

INFORME FINAL DE INVESTIGACIÓN

SISTEMA TELEMÁTICO DE MONITOREO DE CALIDAD DEL AIRE EN ZONAS REMOTAS. DESARROLLO DE UNA PLATAFORMA PARA EL ANÁLISIS DE DATOS Y GENERACIÓN DE MODELOS POR MEDIO DE TÉCNICAS DE BIG DATA Y MACHINE LEARNING

En asocio con la Universidad Tecnológica de El Salvador y el Ministerio de Medio Ambiente y Recursos Naturales

DOCENTE INVESTIGADOR PRINCIPAL:
ING. ELVIS MOISÉS MARTÍNEZ PÉREZ

DOCENTE CO-INVESTIGADORA:
INGA. RINA ELIZABETH LÓPEZ MENÉNDEZ

ESCUELA DE INGENIERÍA EN COMPUTACIÓN
ITCA-FEPADE SEDE CENTRAL

ENERO 2020

INFORME FINAL DE INVESTIGACIÓN

SISTEMA TELEMÁTICO DE MONITOREO DE CALIDAD DEL AIRE EN ZONAS REMOTAS. DESARROLLO DE UNA PLATAFORMA PARA EL ANÁLISIS DE DATOS Y GENERACIÓN DE MODELOS POR MEDIO DE TÉCNICAS DE BIG DATA Y MACHINE LEARNING

En asocio con la Universidad Tecnológica de El Salvador y el Ministerio de Medio Ambiente y Recursos Naturales

DOCENTE INVESTIGADOR PRINCIPAL:
ING. ELVIS MOISÉS MARTÍNEZ PÉREZ

DOCENTE CO-INVESTIGADORA:
INGA. RINA ELIZABETH LÓPEZ MENÉNDEZ

ESCUELA DE INGENIERÍA EN COMPUTACIÓN
ITCA-FEPADE SEDE CENTRAL

ENERO 2020

Rectora

Licda. Ely Escolar Santo Domingo

Vicerrector Académico

Ing. Carlos Alberto Arriola Martínez

Vicerrectora Técnica Administrativa

Inga. Frineé Violeta Castillo

Director de Investigación y Proyección Social

Ing. Mario W. Montes Arias

Dirección de Investigación y Proyección Social

Ing. David Emmanuel Ágreda Trujillo

Inga. Ingrid Janeth Ulloa de Posada

Sra. Edith Aracely Cardoza de González

Directora de Escuela de Ingeniería de Computación

Inga. Marta Corina Quijano de García

363.739 202 85

M385s Martínez Pérez, Elvis Moisés, 1977 -

slv Sistema telemático de monitoreo de calidad del aire en zonas remotas. Desarrollo de una plataforma para el análisis de datos y generación de modelos por medio de técnicas de Big Data y Machine Learning [recurso electrónico] : en asocio con la Universidad Tecnológica de El Salvador y el Ministerio de Medio Ambiente y Recursos Naturales / Elvis Moisés Martínez Pérez, Rina Elizabeth López Menéndez, coaut. -- 1ª ed. -- Santa Tecla, La Libertad, El Salv. : ITCA Editores, 2020.

1 recurso electrónico (70 p. : il. col. ; 28 cm.)

Datos electrónicos (1 archivo : pdf, 8.9 mb). –
<https://www.itca.edu.sv/produccion-academica/>
ISBN : 978-99961-39-40-6 (E-Book, pdf)
ISBN : 978-99961-39-26-0 (Impreso)

1. Control de calidad del aire – Equipo. 2. Programas integrados Para computador. 3. Inteligencia artificial. 4. Sistemas de telecomunicación. I. López Menéndez, Rina Elizabeth, coaut. II. Título.

Autores

Ing. Elvis Moisés Martínez Pérez (ITCA-FEPADE)

Ing. Rony Cortez (UTECS)

Co Autores

Inga. Rina Elizabeth López de Jiménez (ITCA-FEPADE)

Ing. Omar Otoniel Flores Cortez (UTECS)

Docentes Participantes

Ing. Eduardo Antonio Amaya (ITCA-FEPADE)

Inga. Verónica Idalia Rosa (UTECS)

Tiraje: 13 ejemplares

Año 2020

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica, el sector empresarial y la sociedad, como un aporte al desarrollo del país. Para referirse al contenido debe citar el nombre del autor y el título del documento. El contenido de este Informe es responsabilidad de los autores.



Atribución-No Comercial
Compartir Igual
4.0 Internacional

Esta obra está bajo una licencia Creative Commons. No se permite el uso comercial de la obra original ni de las posibles obras derivadas, cuya distribución debe hacerse mediante una licencia igual que la sujeta a la obra original.

Escuela Especializada en Ingeniería ITCA-FEPADE

Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio Web: www.itca.edu.sv

TEL: (503)2132-7423

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA.....	5
2.1.	DEFINICIÓN DEL PROBLEMA	5
2.2.	ANTECEDENTES / ESTADO DE LA TÉCNICA	6
2.3.	JUSTIFICACIÓN	6
3.	OBJETIVOS	6
3.1.	OBJETIVO GENERAL	6
3.2.	OBJETIVOS ESPECÍFICOS	7
4.	HIPÓTESIS	7
5.	MARCO TEÓRICO	7
6.	METODOLOGÍA DE INVESTIGACIÓN	13
6.1.	PREPARACIÓN DEL SERVIDOR	18
6.1.1.	REQUERIMIENTOS MÍNIMOS DEL SERVIDOR	19
6.1.2.	REQUERIMIENTOS MÍNIMOS DEL CLIENTE.....	19
6.1.3.	CONFIGURACIÓN DE SERVIDOR LAMP (LINUX, APACHE, MONGODB, PHP).....	19
6.2.	PLAN DE IMPLEMENTACIÓN	29
6.3.	PLAN DE CAPACITACIONES	30
7.	RESULTADOS	30
7.1.	ESTACIONES DE MONITOREO DE CALIDAD DEL AIRE	30
7.2.	APLICACIÓN WEB PARA EL MONITOREO DE VARIABLES AMBIENTALES	37
7.3.	ALGORITMO DE PREDICCIÓN DE MACHINE LEARNING	44
7.4.	CONSTRUCCIÓN Y PROGRAMACIÓN DE ESTACIONES DE MONITOREO	57
8.	CONCLUSIONES	65
9.	RECOMENDACIONES	66
10.	GLOSARIO	66
11.	REFERENCIAS BIBLIOGRÁFICAS.....	69
12.	ANEXOS	70

1. INTRODUCCIÓN

El presente proyecto de investigación fue desarrollado por la Escuela de Ingeniería en Computación de la *Escuela Especializada en Ingeniería ITCA-FEPADE* y la *Universidad Tecnológica de El Salvador (UTEC)*; surge como una oportunidad para brindarle al país dispositivos tecnológicos de bajo costo, pero con un alto potencial de impacto en el bienestar de la población salvadoreña.

El objetivo del proyecto fue desarrollar una aplicación para el análisis, interpretación y predicción de datos proveniente de estaciones de monitoreo remoto de contaminación ambiental por material particulado PM 2.5 y PM 10 por medio de técnicas de Machine Learning y Big Data para el Ministerio de Medio Ambiente y Recursos Naturales de El Salvador.

Para lograr el objetivo planteado, tenemos como socio estratégico al *Ministerio de Medio Ambiente y Recursos Naturales* de El Salvador (MARN), con el cual se ha validado toda la información y el funcionamiento de los dispositivos. El diseño del dispositivo fue desarrollado por parte de la UTEC en una primera etapa denominada "*Sistema telemático de monitoreo de calidad del aire en zonas remotas, utilizando técnicas IoT y Big data*" [1].

Este proyecto consistió en instalar 8 estaciones de monitoreo en: ITCA-FEPADE Santa Tecla, San Miguel y La Unión; UTEC San Salvador, Alcaldía Municipal de Santa Ana, Instituto Nacional de Antigua Guatemala, Colegio Adventista de Chalatenango y en la Universidad de El Salvador. Con las estaciones instaladas se procedió con la captura de datos en formato para Big Data los cuales son procesados por medio de una plataforma web en donde se grafica el estado de la calidad del aire según la zona seleccionada. Se elaboró un algoritmo de Machine Learning de predicción, el cual es un complemento para las estaciones y realiza una predicción de la calidad del aire para el término de un mes. Según aumente la cantidad de muestras así será el potencial de predicción para día, semana, mes o año. Estos dispositivos miden el nivel de contaminación del aire por material particulado de 2.5 y 10 Micras de diámetro en las zonas donde están instalados.

Se desarrolló una aplicación web para el monitoreo y almacenamiento de variables ambientales que es capaz de ejecutarse en cualquier tipo de pantalla, que van desde teléfonos inteligentes, tabletas o hasta computadoras de escritorio. Además, se diseñó un algoritmo de predicción de Machine Learning. Para la realización del algoritmo se utilizó el lenguaje de programación R, mediante un framework de desarrollo llamado Júpiter. Cabe mencionar que el algoritmo se ha realizado con datos proveniente solamente de la estación instalada en ITCA-FEPADE Sede Central y de la cual solo se han tomado las variables: Fecha, Hora y Material Particulado 2.5. Se cuentan con datos desde el mes de junio a diciembre de 2019. Por el momento se encuentra en versión de prototipo y en etapa de entrenamiento; ya que para obtener mejores resultados de predicción se requiere de una gran cantidad de datos de aproximadamente 5 años de preferencia para que las predicciones sean más exactas que las que se generan al momento.

La aplicación de los resultados de este proyecto con la construcción de más estaciones de monitoreo, permitirá lograr una cobertura a nivel nacional y medir con más detalle la calidad del aire que se respira en El Salvador, esto permitirá mejorar la toma de decisiones respecto al combate de las enfermedades respiratorias. Se ha logrado proveer al país de un diseño de estación a bajo costo para medición de material particulado 2.5 y 10, con su respectiva plataforma de interpretación de datos y un algoritmo de Machine Learning para predicción de contaminación ambiental.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. DEFINICIÓN DEL PROBLEMA

Actualmente la vigilancia epidemiológica nacional es un tema complejo, y las enfermedades respiratorias según el boletín epidemiológico 2018 del Ministerio de Salud Pública son de las más consultadas por la población salvadoreña.

Hasta la primera semana de diciembre de 2018, el Ministerio de Salud recibió 1,767,030 consultas por Infecciones respiratorias agudas, estas comprenden desde un resfriado leve, un catarro hasta el grado más delicado como una neumonía.

Los niveles de contaminación del aire en El Salvador son mayoritariamente en el Departamento de San Salvador, por ser un área urbana donde el transporte público y privado colaboran en contaminar todos los días el ambiente.

El país tiene solo tres estaciones automáticas que componen la Red de Monitoreo de la Calidad del Aire (REDCA), para medir las inmisiones atmosféricas del contaminante PM 2.5 en el Área Metropolitana de San Salvador. Estas estaciones están en el Centro de Gobierno, Universidad Don Bosco, y en el Comando de Doctrina y Educación Militar.

El Ministerio de Medio Ambiente y Recursos Naturales tiene un registro de las mediciones del PM 2.5, que son partículas 20 veces más pequeñas que el grosor de un cabello, por lo que pueden ingresar con facilidad a los pulmones y provocar serios problemas en la salud. Para que la calidad del aire sea buena la cantidad de medición de las partículas contaminantes debe mantenerse en un rango de 0 a 15.3 $\mu\text{g}/\text{m}^3$.

La información que proporcionan estas estaciones puede ser consultada en el sitio web <http://www.snet.gob.sv/ver/meteorologia/monitoreo/calidad+del+aire/> donde solo muestra una tabla con los índices de calidad del aire en la zona metropolitana de San Salvador, los cuales son: buena, satisfactoria, no satisfactoria, dañina a la salud, muy dañina a la salud y peligroso.

El problema actual es que la información de las estaciones no es dinámica, no se puede consultar el estado del aire en tiempos pasados a manera de comparación ni mucho menos hacer una proyección a futuro sobre qué tan buena será el nivel de pureza del aire.

Aunado a este problema está la lentitud en el soporte que reciben estas estaciones, ya que los datos son proporcionados desde Europa, lo que dificulta su obtención.

El objetivo del proyecto no fue reemplazar las estaciones actuales, sino agregar 8 equipos más que ayuden a monitorear el ambiente y contribuir con la población salvadoreña a tener mayor información sobre la pureza del aire y mejorar la toma de decisiones respecto al uso de equipo de protección para las vías respiratorias o cambiar de ruta en ciertas horas del día cuando los niveles de contaminación están más elevados.

Es por este motivo que la *Universidad Tecnológica de El Salvador* y la *Escuela Especializada en Ingeniería ITCA-FEPADE* unieron esfuerzos para completar la segunda etapa de este proyecto contribuyendo en construir e instalar 8 estaciones de monitoreo de calidad del aire con su respectiva plataforma de interpretación de datos interactiva para el usuario.

2.2. ANTECEDENTES / ESTADO DE LA TÉCNICA

Desde inicios de 1950 se observó en los países de América Latina y el Caribe una preocupación por la contaminación del aire. Las universidades y dependencias de los ministerios de salud fueron los organismos que realizaron las primeras mediciones de contaminación en el aire.

Actualmente se conocen las siguientes iniciativas para abordar esta problemática a nivel internacional:

- La Red Panamericana de Muestreo Normalizado de la Contaminación del Aire (REDPANAIRE) inició sus operaciones en junio de 1967, con la recolección de muestras mensuales de polvo sedimentable (PS) y muestras diarias de partículas totales en suspensión (PTS) y de SO₂. La REDPANAIRE comenzó con ocho estaciones y a fines de 1973 tenía un total de 88 estaciones distribuidas en 26 ciudades de 14 países.
- El AMIS es un programa desarrollado por la OMS como parte del programa de Ciudades Saludables. El objetivo del AMIS es actuar como un sistema global de intercambio de información sobre la calidad del aire.
- En El Salvador, el Ministerio de Medio Ambiente y Recursos Naturales es la entidad encargada de realizar las mediciones y divulgación de datos referente a la calidad del aire en el Área Metropolitana de San Salvador (AMSS).

2.3. JUSTIFICACIÓN

El alto índice de enfermedades respiratorias en la población salvadoreña y la necesidad en hacer uso de nuevas tecnologías para mejorar nuestra calidad de vida justifican el desarrollo de este proyecto; adicionalmente al costo beneficio que posee. Una estación del MARN actualmente cuesta alrededor de unos US\$15,000 y las propuestas US\$700.00. Este precio puede variar en función de futuras actualizaciones que se le puedan realizar, tales como agregar más sensores para monitorear nuevas variables, por ejemplo, establecer la calidad del agua. En un promedio de 8 años, el país no ha tenido buena calidad del aire, provocando con esto que la salud de los salvadoreños este bajo riesgo de contraer muchas enfermedades respiratorias [2]. Adicional a lo anterior, el hecho de poder contar con dispositivos creados en el país hace que su mantenimiento sea más barato y factible para el MARN en poder instalar más aparatos en puntos estratégicos que ayuden a tomar decisiones para mejorar la calidad del aire de la población en general.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar una aplicación para el análisis, interpretación y predicción de datos proveniente de estaciones de monitoreo remoto de contaminación ambiental por material particulado PM 2.5 y PM 10 por medio de técnicas de Machine Learning y Big Data para el Ministerio de Medio Ambiente y Recursos Naturales de El Salvador.

3.2. OBJETIVOS ESPECÍFICOS

ITCA-FEPADE

- a) Construir e instalar dos estaciones de monitoreo en las regionales de San Miguel y La Unión para medir la contaminación del aire por material particulado en dichas zonas.
- b) Instalar una estación de monitoreo remoto dentro de las instalaciones de ITCA-FEPADE Santa Tecla proporcionado por la Universidad Tecnológica de El Salvador.
- c) Investigar e implementar herramientas de modelado Big Data para el diseño de la base de datos del proyecto.
- d) Programar un algoritmo de predicción para material particulado PM 2.5 y PM 10 de Machine Learning que ayude a obtener predicciones en términos de un periodo establecido.
- e) Desarrollar una aplicación de fácil acceso y uso para el usuario, integrando base de datos Big Data para la correcta interpretación de los datos proporcionado por las estaciones de monitoreo instaladas.

UTEC

- a) Realizar un estudio de requerimientos y factibilidades en la selección de ubicaciones para el montaje de las estaciones de monitoreo.
- b) Construir 6 estaciones de monitoreo de calidad del aire de bajo costo para la medición de contaminante PM 2.5, PM 10 y temperatura.
- c) Impartir una capacitación a todo el equipo de trabajo de ambas IES sobre Machine Learning.
- d) Capacitar a docentes de ITCA-FEPADE en el ensamblado y mantenimiento de los equipos de monitoreo.

4. HIPÓTESIS

¿Puede mejorar el control de la calidad del aire al incorporar estaciones de monitoreo y recolección de datos, para su posterior análisis con técnicas de machine Learning y Big Data?

5. MARCO TEÓRICO

En un promedio de ocho años San Salvador no ha tenido ningún año con buena calidad del aire como promedio anual, lo que provoca un detrimento en la salud de la población salvadoreña. Los efectos pueden ser inmediatos, al respirar aire muy contaminado o de largo plazo, al estar expuestos por mucho tiempo a un aire que apenas tiene una calidad satisfactoria.

Según el Ministerio de Medio Ambiente y en base a las mediciones que realizan las estaciones de monitoreo, las concentraciones de material particulado PM2.5 en la inmisión atmosférica (concentración de contaminantes en la atmosfera) puede clasificarse en base al siguiente gráfico:



Figura 5.1: Clasificación del nivel de concentración de material particulado PM2.5 en la atmósfera

¿Qué es el contaminante PM 2.5?

