archivos “.RSM”. Cada

archivo contiene el registro de un mes y se actualiza cada 10 minutos una línea que apunta por cortes de 10 minutos. La lectura de la estructura de carpetas es similar a la de gases. Contiene un filtro que comprueba si el código existe en la base de datos y que debe ser ingresado desde el sistema Web, de no existir omite la lectura a fondo.

- Dentro del archivo de gases “.txt” existe una estructura compleja de captura de datos de 180 grados, el cual representa un registro al día en una hora determinada; para esto se definió que un archivo con un tipo de gas representaba un registro por hora en una fecha determinada, en el cual se creó una serie de columnas representando los grados; por cada grado existen cuatro datos (concentración, error de la medida, ajuste e indicador de medición correcta) que se almacenan como texto concatenado por un símbolo “/”.
- Dentro del archivo “.RSM” existen todos los registros para un mes divididos en bloques de 10 minutos. Existen múltiples registros en el cual la hora marca 99:99:99, que tiene dos significados, el primero que el registro aún no ha sido actualizado ya que el módulo de Earthworm crea los registros a futuro por día, segundo nunca se actualizo el registro, ya que Eartworm trabaja con un ring, si no captura el dato en ese momento lo perderá y seguirá con el siguiente registro. Para solventar ese problema se creó un algoritmo sobre el último archivo generado y sobre el nuevo a leer, ya que los registros en 99:99:99 se debían de omitir, creando entre estos espacio sin información un registro de la hora y con un valor en 0.
- Dentro del archivo “.RSM”, se contiene todas las fechas de un mes con información a futuro, pero cada fecha representa un día de datos que se traduce en un registro por día; al crear este registro se debería de estar actualizando según los nuevos datos generados, y se creó una estructura de texto, concatenando un símbolo “/”; esto fue para almacenar las horas y la otra columna en la tabla almacenaba valores con la misma estructura de texto plano concatenando el símbolo “/”.

- La definición de tiempo no es la correcta a la hora local contra los archivos “.RSM”, se investigó que las horas no correspondían a la hora local. Para solventar se creó un algoritmo con definición de un parámetro para agregar o restar horas, adicional que el registro moviera fechas si fuese necesario y cuadrar estas horas de creación contra la hora local.
- Se creó un método que enviara mensajes con un texto de alerta al momento de insertar o actualizar la información contra un valor ingresado desde el sistema Web, que verifica si un promedio excede para decidir si enviar un mensaje email. Definido a una lista de usuarios agregado para recibir notificaciones de alertas. Esto se definió para la lectura de gases y de datos RSAM.
- Se definió la creación de archivos de texto plano para guardar el historial de lo que sucede en la aplicación, ya que estará en modo automático puede presentar errores por falta de parámetros en la base de datos, errores en la estructura de carpeta y errores propios de programación no consideradas. Este archivo se actualiza luego de cada ejecución de lectura para ambos tipos de datos gases y rsam. Creando un nuevo archivo por día de ejecución.
- Se trabajó en la solución de múltiples y diferentes escenarios problemáticos, considerando que la información produzca fallos en la aplicación. Se construyeron métodos de captura de fallos que no deban inutilizar a la aplicación y pueda proseguir si la información externa es correctamente configurada.
- El sistema Earthworm puede presentar fallos en la captura del ring de datos, si no lo captura en un tiempo determinado perderá el valor a asignar para un tiempo definido, esto representa un flujo discontinuo al cargar la información a la base de datos.

#### ***4.3.6. Actividades de ajuste de desarrollo***

- Modificación de código fuente de aplicación, para reparar puntos de mejora identificados en el punto anterior.
- Modificación de código fuente de vista de mantenimiento.
- Creación de vista de la aplicación mostrando las acciones que el programa realiza internamente.
- Ejecución de pruebas de varios escenarios que puedan afectar el flujo de la lectura del sistema de archivos para gases y RSAM.
- Pruebas de carga masiva de datos, y ejecución de varios días en modo automático.

#### ***4.3.7. Limitantes de desarrollo***

- No se contaba a total disposición con el sensor ScanDoas, por lo que se iban haciendo ajustes al código después de realizar pruebas de mediciones de gases. Estas pruebas se realizaban de 2 a 3 veces por mes, por lo que la accesibilidad al hardware fue una grave limitante.
- Se requiere de total acceso a internet, para el funcionamiento de la aplicación. Aunque, de producirse un error, se puede realizar cargas manuales o semiautomáticas, colocando los archivos y su estructura de carpetas correctamente.
- Para la lectura de datos de gases y RSAM, deben estar correctamente configurados los servicios externos de carga de datos de gases vía FTP, el robot MAVAMSI\_BOT2 en al menos una estación de observación, y la correcta configuración del sistema Eartworm hacia el servidor principal.

## **4.4. DESARROLLO DE MAVAMSI**

Luego de pasar por los dos procesos que alimentan la base de datos del sistema, llega el momento de definir las actividades de desarrollo de la aplicación web, que será la cara al usuario de todo el esfuerzo descrito en el presente documento. La aplicación web MAVAMSI, tendrá la obligación de representar los datos recopilados por medio de gráficas y tablas que muestren el comportamiento de un volcán en específico.

A continuación, describimos el desarrollo y los retos enfrentados para crear este robot.

### ***4.4.1. Recursos necesarios***

- 2 programadores de software.
- 2 máquinas para desarrollo de software.

### ***4.4.2. Actividades previas y de preparación***

- Repaso de tecnologías Java, sintaxis de programación, aplicaciones de escritorio, entorno de programación NetBeans.
- Repaso de sistemas de gestión de base de datos relacional MySQL.
- Aprendizaje de configuración y funcionamiento de servidor GlassFish.
- Aprendizaje de tecnología JSF (Java Server Faces) y JSP (Java Server Pages).
- Aprendizaje de Framework PrimeFaces.

### ***4.4.3. Actividades de desarrollo***

- Desarrollo de base de datos.
- Desarrollo de plantillas HTML.

- Desarrollo de módulo de accesos.
- Desarrollo de módulo de gestión de volcanes.
- Desarrollo de módulo de análisis de gases.
- Desarrollo de módulo de análisis de sismicidad.
- Desarrollo de módulo de análisis comparativos.

#### ***4.4.4. Actividades de pruebas de desarrollo***

- Prueba de conexión a base de datos, consulta, inserción y actualización de datos.
- Prueba de gráficas de datos en tiempo real.
- Prueba de gráficas de emisión de gas.
- Prueba de gráficas de RSAM.
- Prueba de gráficas de análisis comparativo.
- Prueba de configuración de usuarios.
- Prueba de configuración de alertas.
- Prueba de sitio web en diferentes dispositivos.

#### ***4.4.5. Retos surgidos durante desarrollo y pruebas***

Debido a que esta aplicación, es el producto final de la recopilación de datos aplicada por los procesos automáticos del sistema, todos los retos surgidos en el desarrollo de MAVAMSI\_BOT1 y MAVAMSI\_BOT2, también afectaron, tanto el funcionamiento, como las fechas de desarrollo del sitio web.

Además de todos los retos surgidos en las otras 2 aplicaciones in house de la solución, también surgieron los siguientes puntos propios de MAVAMSI:

- Durante las pruebas de los gráficos de promedio hora se llegaba a un fallo cuando faltaba algún dato en función al tiempo, por lo que se creó un

proceso para que si en alguna hora determinada no existieran datos se les asignara el valor de 0.

- Inicialmente en el gráfico de concentración de SO<sub>2</sub> se mostraría en color naranja las mediciones que no eran mediciones buenas, pero debido a la diferencia entre los rangos aceptados y los datos erróneos el grafico se deformaba por lo que se optó por eliminar del grafico los datos con errores.
- Se crearon clases auxiliares de configuración para poder acoplar los datos de los gases y mediciones de RSAM al modelo de las tablas que exportan la información en los módulos de análisis.
- Debido al deterioro del escáner de emisiones de gas, se contó con muy poca información de concentración de gas para realizar pruebas en conjunto con mediciones de RSAM.
- Las fallas en los formatos de los datos dependían del funcionamiento del MAVAMSI\_BOT1. Por lo que se debía pausar el desarrollo a la espera de las correcciones en el robot.

#### ***4.4.6. Actividades de ajuste de desarrollo***

- Modificación de código fuente de aplicación, para reparar puntos de mejora identificados en el punto anterior.
- Modificación de modelo de base de datos, para reparar puntos de mejora identificados en el punto anterior.
- Creación de análisis de promedio hora.
- Creación de análisis de RSAM vs PPM promedio hora.

#### ***4.4.7. Limitantes de desarrollo***

- Se debe contar con información actualizada tanto de RSAM como de emisiones de gas. Ya que la aplicación web como tal, no realiza extracción, obtención o creación de datos de estas mediciones.

## 4.5. PLAN DE ACTIVIDADES DE DESARROLLO REAL

Figura 4.2 - Diagrama de plan de actividades de desarrollo real de aplicaciones in house

TAREA	RECURSOS	SEMANAS																																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
Repaso de Tecnologías Java	4 Programadores, 4 Máquinas																																												
Repaso de sentencias SQL y bases de datos MySQL	4 Administradores de base de datos, 4 Máquinas																																												
Aprendizaje de tecnologías de transferencia de archivos	1 Programador, 1 Máquina																																												
Aprendizaje de protocolo FTP	1 Programador, 1 Máquina																																												
Aprendizaje de software FilleZila	1 Programador, 1 Máquina																																												
Aprendizaje de software Novac Program	4 Programadores, 4 Máquinas																																												
Aprendizaje de protocolo SMTP	2 Programadores, 2 Máquinas																																												

TAREA	RECURSOS	SEMANAS																																										
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40			
Aprendizaje de librerías de lectura y escritura de archivos de texto plano	1 Programador, 1 Máquina																																											
Aprendizaje de lectura de sistema de carpetas y archivos	1 Programador, 1 Máquina																																											
Aprendizaje de temporizadores	1 Programador, 1 Máquina																																											
Aprendizaje de Batch	1 Programador, 1 Máquina																																											
Aprendizaje de GlassFish	1 Programador, 1 Máquina																																											
Aprendizaje de JSF y JSP	3 Programadores, 3 Máquinas																																											
Aprendizaje de PrimeFaces	1 Programador, 1 Máquina																																											
Aprendizaje de EarthWorm	4 Programadores, 4 Máquinas																																											
Gestión administrativa con MARN para acceso a EarthWorm	4 Analistas																																											

TAREA	RECURSOS	SEMANAS																																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40					
Estudio y comprensión de señales RSAM y emisiones de gases	4 Analistas	█	█	█	█	█	█	█	█	█																																				
Análisis de requerimientos de usuario	5 Analistas	█	█	█	█	█	█	█	█	█																																				
Diseño de Base de Datos	4 Administradores de base de datos, 4 Máquinas										█	█																																		
Desarrollo de MAVAMSI_BOT2	1 Programador, 1 Máquina													█	█	█	█																													
Desarrollo de MAVAMSI_BOT1	1 Programador, 1 Máquina													█	█	█	█	█	█	█	█																									
Comunicación de MAVAMSI_BOT1 con Base de Datos	1 Programador, 1 Máquina																																													
Desarrollo de vista configuración para MAVAMSI_BOT1	1 Programador, 1 Máquina																																													
Desarrollo de Plantillas HTML/CSS	1 Programador, 1 Máquina																																													
Desarrollo de módulos de	1 Programador, 1 Máquina																																													



En el capítulo 3 del presente trabajo, en el apartado Software In House de la Solución (Apartado 3.1.1); describimos al software In House, como todos los componentes que necesitan ser desarrollados por el equipo de trabajo, para el correcto funcionamiento de la aplicación. Esto, nos ayudará a diferenciar a las aplicaciones propias del proyecto con las aplicaciones de terceros; las cuales, también son imprescindibles para el funcionamiento del sistema, pero que ya estaban previamente desarrolladas por otros proyectos y equipos.

La figura 4.2 muestra el cronograma de actividades de desarrollo, con significativas modificaciones con respecto al diagrama analizado en la figura 4.1. Los diferentes retos enfrentados durante la parte inicial de desarrollo y pruebas del sistema, obligaron a que se tuvieran que hacer ajustes en la programación inicial del mismo.

El avance marcado en rojo en el figura 4.2, representa actividades de ajuste, que se tuvo la necesidad de programarlas en el plan de trabajo, en vista de las dificultades enfrentadas en el desarrollo del sistema. Esto, ocasionó un atraso de casi 4 meses en el plan inicial de desarrollo de la aplicación.

Como pudo verse de manifiesto en los apartados anteriores del presente capítulo, cada aplicación in house del sistema, tuvo que enfrentar diversos retos que obligaron a estirar los tiempos de desarrollo de la misma, atrasando la fase de programación e implementación del sistema en conjunto.

A esto, además, se vio la necesidad de agregar líneas de código a la aplicación en base a los resultados obtenidos en cada prueba realizada.

La incorporación de aplicaciones de terceros, utilizar hardware ad hoc a las necesidades del proyecto, y la gestión de información con entidades externas, fueron factores clave para la modificación en la planeación del mismo, y provocaron que la dificultad de desarrollar este tipo de sistemas aumentara.

A pesar de todos los retos que el equipo de trabajo tuvo que enfrentar, al final de esta escabrosa planificación de desarrollo, se logró tener una herramienta de software robusta, escalable, bajo los requerimientos tanto funcionales como no funcionales de la misma, y que garantice el correcto funcionamiento de todos los componentes que la forman; para así, entregar un producto final a los usuarios que ayude en el entendimiento de la actividad volcánica y que ayude a tomar rápida acción contra cualquier catástrofe natural que afecte nuestro medio.

---

# **CAPÍTULO 5: PLAN DE IMPLEMENTACIÓN**

---

## 5. PLAN DE IMPLEMENTACIÓN

En los capítulos anteriores, buscamos dejar por sentado en qué consiste MAVAMSI, cuál será su utilidad, cómo se diseñó, cuáles son sus limitantes y cómo fue su etapa de desarrollo y pruebas. Ahora es momento de dar a conocer cómo se implementó esta aplicación. Antes de iniciar, veamos un breve repaso de todo lo que estamos por implementar.

El Sistema MAVAMSI (Monitoreo y Análisis de Variables Multiparamétricas y Sísmicas), tiene como objetivo primordial apoyar el centro de vulcanología de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente; en el análisis de la actividad sísmica y emisión de gases volcánicos de las estaciones volcánicas. MAVAMSI comprende los siguientes componentes:

1. Procesamiento de datos sísmicos
2. Procesamiento de datos de gases volcánicos
3. Representación gráfica de datos sísmicos
4. Representación gráfica de gases volcánicos
5. Comparación grafica entre datos sísmicos y gases volcánicos
6. Generación de reportes
7. Administración de usuarios

Estos componentes del sistema no solo pretenden agilizar los procesos, sino que además proveer información que apoye en la toma de decisiones. El acceso será a través de una aplicación Web. Esta le permitirá al usuario acceder a sus diferentes opciones de menú dependiendo del usuario con el que haya iniciado el sistema; esto de acuerdo a roles asignados a los usuarios.

El sistema como tal, es un conjunto de aplicaciones que tiene el objetivo de extraer, transformar y cargar información en una base de datos centralizada, para posteriormente ser utilizada para representaciones gráficas por medio de un sitio web.

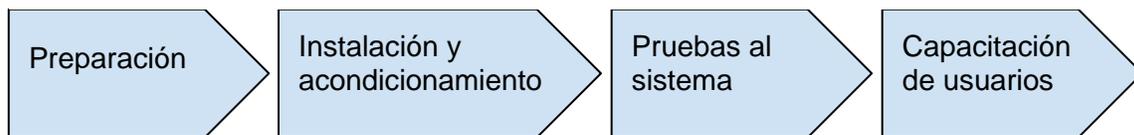
MAVAMSI, es sólo la cara, en forma de aplicación web de todo un proceso de unificación de datos que se ejecuta de trasfondo en un servidor web. Además, es el producto final de los datos producidos por software de terceros.

A continuación, describiremos el proceso de implementación de los diferentes componentes del sistema.

## 5.1. METODOLOGÍA DE IMPLEMENTACIÓN

Para la implementación del sistema se ha considerado seguir cuatro etapas las cuales indican el plan a seguir para el desarrollo de la implementación de MAVAMSI. Las cuatro etapas se presentan en la siguiente figura:

*Figura 5.1 - Metodología de Implementación*



### 5.1.1. Preparación

#### 5.1.1.1. Objetivos de la etapa

- Determinar el recurso humano que participara en el proyecto de implementación del sistema.
- Adquirir el soporte para lograr las condiciones necesarias para la implementación del sistema.
- Poseer el acceso a los datos necesarios para asegurar un buen procesamiento de datos en la implementación del sistema.

#### 5.1.1.2. Actividades de la etapa

- Establecer las metas, objetivos, y estrategias para la implementar y poner en marcha el sistema informático.
- Asignación del administrador de MAVAMSI, actividad en la que se define que empleado o miembro del centro de vulcanología de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente será el encargado de administrar el sistema posteriormente de la implementación y capacitación.

- Gestionar permanentemente el acceso a datos sísmicos proporcionados por el MARN y el acceso a los datos de emisión de gases obtenidos del equipo instalado en las estaciones de medición.

## ***5.1.2. Instalación y acondicionamiento***

### ***5.1.2.1. Objetivo de la etapa***

Establecer un entorno adecuado para la implementación del sistema, teniendo en cuenta las condiciones de red, software y hardware que permitan la operatividad del sistema.

### ***5.1.2.2. Actividades de la etapa***

- Identificación de los recursos necesarios.
- Selección del personal para la instalación de recursos.
- Configuración de servidor de aplicación o máquina virtual.
- Configuración de software de terceros.
- Instalación de software In House.

Entraremos en mayor detalle de estas actividades en el próximo apartado del presente documento.

## ***5.1.3. Pruebas al sistema***

### ***5.1.3.1. Objetivo de la etapa***

Crear un plan de pruebas con la mayoría de casos que se puedan dar dentro de los procesos de administración de usuarios, administración de roles, administración de volcanes, administración de estaciones, administración de gases, configuración de estaciones de gas, concentración de gas, análisis de PPM por rango, análisis de amplitud sísmica RSAM, análisis de RSAM por rango de tiempo y el análisis

comparativo de RSAM contra gases buscando identificar los posibles errores en la ejecución de MAVAMSI.