El material particulado PM 2.5 está formado por partículas que se encuentran en el aire, capaces de provocar serios problemas en la salud. La mayoría se producen por las emisiones de los vehículos diésel.[2]

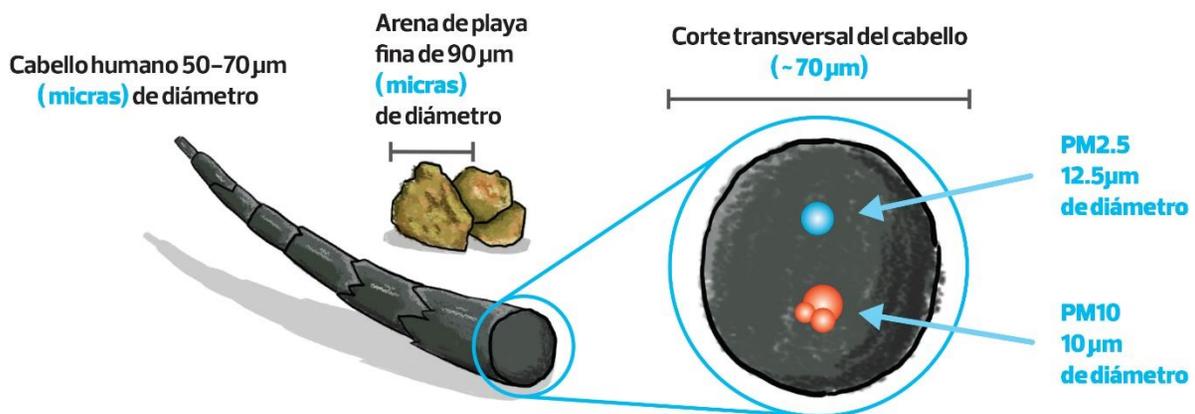


Imagen 5.2: Comparación de tamaño de PM2.5 con cabello humano y grano de arena para comparar el tamaño de dichas partículas.

Fuente: <https://sinapsismx.com/que-son-las-particulas-pm-2-5-y-como-afectan-tu-salud/>

La contaminación provocada por el humo de los vehículos, o de otras fuentes de emisión, a largo plazo se puede convertir en un daño a la salud más grave. La mala calidad del aire puede tener efectos inmediatos o a largo plazo en la salud de las personas. Todos pueden enfermarse a causa de la contaminación del aire, pero las personas sensibles que ya padecen alguna enfermedad, los niños y adultos mayores, pueden sufrir complicaciones en su salud con rapidez que pueden ser mortales.

El Ministerio de Medio Ambiente presentó en el 2014 el protocolo para actuar ante emergencias por contaminación del aire.



Imagen 5.3: Niveles de alerta según la cantidad de concentración de contaminante PM2.5

BIG DATA

Es una disciplina que se ocupa de todas las actividades relacionadas con los sistemas que manipulan grandes conjuntos de datos. Las dificultades más habituales vinculadas a la gestión de estas cantidades de datos se centran en la recolección y el almacenamiento, búsqueda, compartición, análisis, y visualización. La tendencia a manipular enormes cantidades de datos se debe a la necesidad en muchos casos de incluir dicha información para la creación de informes estadísticos y modelos predictivos utilizados en diversas materias, como los análisis de negocio, publicitarios, los datos de enfermedades infecciosas, el espionaje y seguimiento a la población o la lucha contra el crimen organizado. Big Data maneja conjuntos de datos enormes que crecen tan rápido que se vuelve muy difícil manipular y analizar a una granularidad tal donde los procesos colapsan. Esta nueva tecnología no solo viene a resolver los problemas de almacenamiento y gestión que plantean las redes sociales, sino que también auxilia a otros sectores que también presentaban las mismas dificultades como el científico, el médico, el mercadológico, etc.

Tipos de Datos

Existen tres tipos de datos en el ambiente de Big Data:

- **Estructurados:** Son datos que tienen bien definidos su tipo, ya sea texto, numérico, fecha, etc. Por lo general estos datos son almacenados en tablas en un sistema de base de datos relacional.

- No estructurados: Son datos que conservan el formato con el que fueron recolectados, carecen de un formato específico. No se pueden almacenar dentro de una tabla ya que no se puede desgarnar su información a tipos básicos de datos. Algunos ejemplos son los PDF, documentos multimedia, e-mails, etc.
- Semi estructurados: Son datos que no se limitan a tipos determinados, pero que contiene marcadores para separar los diferentes elementos. Es información poco regular como para ser gestionada de una forma estándar. Estos datos poseen sus propios metadatos semiestructurados que describen los objetos y las relaciones entre ellos y pueden acabar siendo aceptados por convención. Algunos ejemplos son HTML, XML y JSON.

Almacenamiento NoSQL

El término NoSQL significa Not Only SQL y son sistemas de almacenamiento que no cumplen con el esquema entidad-relación. Proveen un sistema de almacenamiento mucho más flexible y concurrente y permiten manipular grandes cantidades de información de manera mucho más rápida que las bases de datos relacionales.

Existen cuatro tipos de almacenamiento NoSQL:

- 1 **Almacenamiento Clave-Valor (Key-Value)**. Son sistemas de almacenamiento donde se accede al dato a partir de una clave única. Los valores son aislados e independientes entre ellos y no son interpretados por el sistema. Pueden ser enteros, caracteres u objetos. Por otro lado, este sistema de almacenamiento carece de una estructura de datos clara y establecida, por lo que no requiere un formateo de los datos muy estricto. Son útiles para operaciones simples basadas en claves. Un ejemplo es el aumento de velocidad de carga de un sitio web que pueden utilizar diferentes perfiles de usuario, teniendo mapeados los archivos que hay que incluir según el id de usuario y que han sido calculados con anterioridad. Cassandra es la tecnología de almacenamiento clave-valor más reconocida por los usuarios.
- 2 **Almacenamiento Documental**. Bases de datos con este sistema de almacenamiento guardan un gran parecido con las bases de datos Clave-Valor, diferenciándose en el dato que guardan. Si en la anterior no requería una estructura de datos concreta, en este caso si se guardan datos semiestructurados. Estos datos pasan a llamarse documentos, y pueden estar formateados en XML, JSON, o en el formato que acepte la misma base de datos. Un ejemplo de este tipo de almacenamiento es un blog: se almacena el autor, la fecha, el título, el resumen y el contenido del post. CouchDB o MongoDB son las bases de datos documentales más conocidas.
- 3 **Almacenamiento en Grafo**. Las bases de datos en grafo rompen con la idea de tablas y se basan en la teoría de grafos, donde se establece que la información son los nodos y las relaciones entre la información son las aristas. Relacionan grandes cantidades de datos que pueden ser muy variables. Por ejemplo, los nodos pueden contener objetos, variables y atributos diferentes unos de los otros. Las uniones se sustituyen por recorridos a través del grafo, y se guarda una lista de adyacencias entre los nodos. Un ejemplo es el Facebook, donde cada usuario es un nodo que puede tener aristas de amistad con otros usuarios, o aristas de publicación con nodos de contenidos. Soluciones como Neo4J y GraphDB son las más conocidas dentro de las bases de datos en grafo.

- 4 **Almacenamiento Orientado a Columnas.** Este sistema de almacenamiento es similar al Documental. Su modelo de datos es definido como “un mapa de datos multidimensional poco denso, distribuido y persistente. Se orienta a almacenar datos con tendencia a escalar horizontalmente, por lo que permite guardar diferentes atributos y objetos bajo una misma Clave. A diferencia del Documental y el KeyValue, en este caso podremos almacenar varios atributos y objetos, pero no serán interpretables directamente por el sistema. Permite agrupar columnas en familias y guardar la información cronológicamente, mejorando el rendimiento. Esta tecnología se utiliza en casos de contar 100 o más atributos por clave. Su precursor es BigTable de Google, pero han aparecido nuevas soluciones como HBase o HyperTable.

Inteligencia Artificial

Los dispositivos que cuentan con inteligencia artificial pueden ejecutar distintos procesos análogos al comportamiento humano, como la devolución de una respuesta por cada entrada (similar a los reflejos de los seres vivos), la búsqueda de un estado entre todos los posibles según una acción o la resolución de problema mediante una lógica formal. Cuando se otorga a estos dispositivos la habilidad de aprender y de discernir, se les convierte en entidades que rozan las capacidades de un superhombre, dado que alcanzan velocidades de procesamiento imposibles para los humanos, y no necesitan descansar para funcionar, entre otras ventajas que los ubican por sobre los seres vivos en este contexto. [3]

Machine Learning

El Aprendizaje Automático consiste en una disciplina de las ciencias informáticas, relacionada con el desarrollo de la Inteligencia Artificial, y que sirve, como ya se ha dicho, para crear sistemas que pueden aprender por sí solos. Es una tecnología que permite hacer automáticas una serie de operaciones con el fin de reducir la necesidad de que intervengan los seres humanos. Esto puede suponer una gran ventaja a la hora de controlar una ingente cantidad de información de un modo mucho más efectivo.

Lo que se denomina aprendizaje consiste en la capacidad del sistema para identificar una gran serie de patrones complejos determinados por una gran cantidad de parámetros.

Es decir, la máquina no aprende por sí misma, sino un algoritmo de su programación, que se modifica con la constante entrada de datos en la interfaz, y que puede, de ese modo, predecir escenarios futuros o tomar acciones de manera automática según ciertas condiciones. Como estas acciones se realizan de manera autónoma por el sistema, se dice que el aprendizaje es automático, sin intervención humana.

¿Cómo funciona el Machine Learning?

En la informática clásica, el único modo de conseguir que un sistema informático hiciera algo era escribiendo un algoritmo que definiera el contexto y detalles de cada acción.

En cambio, los algoritmos que se usan en el desarrollo del Machine Learning realizan buena parte de estas acciones por su cuenta. Obtienen sus propios cálculos según los datos que se recopilan en el sistema, y cuantos más datos obtienen, mejores y más precisas serán las acciones resultantes.

Las computadoras se programan a sí mismas, hasta cierto punto, usando dichos algoritmos. Estos funcionan como ingenieros que pueden diseñar nuevas respuestas informáticas, como respuesta a la información que se les suministra a través de su interfaz u otros medios. Todo nuevo dato se convierte en un nuevo algoritmo, y a más datos, mayor complejidad y efectividad de cálculo puede proporcionar el sistema informático.

Tipos de Machine Learning

Un sistema informático de Aprendizaje Automático se sirve de experiencias y evidencias en forma de datos, con los que comprender por sí mismo patrones o comportamientos. De este modo, puede elaborar predicciones de escenarios o iniciar operaciones que son la solución para una tarea específica. Existen **tres tipos principales** de Aprendizaje Automático:

1. Aprendizaje supervisado:

Este tipo de aprendizaje se basa en lo que se conoce como información de entrenamiento. **Se entrena al sistema** proporcionándole cierta cantidad de datos definiéndolos al detalle con etiquetas. Por ejemplo, proporcionando a la computadora fotos de perros y gatos con etiquetas que los definen como tales. Una vez que se le ha proporcionado la suficiente cantidad de dichos datos, podrán introducirse nuevos datos sin necesidad de etiquetas, en base a patrones distintos que ha venido registrando durante el entrenamiento. Este sistema se conoce como clasificación. Otro método de desarrollo del Aprendizaje Automático consiste en **predecir un valor continuo**, utilizando parámetros distintos que, combinados en la introducción de nuevos datos, permite predecir un resultado determinado. Este método se conoce como regresión.

2. Aprendizaje no supervisado:

En este tipo de aprendizaje no se usan valores verdaderos o etiquetas. Estos sistemas tienen como finalidad la **comprensión y abstracción** de patrones de información de manera directa. Este es un modelo de problema que se conoce como clustering. Es un método de entrenamiento más parecido al modo en que los humanos procesan la información.

3. Aprendizaje por refuerzo:

En la técnica de aprendizaje mediante refuerzo, los sistemas aprenden a partir de la experiencia. Como ejemplo se puede observar el comportamiento de un coche autónomo. Cuando el vehículo toma una decisión errónea, es penalizado, dentro de un sistema de registro de valores. Mediante dicho **sistema de premios y castigos**, el vehículo desarrolla una forma más efectiva de realizar sus tareas. [4].

Metodología para modelos de Machine Learning

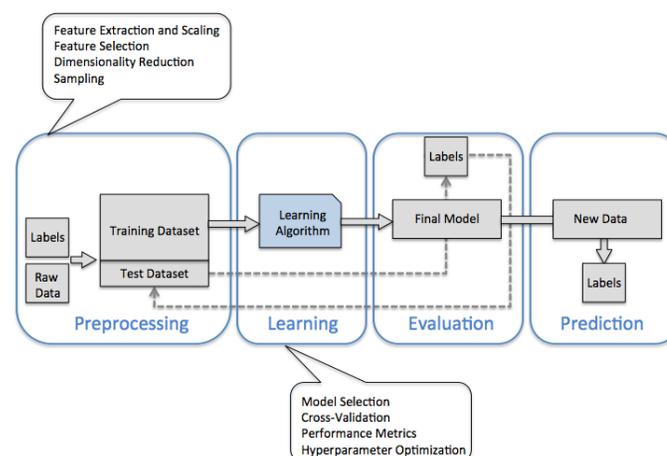


Imagen 5.4. Metodología General para construir modelos de Machine Learning

- **Preprocesamiento.** Este es uno de los pasos más importantes en cualquier aplicación de Machine Learning. Usualmente los datos se presentan en formatos no óptimos (o incluso inadecuados) para ser procesados por el modelo, por esa razón se deben de limpiar y depurar.
- **Entrenamiento y pruebas.** Es esencial comparar los diferentes algoritmos de un grupo para entrenar y seleccionar el de mejor rendimiento. Para realizar esto, es necesario seleccionar una métrica para medir el rendimiento del modelo.
- **Selección del modelo.** Se debe seleccionar el mejor modelo posible dependiendo de la serie de datos con los que se cuenten y el tipo de análisis a realizar.
- **Evaluación.** Consiste en la estimación de datos para ver si el modelo seleccionado es el más óptimo o si debe de cambiar [5].

Series Temporales

Por definición, una serie temporal es una sucesión de observaciones de una variable realizadas a intervalos regulares de tiempo. Según realicemos la medida de la variable considerada podemos distinguir distintos tipos de series temporales:

- a) **Discretas o Continuas,** en base al intervalo de tiempo considerado para su medición.
- b) **Flujo o Stock.** En Economía, se dice que una serie de datos es de tipo flujo si está referida a un período determinado de tiempo (un día, un mes, un año, etc.).

6. METODOLOGÍA DE INVESTIGACIÓN

Como resultado del proceso de trabajo se instalaron 7 estaciones de monitoreo de calidad del aire que se construyeron por parte de la Universidad Tecnológica de El Salvador y la Escuela Especializada en Ingeniería ITCA-FEPADE, las cuales envían datos de temperatura ambiente y de calidad del aire PM 2.5 y PM 10 a un servidor en la nube y a una hoja electrónica de Google Sheets como respaldo. En el servidor se tiene el servicio de Big Data con MongoDB y una aplicación multiplataforma encargada de mostrar los valores enviados por las estaciones en modo gráfico y dinámico según requerimientos del usuario.

Finalmente, se desarrolló un algoritmo de Machine Learning del tipo de Series de tiempo para la predicción de posibles estados de la calidad del aire en la región. Para el desarrollo de la aplicación se contó con la ayuda de un grupo de alumnos destacados de las carreras de Ingeniería en Desarrollo de Software y Técnico en Ingeniería de Sistemas Informáticos de ITCA-FEPADE. La metodología empleada es la que se describe a continuación [6]:

- **Investigación Técnica.** Permitió la recolección de la información necesaria para determinar qué tecnologías ayudarían al desarrollo de la aplicación con conexión a base de datos Big Data y sobre todo, qué tipo de algoritmo de Machine Learning emplear para la predicción de variables que manejan las estaciones de monitoreo. En esta fase, el equipo de trabajo de ITCA-FEPADE coordinó esfuerzos con docentes investigadores de la UTEC para obtener los requerimientos funcionales del proyecto.



Imagen 6.1. Docentes investigadores de ITCA-FEPADE y UTEC en reunión para establecer requerimientos funcionales del proyecto.

- **Establecimiento de socio estratégico.** En esta etapa ya se contaba con el socio estratégico para el proyecto, ya que en una etapa inicial la UTEC había iniciado con el diseño y construcción del prototipo de las estaciones de monitoreo [1], contando con el asocio con el Ministerio de Medio Ambiente y Recursos Naturales de El Salvador (MARN).

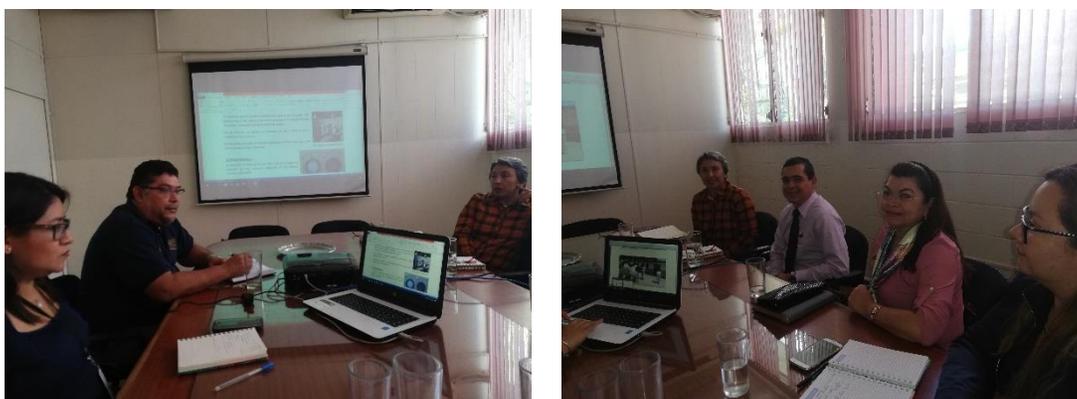


Imagen 6.2. Técnicos especialistas del Ministerio de Medio Ambiente y Recursos Naturales con personal de ITCA-FEPADE durante reunión de establecimiento de necesidades para el proyecto de investigación.

- **Capacitación de estudiantes participantes.** Se seleccionaron en total a 3 estudiantes de las carreras de Ingeniería en Desarrollo de Software y Técnico en Ingeniería de Sistemas Informáticos, ambas carreras pertenecientes a la Escuela Especializada en Ingeniería ITCA-FEPADE quienes apoyaron en el diseño de la base de datos en Big Data, desarrollo de la aplicación multiplataforma y configuración del servidor. Para seleccionar a los alumnos participantes se elaboró un perfil que debían de cumplir como requisitos mínimos (Ver anexo 1). Se impartieron capacitaciones en Big Data por parte de ITCA-FEPADE y de Machine Learning por parte de la UTEC. También se contó con la capacitación en programación y armado de dispositivos de IoT por parte de UTEC. Esta última fue impartida tanto a estudiantes como docentes investigadores en el proyecto.

- Por parte de ITCA-FEPADE se contó con el apoyo de los ingenieros Morris Díaz y Eduardo Amaya quienes apoyaron en la parte electrónica para las estaciones que aportó la institución.



Imagen 6.3. Alumnos de ITCA-FEPADE y UTEC que recibieron capacitación de Big Data por parte de Licda. Lilian Sandoval de ITCA-FEPADE.



Imagen 6.4. Alumnos de UTEC que recibieron capacitación de Big Data.



Imagen 6.5. Alumnos de ITCA-FEPADE que recibieron capacitación de Big Data.



Imagen 6.6. Docentes de ITCA-FEPADE Sede Central y Centro Regional MEGATEC Zacatecoluca que recibieron capacitación de Big Data por parte de ITCA-FEPADE.



Imagen 6.7. Docentes de ITCA-FEPADE Sede Central recibiendo capacitación sobre diseño, montaje y programación de estaciones de monitoreo por parte de Ing. Omar Flores e Ing. Rony Cortez de UTEC.



Imagen 6.8. Docentes de ITCA-FEPADE Sede Central en prácticas de ensamblado y programación de estaciones de monitoreo.

- Recopilación de datos.** En esta etapa también se trabajó en conjunto con la UTEC para establecer los sitios de interés en donde se alojarían las estaciones de monitoreo, así como las condiciones óptimas para su resguardo y conectividad con el servidor. Se visitaron las instalaciones de ITCA-FEPADE como son Sede Central Santa Tecla, Centro Regional MEGATEC Zacatecoluca, y Centro Regional San Miguel en un inicio. Posteriormente se cambió la ubicación del MEGATEC Zacatecoluca por el Centro Regional MEGATEC La Unión a recomendación del personal del MARN por considerarse una zona de bajo nivel de contaminante del aire, por su cercanía con Honduras y por la carencia de datos de esa zona.



Imagen 6.9. Docentes de ITCA-FEPADE y UTEC en establecimiento de requerimientos para montaje de estación de monitoreo de calidad del aire.

- Análisis y desarrollo del proyecto.** Establecidos los requerimientos funcionales para los dispositivos electrónicos y la aplicación multiplataforma, se procedió con el diseño del circuito según indicaciones por parte del Ing. Omar Flores de UTEC. Se construyeron las 2 estaciones de monitoreo y se instalaron en los Centros Regionales de San Miguel y MEGATEC La Unión. Por parte de UTEC se instaló una estación en la Sede Central de ITCA-FEPADE.

Se diseñó la aplicación, la base de datos y el algoritmo de Machine Learning, los cuales fueron integrados con las estaciones de monitoreo.



Imagen 6.10. Docentes investigadores de ITCA-FEPADE y alumnos participantes durante revisión de avances en desarrollo del proyecto.

- **Desarrollo y prueba de los módulos.** Construidas e instaladas las estaciones y programado el aplicativo y el algoritmo de Machine Learning se realizaron las pruebas de conectividad entre todos los módulos involucrados con el servidor configurado por ITCA-FEPADE para tal fin.
- **Implementación.** Consiste en la preparación de todo el ambiente de producción en el Ministerio de Medio Ambiente y Recursos Naturales. Se establecen políticas de seguridad dentro del servidor donde estará alojada la aplicación y al algoritmo de Machine Learning. Este proceso es en conjunto entre todas las instituciones involucradas en el proyecto (MARN, ITCA-FEPADE y UTEC).

Con el afán de contribuir al fomento del uso de aplicaciones tecnológicas dentro de instituciones públicas o privadas y de brindar un aporte a la sociedad, habiendo terminado con el desarrollo, prueba e implementación del proyecto, se procedió a llevar a cabo la *Difusión del mismo* para socializar el resultado obtenido.

A continuación, se presenta un esquema básico que muestra el desarrollo metodológico de la investigación:

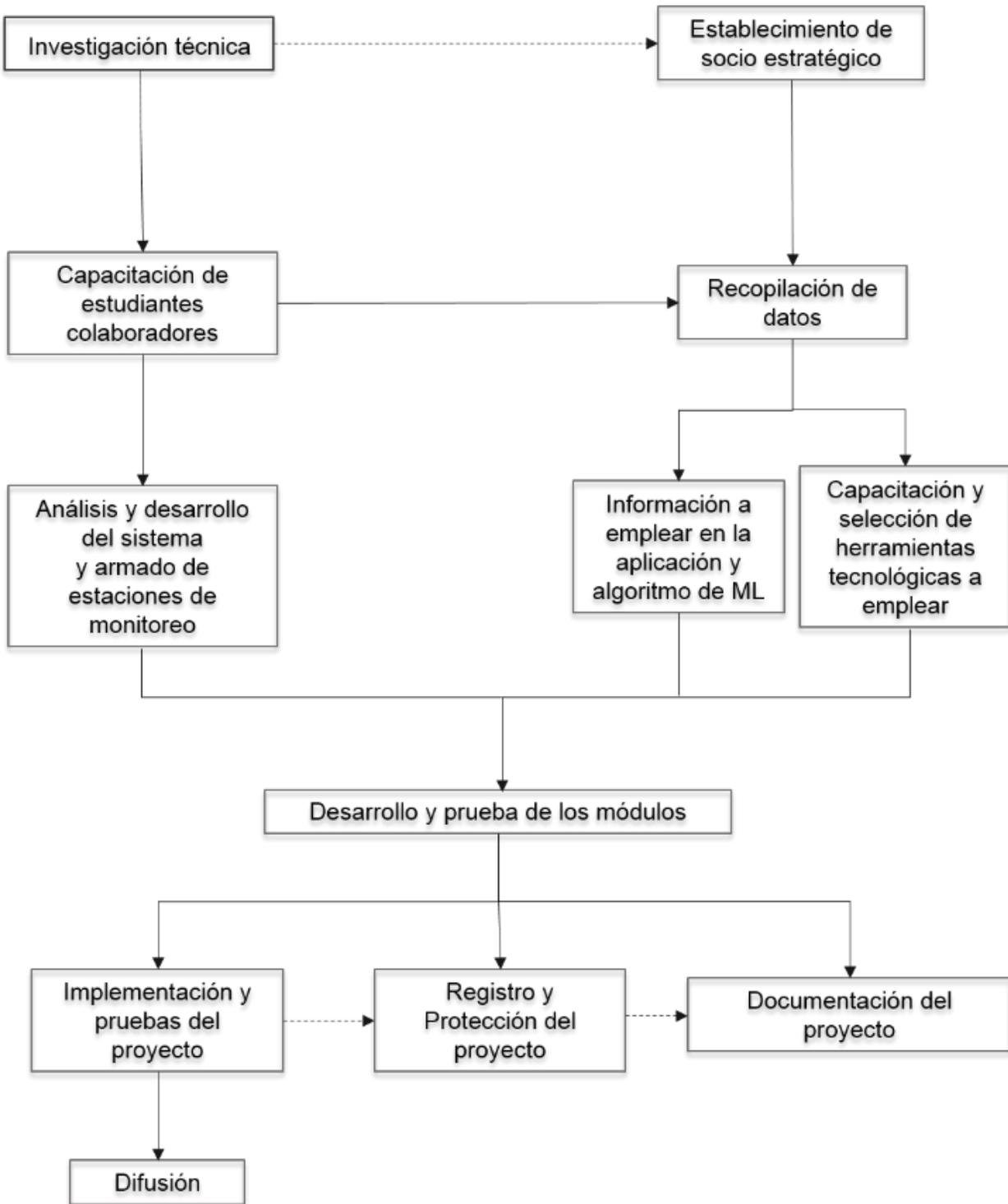


Imagen 6.11. Esquema metodológico de la investigación desarrollada.

6.1. PREPARACIÓN DEL SERVIDOR

La aplicación, base de datos y algoritmo de Machine Learning estarán alojados en un servidor Linux, con base de datos MongoDB.

6.1.1. Requerimientos mínimos del servidor

1. 2 GB de RAM.
2. Procesador de doble núcleo (2 GHz).
3. 25 GB de espacio libre en el disco duro.
4. Conexión a internet.
5. Privilegios de administrador.
6. Sistema operativo Ubuntu o Centos.

6.1.2. Requerimientos mínimos del cliente

- Acceso a internet.
- PC de escritorio, laptop, dispositivo móvil o tableta.
- Un navegador (De preferencia Mozilla o Chrome).

6.1.3. Configuración de servidor LAMP (Linux, Apache, MongoDB, Php)

LAMP es un acrónimo de **Linux, Apache, MongoDB, PHP**. Es una pila popular para crear y desplegar aplicaciones web dinámicas. En esta pila (stack), Linux sirve como el sistema operativo para la aplicación web. MongoDB se utiliza como base de datos. Apache se utiliza como servidor web. PHP se utiliza para procesar contenido dinámico. En algunas otras variantes de esta pila, Perl se utiliza en lugar de PHP o Python. En otras pilas se suele utilizar MySQL como base de datos; y es que, este es la stack más famosa de todas, pero debido a que en nuestro proyecto se está haciendo uso de bases de datos no relacionales, utilizaremos MongoDB como gestor. Para el proyecto se ha utilizado Ubuntu 18.04 como versión del sistema operativo Linux [7].

Instalación de Apache

- a. Abrir una terminal.
- b. Actualizar el administrador de paquetes del sistema:

```
sudo apt update
```

- c. Instalar Apache2:

```
sudo apt install apache2
```

- d. Mostrar configuraciones disponibles:

```
sudo ufw app list
```

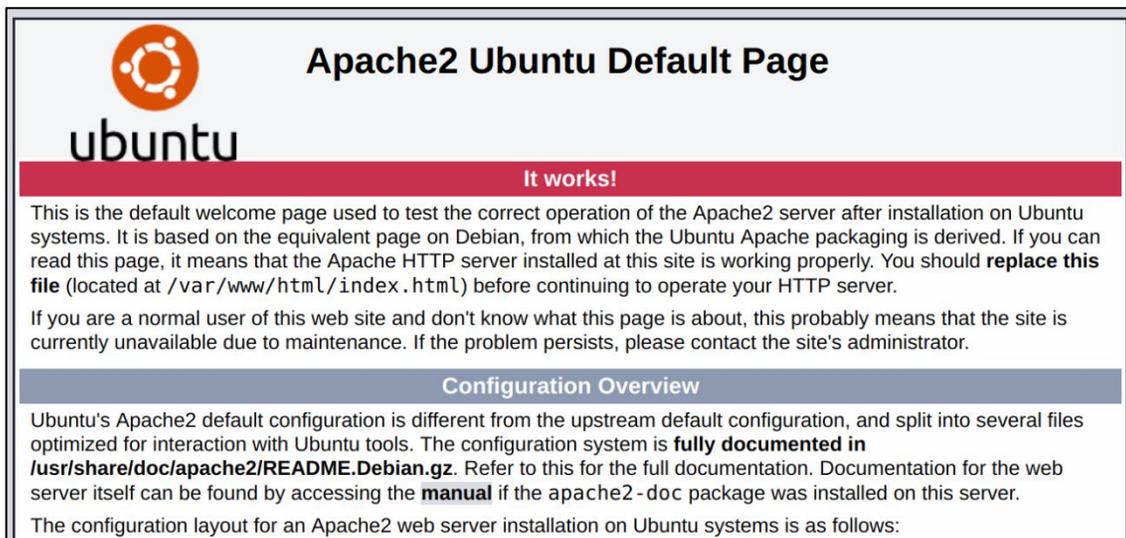
Se mostrará una pantalla como esta:

```
aire@aire-VirtualBox: ~
File Edit View Search Terminal Help
(base) aire@aire-VirtualBox:~$ sudo ufw app list
available applications:
Apache
Apache Full
Apache Secure
CUPS
(base) aire@aire-VirtualBox:~$
```

e. Seleccionar "Apache Full" con el siguiente comando:

```
sudo ufw app info "Apache Full"
```

f. Acceder desde un navegador a la URL "http://localhost" y verificar que se muestra la siguiente pantalla:



Instalación de PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación interpretado que se utiliza para la generación de páginas web de forma dinámica. Éste código se ejecuta al lado del servidor y se incrusta dentro del código HTML. Cabe destacar que es un lenguaje de código abierto, gratuito y multiplataforma.

a. Abrir una terminal.

b. Instalar dependencias de PHP:

```
sudo apt install php libapache2-mod-php
```

c. En la mayoría de los casos, se deseará modificar la forma mediante la cual Apache sirve archivos cuando un directorio es solicitado. En este momento, si un usuario solicita un directorio del servidor, Apache buscará, en primera instancia, un archivo llamado index.html. Nosotros queremos que el servidor web les dé prioridad a los archivos PHP sobre cualquier otro archivo. Para lo cual haremos que el Apache busque el archivo **index.php** en primer lugar:

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

d. Sustituir todo por las siguientes líneas:

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

e. Presionar **Ctrl + X** y luego **'Y'** para guardar los cambios.

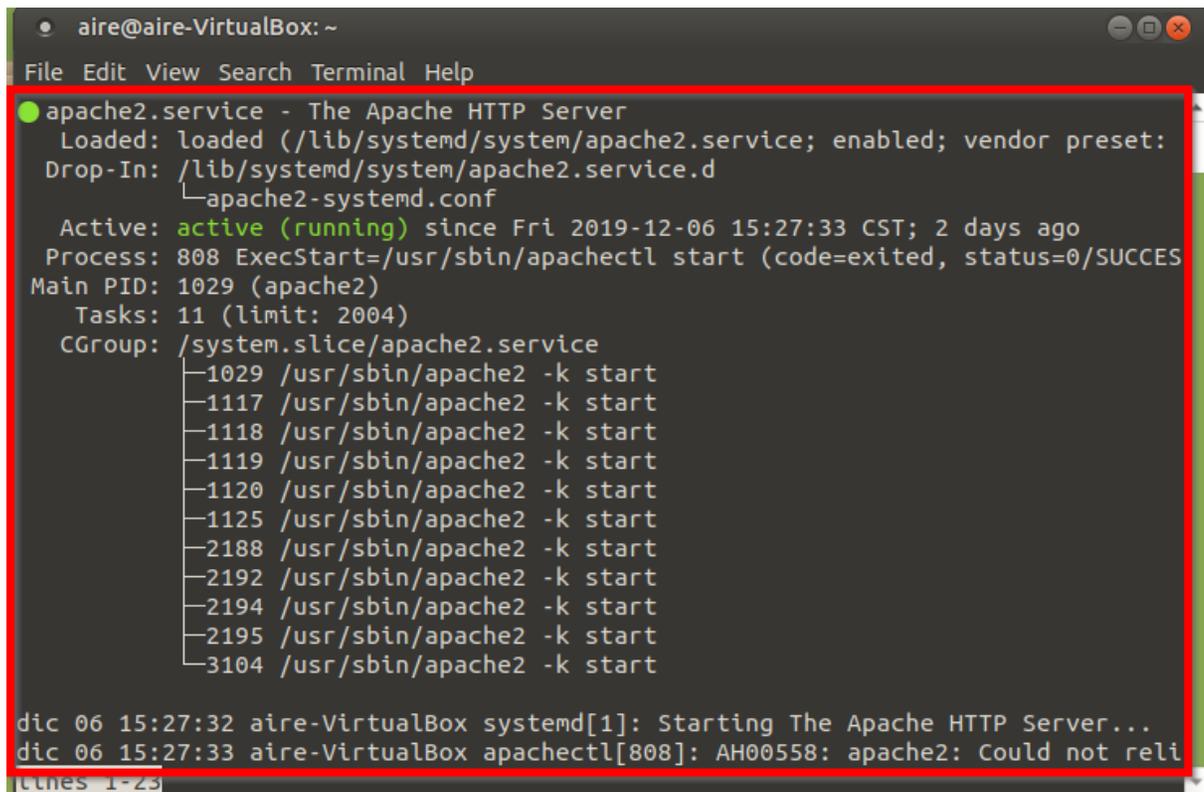
f. Reiniciar el servidor Apache:

```
sudo systemctl restart apache2
```

g. Verificar el estado de Apache para estar seguros de que el servidor se está ejecutando:

```
sudo systemctl status apache2
```