### ***5.1.3.2. Actividades de la etapa***

- **Pruebas con la conexión a la base de datos:** Las pruebas relacionadas con la conexión a la base de datos se realizara de manera remota para asegurar su acceso a través del internet.
- **Pruebas de error esperado:** Pruebas con alta probabilidad de mostrar un error no descubierto hasta el momento de las pruebas.
- **Pruebas de carga:** Pruebas con múltiples usuarios a la aplicación, para probar los requerimientos no funcionales de la aplicación.
- **Pruebas en diversos dispositivos:** Ejecuciones de la aplicación en diferentes dispositivos con diversos tamaños de pantalla.
- **Pruebas funcionales:** pruebas pilotos que permita tener a MAVAMSI en las condiciones óptimas para ponerse en producción, entre las pruebas a realizar están:
  - Comunicación de las Estaciones de medición con el servidor de las aplicaciones.
  - Conexión de MAVAMSI con el hardware de medición de gases volcánicos.
  - Comunicación de MAVAMSI con los bots de automatización.
  - Conexión de MAVAMSI con la base de datos.
  - Acceso a MAVAMSI por medio de diferentes navegadores web.
- **Análisis de datos:** Análisis de los resultados de pruebas comprobar los resultados alcanzados en las pruebas efectuadas y verificar las correcciones pertinentes.

## **5.1.4. Capacitación de usuarios**

### **5.1.4.1. Objetivo de la etapa**

Transmitir el conocimiento pleno del sistema, por medio de video tutoriales adecuados para dar a entender a cualquier usuario potencial de la aplicación, el funcionamiento completo de la misma.

### **5.1.4.2. Actividades de la etapa**

La tabla muestra los puntos a tomar en cuenta en las labores de capacitación, que se buscará cumplir con la creación de video tutoriales que cumplan la función de dar a conocer el funcionamiento de la aplicación.

*Tabla 5.1 - Actividades de capacitación*

<b>TEMA</b>	<b>TEMAS A TRATAR</b>
Gestión de accesos	-Creación de usuarios -Asignación de roles -Administración de usuarios
Gestión de vulcanología	-Registro de volcán -Registro de estaciones sísmicas -Registro gases a analizar -Configuración de estaciones de gas
Análisis de gases	-Configuración de alertas de gases -Análisis de concentración de gas en el aire -Concentración de gas en rango de tiempo
Análisis de sismicidad	-Amplitud sísmica en tiempo real (RSAM) -Análisis de RSAM por rango de tiempo
Análisis comparativo	-Análisis comparativo de RSAM contra emisión de gases por minuto

## 5.2. ACTIVIDADES DE IMPLEMENTACIÓN DE SOFTWARE

Como lo hemos repetido a lo largo de todo éste documento, nuestro sistema es la recopilación de diversas estructuras de software. Si retomamos el diagrama de componentes mostrado en la figura 3.1, podremos ver de hecho que el sistema requiere del funcionamiento de diversas herramientas que sólo funcionarán en conjunto.

La figura 3.1, además, nos permite identificar todos los componentes de software que debemos ya sea configurar, o implementar, para que el sistema pueda funcionar tal cual se había diseñado.

Cabe mencionar también, la importancia de este plan de implementación, ya que con una sola herramienta de software que no sea implementada, el sistema perderá su validez.

El cuadro 5.1 nos mostrará todas las aplicaciones que debemos considerar para el plan de implementación, y que requerirán de tiempo y recursos para llevar a cabo su correcta configuración.

Comparando el contenido del cuadro 5.1 con la figura 3.1, podremos notar que no todos los componentes necesitan de algún tipo de configuración del lado del equipo de trabajo, para el funcionamiento del sistema.

El servidor del SNET, y la instancia de Earthworm instalada en él, será responsabilidad de la entidad, y queda únicamente de nuestro lado, entablar comunicación con ella vía protocolo TCP, por medio de internet.

El servidor GMAIL, también queda fuera de los alcances de la instalación, que el robot MAVAMSI\_BOT1 se encarga de configurar todo lo que sea necesario para entablar configuración con el servidor de envío de correos electrónicos.

Cuadro 5.1 - Componentes de software a implementar

	<b>APLICACIÓN</b>	<b>TIPO DE APLICACIÓN</b>	<b>SERVIDOR</b>
<b>Servidores</b>	AMAZON_VM	Máquina virtual	-
	MAQUINA_ESTACION	Computadora personal	-
<b>Servidores de aplicaciones</b>	Filezilla	Servidor FTP	AMAZON_VM
	GlassFish	Servidor de aplicaciones web	AMAZON_VM
	MySQL	Servidor de base de datos	AMAZON_VM
<b>Software de terceros</b>	Novac Program	Lector de mediciones de emisiones de gases	MAQUINA_ESTACION
	Earthworm	Receptor de información de movimiento sísmico	AMAZON_VM
<b>Software in house</b>	MAVAMSI_BOT1	ETL de unificación de información	AMAZON_VM
	MAVAMSI_BOT2	Transportador de información de emisión de gases de MAQUINA_ESTACION a AMAZON_VM	MAQUINA_ESTACION
	MAVAMSI	Aplicación web	AMAZON_VM

El servidor GMAIL, también queda fuera de los alcances de la instalación, que el robot MAVAMSI\_BOT1 se encarga de configurar todo lo que sea necesario para entablar configuración con el servidor de envío de correos electrónicos.

Volviendo al cuadro 5.1, las filas marcadas en la parte superior, representan servidores de software que será necesario configurar previamente, antes de realizar cualquier instalación o configuración de software. Estos servidores se encargaran de alojar el resto de herramientas de la solución que le darán sentido al funcionamiento de las mismas.

No hay que perder de vista el cuadro 5.1, ya que se convertirá en el mapa de referencia de todo el proceso de implementación que estamos por explicar.

## 5.3. CONFIGURACIÓN DE SERVIDORES

### 5.3.1. *Configuración de Máquina Virtual*

En el capítulo 3, se describió de forma detallada la utilización de una máquina virtual en nuestro proyecto. Explicábamos la necesidad de implementar una máquina, debido a que un simple servidor para aplicaciones web no bastaba, ya que era necesaria la configuración de una computadora que se encargara de albergar todos los componentes de software del sistema y mantenerlos en ejecución.

En entornos ideales, lo más apropiado es que el proyecto hubiese tenido una máquina que se pudiera utilizar como servidor 24/7 para ejecutar todos los componentes; sin embargo, como se explicaba en el capítulo 3, el proyecto no tenía asignado ningún equipo, ni presupuesto para la implementación de algo. Fue de esta forma que se optó por adquirir los servicios de la computación en la nube de Amazon.

La tabla 3.1 nos mostraba un comparativo de todos los proveedores de servicios en la nube que se adaptaban a las necesidades del proyecto; y buscaba explicar por qué Amazon fue la mejor alternativa.

Configurar de forma óptima la máquina virtual, garantizará que los componentes de software se ejecuten de forma funcional, por lo que se espera no perder de vista los siguientes puntos de configuración.

Cabe mencionar también, que estos puntos de configuración de la máquina virtual, serán explicados desde la perspectiva de la EC2 de Amazon, sin embargo, los puntos son independientes de cualquier proveedor de servicios en la nube o de la implementación de un servidor físico.

### **5.3.1.1. Sistema Operativo**

El primer punto, es la creación de una instancia de máquina virtual. Una instancia, no es más que la creación de una computadora en la nube, con las características de hardware y software que tendría una computadora física.

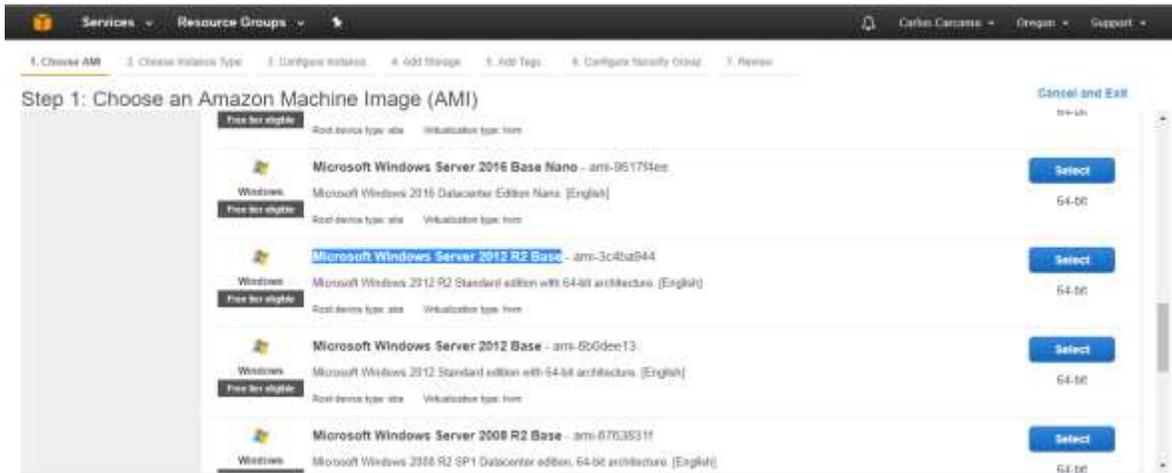
La instancia debe crearse a partir de los requerimientos de hardware descritos en el capítulo 2 del presente trabajo, los cuales, velan por el correcto funcionamiento de todos los componentes de software de la solución.

El punto de inflexión para la creación de la instancia, es la elección del sistema operativo. A pesar que todos los componentes de software que se han de implementar en la máquina virtual, el equipo recomienda siempre seleccionar sistemas operativos de Microsoft.

Aplicaciones a instalar en el servidor como GlassFish o MySQL tienen bastante documentación en internet respecto a su uso e implementación; sin embargo, aplicaciones como Earthworm, basan la mayoría de su documentación a pruebas y datos de implementación propios de Windows; por lo que la implementación en otras plataformas puede ser un verdadero dolor de cabeza.

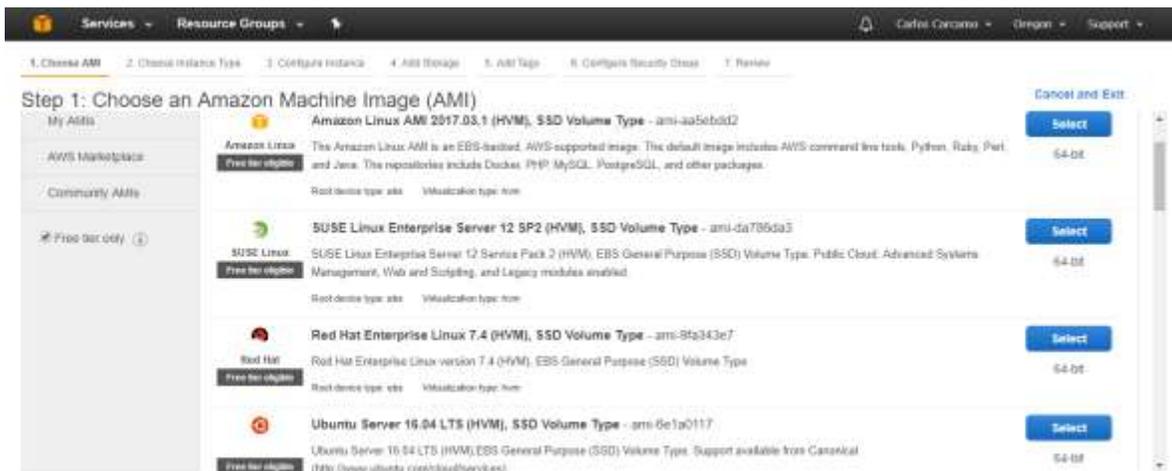
La figura 5.2, muestra un par de opciones de sistemas operativos de Microsoft que se pueden elegir al momento de crear una instancia en EC2. Sin embargo, también se puede elegir sistemas operativos libres como lo muestra la figura 5.3. Todo dependerá de las experiencias y preferencias del equipo implementador de la solución.

Figura 5.2 - Elección de sistema operativo Windows Server de instancia de máquina virtual en Amazon EC2



Nota importante: Si se implementa una instancia de máquina virtual de Amazon, se recomienda siempre elegir sistemas operativos que apliquen a la capa gratuita de soluciones EC2.

Figura 5.3 - Elección de sistema operativo Linux de instancia de máquina virtual en Amazon EC2

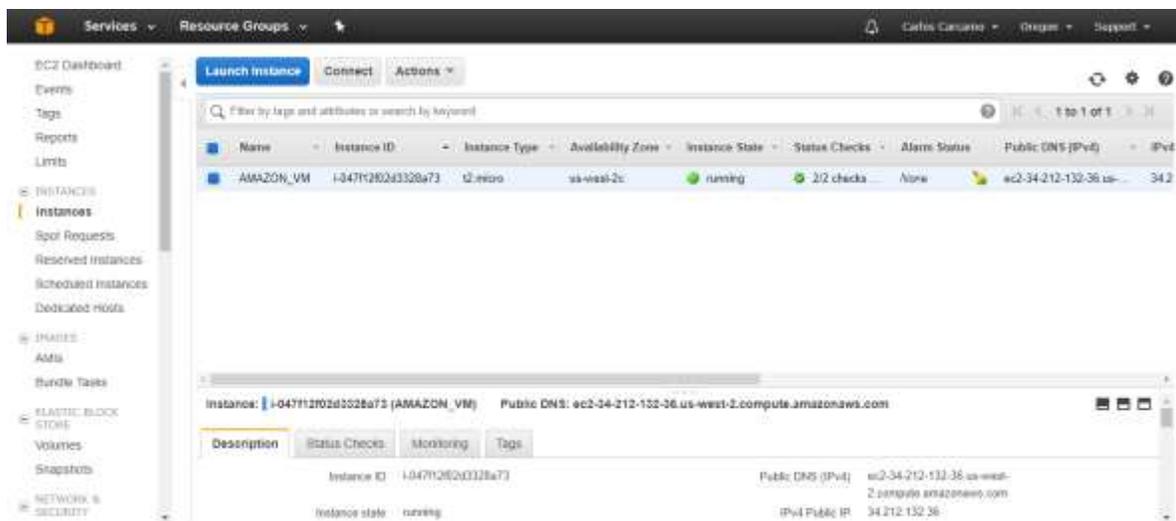


### 5.3.1.2. Puesta en Marcha y Monitoreo de Instancia de Máquina Virtual

Una vez creada la instancia, se debe tener en cuenta que la misma debe estar en constante monitoreo para podernos percatar de cualquier tipo de falla o error en la misma. EC2, ofrece un mantenimiento para tener siempre en la mira el estado de nuestra instancia. Si la instancia llega a fallar, todos los componentes que se alberguen en ella carecerán de funcionalidad.

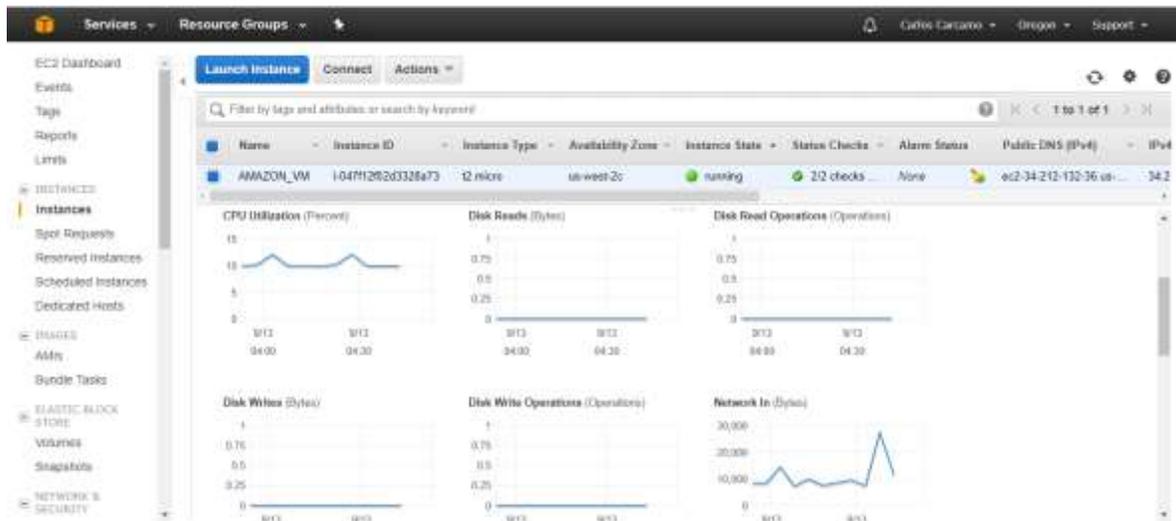
La figura 5.4, nos muestra la instancia creada de la máquina virtual AMAZON\_VM, que funge las acciones de servidor eje en el funcionamiento de nuestra aplicación (ver figura 3.1).

Figura 5.4 - Monitoreo de instancia de máquina virtual



El campo con el estado de la instancia (“Instance Status”) nos dará el panorama de cómo está nuestra máquina virtual. Lo normal es que el indicador esté siempre en verde, con su estado siempre en ejecución. De ésta forma garantizaremos siempre que la instancia está funcional.

Figura 5.5 - Monitoreo de recursos de instancia de maquina virtual



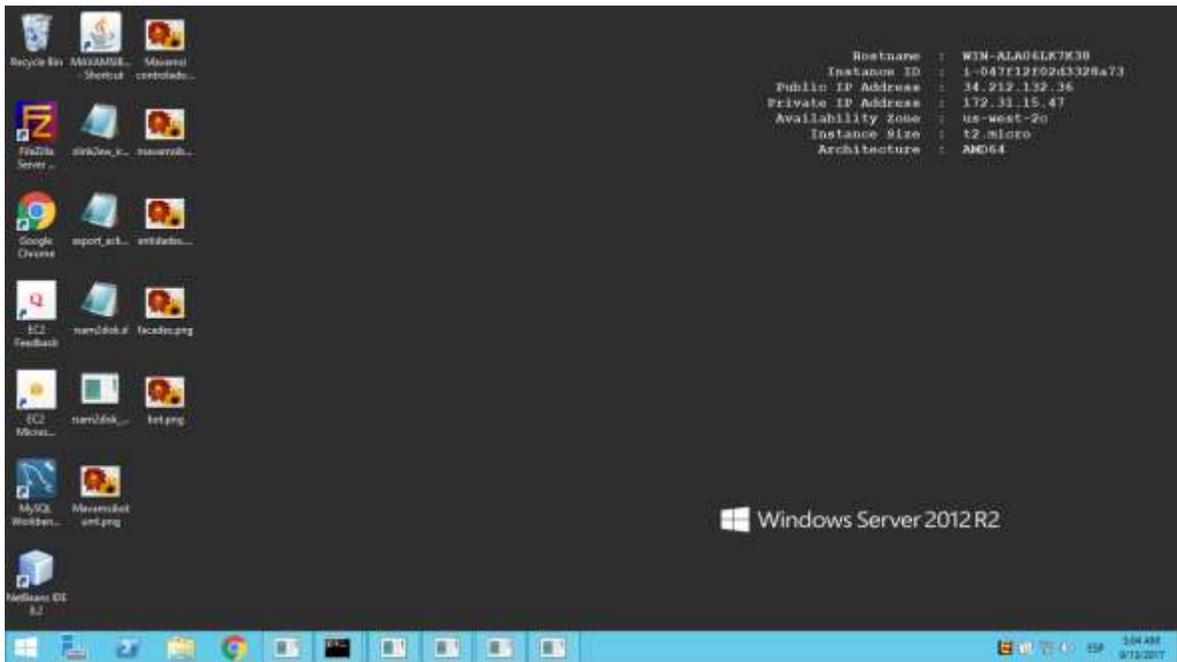
Una vez se tiene la máquina virtual implementada, podemos acceder a ella, por medio de Escritorio Remoto. La figura 5.6 muestra un ejemplo de acceso.