Verificar que se muestre un estado similar al de la imagen:



```
aire@aire-VirtualBox: ~
File Edit View Search Terminal Help
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Fri 2019-12-06 15:27:33 CST; 2 days ago
   Process: 808 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1029 (apache2)
     Tasks: 11 (limit: 2004)
    CGroup: /system.slice/apache2.service
            └─1029 /usr/sbin/apache2 -k start
              1117 /usr/sbin/apache2 -k start
              1118 /usr/sbin/apache2 -k start
              1119 /usr/sbin/apache2 -k start
              1120 /usr/sbin/apache2 -k start
              1125 /usr/sbin/apache2 -k start
              2188 /usr/sbin/apache2 -k start
              2192 /usr/sbin/apache2 -k start
              2194 /usr/sbin/apache2 -k start
              2195 /usr/sbin/apache2 -k start
              3104 /usr/sbin/apache2 -k start

dic 06 15:27:32 aire-VirtualBox systemd[1]: Starting The Apache HTTP Server...
dic 06 15:27:33 aire-VirtualBox apachectl[808]: AH00558: apache2: Could not reli
LINES 1-23
```

h. Instalar PHP-CLI:

```
sudo apt install php-cli
```

i. Verificación del funcionamiento de PHP dentro del servidor. Crear un nuevo archivo PHP con el siguiente comando:

```
sudo nano /var/www/html/info.php
```

j. Ingresar las siguientes líneas de código que mostrarán toda la información de PHP:

```
<?php
phpinfo();
?>
```

k. Ingresar la siguiente URL en un navegador: <http://localhost/info.php>

Verificar que se muestra la siguiente información en pantalla



PHP Version 7.2.3-1ubuntu1	
System	Linux LAMP-1804-test 4.15.0-15-generic #16-Ubuntu SMP Wed Apr 4 13:58:14 UTC 2018 x86_64
Build Date	Mar 14 2018 22:03:58
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-intl.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-xmldrpc.ini

Instalación de MongoDB

MongoDB es un sistema de base de datos NoSQL orientado a documentos de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. Los pasos a seguir son [8]:

- Abrir una terminal.
- Actualizar e instalar MongoDB:

```
sudo apt update
sudo apt install -y mongodb
```

- Comprobamos que el servicio se esté ejecutando de forma correcta:

```
sudo systemctl status mongodb
```

Verificar que se muestra un resultado similar al de la imagen:

```
aire@aire-VirtualBox: ~  
File Edit View Search Terminal Help  
(base) aire@aire-VirtualBox:~$ sudo systemctl status mongodb  
● mongodb.service - An object/document-oriented database  
  Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset:  
  Active: active (running) since Fri 2019-12-06 15:27:32 CST; 2 days ago  
    Docs: man:mongod(1)  
  Main PID: 803 (mongod)  
    Tasks: 29 (limit: 2004)  
   CGroup: /system.slice/mongodb.service  
           └─803 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/m  
dic 06 15:27:32 aire-VirtualBox systemd[1]: Started An object/document-oriented  
lines 1-10/10 (END)
```

Instalación de R

R es un entorno y lenguaje de programación con un enfoque al análisis estadístico. Se trata de uno de los lenguajes de programación más utilizados en investigación científica, siendo además muy popular en los campos de aprendizaje automático (Machine Learning), minería de datos, investigación biomédica, bioinformática y matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y graficación.

- a. Abrir una terminal e ingresar los siguientes comandos:

```
sudo apt update sudo
```

```
apt -y install r-base
```

Instalación de Anaconda

Anaconda es una distribución libre y abierta de los lenguajes Python y R [9], utilizada en ciencia de datos, y aprendizaje automático (machine learning). Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software. Pasos para instalar Anaconda [10]:

- a. Descargar Anaconda de la siguiente página web:

```
curl -O https://repo.anaconda.com/archive/Anaconda3-5.3.0-Linux-x86_64.sh
```

- b. Luego de haber descargado el archivo, ingresar a la carpeta de descargas:

```
cd Descargas
```

```
ls
```

```
Anaconda3-5.3.0-Linux-x86_64.sh
```

- c. Identificado el paquete, dentro de una terminal, digitar el siguiente comando:

```
sh Anaconda3-5.3.0-Linux-x86_64.sh
```

- d. Al ejecutarlo se mostrará el siguiente mensaje:

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Welcome to Anaconda3 5.3.0

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> |
```

- e. Presionar la tecla Enter para ver todos los términos de la licencia y finalmente digitar “yes” para aceptar y continuar con el proceso de instalación.
- f. Posteriormente se solicitará confirmar la ubicación de los archivos de instalación, para el presente caso usaremos los predeterminados por lo que simplemente se presiona la tecla Enter.
- g. A continuación, pregunta si se desea que el instalador inicialice Anaconda3 en nuestro bashrc a lo que en este caso diremos que sí, “yes”.
- h. Finalmente procedemos con la activación de la instalación de anaconda ejecutando el siguiente comando:

```
source ~/.bashrc
```

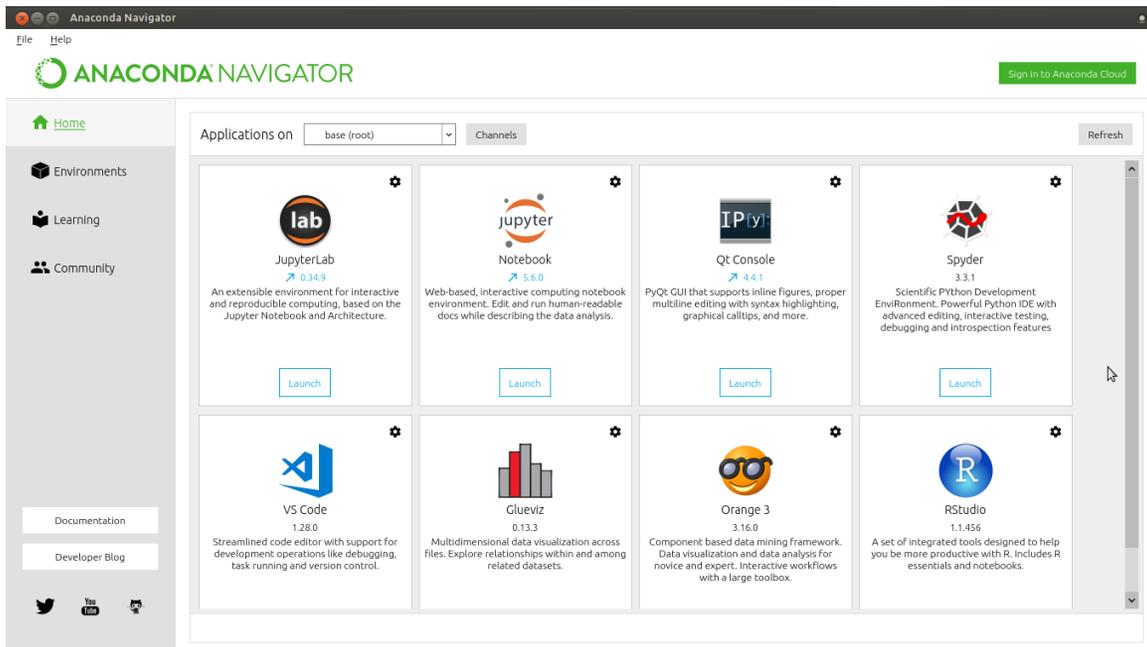
- i. Confirmar con el comando:

```
conda info
```

- j. Para ejecutar anaconda ingresar el siguiente comando desde la terminal:

```
anaconda-navigator
```

Verificar que se carga correctamente en el navegador:



Instalación de Laravel

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC. Los pasos para instalar Laravel son los siguientes [11]:

- a. Ingresar a una terminal y actualizar la paquetería del servidor:

```
sudo apt update && sudo apt upgrade
```

- b. Instalar curl en el caso de que no se encuentre en el servidor:

```
sudo apt install curl
```

- c. Instalar composer con el siguiente comando:

```
curl -s https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin -  
filename=composer
```

- d. Verificar la versión de composer:

```
composer --version
```

- e. Debería de mostrarse algo similar a esto:

```
Composer version 1.8.0 2018-12-03 10:31:16
```

- f. Crear el proyecto de Laravel con el siguiente comando:

```
composer create-project --prefer-dist laravel/laravel my_app
```

- g. El comando anterior buscará todos los paquetes php necesarios. El proceso puede tardar unos minutos y, si tiene éxito, el final de la salida debería tener el siguiente aspecto:

```
Package manifest generated successfully.
```

```
@php artisan key:generate --ansi
```

```
Application key set successfully.
```

- h. Abrir la carpeta del proyecto y ejecutar el servidor:

```
cd my_app
```

```
php artisan server
```

- i. La salida sería algo parecido a esto:

```
Laravel development server started: <http://127.0.0.1:8000>
```

- j. Ingresar la siguiente URL en un navegador y verificar que se carga la pantalla inicio de Laravel:

```
http://127.0.0.1:8000
```

Instalación y configuración del controlador de MongoDB para Laravel

Para utilizar MongoDB con Laravel, es necesario instalar el driver de MongoDB para que ambas tecnologías sean capaces de comunicarse entre sí [12].

- a. Abrir una terminal y ejecutar el siguiente comando:

```
sudo pecl install mongo
```

- b. Abrir el archivo php.ini y agregar la referencia del archivo de MongoDB:

```
sudo nano /etc/php/7.2/apache2/conf.d
```

- c. Agregar esta línea al final del archivo:

```
extension=mongo.so
```

Instalación y configuración de Laravel con Apache (Producción)

Para llevar el proyecto de Laravel a producción con Apache2 es necesario realizar las siguientes configuraciones.

- a. Verificar que todo este actualizado en el servidor:

```
sudo apt-get install update
```

```
sudo apt-get dist-upgrade
```

- b. Mover la carpeta del proyecto a `/var/www`, en este caso se encuentra en el escritorio:

```
sudo cp -r /home/aire/Desktop/aire /var/www/
```

- c. Habilitar el modo rewrite de apache:

```
sudo a2enmod rewrite
```

- d. Cambiar configuración del archivo de configuración por defecto de Apache:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

- e. En este archivo cambiar las siguientes configuraciones:

```
<VirtualHost *:80>
```

```
    # The ServerName directive sets the request scheme, hostname and port that
```

```
    # the server uses to identify itself. This is used when creating
```

```
    # redirection URLs. In the context of virtual hosts, the ServerName
```

```
    # specifies what hostname must appear in the request's Host: header to
```

```
    # match this virtual host. For the default virtual host (this file) this
```

```
    # value is not decisive as it is used as a last resort host regardless
```

```
    # However, you must set it for any further virtual host explicitly
```

```
    #ServerName www.example.com
```

```

ServerAdmin webmaster@localhost
DocumentRoot /var/www/aire/public
<Directory /var/www/aire/public>
Options Indexes FollowSymLinks MultiViews
AllowOverride all
Allow from all
RewriteEngine On
RewriteBase /var/www/aire/public
</Directory>

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example, the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

f. Reiniciar el servidor apache:

```
sudo service apache2 restart
```

g. Asignar los permisos a la carpeta storage:

```
sudo chown -R www-data: storage
```

```
sudo chown -R 775 storage
```

h. Cambiar configuración del archivo `.env`, que se encuentra en `/var/www/aire`:

```
sudo nano .env
```

i. Cambiar las siguientes configuraciones:

```
APP_ENV = production
```

```
APP_DEBUG = false
```

- j. Copiar el archivo `.htaccess` e `index.php` a la carpeta raíz del proyecto:

```
cd /var/www/html/aire/public
```

```
cp .htaccess /var/www/aire
```

```
cp index.php /var/www/aire
```

- k. Modificar el archivo `.htaccess` alojado en `/var/www/aire`:

```
sudo nano .htaccess
```

- l. Eliminar todo el contenido y agregar las siguientes líneas:

```
<IfModule mod_rewrite.c>
```

```
    RewriteEngine on
```

```
    RewriteRule ^(.*)$ public/$1 [L]
```

```
</IfModule>
```

- m. Renombrar el archivo `server.php` a `index.php`:

```
sudo mv server.php index.php
```

- n. Modificar `.htaccess` en la carpeta `public`:

```
<IfModule mod_rewrite.c>
```

```
    <IfModule mod_negotiation.c>
```

```
        Options -MultiViews
```

```
    </IfModule>
```

```
    RewriteEngine On
```

```
    # Redirect Trailing Slashes...
```

```
    RewriteRule ^(.*)/$ /$1 [L,R=301]
```

```
    # Handle Front Controller...
```

```
    RewriteCond %{REQUEST_FILENAME} !-d
```

```
    RewriteCond %{REQUEST_FILENAME} !-f
```

```
    RewriteRule ^ index.php [L]
```

```
</IfModule>
```

Desde un navegador ingresar la URL “`http://localhost`”; el proyecto debería estar ejecutándose correctamente.

NOTA: Ejecutar las siguientes configuraciones **ÚNICAMENTE** si al ingresar a “localhost” no se muestra el proyecto correctamente:

o. Deshabilitar *mpm event*

a2dismod mpm_event

p. Habilitar php7.2

a2enmod php7.2

q. Reiniciar apache2

systemctl restart apache2

r. Ingresar nuevamente al navegador y digitar la dirección “http://localhost” y el proyecto debería estar funcionando correctamente.

6.2. PLAN DE IMPLEMENTACIÓN

Para la implementación de este proyecto se trabajó en conjunto con la Unidad de Informática del Ministerio de Medio Ambiente y Recursos Renovables (MARN). Los pasos realizados fueron los siguientes:

1. Preparación de un servidor con salida a internet exclusivo para el alojamiento del módulo. Los requisitos que debe cumplir dicho servidor son:
 - Sistema Operativo: Linux (recomendable Ubuntu 18.04)
 - Versión de Apache: Apache 2.4.6
 - Versión de PHP: PHP 7.2 o superior
 - Versión de MongoDB Compass: 1.19.12
 - Laravel: 6.0 o superior.
 - R: 3.5.2
 - Espacio en Disco: 50 GB
2. Asignar una dirección DNS al servidor para poder utilizar con los usuarios. Por ejemplo: **https://aire.marn.gob.sv**
3. Asignar los permisos requeridos de escritura y lectura a las carpetas del proyecto donde se almacenará en el servidor (*/var/www/aire*).
4. Crear el usuario y permisos respectivos en el servidor para el acceso a la base de datos.
5. Montar la base de datos en el servidor.
6. Asignar una dirección IP de tipo pública a cada estación de monitoreo para el envío de datos al servidor de las mediciones obtenidas cada 20 min de PM 2.5, PM 10 y Temperatura.
7. Realizar pruebas de integración y funcionalidad en la aplicación y las estaciones de monitoreo.

8. Socializar el proyecto con el resto de Unidades del Ministerio de Medio Ambiente y población en general.
9. Capacitar al personal del MARN sobre el uso de la aplicación y mantenimiento de las estaciones de monitoreo.
10. Se hará entrega en un acto formal de los instaladores del proyecto, así como la documentación técnica requerida para el mismo.

6.3. PLAN DE CAPACITACIONES

Las capacitaciones estarán sujetas a petición y coordinación del Ministerio de Medio Ambiente y Recursos Naturales. Se definirán fechas y tiempos para poderse impartir.

Objetivo general:

Capacitar a todo el personal seleccionado del Ministerio de Medio Ambiente en el uso y mantenimiento del módulo informático y estaciones de monitoreo del proyecto.

Alcances:

La capacitación estará orientada al uso y mantenimiento del módulo informático del proyecto y a las estaciones de monitoreo instaladas en las distintas zonas del país, específicamente con el personal de la Unidad de Informática del Ministerio de Medio Ambiente y Recursos Naturales.

Actividades a realizar:

1. Capacitar al personal de Informática en la instalación y configuración del módulo informático.
2. Capacitar al personal en la instalación y mantenimiento de las estaciones de monitoreo tanto de hardware como software.
3. Explicar la metodología de programación empleada al personal técnico de informática, tanto en la aplicación como en las estaciones de monitoreo.

7. RESULTADOS

Los resultados obtenidos al finalizar el proyecto fueron los siguientes:

7.1. ESTACIONES DE MONITOREO DE CALIDAD DEL AIRE

Se construyeron e instalaron 8 estaciones de monitoreo de calidad del aire en total, dos de las cuales son aporte de ITCA-FEPADE y están ubicadas en los Centros Regionales de San Miguel y MEGATEC La Unión de ITCA-FEPADE. El resto de estaciones son proporcionadas por UTEC. Las ubicaciones de las estaciones son:

Estación ITCA-FEPADE (Sede Central)



Imagen 7.1.1. Vista satelital de la ubicación de las instalaciones de ITCA-FEPADE Sede Central ubicada en Carr. Panamericana, Km 11 ½ Santa Tecla (Lat: 13.673875, Lon: 89.279146). Fuente: Google Earth



Imagen 7.1.2. Docentes de ITCA-FEPADE y UTEC configurando estación de monitoreo. Ing. Elvis Martínez e Ing. Omar Flores



Imagen 7.1.3. Docentes de ITCA-FEPADE y alumnos colaborando en instalación de la estación.



Imagen 7.1.4. Estación de monitoreo instalada en Edificio "C" de ITCA-FEPADE.

Estación ITCA-FEPADE (Sede Regional San Miguel)



Imagen 7.1.5. Vista satelital de la ubicación de las instalaciones de ITCA-FEPADE San Miguel ubicada en Ruta De Paz, San Miguel (**Lat:** 13.500032, **Lon:** -88.163438). Fuente: Google Earth



Imagen 7.1.6. Docentes de ITCA-FEPADE en la búsqueda del lugar donde se instalará la estación de monitoreo en Centro Regional San Miguel de ITCA-FEPADE.

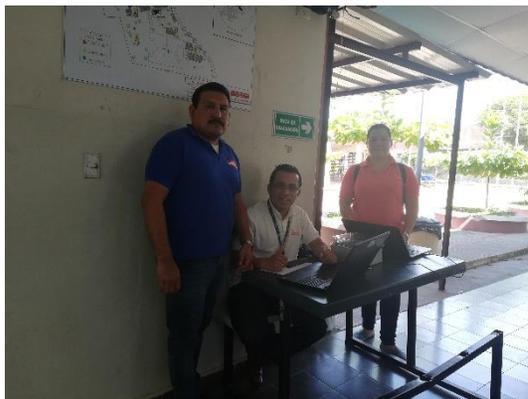


Imagen 7.1.7. Docentes de ITCA-FEPADE Sede Central en proceso de configuración de estación de monitoreo. Ing. Elvis Martínez, Lic. Oswaldo Hernandez e Inga. Rina López



Imagen 7.1.8. Estación de monitoreo instalada en ITCA-FEPADE Centro Regional San Miguel.

Estación ITCA-FEPADE (Sede Regional MEGATEC La Unión)



Imagen 7.1.9. Vista satelital de la ubicación de las instalaciones de ITCA-FEPADE La Unión ubicada en Calle Santa María, La Unión (**Lat:** 13.339447, **Lon:** -87.858472). Fuente: Google Earth



Imagen 7.1.10. Personal de mantenimiento, soporte técnico y docentes investigadores de ITCA-FEPADE Sede Central y MEGATEC La Union en la búsqueda del lugar donde se instalará la estación de monitoreo.



Imagen 7.1.11. Estación de monitoreo instalada en ITCA-FEPADE Centro Regional MEGATEC La Unión.

Estación UTEC (San Salvador)



Imagen 7.1.12. Vista satelital de la ubicación de las instalaciones de Edificio de Vicerrectoría de Investigación y Proyección Social de UTEC ubicada en Calle Arce y 19 Av. Sur, San Salvador (**Lat:** 13.699783, **Lon:** -89.201084). Fuente: Google Earth

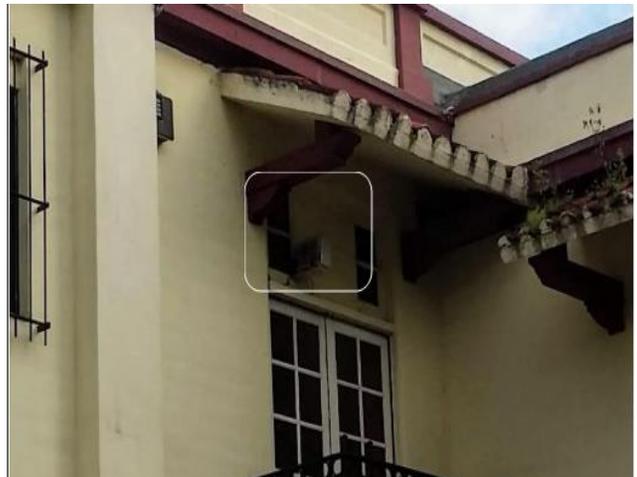


Imagen 7.1.13. Instalación en fachada exterior lateral Edificio de Investigaciones, Campus Central UTEC, San Salvador. Fuente: <https://sites.google.com/mail.utec.edu.sv/estacion-pm-utec/>



Imagen 7.1.14. Estación de monitoreo instalada en Campus Central UTEC. Fuente: <https://sites.google.com/mail.utec.edu.sv/estacion-pm-utec/>

Estación Alcaldía Municipal de Santa Ana



Imagen 7.1.15. Vista satelital de la ubicación de las instalaciones de la Alcaldía Municipal de Santa Ana ubicada en Avenida Independencia Norte y Calle Libertad Poniente, Santa Ana (**Lat:** 13.9947285, **Lon:** -89.5574126). Fuente: Google Earth



Imagen 7.1.16. Estación de monitoreo instalada en muro exterior del edificio de la Alcaldía Municipal de Santa Ana. Fuente: <https://sites.google.com/mail.utec.edu.sv/estacionsantaana>

Estación Instituto Nacional de Antiguo Cuscatlán



Imagen 7.1.17. Vista satelital de la ubicación de las instalaciones del Instituto Nacional de Antiguo Cuscatlán ubicada en Ave. Antiguo Cuscatlán, La Libertad (**Lat:** 13.675338, **Lon:** -89.241452). Fuente: Google Earth



Imagen 7.1.18. Estación de monitoreo instalada en Instituto Nacional de Antiguo Cuscatlán. Ing. Rony Cortez supervisa el montaje. Fuente: <https://sites.google.com/mail.utec.edu.sv/calidad/aire/p%C3%A1gina-principal>

Estación Colegio Adventista (Chalatenango)



Imagen 7.1.19. Vista satelital de la ubicación de las instalaciones del Colegio Adventista ubicado en Calle Prof. Juan Ramón Mena, Chalatenango (**Lat:** 14.043241, **Lon:** -88.937142). Fuente: Google Earth



Imagen 7.1.20. Estación de monitoreo instalada en muro exterior del Colegio Adventista, Chalatenango. Fuente: <https://sites.google.com/mail.utec.edu.sv/calidad/aire/p%C3%A1gina-principal>

Estación Universidad de El Salvador (San Salvador)



Imagen 7.1.21. Vista satelital de la ubicación de las instalaciones de la Universidad de El Salvador ubicado en Final 25 Avenida Norte, San Salvador (Lat: 13.7161024, Lon: -89.2056577). Fuente: Google Earth



Imagen 7.1.22. Estación de monitoreo instalada en muro exterior del Edificio de Ingeniería Eléctrica, Facultad de Ingeniería y Arquitectura, San Salvador.

Fuente:

<https://sites.google.com/mail.utec.edu.sv/calidadair/e/p%C3%A1gina-principal>

7.2. APLICACIÓN WEB PARA EL MONITOREO DE VARIABLES AMBIENTALES

La aplicación desarrollada, es capaz de ejecutarse en cualquier tipo de pantalla, que van desde teléfonos inteligentes, tabletas o hasta computadoras de escritorio. La descripción de la funcionalidad de las pantallas del aplicativo son las siguientes:

PANTALLA DE INICIO

En esta pantalla se muestra al usuario, información valiosa para entender la forma en que la calidad del aire es medida; asimismo, definiciones y como la contaminación de este afecta a los seres vivos.

Calidad del aire

Inicio Nosotros Iniciar Sesión

Calidad del aire

¿Sabías qué?

- En 2016, el 93% de la población está en lugares donde no se respetaron las Directrices de la OMS sobre la calidad del aire.
- Cuanto más rápido sean los niveles de contaminación del aire mejor será la salud cardiovascular y respiratoria de la población, tanto a largo como a corto plazo.
- Un 93% de esas defunciones prematuras se producen en países de bajos y medianos ingresos, y las mayores tasas de mortalidad se registran en las regiones de Asia Sudoriental y el Pacífico Occidental de la OMS.



Índice	Interpretación
Buena	Sin impactos sobre la salud.
Satisfactoria	Sin impactos sobre la salud.
No satisfactoria	Dañino para niños y personas con enfermedades respiratorias, tales como asma.
No saludable	Personas con enfermedades respiratorias, tales como asma, deben evitar salir al aire libre.
Dañino para la salud	Personas con enfermedades respiratorias, tales como asma, deben evitar estar al aire libre.

Partículas (PM)

Definición y fuentes principales



Las PM son un indicador representativo común de la contaminación del aire. Afectan a más personas que cualquier otro contaminante. Los principales componentes de las PM son los sulfatos, los nitratos, el amoníaco, el carbono de soot, el hollín, los polvos minerales y el agua. Consisten en una compleja mezcla de partículas sólidas y líquidas, de sustancias orgánicas e inorgánicas suspendidas en el aire. Si bien las partículas con un diámetro de 10 micrones o menos (a PM₁₀) pueden penetrar y alojarse profundamente dentro de los pulmones, existen otras partículas aún más dañinas para la salud, que son aquellas con un diámetro de 2,5 micrones o menos (a PM_{2.5}). Las PM_{2.5} pueden atravesar la barrera pulmonar y entrar en el sistema sanguíneo (a exposición crónica a partículas) y contribuir al riesgo de desarrollar enfermedades cardiovasculares y respiratorias, así como cáncer de pulmón.

Generalmente, las mediciones de la calidad del aire se notifican como concentraciones medias diarias o anuales de partículas PM₁₀ por metro cúbico (m³) de aire. Las mediciones científicas de la calidad del aire describen esas concentraciones de PM expresadas en microgramos (µg/m³). Cuando se dispone de instrumentos de medición suficientemente sensibles, se notifican también las concentraciones de partículas finas (PM_{2.5} o más pequeñas).

Más información

[¿Quieres saber cómo está la calidad del aire en nuestro país? ¡Clícalo aquí!](#)

Copyright © 2019 MARN | ITCA-FEPADE. Todos los derechos reservados.

Pantalla de créditos

Se muestran los integrantes que han sido parte del equipo para llevar a cabo este proyecto y una breve descripción de cómo nace la idea.

Calidad del aire

Inicio Nosotros Iniciar Sesión



Proyecto de Investigación
ITCA-FEPADE | UTEC

Coordinadores de la Investigación

Ing. Elvis Marías Martínez Pérez (Investigador principal)
Ing. Rina Elzabeth López (Co-Investigadora)
Ing. Ronny Cortez (Investigador principal)
Ing. Dinaer Damiel Flores (Co-Investigador)

Equipo de Investigación ITCA-FEPADE

Leonel Antonio López Valencia
Manuel Alexander Delgado Hernández
Jorge Daniel Ángel Fernández

Copyright © 2019 MARN | ITCA-FEPADE. Todos los derechos reservados.

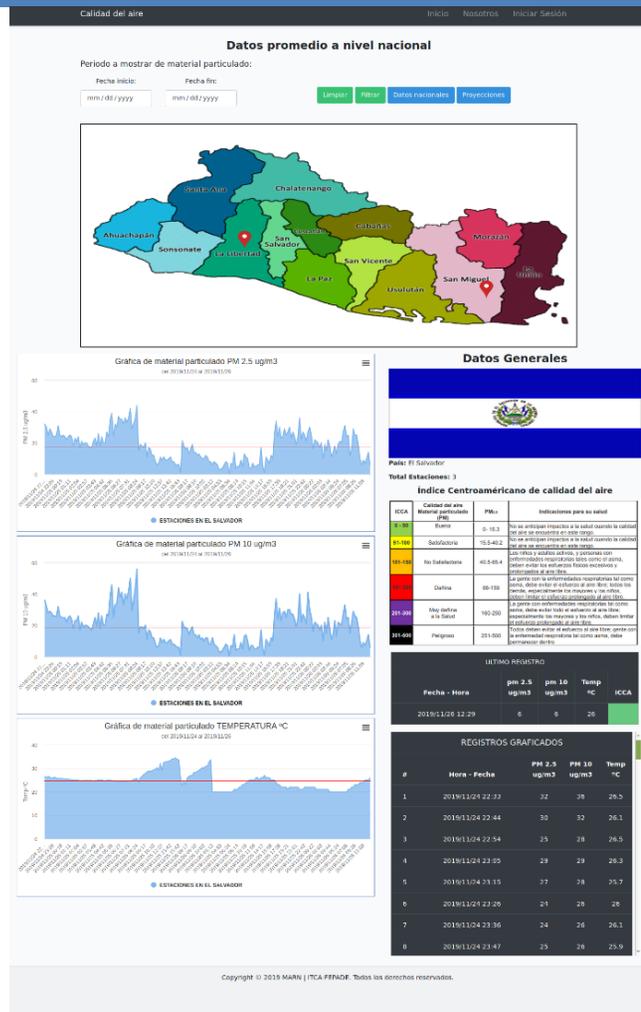
Mapa de distribución de las estaciones

Dentro de esta área se muestran todas las estaciones disponibles con información recopilada por cada uno de los dispositivos. Al pasar el puntero por encima de una estación, se desplegará un pequeño cuadro el cual le dará toda la información de la estación. De esta manera para el usuario mucho más fácil saber cuál de las estaciones es la que desea que se le muestre.



Pantalla de datos

Esta pantalla consta de distintas áreas que son esenciales para el funcionamiento de la aplicación (Mapa de distribución de las estaciones, Área de filtro por fechas, Gráficas, Datos Generales y Tabla de Datos). En la imagen se muestran resultados de medición a nivel promedio de todas las estaciones para el país, pero también se puede obtener información individual por estación.



Mostrar una estación específica

Esta opción le permite al usuario mostrar la información recopilada por una estación en particular. Para llevar a cabo esta acción debe seleccionar el marcador de la estación que desee y éste, automáticamente se actualizará con los datos que han sido recopilados a la fecha; asimismo, cambiará el título mostrándole la información general de la estación seleccionada.

Área de filtro por fechas

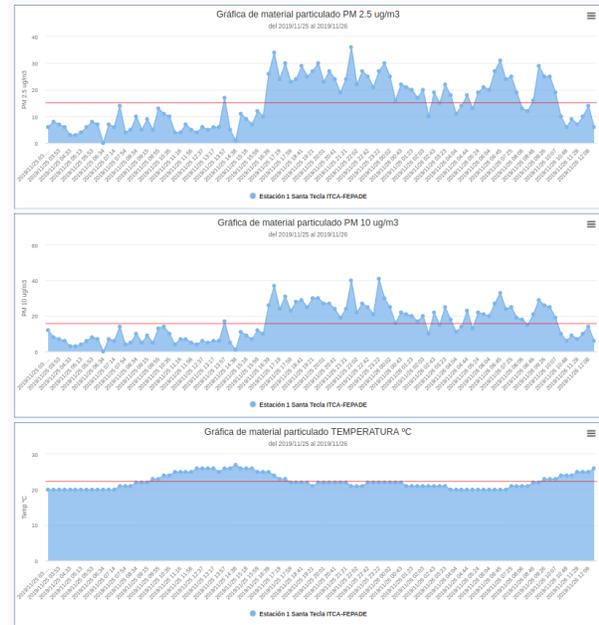
Esta opción le permite filtrar entre fechas específicas, para realizar esta acción debe de ingresar las fechas de inicio y de fin entre las cuales se requiere que la información se muestre. Al haber ingresado los datos necesarios, presionar el botón de filtrar y toda la información será actualizada con los nuevos parámetros.

Área de información

Esta área le muestra la información de la estación seleccionada. En el caso de que no haya ninguna estación seleccionada se mostrará una bandera de El Salvador indicándole que no hay ninguna estación seleccionada y que la información mostrada es recopilada por todas las estaciones a nivel nacional.

Mapa de distribución de las estaciones

En esta sección se muestran cada uno de los datos obtenidos de forma gráfica; para que el usuario sea capaz de ver de una forma sencilla, los cambios del particulado a lo largo del tiempo.



Sección de calidad del material particulado

En esta sección se muestran las categorías del ICCA (Índice Centroamericano de la calidad del aire) y la forma en la que la contaminación del aire afecta a la salud. En la siguiente tabla se muestra el último valor de los registros junto con sus datos respectivos; además, de su ICCA.

Índice Centroamericano de calidad del aire				
ICCA	Calidad del aire Material particulado (PM)	PM _{2.5}	Indicaciones para su salud	
0 - 50	Buena	0- 15.3	No se anticipan impactos a la salud cuando la calidad del aire se encuentra en este rango.	
51-100	Satisfactoria	15.5-40.2	No se anticipan impactos a la salud cuando la calidad del aire se encuentra en este rango.	
101-150	No Satisfactoria	40.5-65.4	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar los esfuerzos físicos excesivos y prolongados al aire libre.	
151-200	Dañina	66-159	La gente con la enfermedades respiratorias tal como asma, debe evitar el esfuerzo al aire libre; todos los demás, especialmente los mayores y los niños, deben limitar el esfuerzo prolongado al aire libre.	
201-300	Muy dañina a la Salud	160-250	La gente con enfermedades respiratorias tal como asma, debe evitar todo el esfuerzo al aire libre; especialmente los mayores y los niños, deben limitar el esfuerzo prolongado al aire libre.	
301-500	Peligroso	251-500	Todos deben evitar el esfuerzo al aire libre; gente con la enfermedad respiratoria tal como asma, debe permanecer dentro	

ULTIMO REGISTRO				
Fecha - Hora	pm 2.5 ug/m3	pm 10 ug/m3	Temp °C	ICCA
2019/11/26 12:29	6	6	26	50

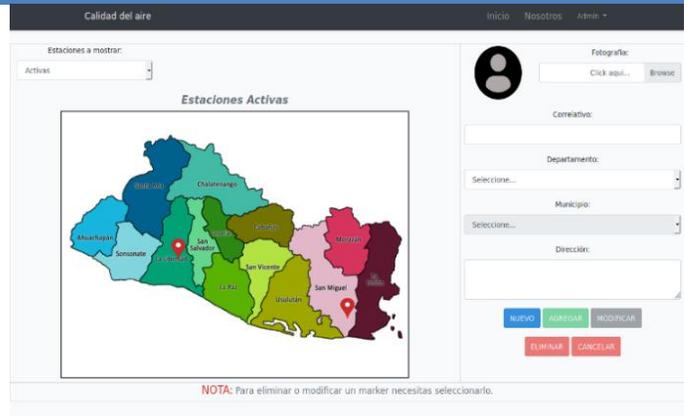
Tabla de datos

En la tabla de datos se especifican cada uno de los valores que se han obtenido de la estación en cuestión. Se muestran la fecha y hora a la que fueron ingresados, el valor de PM 2.5, PM10 y la temperatura.