“Con Conexión a Escritorio remoto, puedes conectarse a un equipo que ejecute Windows desde otro equipo que ejecute Windows que esté conectado a la misma red o a Internet. Por ejemplo, puedes usar todos los programas, archivos y recursos de red del equipo del trabajo desde el equipo de tu casa, como si estuvieras sentado delante del equipo del trabajo.

Para conectarte a un equipo remoto, este debe estar encendido y tener una conexión de red, Escritorio remoto debe estar habilitado y debes tener acceso de red al equipo remoto (que puede ser a través de Internet) y permiso para conectarte. Para tener permiso para conectarte, debes estar en la lista de usuarios. Antes de iniciar una conexión, es recomendable que busques el nombre del equipo al que te vas a conectar y que te asegures de que se permiten conexiones de Escritorio remoto a través de su firewall.” (Microsoft, 2016)

Sobre esta máquina virtual, se instalarán las aplicaciones, según corresponden en el cuadro 5.1.

Figura 5.6 - Vista de máquina virtual AMAZON\_VM



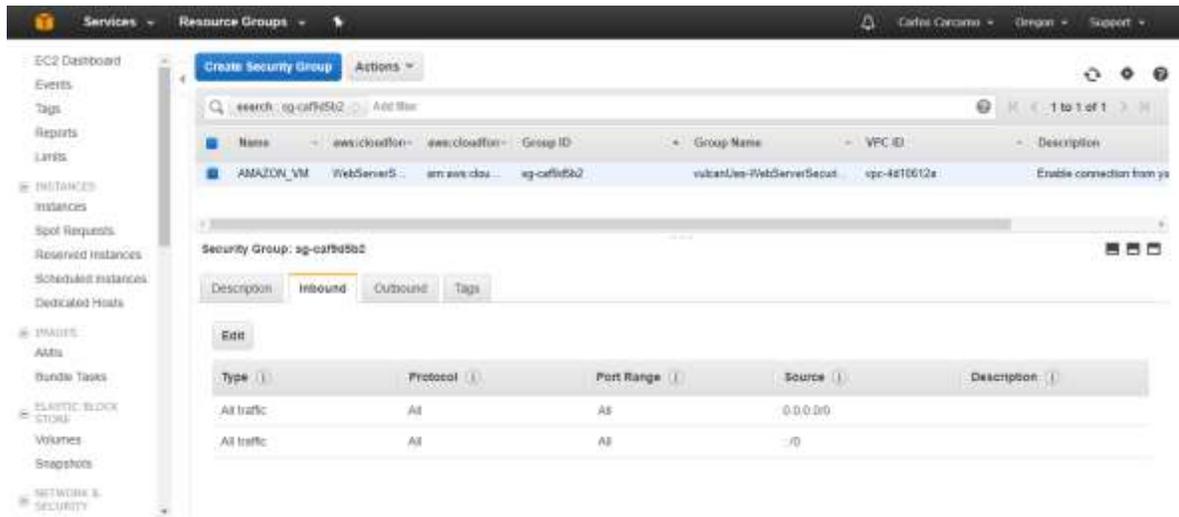
### 5.3.1.3. Configuración de red

Finalmente, será necesario configurar las condiciones de red de la máquina virtual para permitir que cualquier dispositivo desde cualquier IP, puede realizar peticiones sobre la máquina.

Aunque esto supone un riesgo informático, es una necesidad que la máquina virtual esté accesible desde cualquier IP, puesto que se comunicará por medio de internet con otros servidores (MAQUINA\_ESTACION, SNET y GMAIL), los cuales, pueden cambiar de dirección en cualquier momento y sin previo aviso.

Si se quiere evitar este riesgo, se pueden agregar las direcciones IP desde las cuales se permitirá el acceso a la máquina virtual. La figura 5.7 muestra las opciones de configuración de red.

Figura 5.7 - Configuraciones de red de máquina virtual



La figura 5.7, muestra cómo la máquina permitiría entrada de peticiones desde cualquier IP y por cualquier puerto de comunicación, Sin embargo, por medio de esta vista se puede delimitar las direcciones a usar.

### 5.3.2. Configuración de Máquina en Estación de Observación

El siguiente servidor a configurar será la máquina de la estación de observaciones, la cual, será la responsable de capturar los datos arrojados por el escáner ScanDoas, transformarlos a formato de texto plano por medio de Novac Program, y finalmente enviarlos vía FTP a la máquina de la solución por medio de MAVAMSI\_BOT2. En esta máquina se implementará Novac Program.

La configuración de la máquina es simple. Únicamente, necesita cumplir con dos requerimientos no funcionales que garantizarán el correcto funcionamiento:

1. Estar encendida durante el tiempo que se lleven a cabo las mediciones de emisiones de gases.

2. Tener acceso ininterrumpido a internet, para la transferencia de archivos.

La implementación de Novac Program y de MAVAMSI\_BOT2 se explicará a detalle en posteriores apartados de éste capítulo.

El sistema operativo que debe poseer ésta máquina, debe ser cualquier versión de Microsoft Windows XP o superior. Ésta condición es delimitada por la implementación de Novac Program, ya que, la aplicación de la que se dispone, es un ejecutable de Windows (.exe).

El sistema operativo, además, debe ser capaz de soportar la instalación de JVM, para la ejecución de MAVAMSI\_BOT2.

El equipo de trabajo, recomienda utilizar Windows 7 o superior, e instalar la versión de JRE 7 o superior, para obtener óptimos resultados con la ejecución de Novac Program y MAVAMSI\_BOT2.

El único requerimiento de hardware que debe cumplir esta máquina es que posea una entrada de cable de red (Ethernet); ya que es por medio de esta interfaz que la máquina se conecta con ScanDoas.

## 5.4. CONFIGURACIÓN DE SERVIDORES DE SOFTWARE

Las siguientes aplicaciones, serán necesarias para la coacción entre las aplicaciones de terceros y las aplicaciones in house de la solución. Se trata de servidores de aplicaciones para fines específicos y su correcta implementación serpa crucial para el proyecto.

Toda la implementación y configuración de estas herramientas, será del lado del servidor AMAZON\_VM.

### 5.4.1. *FileZilla*

“FileZilla es un cliente FTP multiplataforma de código abierto y software libre, licenciado bajo la Licencia Pública General de GNU. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS).

FileZilla permite a un usuario crear una lista de sitios FTP o SFTP (estableciendo una conexión cifrada que utiliza el protocolo SSH) con sus datos de conexión, como el número de puerto a usar, o si se utiliza inicio de sesión normal o anónima. Para el inicio normal, se guarda el usuario y, opcionalmente, la contraseña.” (FileZilla, 2017)

FileZilla ayudará a la configuración de la máquina virtual, para que tome el rol de un servidor FTP. De ésta forma, permitirá a MAVAMSI\_BOT2 conectarse a la máquina virtual, y poder transferir los archivos de emisiones de gases por este protocolo.

FileZilla es una aplicación fácil de instalar y de mantener, que no ocupa mucho espacio en memoria y casi no genera tiempo de procesamiento al CPU de la máquina virtual, por lo que se convierte es una perfecta opción como servidor FTP de la solución.

La implementación necesaria de FileZilla es bastante simple, tal y como se describirá a continuación.

#### **5.4.1.1. Instalación de FileZilla Server**

La instalación de FileZilla, es tan simple como descargar el software de la página principal del proveedor, y utilizar el wizard de instalación de la herramienta.

Como se recomendó en puntos anteriores la utilización de Windows Server, se debe descargar la versión correcta para el sistema operativo donde se instale el servidor FTP.

Se recomienda descargar la aplicación del sitio web oficial del proveedor de software: <https://filezilla-project.org/index.php> (sitio validado al 13 de septiembre de 2017). Al momento de descargar, se debe descargar la versión Servidor de la aplicación, que es la que ayudará a configurar la máquina virtual como servidor.

Una vez instalada la aplicación, se procederá a realizar la configuración necesaria.

Para mayor información sobre la instalación y configuración de la aplicación, favor remitirse a los siguientes enlaces de interés:

- Instalación de aplicación: [https://wiki.filezilla-project.org/Client\\_Installation](https://wiki.filezilla-project.org/Client_Installation) (sitio validado al 13 de septiembre de 2017)
- Configuración de red de servidor: [https://wiki.filezilla-project.org/Network\\_Configuration](https://wiki.filezilla-project.org/Network_Configuration) (sitio validado al 13 de septiembre de 2017)

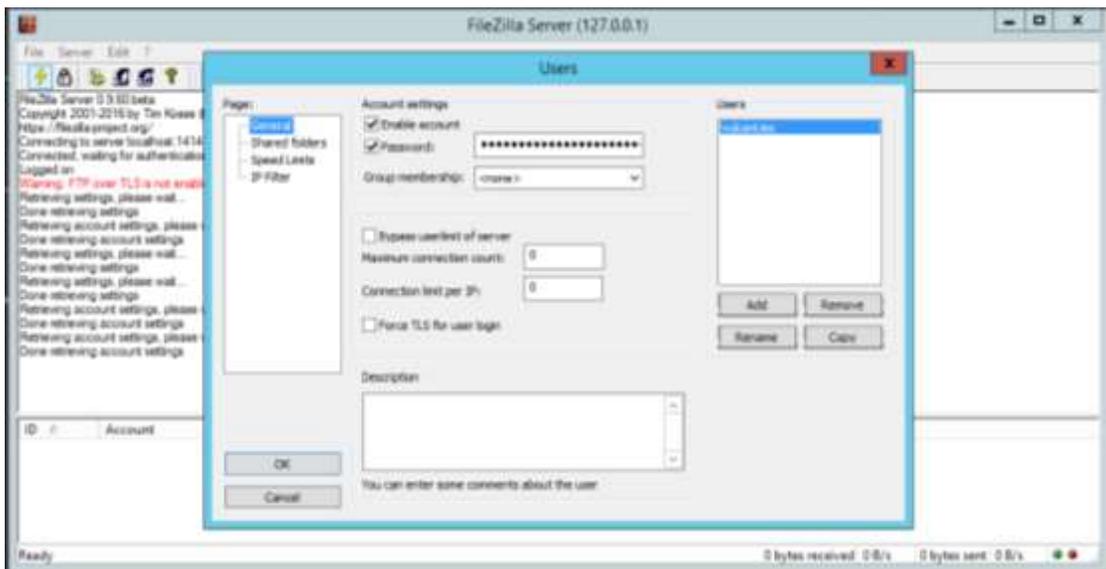
#### **5.4.1.2. Configuración de usuario FTP**

Para que MAVAMSI\_BOT2 se pueda conectar al servidor, será necesaria la creación de un usuario FTP, con sus respectivas credenciales, para poder iniciar la comunicación por medio de éste protocolo.

FileZilla Server, permite la creación de múltiples usuarios, a los cuales, se les puede delegar ciertos privilegios sobre la estructura de carpetas configurada en el servidor. Para nuestro caso, solo necesitaremos un usuario por medio del cual podamos comunicarnos con la máquina virtual desde la máquina de la estación.

La creación y administración de usuarios, se realiza por medio de la opción “Users” de FileZilla server tal y como lo muestra la figura 5.8.

*Figura 5.8 - Administración de usuarios en FileZilla Server*



Se recomienda no modificar la contraseña o el nombre del usuario que se cree en éste apartado, pues es implicará cambios en todas las estaciones que utilicen ese usuario como medio de comunicación entre el servidor y las máquinas de la estación.

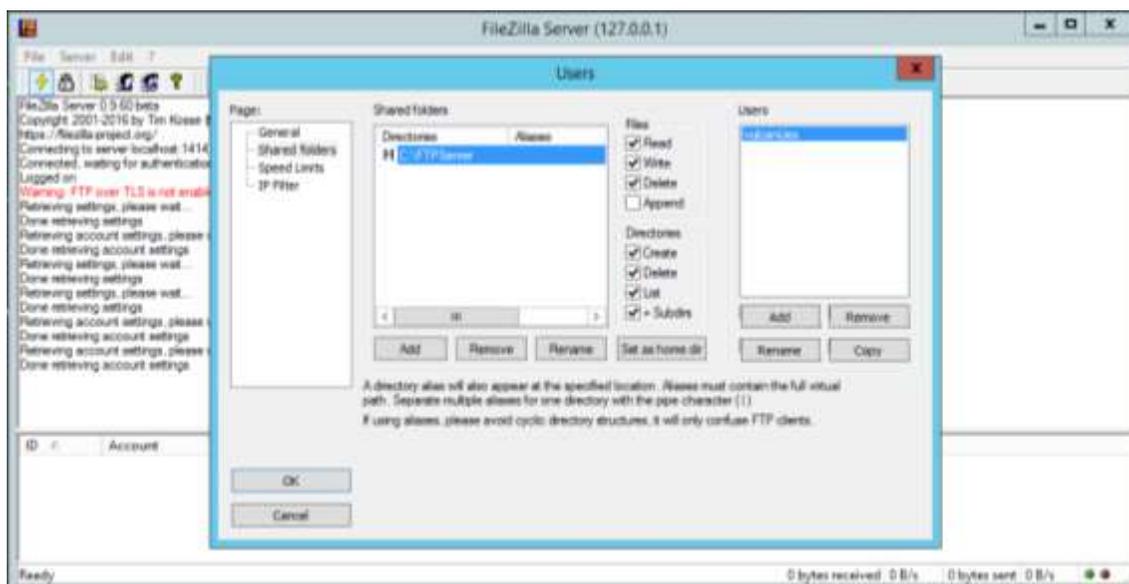
El equipo de trabajo recomienda también, crear un usuario FTP por cada máquina que se conecte con la máquina virtual vía FTP. De ésta forma se tendrá más control sobre la transacción de archivos que se envíen al servidor.

### 5.4.1.3. Configuración de estructura de carpetas en servidor

Una vez creado el usuario, será necesario otorgar accesos a la ruta de almacenamiento físico de la máquina virtual donde se almacenarán los archivos procedentes de la máquina de la estación.

La figura 5.9, muestra la ventana de configuración de éstos permisos, siempre en la opción “Users”, menú “Shared Folders” de FileZilla.

Figura 5.9 - Configuración de carpetas compartidas en FileZilla



El equipo recomienda colocar la carpeta central de acopio de archivos en la raíz del servidor, para que su ubicación no dependa de ningún usuario de sistema operativo y su ubicación sea más simple y rápida.

El equipo también recomienda la aplicación de todos los privilegios al usuario creado en el punto anterior sobre la carpeta seleccionada.

Nota: Cabe aclarar que en el servidor FTP no se realizará ninguna creación previa de estructura de carpetas para el almacenamiento de archivos de gases. LA creación de la estructura de carpetas la crea y la administra MAVAMSI\_BOT2.

## **5.4.2. GlassFish**

GlassFish será el servidor de aplicaciones web a utilizar en nuestro sistema. La función de GlassFish es de suma importancia en el modelo de nuestra solución, ya que será por medio de éste que se interpretará el código de la aplicación web MAVAMSI para poder ser ejecutado desde cualquier navegador web en cualquier momento.

### **5.4.2.1. Instalación de GlassFish**

La instalación de GlassFish difiere un poco con la instalación y configuración de otras aplicaciones. Su funcionalidad está ligada con NetBeans, ya que está implícita al momento de compilar la aplicación web desde el IDE donde fue desarrollado.

El equipo recomienda leer la guía de instalación de GlassFish creada por el proveedor de software (Java) donde se detallan los pasos a seguir.

La guía de instalación se adjunta en el disco compartido entregado junto con el presente trabajo de grado, sin embargo, a continuación se detalla su contenido en internet:

- Descarga de GlassFish de sitio oficial de proveedor de software: <https://javaee.github.io/glassfish/download> (sitio validado al 13 de septiembre de 2017).
- Guía oficial de Instalación de GlassFish: <https://javaee.github.io/glassfish/doc/4.0/installation-guide.pdf> (sitio validado al 13 de septiembre de 2017).
- Guía recomendada de instalación de GlassFish: <https://www.adictosaltrabajo.com/tutoriales/instalacion-configuracion-glassfish/> (sitio validado al 13 de septiembre de 2017).

Figura 5.10 - GlassFish ejecutándose en máquina virtual



La figura 5.10, nos muestra la instancia de GlassFish corriendo en el servidor AMAZON\_VM. Sobre esta instancia, configuraremos la compilación de MAVAMSI; entraremos en más detalles de esta implementación un par de apartados adelante.

### 5.4.3. MySQL

Antes de instalar los componentes de software, es necesario e indispensable instalar el servidor de base de datos que albergará el modelo entidad relación de los datos de la aplicación. En el capítulo 3, se habló de las ventajas que presenta la implementación de MySQL como sistema gestor de base de datos en relación a la comparativa realizada en la tabla 3.6.

El modelo de la base de datos, será entregado junto con el presente trabajo en un disco compacto. El modelo de la base, será entregado en formato de archivo de Workbench (.mwb), por lo que será necesaria la instalación de este software para la importación de toda la base que utilizará la herramienta. A continuación detallamos los aspectos de instalación y configuración necesarios para utilizar la base de datos.

### **5.4.3.1. Instalación de MySQL Server**

El primer punto será instalar MySQL Server en la máquina virtual de la solución. Con este punto, se garantiza que la máquina funcionará como un servidor de base de datos, que le permitirá compartir esta información con los usuarios por medio de la aplicación web MAVAMSI.