REGISTROS GRAFICADOS				
#	Hora - Fecha	PM 2.5 ug/m3	PM 10 ug/m3	Temp °C
1	2019/11/25 03:12	6	12	20
2	2019/11/25 03:32	8	8	20
3	2019/11/25 03:53	7	7	20
4	2019/11/25 04:13	6	6	20
5	2019/11/25 04:33	3	3	20
6	2019/11/25 04:53	3	3	20
7	2019/11/25 05:13	4	4	20
8	2019/11/25 05:33	6	6	20

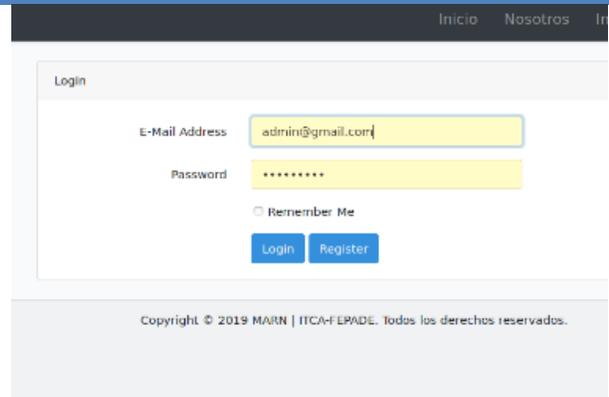
Pantalla de administración de estaciones

Esta pantalla le permite al usuario crear nuevas estaciones, modificar las existentes (ya sea información o cambiando de lugar la estación) o deshabilitar las existentes. Para acceder a esta pantalla es requisito que tenga las credenciales de acceso como administrador.



Iniciar sesión

El inicio de sesión es únicamente para usuarios administradores. Al identificarse dentro del sistema se podrán gestionar las estaciones y de igual forma la consulta de datos.



Habilitar una estación

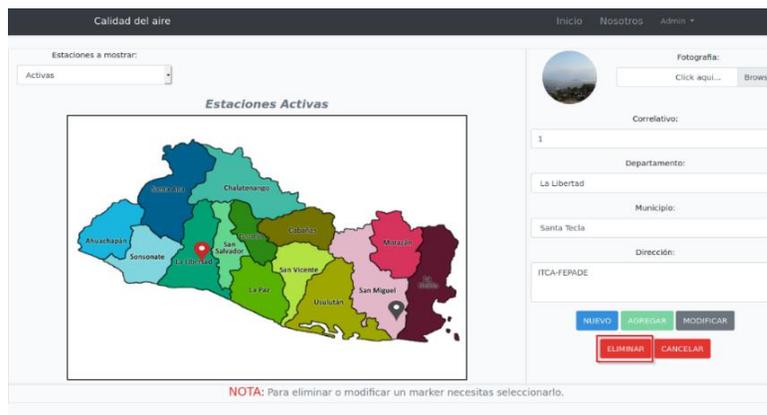
Esta sección, permite habilitar una estación que este eliminada o inactiva. Basta con elegir el parámetro "Inactivas" y las estaciones con este estado se mostraran en el mapa, seleccionar el marcador respectivo y presionar el botón restaurar.



Deshabilitar una estación

Si se desea deshabilitar una estación, se debe de seleccionar su marcador respectivo en el mapa y luego presionar el botón “Eliminar”.

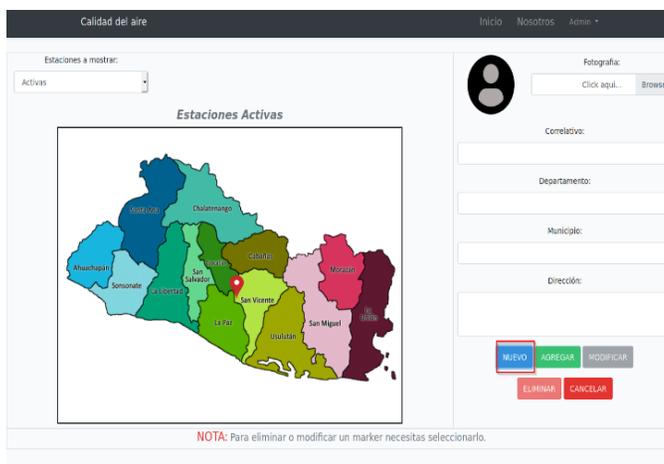
Es importante tener en cuenta que, si se deshabilita la estación, la información de esta no se elimina por completo y puede habilitarse de nuevo.



Crear o modificar una nueva estación

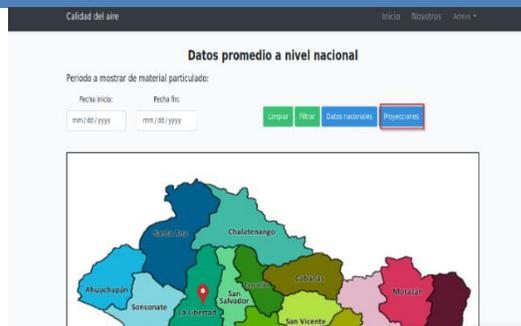
Para crear una nueva estación se debe de presionar el botón “Nuevo”. Este le habilitará cada uno de los campos necesarios para que sean llenados con la información requerida. Al terminar presionar el botón “Agregar” para guardar los cambios. Si desea cancelar esta acción presione el botón de “Cancelar”.

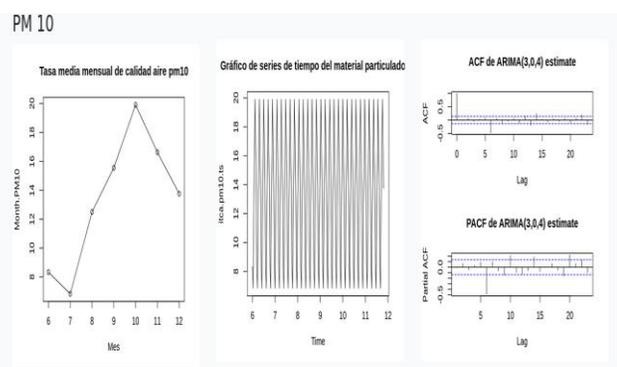
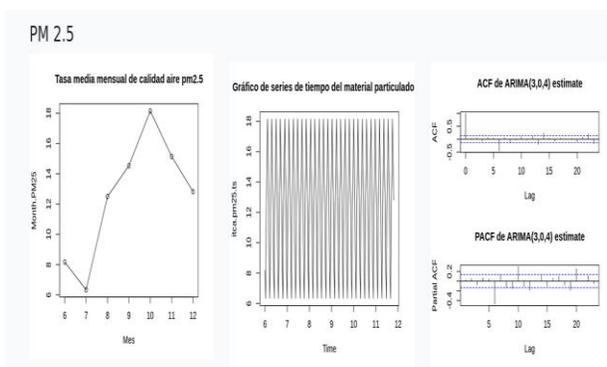
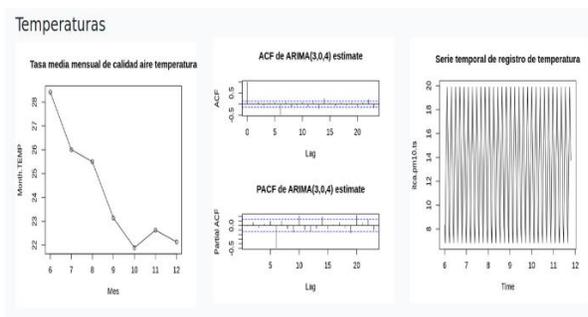
Si se requiere modificar algún valor de una estación, solo se selecciona su marcador, los datos son cargados en el formulario, actualizar el dato requerido y presionar el botón “Modificar”.



Proyecciones

Para mostrar las predicciones, basta con solo presionar el botón “Proyecciones” y este le mostrará cada una de estas en el estado actual de los últimos 30 min. A su vez, se le mostrarán los promedios del material particulado y la temperatura del aire





Ejemplo de gráficas con la implementación de algoritmo de Machine Learning para la predicción de variables PM 2.5, PM 10 y Temperatura.

7.3. ALGORITMO DE PREDICCIÓN DE MACHINE LEARNING

Para la realización del algoritmo se utilizó el lenguaje de programación R, mediante un framework de desarrollo llamado Júpiter. Cada uno de los pasos de la solución se detallan a continuación:

a) Análisis del Problema

La calidad del aire en El Salvador es un tema de interés nacional, ya que actualmente no se cuenta con mecanismos que permitan su fácil conocimiento y análisis de los datos, por lo que este algoritmo deberá permitir analizar la fuente de datos provenientes de estaciones de monitoreo de calidad del aire.

b) Entendimiento de los Datos

Con el fin de analizar y tratar de predecir la calidad del aire en Santa Tecla, se analizaron los datos provenientes de una estación de monitoreo, la cual mide material particulado **2.5** y **10**, así como temperatura, además se describen las variables utilizadas:

Fecha Hora: tipo fecha y hora, con mediciones programadas cada 20 min.

PM25: numérica decimal, 8.0,8.8,9.0, etc.

PM10: numérica decimal, 8.0,8.8,9.0, etc.

Temp: numérica decimal, 8.0,8.8,9.0, etc.

Mes: numérica entera del 1 al 12.

Cabe mencionar que el algoritmo se ha realizado con datos proveniente solamente de la estación instalada en ITCA-FEPADE Sede Central y de la cual solo se han tomado las variables Fecha Hora y Material particulado 2.5. Se cuentan con datos desde el mes de junio a diciembre de 2019.

c) Carga y Exploración de datos

```
## Lee el archivo csv con el nombre de registro_pmitca.csv.
unemp.data <- read.csv ('registro_pmitca.csv', header = TRUE, stringsAsFactors =
FALSE)

# Obtener información general sobre el conjunto de datos
str(unemp.data)

# Índice de calidad del aire PM2.5 Se extrae del conjunto de datos las
# variables pm2.5 y Mes
Month.PM25 <- sapply(split(unemp.data$PM25, unemp.data$Mes), mean)
Mes = c(6:12)

# Se grafican las variables para ver la media mensual de calidad de aire,
# con la variable pm2.5
plot(Mes, Month.PM25, main = "Tasa media mensual de calidad aire pm2.5")
lines(Mes, Month.PM25)
```

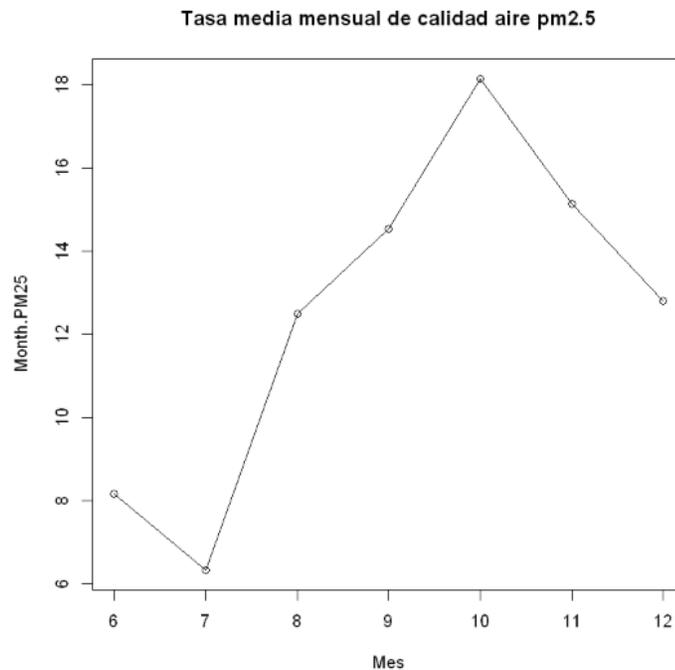


Imagen 7.3.1. Tasa Media Mensual de Material Particulado 2.5.

d) Preparación del conjunto de datos para su análisis.

Este conjunto de datos tiene casi 2500 observaciones recolectadas cada 20 min provenientes de la estación de monitoreo de calidad del aire ubicada en la Sede Central de ITCA-FEPADE, por lo que para su análisis se tomó a bien realizar conjuntos de datos más pequeños por mes.

```
# Dividir en conjuntos de datos más pequeños clasificados por mes desde junio
# a diciembre

sub.6 <- subset(unemp.data, unemp.data[,6] == 6)
sub.7 <- subset(unemp.data, unemp.data[,6] == 7)
sub.8 <- subset(unemp.data, unemp.data[,6] == 8)
sub.9 <- subset(unemp.data, unemp.data[,6] == 9)
sub.10 <- subset(unemp.data, unemp.data[,6] == 10)
sub.11 <- subset(unemp.data, unemp.data[,6] == 11)
sub.12 <- subset(unemp.data, unemp.data[,7] == 12)

#Preparar los datos para el análisis de series de tiempo
# Tasa de calidad del aire mensual promedio ordenada por mes para los datos
# de estación de ITCA.

mean6 <- sapply(split(sub.6$PM25, factor(sub.6$Mes, ordered=TRUE)), mean)
mean7 <- sapply(split(sub.7$PM25, factor(sub.7$Mes, ordered=TRUE)), mean)
mean8 <- sapply(split(sub.8$PM25, factor(sub.8$Mes, ordered=TRUE)), mean)
mean9 <- sapply(split(sub.9$PM25, factor(sub.9$Mes, ordered=TRUE)), mean)
mean10 <- sapply(split(sub.10$PM25, factor(sub.10$Mes, ordered=TRUE)), mean)
mean11 <- sapply(split(sub.11$PM25, factor(sub.11$Mes, ordered=TRUE)), mean)
mean12 <- sapply(split(sub.12$PM25, factor(sub.12$Mes, ordered=TRUE)), mean)

#Crear un marco de datos de tasas medias mensuales para el año 2019

mean.monthly.PM25 =
data.frame('Mes'=c(rep('06',30),rep('07',30),rep('08',30),rep('09',30),rep('10',30),rep('11',
30),rep('12',30)),
'PM25' = c(mean6,mean7,mean8,mean9,mean10,mean11,mean12))

#Las primeras filas de este conjunto de datos

head(mean.monthly.PM25)

str(mean.monthly.PM25)
```

```

Mes PM25
1 06 8.166667
2 06 6.333333
3 06 12.500000
4 06 14.533333
5 06 18.149096
6 06 15.142495

'data.frame': 210 obs. of 2 variables:
 $ Mes : Factor w/ 7 levels "06","07","08",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ PM25: num 8.17 6.33 12.5 14.53 18.15 ...

```

Imagen 7.3.2. Primeras mediciones de material particulado pm2.5 del Set de datos.

TRABAJANDO CON SERIES DE TIEMPO EN R

Se instalan y cargan los paquetes necesarios en este caso es el paquete **repr** y **forecast** que se utilizaron para tratar las series de tiempos.

```

install.packages(repr)
install.packages(forecast)
options(warn=-1)
# Cargar paquetes y funciones requeridos
require(repr)
require(forecast)

```

Creación de grafico en R con las series de tiempo cargadas en el DataSet.

```

#Convierte a series de tiempo usando la función ts
itca.pm25.ts <- ts(mean.monthly.PM25[,2], start = c(6,1), freq =36)

#Gráfico the time series
options(repr.pmales.extlot.width=8, repr.plot.height=4)
plot(itca.pm25.ts, main = 'Gráfico de series de tiempo del material particulado 2.5')

options(repr.pmales.extlot.width=8, repr.plot.height=4)
plot(log(itca. pm25.ts), main = 'Serie temporal de registro del material particulado 2.5')

```

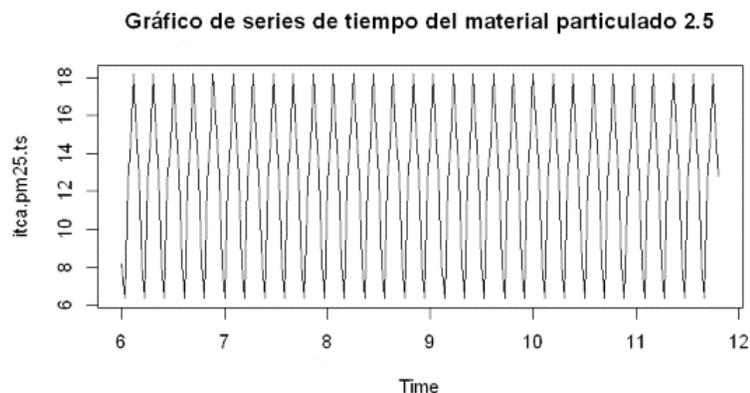


Imagen 7.3.3. Gráfico de series de tiempo de material particulado 2.5 en los meses de junio-diciembre 2019

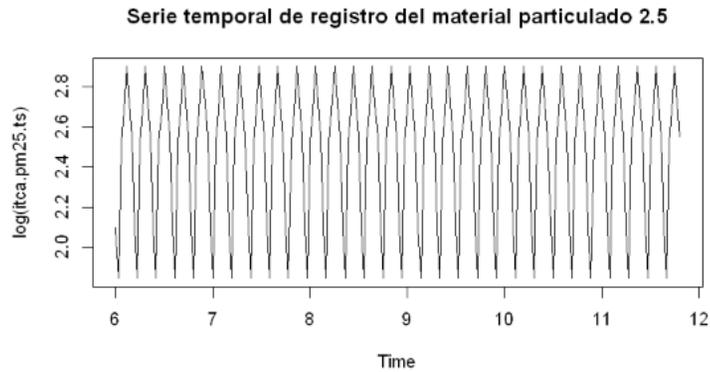


Imagen 7.3.4. Gráfico de serie temporal de registro del material particulado 2.5 en los meses de junio-diciembre 2019

Como se pueden observar ambas graficas muestran la variación de lo medidas de material particulado entre los meses de junio a noviembre, observando que la serie es no estacionaria es decir que no sufre variaciones con el tiempo.