La instalación de MySQL Server, es bastante sencilla, basta con descargar la versión más reciente del sitio web oficial del proveedor y posteriormente instalarla en la máquina a utilizar de servidor.

El equipo recomienda descargar la versión de MySQL Community Server más reciente. Esta versión es totalmente gratuita y de código libre, y al ser una versión desarrollada por la comunidad de MySQL alrededor del mundo, contiene mucha documentación útil de toda su configuración, publicada en internet.

La instalación del gestor, es bastante simple. Al momento de descargar de internet, veremos un wizard de instalación, que ayudará a implementar todos los elementos de forma fácil y rápida. Los enlaces de interés para este punto son los siguientes:

- Descarga de MySQL Community Server de sitio oficial de proveedor de software: <https://dev.mysql.com/downloads/mysql/> (sitio validado al 13 de septiembre de 2017).
- Guía de instalación de MySQL Community Server de sitio oficial de proveedor de software: <https://dev.mysql.com/doc/refman/5.7/en/installing.html> (sitio validado al 13 de septiembre de 2017).

### **5.4.3.2. Instalación de MySQL Workbench**

Como complemento del gestor de base de datos, MySQL ofrece una herramienta para el manejo de modelos de datos, bases relacionales y monitoreo de instancias de base de datos publicadas, se trata de MySQL Workbench.

MySQL server, instala todos los componentes de software necesarios para la administración y mantenimiento de bases de datos, sin embargo, no instala ninguna herramienta para su gestión y monitoreo con entorno gráfico, por lo que Workbench se convierte en una utilidad magnífica que complementa la labor del sistema gestor.

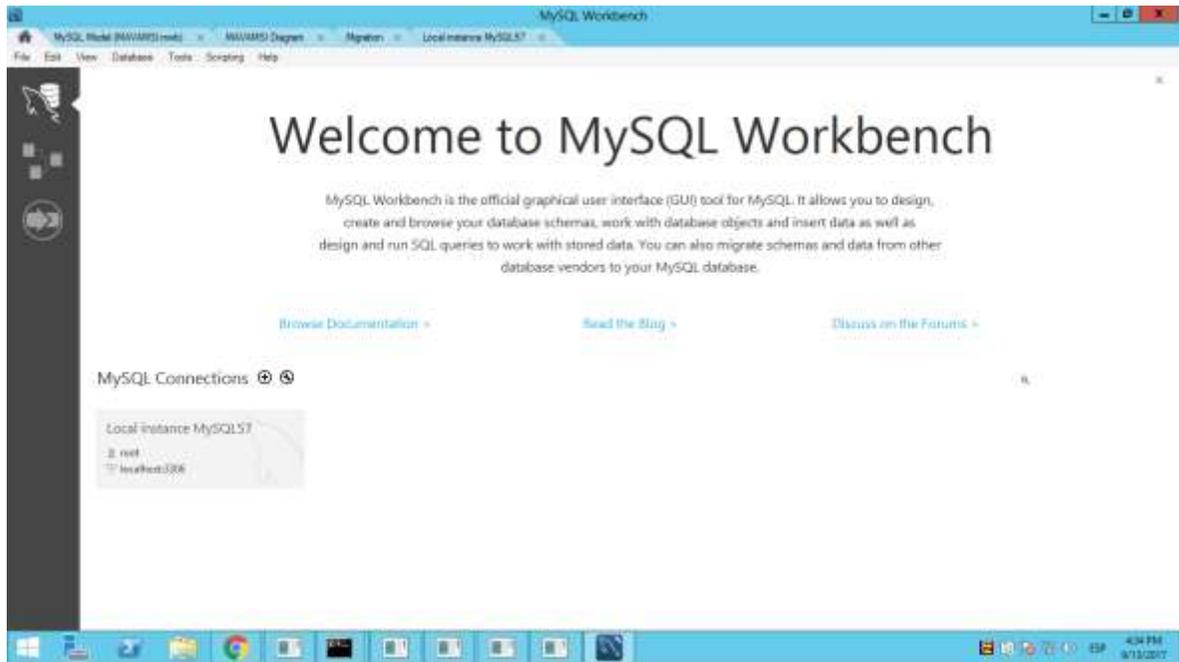
Workbench nos permitirá la implementación del modelo entidad relación de nuestra base de datos, el monitoreo de la instancia de la base, y nos ayudará a realizar cualquier cambios sobre alguna tabla u objeto de la base de forma simple, rápida y segura.

La instalación de Workbench, es bastante simple. Al momento de descargar de internet, veremos un wizard de instalación, que ayudará a implementar todos los elementos de forma fácil y rápida. Los enlaces de interés para este punto son los siguientes:

- Descarga de MySQL Workbench de sitio oficial de proveedor de software: <https://www.mysql.com/products/workbench/> (sitio validado al 13 de septiembre de 2017).
- Guía de instalación de MySQL Community Server de sitio oficial de proveedor de software: <https://dev.mysql.com/doc/refman/5.7/en/installing.html> (sitio validado al 13 de septiembre de 2017).

Una vez instalado Workbench en la máquina virtual, podemos proceder a crear la base de la aplicación.

Figura 5.11 - MySQL Workbench instalado en AMAZON\_VM



### 5.4.3.3. Creación de Instancia de Base de Datos

Una vez instaladas las herramientas de software necesarias para el funcionamiento de la base de datos de la aplicación, es momento de crear una instancia de conexión a base de datos de MySQL.

“Una instancia de base de datos es una copia del motor de la base que se ejecuta como un servicio de sistema operativo. Cada instancia administra varias bases de datos del sistema y una o varias bases de datos de usuario. Cada equipo puede ejecutar varias instancias de Motor de base de datos. Las aplicaciones se conectan a la instancia para realizar el trabajo en una base de datos administrada por la instancia.

Una instancia de base de datos funciona como un servicio que controla todas las solicitudes de aplicación para trabajar con datos de cualquiera de las bases de datos administradas por dicha instancia. Es el destino de las solicitudes de conexión

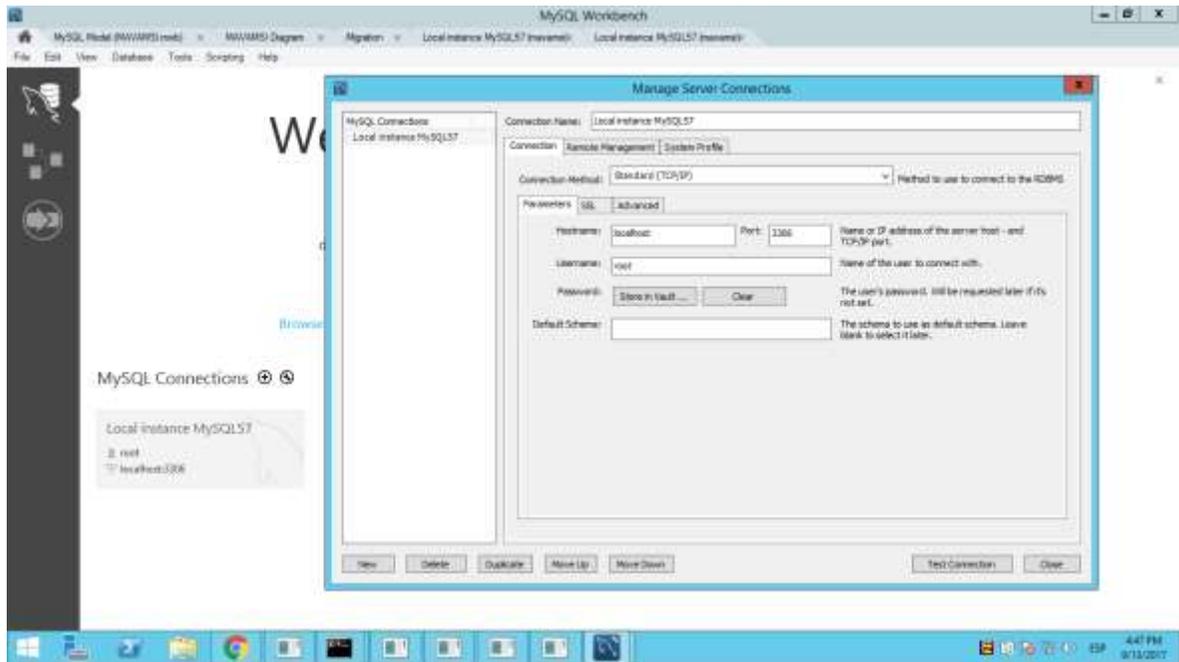
(inicios de sesión) de aplicaciones. La conexión se ejecuta en una conexión de red si la aplicación y la instancia están en equipos independientes. Si la aplicación y la instancia están en el mismo equipo, la conexión de MySQL se puede ejecutar como una conexión de red o una conexión en memoria. Cuando una conexión se ha completado, una aplicación envía instrucciones SQL a través de la conexión hasta la instancia.

Puede ejecutar múltiples instancias de Motor de base de datos en un equipo. Una instancia puede ser la instancia predeterminada. La instancia predeterminada no tiene nombre. Si una solicitud de conexión especifica solo el nombre del equipo, se establece la conexión a la instancia predeterminada. Una instancia con nombre es una instancia en la que se especifica un nombre de instancia al instalar la instancia. Una solicitud de conexión debe especificar el nombre del equipo y el nombre de instancia para conectar a la instancia. No hay ningún requisito para instalar una instancia predeterminada; todas las instancias que se ejecutan en un equipo pueden ser instancias con nombre.” (MySQL, 2017)

Para crear la instancia de conexión a la base, es tan simple como crear una nueva conexión desde MySQL Workbench, y llenar la información de la conexión tal cual y lo muestra la figura 5.12.

Una vez creada la instancia de conexión a la base, ya podemos implementar nuestro modelo de base de datos en el servidor.

Figura 5.12 - Creación de instancia de conexión a la base de datos



#### 5.4.3.4. Implementación de Base de Datos

Adjunto a la entrega del presente trabajo de grado, se entregará un disco compacto que contendrá el modelo entidad relación de la base de datos de la aplicación.

En el capítulo 3 se abordó todo el modelo de la base de datos que requiere ser utilizada para el despliegue del sistema. La figura 3.9 muestra en totalidad el modelo entidad relación que se diseñó para la base de datos; y los cuadros del 3.1 al 3.16 muestran el detalle de cada tabla que compone la base. Todos estos objetos de base de datos, se encuentran unificados en el archivo MAVAMSI.mwb, el cual, es un archivo de modelo de datos de Workbench, que será implementado en la instancia de conexión creada en el punto anterior, por medio de la funcionalidad de Ingeniería Inversa de Workbench.

Al abrir el archivo MAVAMSI.mwb desde Workbench, veremos el modelo entidad relación descrito en la figura 3.9; sin embargo, el esquema de esta base aún no forma parte de nuestra base de datos. Para Importar el esquema a la base de datos del servidor, se debe hacer uso de la metodología de Ingeniería Inversa de Workbench, que se encarga de crear los objetos de la base a partir del archivo .mwb.

“El esquema de una base de datos describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos. En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

El esquema es generalmente almacenado en un diccionario de datos. Aunque generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos.” (MySQL, 2017)

Para llevar a cabo la migración de la base, el equipo recomienda seguir el siguiente tutorial, desarrollado por el proveedor de software, MySQL: <https://dev.mysql.com/doc/workbench/en/wb-reverse-engineer-live.html> (sitio validado al 13 de septiembre de 2017).

Al momento de realizar la ingeniería inversa de la base, el equipo recomienda importar todos y cada uno de los elementos que el asistente indica, tal y como lo muestra la figura 5.13.

Una vez importado el esquema de la base de datos, ya podremos acceder a ella por medio de la instancia de conexión a la base creada en el apartado anterior, y ya podremos tener acceso a la misma, por medio de las aplicaciones MAVAMSI\_BOT1 y MAVAMSI.

Figura 5.13 - Ingeniería Inversa de Workbench para importación de esquema de base de datos MAVAMSI

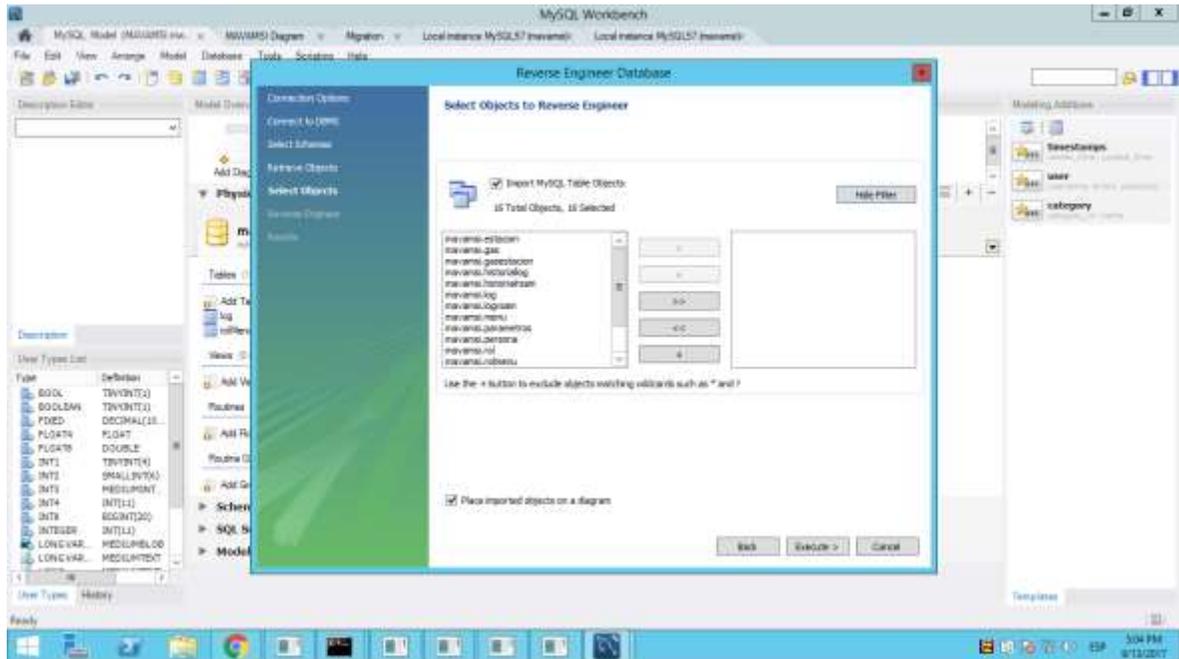
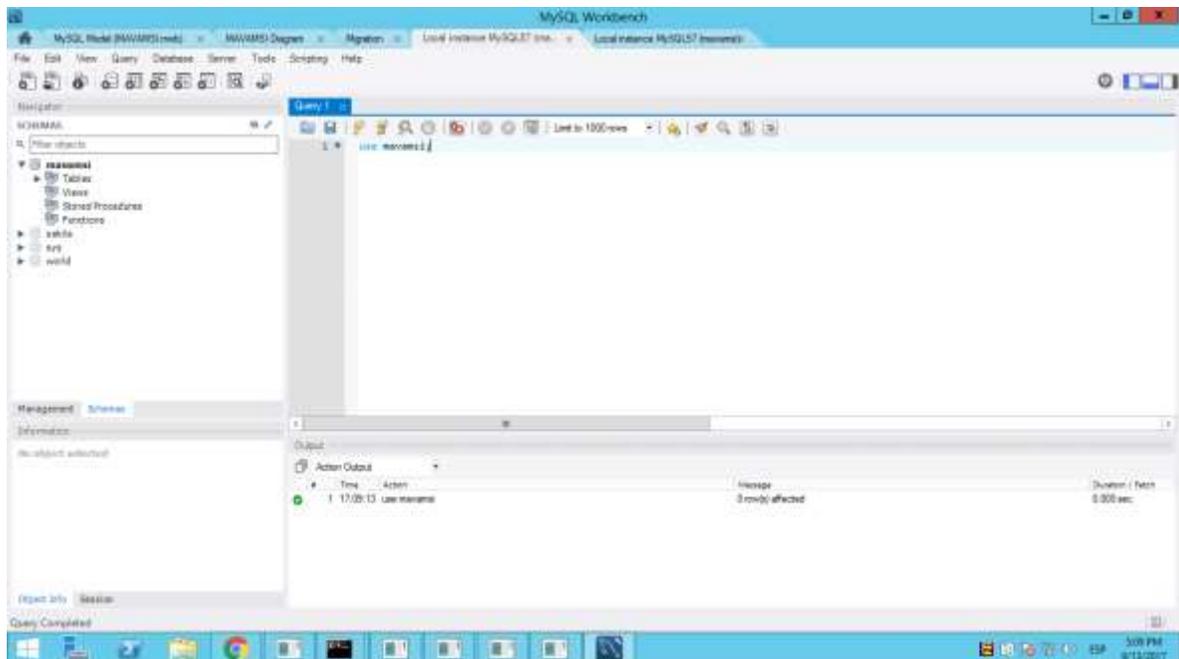


Figura 5.14 - Esquema MAVAMSI importado en base de datos



## **5.5. IMPLEMENTACIÓN DE SOFTWARE DE TERCEROS**

### ***5.5.1. Novac Program***

Sin duda alguna, una de las configuraciones más cruciales para el correcto funcionamiento del sistema, será la implementación de Novac Program en la máquina de la estación de observación.

La configuración de Novac Program es bastante delicada, ya que es un poco compleja y debe ser realizada con cuidado para no perder de vista ningún punto.

En este apartado daremos a conocer la configuración del hardware ScanDoas por medio de Novac Program, y la configuración correcta del mismo.

Cabe aclarar, que Novac Program es un programa ejecutable (.exe) por lo tanto no requiere de ninguna instalación previa para su funcionamiento. La última versión del ejecutable se compartirá por medio del disco compacto que contendrá el presente trabajo de grado.