Detección de patrones

Autocorrelación, también conocida como correlación en serie, es la correlación de una serie de tiempo con una copia retrasada de sí misma. Informalmente, es la similitud entre las observaciones en función del tiempo de retraso entre ellas. El análisis de la autocorrelación es una herramienta matemática para encontrar patrones repetitivos, como la presencia de periodicidad o estacionalidad.

También podemos definir una segunda orden **autocorrelación parcial**. La función de autocorrelación parcial (PACF) proporciona la correlación parcial de una serie de tiempo con sus propios valores rezagados, controlando los valores de la serie de tiempo en todos los rezagos más cortos. Contrasta con la función de autocorrelación, que no controla para otros retrasos. Esta función desempeña un papel importante en los análisis de datos dirigidos a identificar la extensión del retraso en un modelo autorregresivo (AR), que describiremos en breve.

Ahora, vamos a trazar el ACF y PACF de nuestros datos de material particulado 2.5.

```
#Funciones de autocorrelación y autocorrelación parcial
plot.acf <- function(df, col = 'remainder', is.df =TRUE)
{
  if(is.df) temp <- df[, col]
  else temp <- df
  par(mfrow = c(2,1))
  acf(temp, main = paste('ACF de', col))
  pacf(temp, main = paste('PACF de', col))
  par(mfrow = c(1,1))
}
options(repr.pmales.extlot.width=8, repr.plot.height=6)
plot.acf(log(itca.pm25.ts), col = 'Tasa de Material Particulado 2.5', is.df = F)
```

ACF de Tasa de Material Particulado 2.5

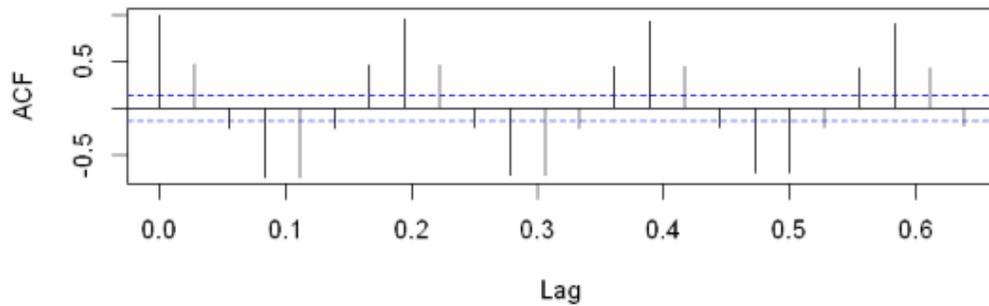


Imagen 7.3.5. Gráfico de autocorrelación ACF de material particulado 2.5

PACF de Tasa de Material Particulado 2.5

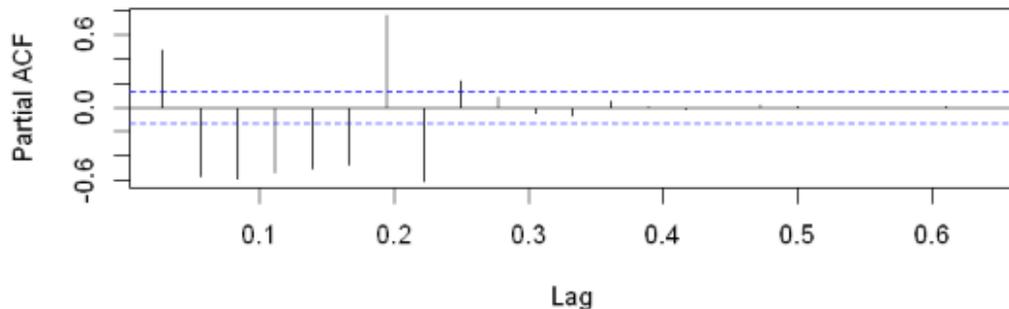


Imagen 7.3.6. Gráfico de autocorrelación parcial PACF de material particulado 2.5.

Las líneas de puntos azules en los gráficos ACF y PACF son los intervalos de confianza del 95% para los valores de correlación. Las líneas verticales muestran la correlación. Cualquier línea vertical que cruce las líneas punteadas azules indica un valor de correlación significativo. Un i.i.d. las series de tiempo normalmente distribuidas no tendrán valores significativos de ACF o PACF más allá del retraso 0. En este caso, el gráfico de ACF tiene varios retrasos significativos no cero y muestra cierto patrón de aumento y caída. También el ACF está disminuyendo o decayendo lentamente. La gráfica PACF muestra valores significativos para más de un retraso, lo que indica que podría haber una tendencia. Esto claramente no es una serie temporal estacionaria.

Descomposición de STL

A veces, la tendencia y los componentes cíclicos se agrupan en uno, llamado componente de tendencia-ciclo. El componente de ciclo de tendencia se puede denominar simplemente componente de "tendencia", aunque puede contener un comportamiento cíclico. Por ejemplo, una descomposición estacional de series de tiempo, conocida como **Seasonal & Trend** descomposición utilizando **Loess** o **STL**, descompone una serie de tiempo en componentes estacionales, de tendencia e irregulares utilizando **loess** y grafican los componentes por separado, por lo que el componente cíclico (si está presente en los datos) se incluye en el gráfico del componente de "tendencia".

Como su nombre indica, este modelo hace lo siguiente:

- La tendencia se elimina utilizando un modelo de regresión LOESS.
- El componente estacional se elimina mediante una regresión en componentes periódicos.
- El resto se conoce como el residual.

A continuación, se muestra la aplicación del algoritmo de descomposición STL.

```
# Función para la descomposición STL de una serie de tiempo en componentes
ts.decomp <- function(df, col = 'Series Name', span = 0.13, Mult = TRUE)
{
  if(Mult) temp = log(df) else temp = df
  spans = span * length(temp)
  fit <- stl(temp, s.window = "periodic", t.window = spans)
  plot(fit, main = paste('Decompositon of', col, 'with loess span = ', as.character(span)))
  fit$time.series
}

itca.pm25.decomp <- ts.decomp((itca.pm25.ts), col = 'Pm2.5 ITCA-FEPADE', Mult = FALSE)
str(itca.pm25.decomp)

#Plot the ACF & PACF of the remainder
options(repr.pmales.extlot.width=8, repr.plot.height=6)
plot.acf(itca.pm25.decomp[, 1], is.df = FALSE)
```

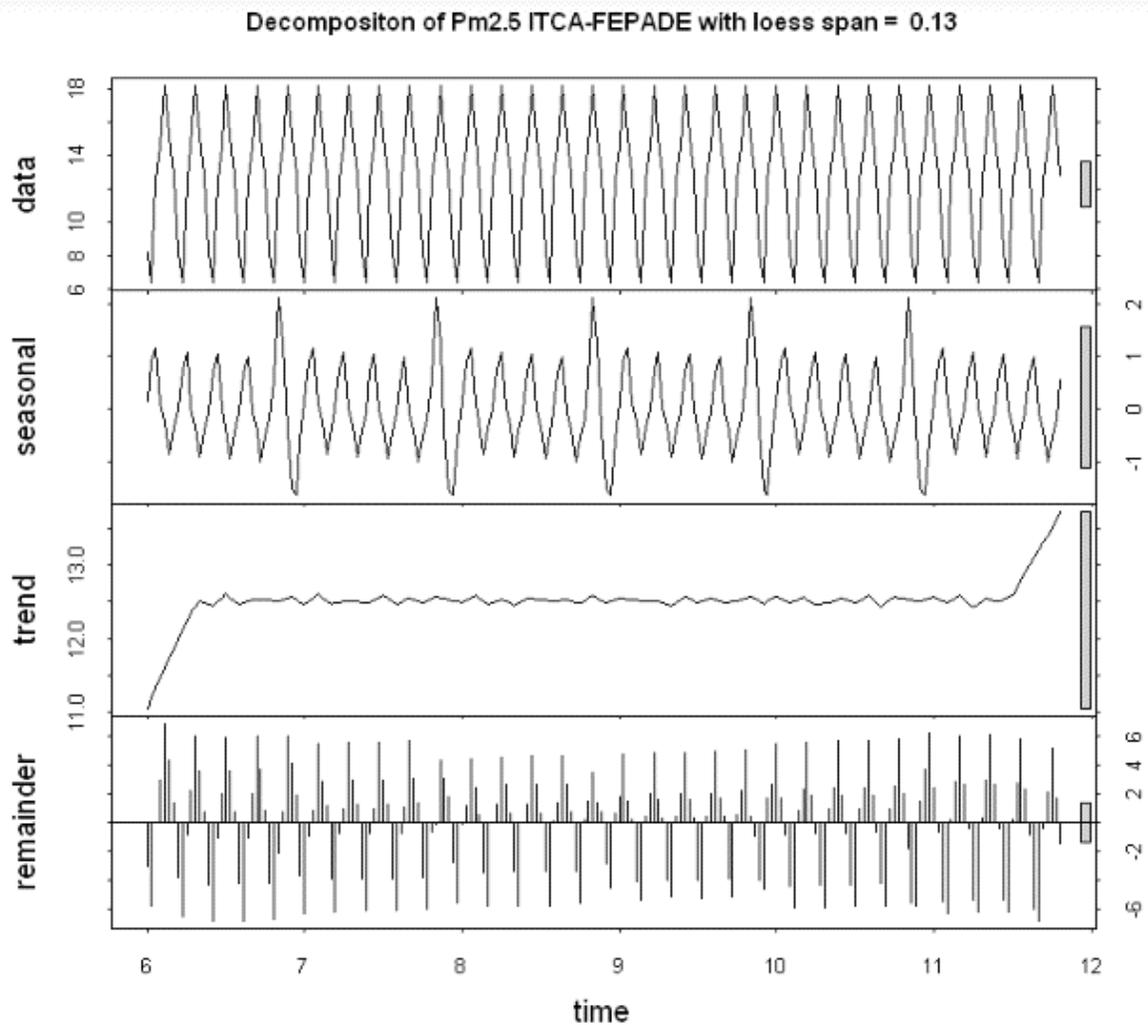


Imagen 7.3.7. Grafica de descomposición STL

Modelo ARIMA para la serie Residual

El modelo de media móvil integrada autorregresiva, o ARIMA agrega un término integrador al modelo ARMA. El componente de integración realiza la diferenciación para modelar un componente de paseo aleatorio.

El componente integrante modela una de las partes no estacionarias de una serie de tiempo. El modelo ARIMA está definido por órdenes p , d , q . El orden del operador de diferenciación del término integrador está definido por d .

Ahora, intentemos estimar los parámetros ARIMA (p , d , q) de la serie de tiempo (recuerde, esta es una serie de tiempo no estacionaria) e imprimimos un resumen del mejor modelo y trazamos el ACF y el PACF del residual.

```
#ARIMA model estimation for STL remainder
ts.model = function(ts, col = 'remainder', order = c(1,1,0))
{
```

```

mod = arima(ts, order = order, include.mean = FALSE)
print(mod)
mod
}

arima.estimate1 <- ts.model(itca.pm25.decomp[, 1], order = c(1,0,1))
#ARIMA(1,0,1) model
arima.estimate2 <- ts.model(itca.pm25.decomp[, 1], order = c(0,1,1))
#ARIMA(0,1,1) model
arima.estimate3 <- ts.model(itca.pm25.decomp[, 1], order = c(1,1,3))
#ARIMA(1,1,3) model
arima.estimate4 <- ts.model(itca.pm25.decomp[, 1], order = c(0,1,5))
#ARIMA(0,1,5) model
arima.estimate5 <- ts.model(itca.pm25.decomp[, 1], order = c(2,1,3))
#ARIMA(2,1,3) model
arima.estimate6 <- ts.model(itca.pm25.decomp[, 1], order = c(3,0,4))
#ARIMA(3,0,4) model
arima.estimate7 <- ts.model(itca.pm25.decomp[, 1], order = c(4,0,5))
#ARIMA(4,0,5) model
arima.estimate8 <- ts.model(itca.pm25.decomp[, 1], order = c(4,1,5))
#ARIMA(4,1,5) model

cat(paste('Sigma^2 of the original series=', as.character(var(log(itca.pm25.ts)))))

```

```

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ar1      ma1
  0.3323  0.5680
s.e.  0.1014  0.1052

sigma^2 estimated as 0.3247: log likelihood = -180.29, aic = 366.57

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ma1
  0.3527
s.e.  0.0595

sigma^2 estimated as 0.4586: log likelihood = -215.15, aic = 434.31

```

```

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ar1      ma1      ma2      ma3
  0.1648 -0.1381 -0.0291 -0.8328
s.e.  0.0822  0.0575  0.0923  0.0510

sigma^2 estimated as 0.2333: log likelihood = -147.46, aic = 304.93

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ma1      ma2      ma3      ma4      ma5
-0.2368 -0.7441 -0.8476 -0.0885  0.9316
s.e.  0.0337  0.0353  0.0339  0.0355  0.0303

sigma^2 estimated as 0.1221: log likelihood = -86.71, aic = 185.43

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ar1      ar2      ma1      ma2      ma3
  0.2642 -0.3539 -0.1802  0.1802 -1.0000
s.e.  0.0660  0.0667  0.0228  0.0237  0.0248

sigma^2 estimated as 0.1951: log likelihood = -132.43, aic = 276.87

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ar1      ar2      ar3      ma1      ma2      ma3      ma4
  0.6956 -0.2132 -0.5519 -0.548 -0.4097  0.4143 -0.4566
s.e.  0.1246  0.1632  0.1123  0.156  0.1348  0.1261  0.1030

sigma^2 estimated as 0.08215: log likelihood = -40.26, aic = 96.53

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ar1      ar2      ar3      ar4      ma1      ma2      ma3      ma4
-0.3936  0.2906 -0.3756 -0.8268  0.8188 -0.5817 -0.9891 -0.344
s.e.  0.0568  0.0472  0.0420  0.0470  0.0892  0.0961  0.0667  0.094
      ma5
  0.0960
s.e.  0.0691

sigma^2 estimated as 0.06579: log likelihood = -17.81, aic = 55.62

```

```

Call:
arima(x = ts, order = order, include.mean = FALSE)

Coefficients:
      ar1      ar2      ar3      ar4      ma1      ma2      ma3      ma4
-0.4295  0.2847 -0.3809 -0.8296 -0.0968 -1.3875 -0.4858  0.5335
s.e.   0.0450  0.0477  0.0433  0.0429  0.0741  0.0730  0.1354  0.0661
      ma5
      0.4411
s.e.   0.0822

sigma^2 estimated as 0.06703: log likelihood = -24.38, aic = 68.76
Sigma^2 of the original series = 0.118218591096399

```

Imagen 7.3.8. Estimaciones del modelo ARIMA

De los resultados anteriores notamos que:

El modelo ARIMA (3,0,4) con los siguientes coeficientes tiene el menor AIC (Criterios de información de Akaike)

Coefficientes:

	ar1	ar2	ar3	ma1	ma2	ma3	ma4
-	-1.0076	0.2590	0.7093	1.8248	1.3592	0.5135	0.2456
es decir.	0.0557	0.0958	0.0563	0.0696	0.1589	0.1603	0.0702

sigma ^ 2 estimado como 0.0006552: probabilidad de registro = 725.9, aic = -1435.8

Todos los coeficientes son significativos (el error estándar asociado con estos coeficientes es algunas veces más pequeño que el valor del coeficiente en sí)

El *sigma2* o la varianza de la serie residual es un orden de magnitud menor que la serie original. Así que este es el mejor modelo de ajuste para los residuos. Grafiquemos el ACF y el PACF de esto y observemos si este modelo eliminó el componente estacional y de tendencia del residuo o resto.

Graficamos la estimación 6.

```
plot.acf(arima.estimate6$resid[-1], col = 'ARIMA(3,0,4) estimate', is.df = F)
```

ACF de ARIMA(3,0,4) estimate

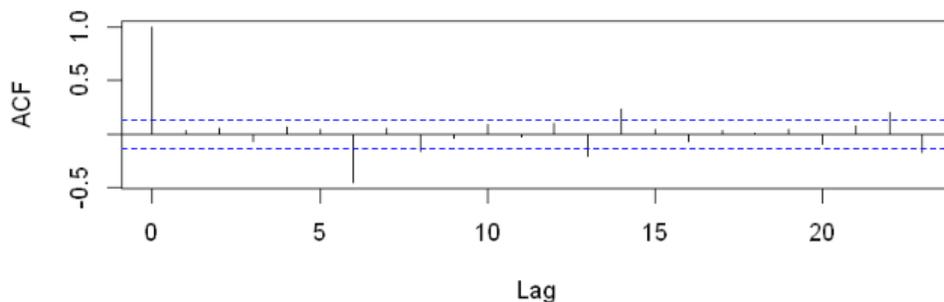


Imagen 7.3.9. Grafica de estimación de ACF ARIMA (3,0,4)

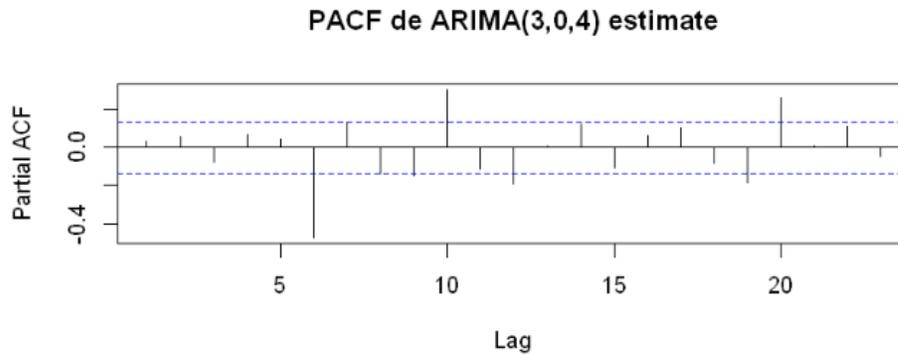


Imagen 7.3.10. Grafica de estimación de PACF ARIMA (3,0,4)

e) Evaluación del modelo

El paquete R **forecast** en particular, tiene herramientas para estimar automáticamente los modelos ARIMA estacionales y utilizar estos modelos para hacer pronósticos futuros.