#### ***5.5.1.1. Configuración de Red***

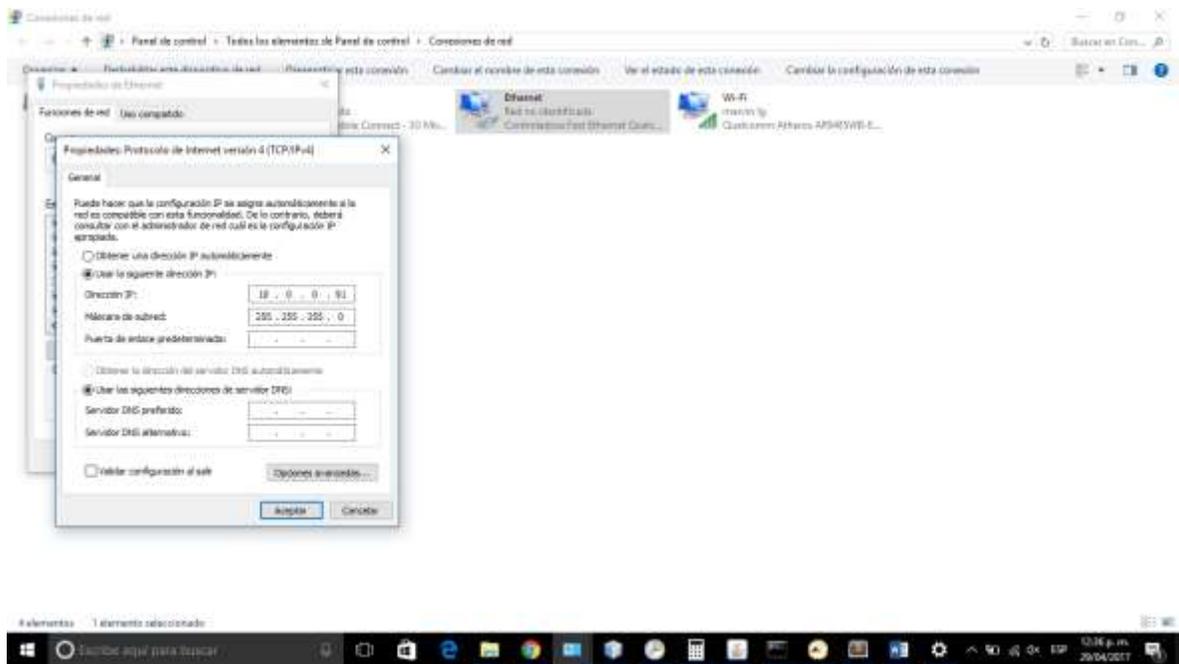
Antes de abrir la aplicación, es necesario realizar una configuración de red, para el adaptador de red Ethernet. La comunicación entre ScanDoas y la máquina, se realiza por medio del protocolo FTP, por medio de un cable de red que comunica los archivos entre ambos dispositivos; donde ScanDoas se convierte en el servidor FTP, y la máquina de la estación en el cliente FTP.

Tal y como se mencionó al inicio del trabajo de grado en el capítulo 1 en el apartado de Limitantes, el desarrollo del proyecto ha sido adaptado para equipo de medición de la serie S2000; para este tipo de equipos, por lo general, la IP de referencia de su puerto Ethernet es la 10.0.0.90; por lo que hay que configurar la

máquina de la estación de medición en alguna dirección del rango que comprenda a esa IP como su puesta de enlace.

El equipo recomienda configurar la máquina de la estación bajo la IP 10.0.0.91, para garantizar una dirección bajo el mismo rango, tal y como lo muestra la figura 5.15.

Figura 5.15 - Configuración de red para conexión de ScanDoas



Además de esto, el equipo recomienda aceptar las directivas de firewall de Windows con respecto a esta conexión, indicando que la conexión es confiable; ya que si esto no se realiza se corre con el riesgo de que la conexión se rechace en cualquier momento.

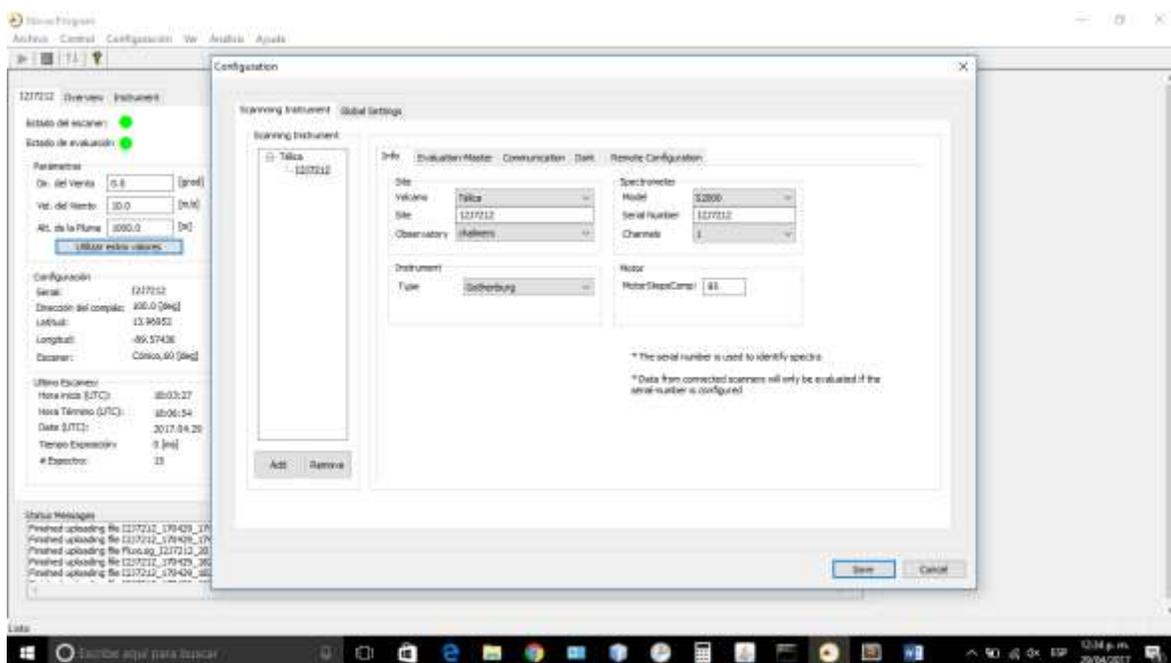
Una vez realizada la conexión, se puede probar la misma por medio de la consola de Windows, realizando ping a la dirección 10.0.0.90, donde debe ser exitosa, demostrando la configuración de red fue correcta.

### 5.5.1.2. Configuración de ScanDoas en Novac Program

Una vez configurada la conexión con el sensor a nivel físico, es momento de configurar el sensor a nivel aplicación desde Novac Program. Al abrir la aplicación por primera vez, podremos ver que no hay ningún dispositivo asociado, por lo que tocará agregarlo desde el menú de Configuración de la herramienta.

Lo primero que hay que hacer, es configurar el modelo del dispositivo y su número de serie, tal y como lo muestra la figura 5.16.

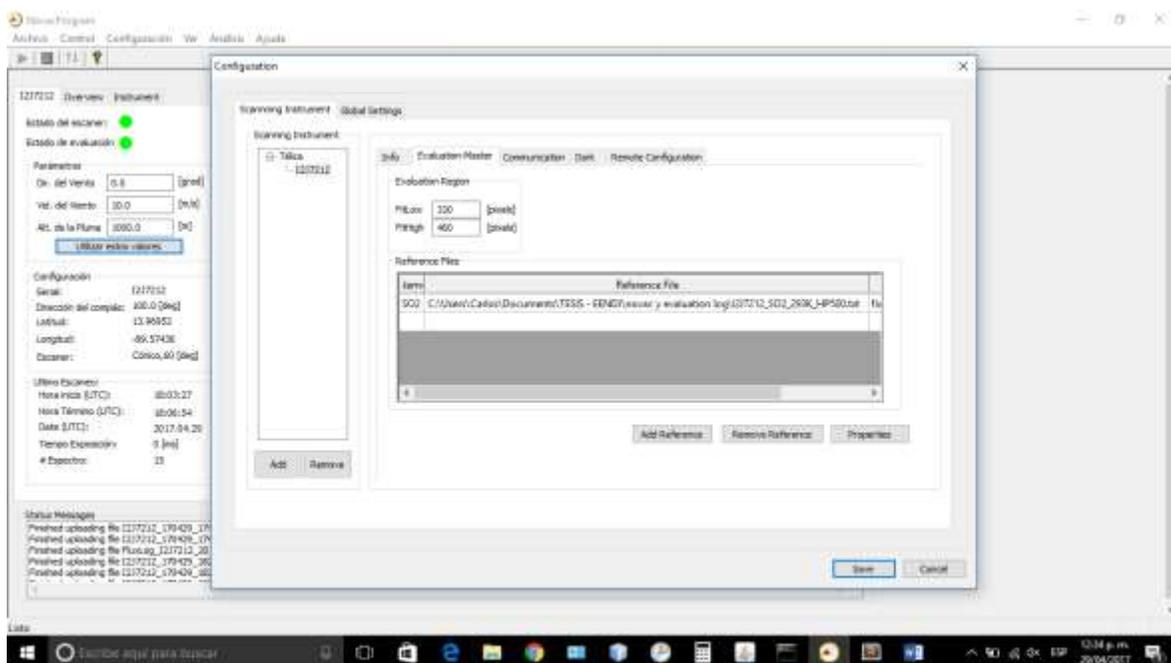
Figura 5.16 - Configuración de modelo de dispositivo ScanDoas en Novac Program



El modelo del dispositivo para el cual se trabajó el proyecto es el S2000. El número de serie del sensor, por lo general, está especificado en la parte superior del sensor. Éste número es importante, ya que cada sensor es diferentes y Novac Necesita adaptar su forma de lectura de datos en base a cada sensor.

El siguiente paso en la configuración, se detalla en la pestaña Evaluation Master de la vista anterior. En este apartado de detallan los archivos de referencia bajo los que se tomarán las mediciones, como lo muestra la figura 5.17.

Figura 5.17 - Configuración de evaluación maestra de ScanDoas en Novac Program

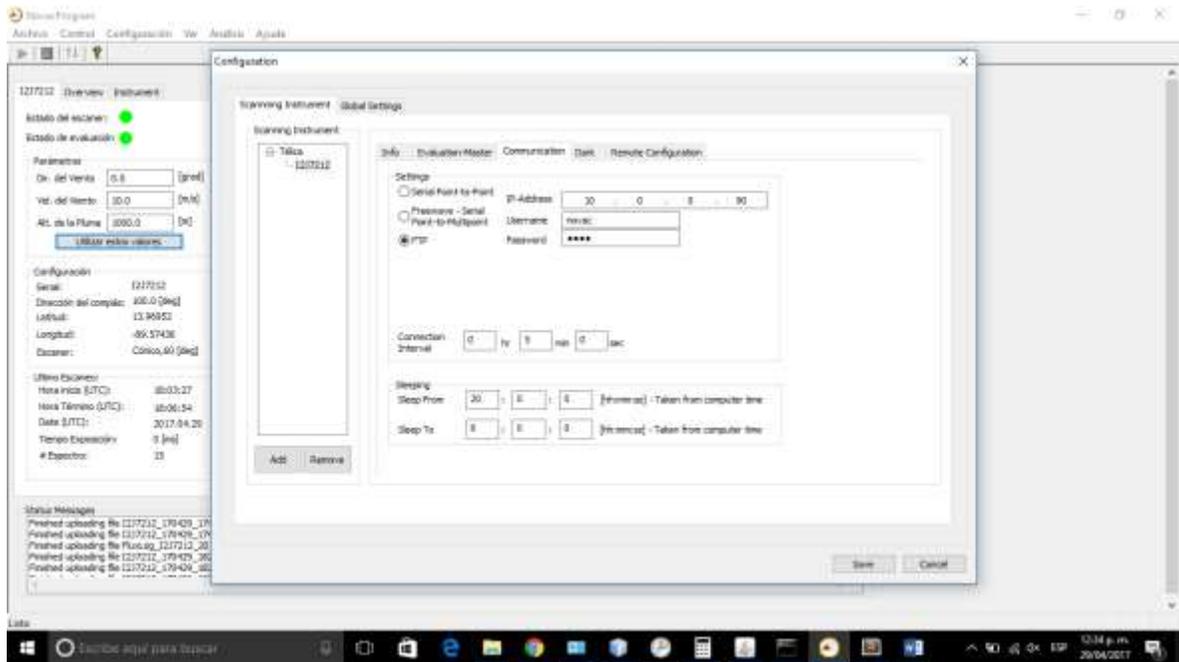


El archivo especificado en este apartado, es un archivo de texto plano que contiene los rangos de mediciones necesarios para calibrar las mediciones del sensor, y que además, le ayudan a Novac Program a interpretar las diferencias de luminosidad capturadas por el sensor.

El archivo de evaluación maestra para el sensor ScanDoas S2000 utilizado en la fase de desarrollo del proyecto, será adjuntado al disco compacto que contiene el presente trabajo de graduación.

Posteriormente, será necesario configurar la comunicación del sensor con Novac Program. Como lo hablamos en el apartado anterior, el sensor funciona como un servidor, y se conecta con la máquina de la solución vía FTP. La figura 5.18 muestra la configuración de este apartado.

Figura 5.18 - Configuración de comunicación entre ScanDoas y Novac Program

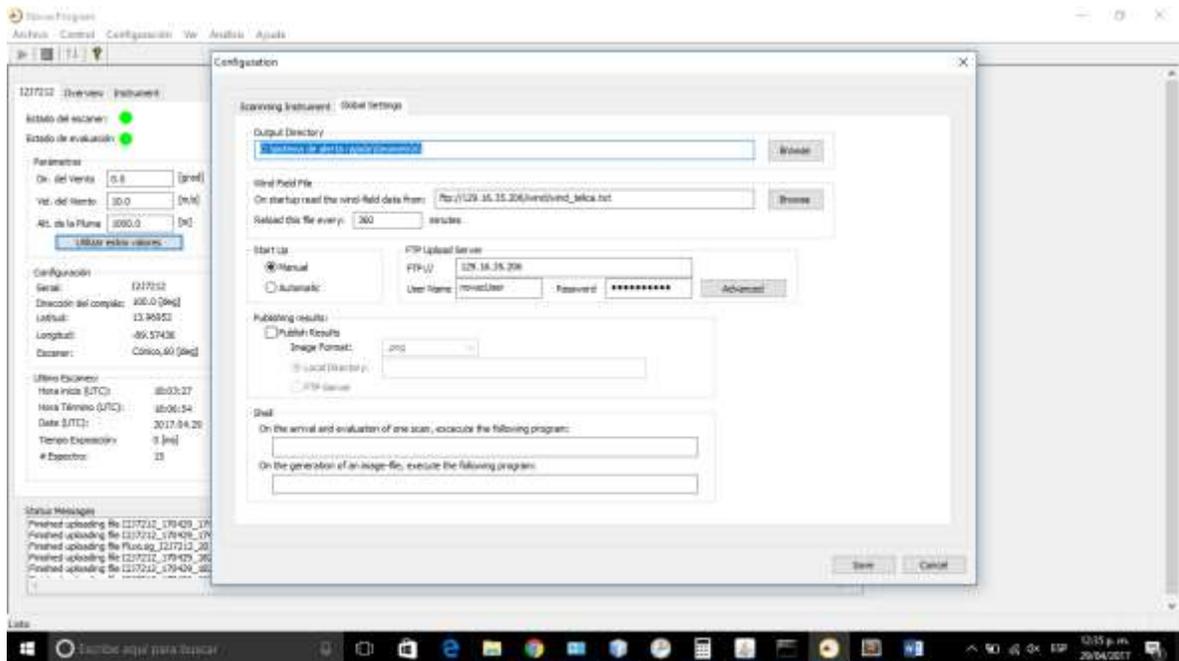


Las credenciales de acceso al servidor FTP, son compartidas por el fabricante del sensor. Para las pruebas creadas con el sensor S2000, se utilizaron las siguientes credenciales:

*Usuario:* novac  
*Contraseña:* 1225

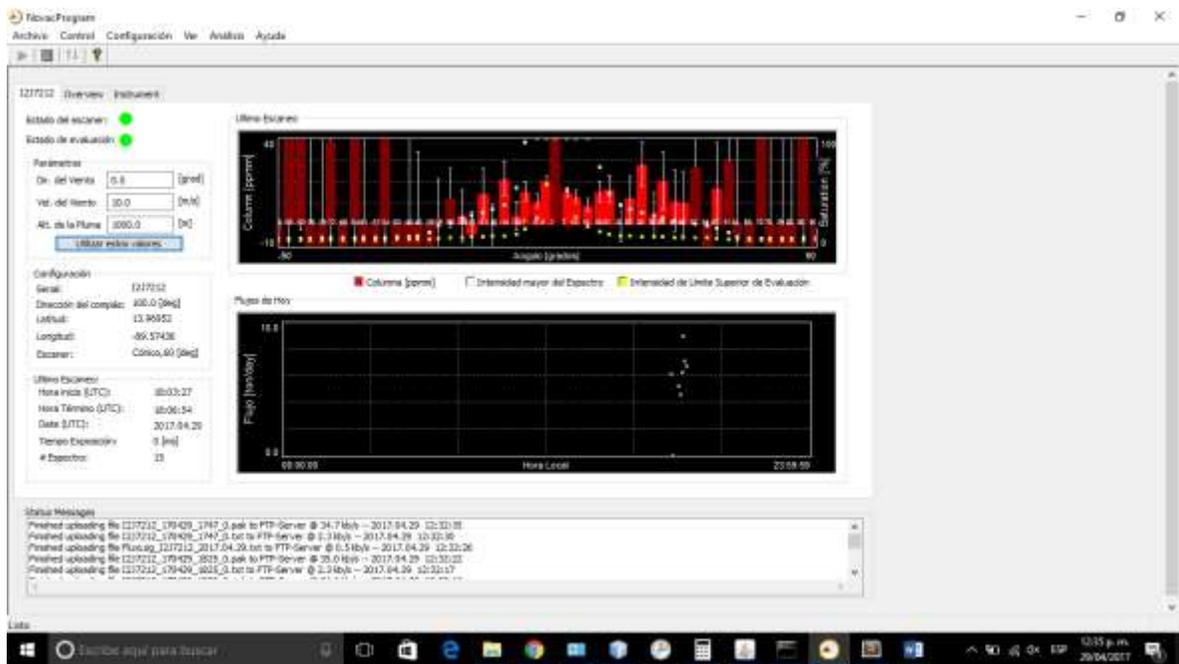
Finalmente, una parte muy importante, la designación de la carpeta de salida de los archivos de texto transformados por Novac Program, Esta configuración se realiza desde la pestaña Global Settings, como se muestra en la figura 5.19.

Figura 5.19 - Configuración de carpeta de salida de archivos de texto de mediciones de emisión de gases en Novac Program



Finalmente, tendremos correctamente configurado e implementado Novac Program y se podrá comunicar con el sensor para la obtención de medidas. Los indicadores de la parte superior izquierda indican, al estar en verde, su correcta configuración. La figura 5.20, muestra un ejemplo de conectividad correcta con ScanDoas, y grafica las primeras obtenciones de registros del sensor.

Figura 5.20 - Configuración finalizada de Novac Program



## 5.5.2. EarthWorm

La configuración de EarthWorm, es mucho más compleja que la implementación de cualquier otra herramienta de software a implementar en la solución; por lo tanto, se ha buscado realizar una guía más detallada con la implementación de cada módulo que forma parte de EarthWorm. Dicha guía se encuentra anexa en el disco compacto que contiene el presente trabajo de grado.

Dicho esto, en el presente apartado daremos a conocer las macrotareas de implementación de EarthWorm en la máquina virtual de la solución.

### 5.5.2.1. Actividades previas y de preparación

- Conocimiento de funcionamiento de EarthWorm, lógica de funcionamiento modular, configuración de estaciones volcánicas, creación de directorio de instalación para EarthWorm.
- Gestión administrativa con el Ministerios de Medio Ambiente y Recursos Naturales para acceso a datos sísmicos.

- Aprendizaje de procesamiento de datos sísmicos.
- Estudio y comprensión de señales RSAM
- Aprendizaje de conversión de datos sísmicos a señales RSAM.
- Aprendizaje acerca de importación de datos sísmicos a servidor web.
- Configuración de dirección IP y puerto de red para exportar datos sísmicos por parte del MARN.
- Configuración del módulo de exportación de datos sísmicos.
- Configuración de dirección IP del servidor para la importación de datos.

#### ***5.5.2.2. Actividades de Implementación y Configuración***

- Configuración y creación de variables globales de EarthWorm en el servidor.
- Creación de diagrama de módulos a utilizar.
- Configuración de espacios de memorias para almacenamiento temporal de datos sísmicos.
- Configuración de módulo de importación de datos.
- Configuración de módulo de conversión de frecuencia de datos sísmicos.
- Configuración de módulo de registro de datos es espacio de memoria.
- Configuración de módulo de conversión de datos sísmicos a medición RSAM.
- Configuración de módulo de registro de datos RSAM en servidor web por medio de archivo de texto.

#### ***5.5.2.3. Actividades de Pruebas de Implementación***

- Prueba de acceso a datos sísmicos
- Prueba de registro de datos sísmicos en espacios de memoria temporales.
- Prueba de acceso y procesamiento de datos sísmicos en espacios de memoria temporales.
- Prueba de procesamiento y registro de datos RSAM.
- Creación automática de archivos “txt” en el servidor con datos RSAM.

#### ***5.5.2.4. Retos Surgidos durante Pruebas e Implementación de EarthWorm***

- La configuración de EarthWorm es modular y por lo tanto se debe de investigar cada módulo con el que se cuenta para entender cuáles son los mejores para lograr los objetivos logrados, por lo que se probaron varios módulos antes de definir los que se usarían finalmente.
- Inicialmente no se contaban con la aprobación de parte del MARN para el acceso a los datos sísmicos ya que tomo mucho tiempo, por lo que se inició a consumir datos sísmicos de una estación sísmica pública.
- Existió el problema al momento de importar los datos sísmicos brindados por el MARN con el módulo que se había definido inicialmente, por lo que se investigó otro módulo para poder hacer la importación y se realizó el cambio de parte del MARN y la configuración necesaria de parte del equipo de desarrollo.
- Los datos sísmicos no se guardaban en un archivo de texto en el servidor para su procesamiento, por lo que se utilizó un módulo personalizado para esta función.
- Durante la fase de prueba se interrumpió la recepción de datos sísmicos debido a que la configuración realizada por el MARN tuvo un error momentáneo.

#### ***5.5.2.5. Ajustes a configuración inicial***

- Se utilizó un módulo personalizado para poder guardar los datos sísmicos en el servidor en un archivo tipo “txt” que se almacena en una estructura de carpetas definida para que los robots desarrollados puedan consumir los datos.
- Se configuro solo una estación sísmica ya que el MARN es la única en la cual tiene sistema de monitoreo en la zona occidental, a pesar que se pueden configurar más estaciones.

#### ***5.5.2.6. Limitantes de Implementación de EarthWorm***

- El tiempo que tomo entre la solicitud de acceso de datos sísmicos a la aprobación llevo más tiempo del esperado.
- Se depende exclusivamente de los datos sísmicos proporcionados por el MARN, por lo cual si es un futuro el MARN cambia su configuración en la exportación de datos se perderá el acceso a los datos sísmicos.
- La documentación de los módulos de EarthWorm es muy básica e incluso en algunos casos nulos, debido a que es un sistema que no cuenta con una documentación oficial respaldada por un equipo de desarrollo constante.
- Es necesario que exista equipo hardware en las estaciones sísmicas de donde se desea capturar los datos para su procesamiento.

## 5.6. IMPLEMENTACIÓN DE SOFTWARE IN HOUSE

### 5.6.1. MAVAMSI\_BOT2

La implementación de MAVAMSI\_BOT2 queda totalmente del lado de la máquina de la estación de observación volcánica.

Con la idea de realizar un sistema escalable, se trabajó en un diseño de software funcional que permitiera comunicar los datos de emisiones de gases desde cualquier estación de observación, y desde cualquier volcán. MAVAMSI\_BOT2 se encarga de cumplir este requerimiento ya que para la implementación de múltiples estaciones, basta con ejecutar MAVAMSI\_BOT2 desde cada una de ellas.

A continuación, veremos los puntos a tomar en cuenta para implementar este robot.

La aplicación MAVAMSI\_BOT2 está formada por los siguientes archivos. Cabe mencionar que estos archivos serán compartidos vía disco compacto.

*Cuadro 5.2 - Componentes a implementar de MAVAMSI\_BOT2*

<b>Archivo</b>	<b>Descripción</b>
<b>MAVAMSI_BOT2.jar</b>	Robot para compartir archivos vía FTP a máquina virtual
<b>params.properties</b>	Archivo de parámetros de robot
<b>NovacTrigger.cmd</b>	Programador de ejecución de Novac Program
<b>FTPRobotScheduler.cmd</b>	Programador de ejecución de MAVAMSI_BOT2

#### 5.6.1.1. Implementación de MAVAMSI\_BOT2

La implementación del robot, consiste simplemente en modificar los valores del archivo de configuración “params.properties”; en el cual, se especificará la siguiente información:

Cuadro 5.3 - Parámetros de archivo de configuración de MAVAMSI\_BOT2

Parámetro	Descripción
<b>ftpServer</b>	Dirección IP del servidor FTP al que se enviarán los archivos (máquina virtual)
<b>ftpUser</b>	Usuario FTP
<b>ftpPass</b>	Contraseña de usuario FTP
<b>ftpPort</b>	Puerto de conexión FTP
<b>novacOutputPath</b>	Ruta de salida de archivos de texto de Novac Program. Debe ser la misma ruta configurada en la implementación del mismo
<b>volcan</b>	Nombre del volcán
<b>estacion</b>	Nombre de la estación de observación

Para la configuración del volcán y la estación de observación, estos deben estar previamente introducidos en la base de datos de la aplicación, por medio del módulo de administración de volcanes de la aplicación web MAVAMSI.

### 5.6.1.2. Configuración de NovacTrigger

Este proceso batch de consola de Windows, se encarga de abrir la aplicación Novac Program. La única labor de configuración que debe realizarse, es modificar el archivo NovacTrigger.cmd, y especificar la ruta de acceso a Novac Program, tal y como se muestra en la figura 5.21.

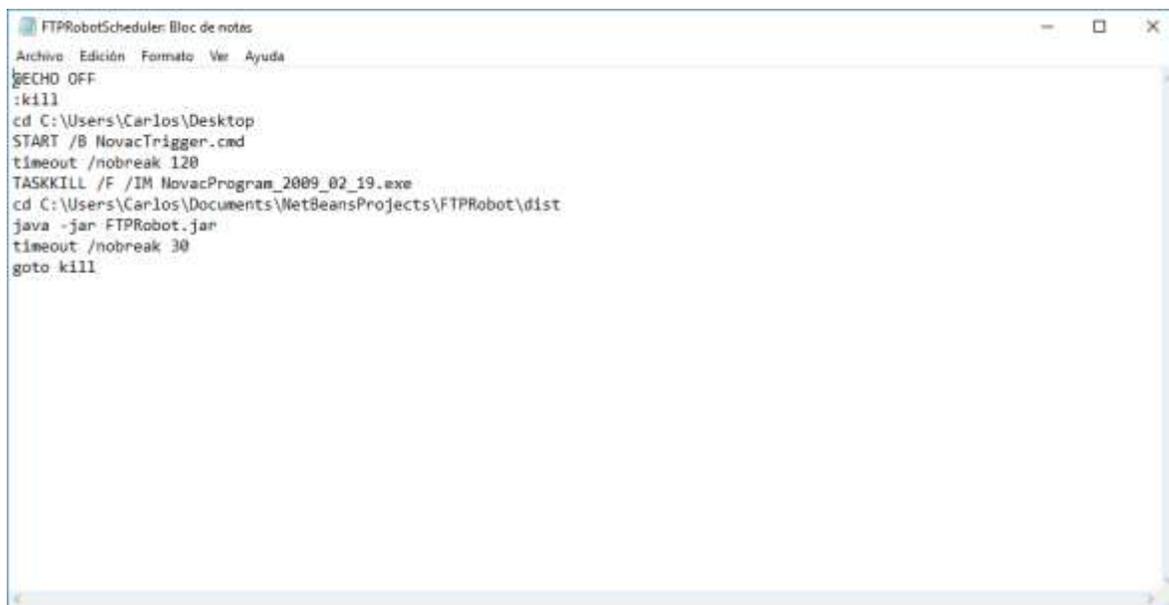
Figura 5.21 - Configuración de NovacTrigger



### 5.6.1.3. Configuración de FTPRobotScheduler

Finalmente, basta con configurar el proceso batch FTPRobotScheduler, que se encarga de ejecutar el robot MAVAMSI\_BOT2, el proceso NovacTrigger.cmd y modificar los rangos de tiempo en los que se quiere ejecutar cada aplicación. La figura 5.22, muestra el contenido de este proceso.

Figura 5.22 - Configuración de FTPRobotScheduler



```
FTPRobotScheduler: Bloc de notas
Archivo Edición Formato Ver Ayuda
@ECHO OFF
:kill
cd C:\Users\Carlos\Desktop
START /B NovacTrigger.cmd
timeout /nobreak 120
TASKKILL /F /IM NovacProgram_2009_02_19.exe
cd C:\Users\Carlos\Documents\NetBeansProjects\FTPRobot\dist
java -jar FTPRobot.jar
timeout /nobreak 30
goto kill
```

Cabe mencionar, que las unidades de tiempo de ejecución de cada proceso de software están en segundos.

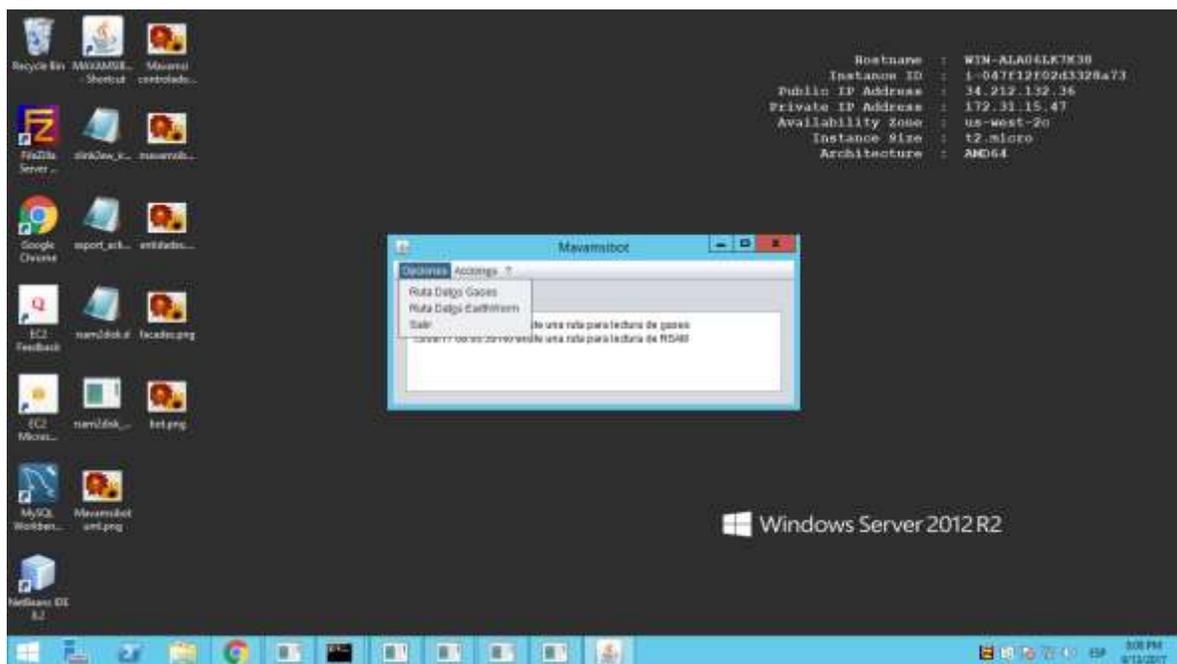
### 5.6.2. MAVAMSI\_BOT1

La configuración de MAVAMSI\_BOT1, sin duda es la más simple de realizar. Este robot, estará ejecutándose de forma permanente y continua en la máquina virtual de la solución.

Lo único que hay que tener en mente para su implementación, son las siguientes consideraciones:

- La aplicación no requiere una instalación, ya que es un archivo ejecutable, solo se necesita tener instalado “Java Runtime Enviroment” (JRE) en su última versión.
- Copiar en una carpeta la capeta lib y el ejecutable “Mavamsibot.jar”, de esta manera puede ser ejecutada.
- La aplicación entra por default en modo automático buscando las rutas donde se almacenan los datos, con las opciones de “rutas” se seleccionan la dirección de la capeta raíz para la lectura de datos de gases como otra ruta raíz de datos RSAM.

*Figura 5.23 - Ejecución de MAVAMSI\_BOT1 en máquina virtual*



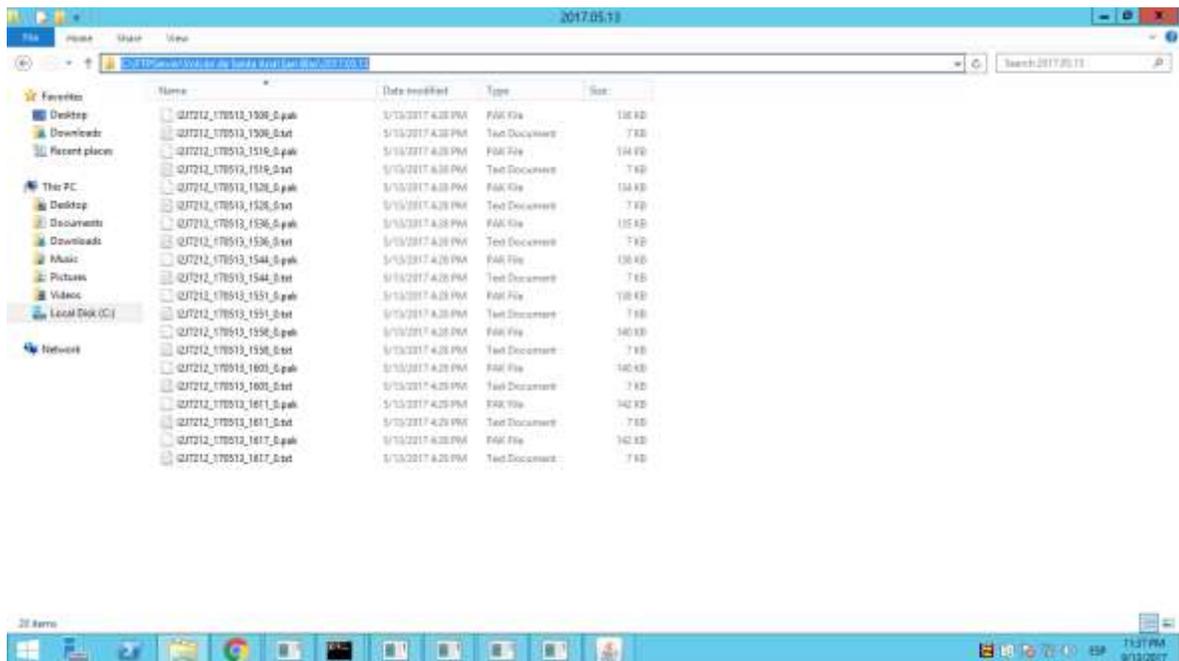
### **5.6.2.1. Configuración de rutas de carpetas de archivos**

La funcionalidad de MAVAMSI\_BOT1, yace en la transformación de archivos de texto plano a registros de datos de la base de la aplicación. Éstos archivos se

almacenan en rutas de carpetas construidas, tanto por MAVAMSI\_BOT1 (para archivos se emisiones de gases) y por EarthWorm (para RSAM) donde se almacenan los archivos que contienen todas las mediciones.

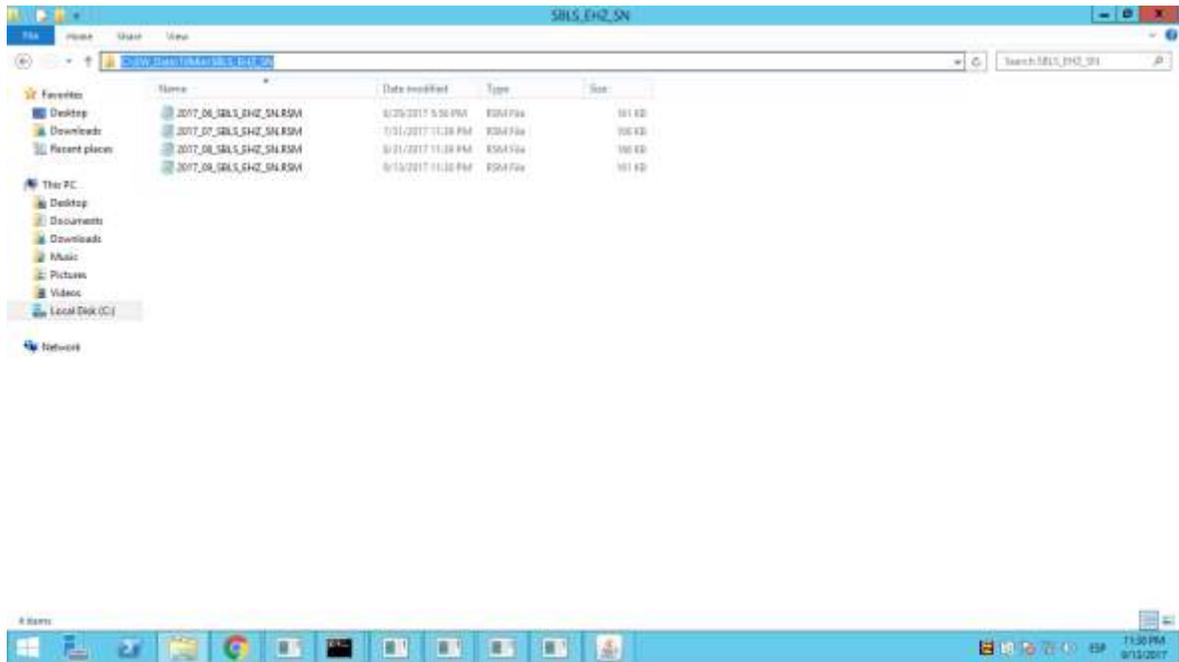
La correcta configuración de estas rutas en MAVAMSI\_BOT1 garantizará su funcionamiento. Las figuras 5.24 y 5.25 muestran la estructura de carpetas para cada fuente de datos.

Figura 5.24 - Estructura de carpetas para archivos de emisiones de gases



Una vez establecidas las rutas de cada carpeta, basta con agregar cada una de ellas en el configurador de MAVAMSI\_BOT1, tal y como lo muestra la figura 5.23.

Figura 5.25 - Estructura de carpetas para archivos de RSAM



### 5.6.3. MAVAMSI

Finalmente, es momento de implementar la aplicación web MAVAMSI. Esta aplicación será implementada en GlassFish, el servidor de aplicaciones que interpretará la aplicación.

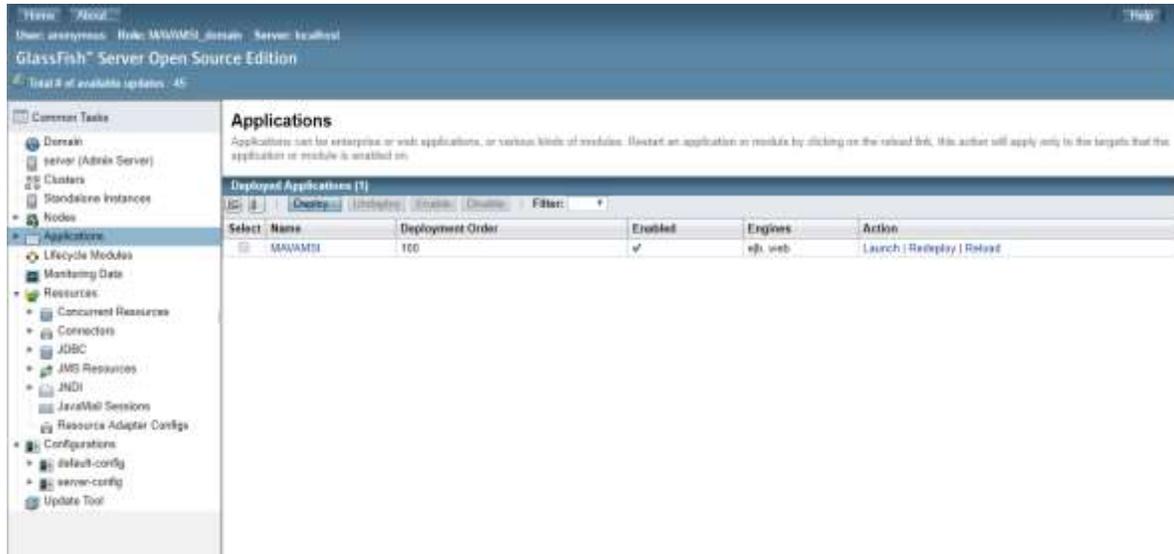
#### 5.6.3.1. Despliegue de MAVAMSI en GlassFish

Para realizar un despliegue de aplicaciones, GlassFish necesita el archivo compilado en formato .war. El archivo compilado de toda la aplicación web, puede encontrar adjunto en el disco compartido entregado junto con el presente trabajo de grado.

El equipo recomienda, poder revisar a detalle la guía de despliegue publicada por el proveedor de software de la herramienta, la cual, se puede consultar mediante el siguiente enlace: <http://docs.oracle.com/cd/E19798-01/821-1757/geyvr/> (sitio validado al 13 de septiembre de 2017).

Si accedemos al apartado Applications, en GlassFish, podremos ver la instancia de nuestra aplicación ejecutándose con toda normalidad.

Figura 5.26 - Implementación de MAVAMSI en GlassFish



### 5.6.3.2. Configuración de JDBC en GlassFish

La última configuración a realizar, será configurar el pool de conexión entre la aplicación web y la base de datos. Para esto se debe configurar JDBC desde GlassFish.

El equipo recomienda, poder revisar a detalle la guía de configuración de JDBC publicada por el proveedor de software de la herramienta, la cual, se puede consultar mediante el siguiente enlace: <https://docs.oracle.com/cd/E19798-01/821-1751/ggnfv/index.html> (sitio validado al 13 de septiembre de 2017).

Las figuras 5.27 y 5.28 muestran la configuración de JDBC que se debe realizar del lado de GlassFish para garantizar la conexión entre la aplicación web y la base de datos.

Figura 5.27 - Configuración de pool de conexión a base de datos desde GlassFish

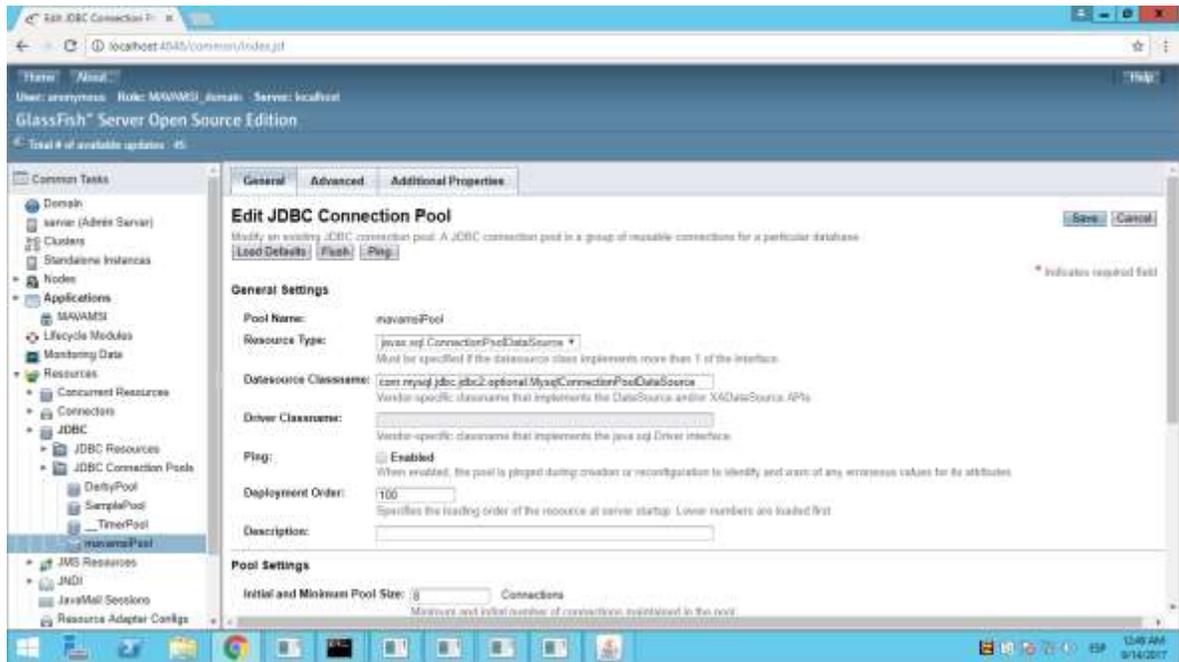
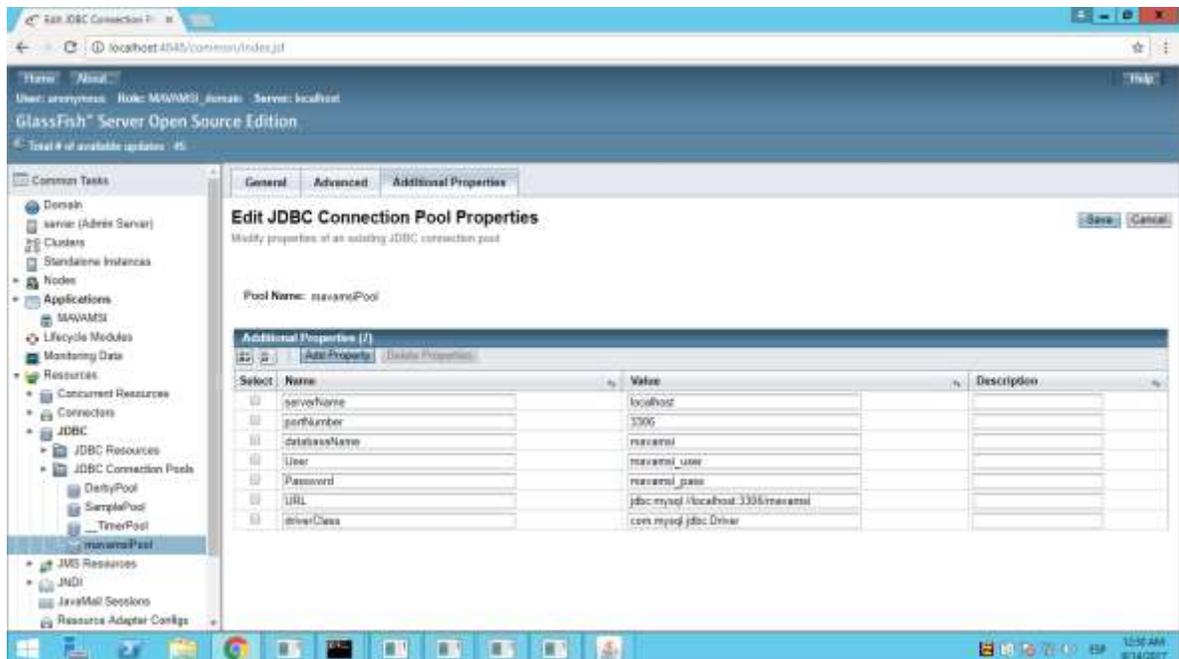


Figura 5.28 - Propiedades de pool de conexión a base de datos desde GlassFish



Como resultado final, tendremos nuestra aplicación correctamente implementada en GlassFish y accesible desde la dirección "localhost" desde la máquina virtual, y or medio de la dirección IP pública de la máquina virtual desde afuera de la misma.

*Figura 5.29 - MAVAMSI implementada correctamente*



## 5.7. ACTIVIDADES DE PLAN DE IMPLEMENTACIÓN

Figura 5.30 – Diagrama de actividades del plan de implementación de la solución

Tarea	Recursos	Días																													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Configuración de Máquina Virtual	1 Programador, 1 Máquina	■	■	■																											
Configuración de red de Máquina Virtual	1 Programador, 1 Máquina				■																										
Configuración de máquina en estación de observación	1 Programador, 1 Máquina					■																									
Instalación y configuración de FileZilla	1 Programador, 1 Máquina						■																								
Instalación y configuración de GlassFish	1 Programador, 1 Máquina							■																							
Instalación y configuración de MySQL Server	1 Administrador de base de datos, 1 Máquina								■																						

Tarea	Recursos	Días																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Instalación y configuración de MySQL Workbench	1 Administrador de base de datos, 1 Máquina																															
Implementación de esquema de base de datos	1 Administrador de base de datos, 1 Máquina																															
Implementación de Novac Program	1 Programador, 1 Máquina																															
Implementación de EarthWorm	1 Programador, 1 Máquina																															
Implementación de MAVAMSI_BOT2	1 Programador, 1 Máquina																															
Implementación de MAVAMSI_BOT1	1 Programador, 1 Máquina																															
Implementación de MAVAMSI en GlassFish	1 Programador, 1 Máquina																															
Configuración de JDBC en GlassFish	1 Administrador de base de																															



# CONCLUSIÓN

La actividad volcánica, es sin duda alguna, un evento que ningún grupo de investigación debería pasar por alto. La historia nos ha demostrado que han existido una infinidad de catástrofes producidas por estos colosos, que, si bien, es imposible predecir con exactitud (y mucho menos evitar), han cobrado consigo las vidas de muchas familias, la salud de otros, y las tierras de muchas poblaciones.

Este tipo de proyecto ayuda a la monitorización constante de cualquier volcán, ya que es una aplicación escalable y de fácil implementación; lo cual genera que el análisis de volcanes se pueda extender a lo largo del país, y así poder estar sabedores de cualquier anomalía, y tomar pronta acción para salvar la mayor cantidad de vidas posible.

La investigación volcánica, es un campo de lento desarrollo en el ámbito de las ciencias naturales en nuestro medio. Esto, ocasiona que se tengas pocos proyectos de ésta índole, y por ende, la documentación sea escasa y sea dificulte más el diseño de un sistema que busca posicionarse en este campo. Sin embargo, a pesar de eso, el equipo de trabajo, logró documentarse bajo las fuentes teóricas necesarias para llevar a cabo el desarrollo de un proyecto que es pionero en su clase y que buscará ayudar a crecer las utilidades informáticas en su medio.

El desarrollo de MAVAMSI cumplió con su doble objetivo. En primer lugar, entregar una aplicación robusta, escalable y responsiva; que muestre la actividad de un volcán en específico mediante el análisis de dos de las condiciones que más marcan su actividad, el movimiento sísmico y la emisión de gases. El segundo, ayudar a entregar cuatro futuros Ingenieros de Sistemas Informáticos a la sociedad; que por medio del diseño, desarrollo e implementación de esta solución, pudieron fortalecer sus habilidades tanto técnicas, como de gestión con entidades, relaciones interpersonales con usuarios finales de aplicación, metodologías de diseño y desarrollo de software, y documentación de procesos y herramientas informáticas.

---

---

# **REFERENCIAS BIBLIOGRÁFICAS**

# REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, L. J. (2017). *Fundamentos de programación: Algoritmos, estructura de datos y objetos*. McGraw Hill. Obtenido de assets.mheducation.es:  
<http://assets.mheducation.es/bcv/guide/capitulo/8448148703.pdf>
- Amazon Web Services. (30 de Julio de 2017). *Amazon EC2*. Obtenido de aws.amazon.com: <https://aws.amazon.com/es/ec2/?p=tile>
- Amazon Web Services. (30 de Julio de 2017). *Cloud Computing con Amazon Web Services*. Obtenido de aws.amazon.com: <https://aws.amazon.com/es/what-is-aws/>
- Andrés, R. (31 de Mayo de 2017). *Qué es una máquina virtual, cómo funciona y para qué sirve*. Obtenido de Computer Hoy:  
<http://computerhoy.com/noticias/software/que-es-virtual-machine-maquina-virtual-46606>
- Apache Commons. (Febrero de 2017). *Apache Commons Net*. Obtenido de Apache Commons: <https://commons.apache.org/proper/commons-net/index.html>
- Atmos. Chem. Phys. (2013). *Volcanic SO2 fluxes derived from satellite data: a survey using OMI, GOME-2, IASI and MODIS*. Bélgica: Atmospheric Chemistric and Physics.
- Belgian Federal Scientific Research Institute. (2016). *The Royal Belgian Institute for Space Aeronomy*. Recuperado el 23 de Junio de 2016, de <http://www.aeronomie.be/en/whatisbira-iasb.htm>

- Bootstrap. (2017). *Getting started*. Obtenido de Bootstrap Documentation: <https://v4-alpha.getbootstrap.com/getting-started/introduction/>
- Dataprix. (25 de Febrero de 2010). *Dataprix*. Obtenido de Herramientas ETL. ¿Que son, para que valen?. Productos mas conocidos. ETL´s Open Source.: <http://www.dataprix.com/blogs/respinosamilla/herramientas-etl-que-son-para-que-valen-productos-mas-conocidos-etl-s-open-sour>
- Dávila Madrid, R. Z. (2011). *Notas Introductorias Sismología*. México: Universidad Autónoma de México.
- Desarrollo Web. (12 de Agosto de 2017). *Desarrollo Web*. Obtenido de Aplicación de Escritorio: <https://desarrolloweb.com/wiki/aplicacion-de-escritorio.html>
- FileZilla. (2017). *Overview*. Obtenido de FileZilla: <https://filezilla-project.org/index.php>
- Forsberg, K., Mooz, H., & Cotterman, H. (2005). *Visualizing Project Management* (Tercera ed.). New York: John Wiley and Sons.
- Google. (2017). *Google reCaptcha*. Obtenido de Google reCaptcha: <https://www.google.com/recaptcha/intro/android.html>
- Google. (2017). *Gmail API Overview*. Obtenido de Google Developers: <https://developers.google.com/gmail/api/guides/>
- Google Cloud. (30 de Septiembre de 2016). *Google Apps for Work ahora se llama G Suite y trae novedades*. Obtenido de G Suite: [https://gsuite.google.com/intl/en\\_uk/products/](https://gsuite.google.com/intl/en_uk/products/)
- Google Cloud. (30 de Julio de 2017). *¿Por qué elegir Google Cloud Platform?* Obtenido de cloud.google.com: <https://cloud.google.com/why-google/?hl=es>

- Google Support. (12 de Agosto de 2017). *Utilizar la configuración de SMTP para enviar correos desde una impresora, un escáner o una aplicación*. Obtenido de Google Support: <https://support.google.com/a/answer/176600?hl=es>
- Ibáñez, J. M., & Carmona, E. (1997). *Sismicidad Volcanica*. Granada: Instituto Andaluz de Geofísica. Universidad de Granada.
- Instrumental Software Technologies, I. (11 de Noviembre de 2016). *EarthWorm*. Obtenido de ISTI.com: <http://www.isti.com/products/earthworm/>
- International Volcanic Health Hazard Network. (2016). *Guía Sobre Gases Volcánicos y Aerosoles*. Recuperado el 20 de Mayo de 2016, de <http://www.ivhhn.org>
- Internet Engineering Task Force. (Marzo de 2014). *IETF Tools*. Obtenido de FILE TRANSFER PROTOCOL (FTP): <https://tools.ietf.org/html/rfc959>
- Java. (Agosto de 2017). *¿Qué es la tecnología Java y para qué la necesito?* Obtenido de Java.com: [https://www.java.com/es/download/faq/whatis\\_java.xml](https://www.java.com/es/download/faq/whatis_java.xml)
- Java. (Agosto de 2017). *Conceptos útiles y glosario de definiciones*. Obtenido de Java.com: [https://www.java.com/es/download/faq/helpful\\_concepts.xml](https://www.java.com/es/download/faq/helpful_concepts.xml)
- Java. (Agosto de 2017). *Conozca más sobre la tecnología Java*. Obtenido de Java.com: <https://www.java.com/es/about/>
- Lippert, A. M., Sevillano Reimers, C., & Millán Flaile, R. (2011). *El IDNA: Estudio de los volcanes*.
- Lorca Mella, E., & Recabarren Herrera, M. (1994). *Terremotos y Tsunamis o Maremotos*. Santiago de Chile: Servicio Hidrográfico y Oceanográfico de la Armada de Chile.

- Lorenzo Salazar, J. M. (2015). *Volcanes, Terremotos y Matemáticas*. España: Universidad de La Laguna.
- Microsoft. (14 de Septiembre de 2016). *Conectarse a otro equipo mediante Conexión a Escritorio remoto*. Obtenido de Support Microsoft: <https://support.microsoft.com/es-es/help/17463/windows-7-connect-to-another-computer-remote-desktop-connection>
- Microsoft Azure. (30 de Julio de 2017). *¿Qué es Azure?* Obtenido de azure.microsoft.com: <https://azure.microsoft.com/es-es/overview/what-is-azure/>
- Microven. (13 de Mayo de 2016). *Máquinas Virtuales en la nube*. Obtenido de Microven.net: <http://www.microven.net/que-es-la-nube/maquinas-virtuales-en-la-nube>
- Mozilla. (10 de Julio de 2017). *Guía de JavaScript - Introducción*. Obtenido de MDN web docs: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>
- Murray, T. L. (1992). *A Real-Time Seismic-Amplitude Measurement System (RSAM)*. U.S. Geological Survey Bulletin.
- MySQL. (2017). *2.10 Postinstallation Setup and Testing*. Obtenido de MySQL 5.7 Reference Manual : <https://dev.mysql.com/doc/refman/5.7/en/postinstallation.html>
- MySQL. (2017). *3.3 Creating and Using a Database*. Obtenido de MySQL 5.7 Reference Manual : <https://dev.mysql.com/doc/refman/5.7/en/database-use.html>

MySQL. (2017). *Chapter 1 Overview of MySQL Connector/J*. Obtenido de MySQL Dev: <https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-overview.html>

NetBeans. (2017). *Bienvenido a NetBeans y www.netbeans.org*. Obtenido de NetBeans: [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html)

NetBeans. (2017). *Información NetBeans IDE 6.1*. Obtenido de NetBeans: [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html)

NOVAC. (2011). *Major scientific achievements*. Obtenido de NOVAC: <http://www.novac-project.eu/project-results.html>

Oracle. (2015). *Understanding Class Members*. Obtenido de Java Documentation: <https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>

Oracle. (2017). *JavaServer Faces Technology Overview*. Obtenido de Java EE: <http://www.oracle.com/technetwork/java/javaee/overview-140548.html>

Oracle. (2017). *Oracle MySQL*. Obtenido de Oracle: <https://www.oracle.com/lad/mysql/index.html>

Perry, J. S. (3 de 12 de 2012). *Conceptos básicos del lenguaje Java*. Obtenido de IBM developerWorks: <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/index.html>

Power Data. (Mayo de 2016). *¿Qué es un gestor de datos y para qué sirve?* Obtenido de Power Data: <http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-un-gestor-de-datos-y-para-que-sirve>

Power Data. (Agosto de 2016). *Data Warehouse: todo lo que necesitas saber sobre almacenamiento de datos*. Obtenido de Power Data. Especialistas en Gestión de Datos: <http://www.powerdata.es/data-warehouse>

- PrimeFaces. (2017). *Why PrimeFaces*. Obtenido de PrimeFaces:  
<https://www.primefaces.org/whyprimefaces/>
- Rumbaugh, J. R. (2005). *Object-Oriented Modeling and Design with UML*.  
Missouri: Pearson.
- Sandoval, S., Montalvo, F., & Rosa Lúe, A. A. (2013). *Estudio de los gases de la cadena volcánica y comparación con gases geotermales en El Salvador*.  
San Salvador: Universidad de El Salvador.
- Servicio Nacional de Estudios Territoriales - SNET. (2016). *Descripción Volcán de Santa Ana*. Recuperado el 5 de Junio de 2016, de  
<http://mapas.snet.gob.sv/geologia/vulcanologia/descripiconVSA.htm>
- SNET. (2016). *Servicio Geológico*. Recuperado el 23 de Junio de 2016, de  
<http://www.snet.gob.sv/Geologia/index.htm>
- Sol, M. (02 de Febrero de 2016). El Salvador: Un volcán Activo. *El Universitario*.  
Recuperado el 24 de Junio de 2016, de  
[http://www.eluniversitario.ues.edu.sv/index.php?option=com\\_content&view=article&id=4651:2016-02-02-21-17-32&catid=41:acontecer&Itemid=30](http://www.eluniversitario.ues.edu.sv/index.php?option=com_content&view=article&id=4651:2016-02-02-21-17-32&catid=41:acontecer&Itemid=30)
- Universidad Carlos III de Madrid. (2008). *Java Server Faces (JSF)*. Obtenido de  
<http://www.it.uc3m.es>:  
<http://www.it.uc3m.es/spickin/docencia/comsoft/presentations/spanish/07-08/JSF.pdf>
- Universidad Católica de Valparaíso. (2016). *Volcanes y Sismicidad en Chile*.  
Recuperado el 19 de Junio de 2016, de  
<http://ciencias.ucv.cl/quimica/mod2/q2m2a007.htm>
- Universidad de Alicante. (2006). *Qué es una aplicación web*. Alicante:  
Departamento de lenguajes y sistemas informáticos.

Universidad de Chile. (2017). *Java Virtual Machine*. Obtenido de Centro de Computación Universidad de Chile:  
<http://www.cec.uchile.cl/~luvasque/edo/java/manuales/JVM%20y%20variables%20de%20entorno.pdf>

Universidad de El Salvador. (2016).  
<http://www.fia.ues.edu.sv/tgeisi/paginas/politicas.html>. Recuperado el 2016, de Ley de Fomento y Protección a la Propiedad Intelectual.

Universidad de El Salvador. (2016). *Instituto de Vulcanología IV-UES*. Recuperado el 24 de Junio de 2016, de  
<http://www.ues.edu.sv/CENTROSINSTITUTIVOS/iv-ues>

Universidad de Navarra. (2003). *Qué es el JDK (Java Development Kit)*. Obtenido de <http://www.esi.unav.es>:  
<http://www.esi.unav.es/Asignaturas/Informat2/Clases/Clases9899/Clase01/JavaEntorno/tsld003.htm>

Universidad del País Vasco. (2017). *La Máquina Virtual Java*. Obtenido de <http://www.sc.ehu.es>:  
<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm>

Vera Lizcano, B. E. (2003). *Evaluación de Errores de Datos Sismológicos: Conceptos, Instrumentación y Observación Sismológica*. Santiago de Cali: Universidad del Valle.

---

---

# **ANEXOS**

# ANEXO 1: SOLICITUD DE INFORMACIÓN DE MOVIMIENTO SÍSMICO AL MARN

Santa Ana, 23 de junio de 2016.

Ing. Celina Kattán  
Directora General del Observatorio Ambiental  
Ministerio de Medio Ambiente  
Presente

Estimada Ing. Kattán:

Reciba un cordial saludo deseándole éxitos en sus labores diarias.

El Infrascrito Jefe del Departamento de Ingeniería y Arquitectura de la Facultad Multidisciplinaria de Occidente hace constar que los estudiantes :

1. Cáceres Barrientos, Luis Alberto
2. Cárcamo González, Carlos Alexander
3. Padilla Hernández, Marvin René
4. Zaldaña Lemus, Melvin Guillermo

Actualmente son egresados de la carrera de Ingeniería en Sistemas informáticos y se encuentran desarrollando el trabajo de grado titulado "Diseño e implementación de un sistema para monitoreo volcánico de variables multiparametricas sísmicas y emisiones de Dióxido de Azufre (SO<sub>2</sub>) en el volcán Santa Ana"

Dada la importancia y naturaleza del trabajo de tesis solicitamos a Ud. Sus buenos oficios para que nos puedan brindar:

1. Las señales analógicas-digitales de las estaciones sísmicas: San Blas (SBL), San José (SNJE), La montaña (MONT) y El Retiro (RTR) correspondientes al volcán Santa Ana (VSA) en el formato adecuado para evaluar las señales en tiempo real.
2. Una charla informativa en la que se detallen:  
Los instrumentos de adquisición de señales analógicas-digitales, tipo, filtraje, ruido, software aplicado, algoritmos para la captura de las velocidades de las ondas S, P, Rayleigh y Love, clasificación de sismos DVT, VT, LP, tremor, Híbridos etc., RSAM, espectrogramas distancia epicentral, cálculo del valor de "b" en la ecuación de Gutenberg -Ritcher, y otros que se consideren pertinentes a la temática.
3. Una charla informativa en la que se detallen:  
El sistema ScanningDOAS, Base de datos de los escaneos en formato digital desde 2005 de la estación del volcán Santa Ana, proceso para la adquisición de los scanning y el cálculo del flujo de Dióxido de Azufre (SO<sub>2</sub>), base de datos digital de la velocidad de viento y modelo a pequeña escala.

Considerando que las dos charlas son extensas, mucho agradeceríamos que se nos concedieran en dos fechas diferentes.

Por la atención a la presente esperando su fina colaboración, me suscribo de Ud, atentamente

Ing. Douglas García Rodezno  
Jefe Interino del Departamento de Ingeniería y Arquitectura  
Facultad Multidisciplinaria de Occidente

cc. Ing. José Roberto Colón Villalta.  
Asesor del Trabajo de Grado

## ANEXO 2: CARTA DE COMPROMISO CON MARN



### CARTA COMPROMISO.

Donación e Intercambio de Información procedente de la  
Dirección General del Observatorio Ambiental.

Nosotros, *Luis Alberto Cáceres Barrientos* con DUI 04097337-4 y Carnet Estudiantil CB07025; *Carlos Alexander Cárcamo González* con DUI 04659826-1 y Carnet Estudiantil CG10015; *Melvin Guillermo Zaldaña Lemus* con DUI 04458235-3 y Carnet Estudiantil ZL0901; *Marvin René Padilla Hernández* con DUI 04132604-1 y Carnet Estudiantil PH07019; extendido en la Universidad de El Salvador, nos comprometemos a las siguientes condiciones con la información que me ha sido entregada por el Ministerio de Medio Ambiente y Recursos Naturales a través de la Dirección General del Observatorio Ambiental:

- 1) No proporcionar a terceras personas
- 2) Utilizarla exclusivamente para el trabajo de Tesis
- 3) No utilizarla con fines de lucro
- 4) Poner la fuente de la información DOA/MARN
- 5) Entregar una copia en formato digital de la tesis a la DOA

Detalle de la información proporcionada:

Se solicita del acceso a la información de movimiento sísmico para la estación de observación y monitoreo volcánico "San Blas", para el volcán de Santa Ana; esto con el fin de poder adaptar esta información a nuestro trabajo de grado titulado Diseño e Implementación de Sistema para Monitoreo Volcánico de Variables Multiparamétricas Sísmicas y Emisiones de dióxido de Azufre en el Volcán de Santa Ana.

Anexar copia del Carnet Universitario

Kilómetro 5½ Carretera a Santa Tecla,  
Calle y Colonia Las Mercedes,  
Edificio MARN, complejo IISTA,  
San Salvador, El Salvador, Centro América.

Tel: (503) 2132-6276  
Correo electrónico: medioambiente@marn.gob.sv  
facebook.com/marn.gob.sv  
Twitter: @MARN\_Oficial\_SV

## ANEXO 3: CARTA DE CONFORMIDAD DE FINALIZACIÓN DE APLICACIÓN

Santa Ana, 16 de septiembre de 2017

A quien Interese:

Yo, Rodolfo Olmos, por este medio, hago constar que los bachilleres Carlos Alexander Cárcamo González, Luis Alberto Cáceres Barrientos, Marvin René Padilla Hernández y Melvin Guillermo Zaldaña Lemus; han desarrollado el trabajo de grado titulado "*Diseño e Implementación de Sistema para Monitoreo Volcánico de Variables Multiparamétricas Sísmicas y Emisiones de Dióxido de Azufre en el Volcán de Santa Ana*"; cuyo resultado final desemboca en la construcción de una aplicación web.

Yo, en mi rol de Especialista en Vulcanología, doy fe que los jóvenes han diseñado y desarrollado el proyecto en mi compañía, donde he aportado de mis conocimientos en el área para el correcto funcionamiento de la aplicación.

Hago constar, que la aplicación web, cumple con los requerimientos funcionales establecidos en la fase de inceptión del proyecto, desempeñando de forma correcta la inserción de información en una base de datos centralizada de mediciones de movimiento sísmico y emisiones de dióxido de azufre; la representación de esta información por medio de reportes con tablas y gráficas de RSAM, concentración de gas de dióxido de azufre y reportes comparativos de ambas variables; y el envío de alertas por correo electrónico.



Licenciado Rodolfo Olmos

## ANEXO 4: VIDEOS DE MANUALES DE USUARIO

Adjunto con el presente trabajo de grado, se entrega el manual de usuario de la aplicación MAVAMSI; explicado por medio de video tutoriales que ayuden a que la comprensión de la información por medio de los usuarios sea más rápida y más interactiva.

El contenido de cada uno de los videos es el siguiente:

- **MAVAMSI – Acceso y Restablecer Contraseñas:** Con este video se pretende mostrar al usuario, los pasos a seguir para iniciar sesión en la herramienta; y de caso de ser necesario, los pasos para restablecer la contraseña en caso de haberla extraviado.
- **MAVAMSI – Módulo “Gestión de Accesos”:** En este vídeo, se estudia a fondo el módulo de Gestión de Accesos de la aplicación; donde se muestran los pasos a seguir para la creación de roles y asignación de permisos a cada módulo de la herramienta, que poseerá cada usuario.
- **MAVAMSI – Módulo “Gestión de Vulcanología”:** En este tutorial, se aprenderá a configurar cada volcán que se desee analizar por medio de la aplicación, en caso de tener fuentes de datos provenientes de múltiples estaciones y volcanes. Se aprenderá a crear un volcán con todas sus propiedades, necesario para el funcionamiento de los procesos de extracción de la herramienta.

- **MAVAMSI – Módulo “Análisis de Gases”:** En esta entrega, se verá cómo se representa la información de emisiones de gases para los volcanes configurados en la aplicación. Se verán los tipos de reportes que se pueden crear, sus respectivas gráficas y tablas, y los filtros que ayuden a segmentar la información. Además, se aprenderá a exportar esta información de la aplicación en diferentes formatos.
- **MAVAMSI – Módulo “Análisis de Sismicidad”:** En esta parte, se verá cómo se representa la información de movimiento sísmico (RSAM) para los volcanes configurados en la aplicación. Se verán los tipos de reportes que se pueden crear, sus respectivas gráficas y tablas, y los filtros que ayuden a segmentar la información. Además, se aprenderá a exportar esta información de la aplicación en diferentes formatos.
- **MAVAMSI – Módulo “Análisis Comparativos”:** En este punto, se verá como realizar reportes que combinen la información de emisiones de gases y movimiento sísmico para los volcanes configurados en la aplicación. Se conocerán los diferentes reportes a crear, con sus gráficas y tablas, y con los posibles filtros a utilizar.
- **MAVAMSI – Configuración de Usuarios:** Finalmente, se mostrarán los pasos a seguir para la administración de usuarios de la aplicación. En este video aprenderemos a crear, modificar, o eliminar usuarios que designen acceso a MAVAMSI.

## **ANEXO 5: EVIDENCIAS DE EQUIPO DE TRABAJO EN DESARROLLO DEL PROYECTO**

*Figura Anexo 1 – Sensor ScanDoas*



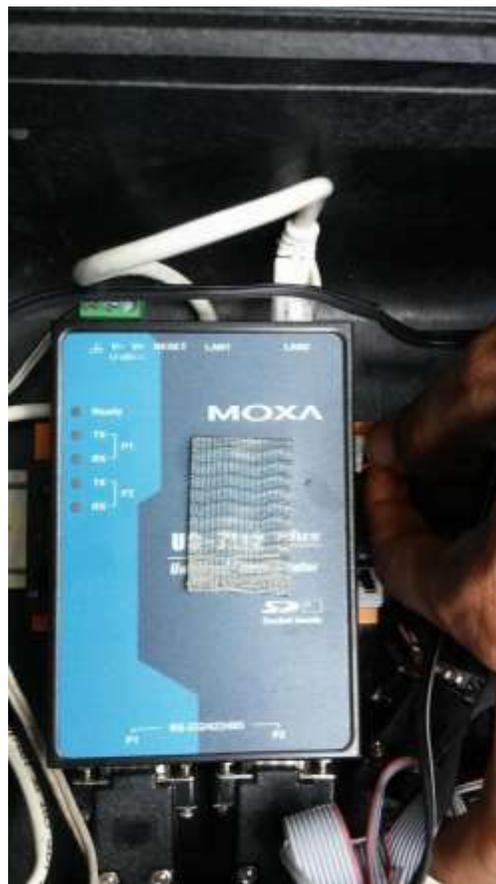
*Figura Anexo 2 - Sensor ScanDoas*



*Figura Anexo 3 - Componentes de sensor de medición de emisiones de gas*



*Figura Anexo 4 - Placa MOXA, de sensor de medición serie S2000*



*Figura Anexo 5 - Instalación de componentes de medición de gas*



*Figura Anexo 6 - Instalación de componentes de medición de gas*



*Figura Anexo 7 - Instalación de componentes de medición de gas*



*Figura Anexo 8 - Configuración de ScanDoas en Novac Program*



*Figura Anexo 9- Configuración de ScanDoas en Novac Program*



*Figura Anexo 10 - Instalación de equipo de mediciones*



*Figura Anexo 11 - Instalación de equipo de mediciones*



*Figura Anexo 12 - Visita técnica de equipo de trabajo a oficinas de MARN*



*Figura Anexo 13 - Visita técnica de equipo de trabajo a oficinas de MARN*