La función **auto.arima** del paquete **forecast** hace lo siguiente:

- Realiza una regresión progresiva a partir del orden máximo de cada coeficiente para encontrar el modelo ARIMA estacional óptimo. La optimización se realiza utilizando los Criterios de Información de Akaike (AIC) y los Criterios de Información Bayesiana (BIC).
- El modelo ARIMA estacional puede escribirse como ARIMA (p, d, q) (P, D, Q). Ya hemos explorado el modelo estándar de ARIMA, ARIMA (p, d, q). Los términos estacionales adicionales modelan el comportamiento estacional y de tendencia:
 - a. P es el orden del modelo autorregresivo estacional.
 - b. D es el orden de un operador diferenciador o integrador, que modela la tendencia.
 - c. Q es el orden del modelo de media móvil estacional.

Se utiliza la función **summary** para ver los resultados de la evaluación del modelo

```
fit.itcapm25 <- auto.arima(itca.pm25.ts)
summary(fit.itcapm25)
```

```

Series: itca.pm25.ts
ARIMA(1,0,1)(1,0,0)[36] with non-zero mean

Coefficients:
      ar1      ma1      sar1      mean
      0.2935  0.2935  0.2190  12.5088
s.e.  0.0953  0.0916  0.1099  0.4539

sigma^2 estimated as 8.848:  log likelihood=-525.94
AIC=1061.87  AICc=1062.17  BIC=1078.61

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.007878438  2.946117  2.461684  -7.094204  23.46541  0.7307725
              ACF1
Training set 0.05086941

```

Imagen 7.3.11. Datos obtenidos de la evaluación del modelo sumariados.

Despues de evaluar el modelo, se grafican los resultados para ver la tendencia del pronostico, usando la funcion **forecast**

```

unemp.forecast <- forecast((fit.itcapm25),h=12)
summary(unemp.forecast)
plot(unemp.forecast)

```

```

Forecast method: ARIMA(1,0,1)(1,0,0)[36] with non-zero mean

Model Information:
Series: itca.pm25.ts
ARIMA(1,0,1)(1,0,0)[36] with non-zero mean

Coefficients:
      ar1      ma1      sar1      mean
      0.2935  0.2935  0.2190  12.5088
s.e.  0.0953  0.0916  0.1099  0.4539

sigma^2 estimated as 8.848:  log likelihood=-525.94
AIC=1061.87  AICc=1062.17  BIC=1078.61

Error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.007878438  2.946117  2.461684  -7.094204  23.46541  0.7307725
              ACF1
Training set 0.05086941

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
11.83333      12.42178  8.609702  16.23386  6.591708  18.25186
11.86111      11.51238  7.092170  15.93259  4.752253  18.27251
11.88889      11.14290  6.674186  15.61162  4.308590  17.97722
11.91667      12.50295  8.030075  16.97582  5.662281  19.34361
11.94444      12.95105  8.477823  17.42428  6.109839  19.79226
11.97222      13.74379  9.270533  18.21705  6.902533  20.58505
12.00000      13.08552  8.612258  17.55878  6.244256  19.92678
12.02778      12.57639  8.103127  17.04965  5.735125  19.41765
12.05556      11.55775  7.084491  16.03101  4.716489  18.39902
12.08333      11.15622  6.682957  15.62948  4.314955  17.99748
12.11111      12.50685  8.033592  16.98012  5.665590  19.34812
12.13889      12.95220  8.478936  17.42546  6.110935  19.79346

```

Imagen 7.3.12. Datos obtenidos de la función para pronósticos de calidad del aire forecast.

Forecasts from ARIMA(1,0,1)(1,0,0)[36] with non-zero mean

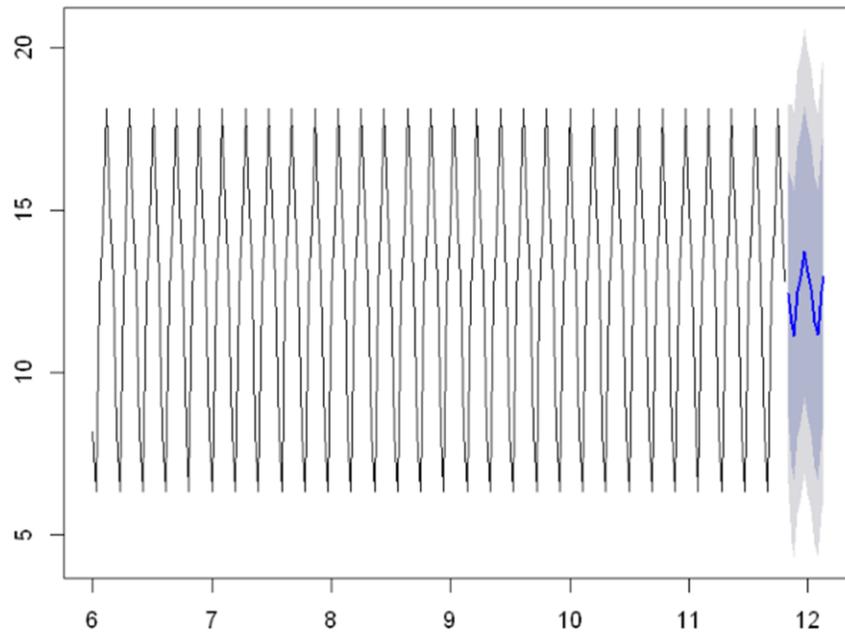


Imagen 7.3.13 Graficas de tasas pronosticadas de material particulado 2.5.

La línea azul en el gráfico anterior muestra las tasas pronosticadas y el frotis gris denota los intervalos de confianza.

Notamos que el 80% y el 95% de los intervalos de confianza son casi proporcionales al valor pronosticado en sí mismo y se hacen cada vez más amplios a medida que avanza el tiempo. Esto implica que la varianza puede estar aumentando con el tiempo, o la heterocedasticidad. Eso significa que esto puede no ser adecuado para nuestros datos de calidad del aire.

7.4. CONSTRUCCIÓN Y PROGRAMACIÓN DE ESTACIONES DE MONITOREO

La elaboración de las estaciones está basado en el diseño de autoría de la Universidad Tecnológica de El Salvador, el cual ya fue presentado y divulgado para lo cual se deja este enlace (<http://hdl.handle.net/11298/1144>) como referencia a el trabajo desarrollado. En esta etapa se contó con la ayuda de docentes investigadores de la Escuela de Ingeniería Eléctrica y Electrónica (Ing. Morris Díaz) y la Escuela de Educación Dual (Ing. Eduardo Amaya).

Por parte de la Universidad Tecnológica de El Salvador se contó con la participación de los docentes investigadores Ing. Omar Flores como el especialista en Electrónica, Ing. Rony Cortez y la Inga. Verónica Idalia Rosa Urrutia ambos del área de informática.

El código de cada estación esta en MicroPython, el cual esta optimizado para ejecutarse en microcontroladores. La codificación es la siguiente:

```

1. # *****
2. # Aim: to control a IoT device to measurement PM Air Quality
3. # Hardware: Lopy Board + PMS5003 sensor + DSB20 sensor + OledDisplay
4. # Main Fuction: take a measurement of PM2.5 and PM10, display it and
5. # send it via Internet to a "Cloud Storage" (Google?)
6. # Author: omar.flores@utec.edu.sv
7. # Last update: sep18
8. # *****
9. # usefull Links
10. # https://www.tiny-dev.com/images/Lopy\_pinout.png
11. # https://github.com/pycom/pycom-micropython-sigfox/blob/master/docs/Lopy/quickref.rst
12. #
13. # Hardware conexions
14. # Exp.Board ----- Peripheral
15. # P10 ----- SCL Oled
16. # P9 ----- SDA Oled
17. # P8 ----- DAT DS18B20
18. # P4 ----- TX pms5003
19. # P11 ----- SET pms5003
20.
21.
22. import gc
23. # import Libraries
24. import machine #to use machine-core fuctions
25. from machine import I2C #to use I2C port onboard
26. from machine import Timer, Pin, PWM
27. import ssd1306 #to manipulated OledDisplay
28. import time #to use time functions
29. import pycom #to use some features of pycom boards
30. #to read PMSx sensors
31. from pms5003 import PMS5003, PMSData
32. import _thread
33. import utime
34. # to use WLAN onboard
35. from network import WLAN
36. import config #file with credentials
37. wlan = WLAN(mode=WLAN.STA, antenna=WLAN.INT_ANT) # configure as Station

```

```

38.
39. #Configuracion de IP estática Sensor PM
40. wlan.ifconfig(config=('***.***.***.***', '255.255.255.0', '192.168.***.***',
'192.168.***.***'))
41.
42. #for read Temp Sensor
43. from onewire import DS18X20
44. from onewire import OneWire
45. #in order to make HTTP urequests
46. import urequests
47. pycom.heartbeat(False)
48. # make some object / declarations
49. # in order to use OledDisplay (sda, scl)
50. i2c = machine.I2C(0, machine.I2C.MASTER, baudrate=400000, pins=('P9','P10'))
51. lcd = ssd1306.SSD1306_I2C(128,32,i2c) #size of OledDisplay
52. lcd.fill(0)
53. lcd.text(" BOOT! ",0,12)
54. lcd.show()
    55. time.sleep(3)
    56. # a welcome message
    57. lcd.fill(0)
    58. lcd.text(" Bienvenido",0,0)
    59. lcd.text("Estación AQ-PM",0,12)
    60. lcd.text(" ITCA-LU",0,24)
    61. lcd.show()
    62. time.sleep(3)
    63. lcd.fill(0)
    64.
    65. # in order to comunicated with PMS sensor
    66. # PMS5003:
    67.
    68. # def __init__(en, tx, rx, rst):
    69. pmSensor = PMS5003(Pin.module.P11, Pin.module.P3, Pin.module.P4, Pin.module.P12)
    70. # in order to read DS18B20 data Line connected to pin P8 of ESP32
    71. ow = OneWire(Pin('P8'))
    72. temp = DS18X20(ow)
    73. tSensor = round(temp.read_temp_async(),1)

```

```

74. time.sleep(1)
75. temp.start_conversion()
76. time.sleep(1)
77. # use to access Google Script
78.
79. #utec ITCA1 script
80. urlGAScript1 =
    "https://script.google.com/a/mail.utec.edu.sv/macros/s/*****/exec?"
81. urlGAScript = "http://****?"
82.
83.
84. def conectarWifi():
85. # setup of WiFi conections
86. print (config.ssid)
87. nets = wlan.scan()
88. for net in nets:
89. if net.ssid == config.ssid:
90. print('Network found!')
91. lcd.fill(0)
92. lcd.text('Connecting ...',0,0)
93. lcd.show()
94. wlan.connect(net.ssid, auth=(net.sec, config.password), timeout=5000)
95. i=0
96. while not wlan.isconnected():
97. print('!')
98. lcd.text('.',i,12)
99. lcd.show()
100. machine.idle() # save power while waiting
101. i=i+1
102. print('WLAN connection succeeded!')
103. break
104. print(wlan.ifconfig())
105. # wifi link message
106. lcd.fill(0)
107. lcd.text("Wifi",0,0)
108. lcd.text(config.ssid,5,12)
109. lcd.text("    done!",0,24)

```

```

110. lcd.show()
111. time.sleep(3)
112. lcd.fill(0)
113. #-----
114.
115.
116.
117.
118. def showDataOled():
119.     lcd.fill(0)
120.     lcd.text("PM2.5 = ",0,0)
121.     lcd.text(str(mean_data.pm25), 64,0)
122.     lcd.text("PM10 = ",0,12)
123.     lcd.text(str(mean_data.pm10), 64,12)
124.     lcd.text("TEMP = ",0,24)
125.     lcd.text(str(tSensor) + " C",64,24)
126.     lcd.text(str(int(temp.read_temp_async())),64,24)
127.     lcd.show()
128.     print("*****")
129.
130.     #*****
131.
132.
133.
134.     #-----
135.     # try to connect to local WiFi
136.     conectarWifi()
137.     #-----
138.     # main loop, runs 4ever
139.     while True:
140.         #---read Temp Sensor---
141.         #read temperature sensor
142.         tSensor = round(temp.read_temp_async(),1)
143.         time.sleep(1)
144.         temp.start_conversion()
145.         time.sleep(1)
146.         tempStr = str(tSensor)

```

```

147. print(tempStr)
148. #-----
149. #---read PM Sensor ---
150. #read PM sensor
151. print("reading PM")
152. pmSensor.wake_up()
153. print("wake PM")
154. frames = pmSensor.read_frames(5)
155. print("a")
156. pmSensor.idle()
157. cpm25 = 0
158. cpm10 = 0
159. pm25 = 0
160. pm10 = 0
161. for data in frames:
162.     cpm25 += data.cpm25
163.     cpm10 += data.cpm10
164.     pm25 += data.pm25
165.     pm10 += data.pm10
166. mean_data = PMSData(cpm25/len(frames), cpm10/len(frames), pm25/len(frames),
    pm10/len(frames))
167. print('cPM25: {}, cPM10: {}, PM25: {}, PM10: {}'.format(mean_data.cpm25,
    mean_data.cpm10, mean_data.pm25, mean_data.pm10))
168.
169. showDataOled()
170. time.sleep(5)
171. #-----
172.
173. while not wlan.isconnected():
174.     conectarWifi()
175.
176. # ***** post to Google Sheets
177. pm25 = str(mean_data.pm25)
178. pm10 = str(mean_data.pm10)
179. urlFinal = urlGAScript + "pm25=" + pm25 + "&pm10=" + pm10 + "&temp=" + tempStr +
    "&estacion=3"
180. urlFinal1 = urlGAScript1 + "pm25=" + pm25 + "&pm10=" + pm10 + "&temp=" + tempStr
181.

```

```

182. #make a request to App Script
183. lcd.fill(0)
184. lcd.text("Posting...",0,0)
185. lcd.show()
186. time.sleep(5)
187. lcd.show()
188. r = None
189. r1 = None
190. try:
191. r = urequests.get(urlFinal)
192. print("post0Ki")
193. r.close()
194.
195. r1 = urequests.get(urlFinal1)
196. print("post10Ki")
197. r1.close()
198.
199. print("post0K")
200. except NotImplementedError:
201. print("NotImplementedError")
202. except Exception as e: # Here it catches any error.
203. if isinstance(e, OSError) and r and r1: # If the error is an OSError the socket has
to be closed.
204. r.close()
205. r1.close()
206. value = {"error": e}
207. pass #print(r.text)
208. gc.collect()
209. lcd.fill(0)
210. lcd.text("Posted !",0,0)
211. lcd.show()
212. time.sleep(5)
213. showDataOled()
214. time.sleep(1200)
215.
216. #-----ed of code-----

```

Como parte de los requerimientos de los técnicos del Ministerio de Medio Ambiente, las estaciones deben estar ubicadas de 2 a 3 mts. de altura como máximo para poder tener una medición más correcta respecto al aire que respiramos, es decir, a una altura promedio de una persona y de preferencia en donde hay mayor circulación de personas.

Algunos de los cuidados que se deben de tener con el montaje de las estaciones son:

1. Deben estar en un lugar protegido, fuera del alcance de las personas a una mala manipulación o hurto de la misma.
2. Deben tener una conexión eléctrica cercana y exclusiva para la estación, evitando desconexión de la misma y pérdida de lecturas de datos ambientales.
3. Deben de tener conectividad Wifi, por el momento no cuentan con conexión Ethernet.
4. Se recomienda colocarlas dentro de una base protectora para el Sol e inclemencias del clima. En el caso de ITCA-FEPADE se crearon unas bases metálicas para su protección.



Imagen 7.4.1. Diseño propuesto de base protectora para el montaje de las estaciones de monitoreo de calidad del aire.



Imagen 7.4.2. Estación de monitoreo de calidad del aire construida por ITCA-FEPADE



Imagen 7.4.3. Ing. Luis Alonso Grande Cedillos colaborando en la construcción de bases protectoras en talleres de ITCA-FEPADE



Imagen 7.4.4. Inga. Rina López mostrando bases protectoras terminadas en talleres de ITCA-FEPADE

8. CONCLUSIONES

ITCA-FEPADE

- a) Se construyeron e instalaron dos estaciones de monitoreo para medir calidad del aire con sensores para PM 2.5 y PM 10 así como la temperatura ambiente. Se programaron en lenguaje Python y se establecieron que enviaran los datos cada 20 min, logrando tener 72 lecturas por día.
- b) Por parte de UTEC se instaló una estación de monitoreo dentro de las instalaciones de ITCA-FEPADE Sede Central en Santa Tecla, programada en lenguaje C.
 - c) Se configuró un servidor con sistema operativo Linux en donde se montó una base de datos en Big Data, utilizando MongoDB como herramienta de gestión.
 - d) Se programó un algoritmo de predicción para material particulado de PM 2.5, PM 10 y temperatura con metodología de Machine Learning para Análisis de serie de tiempos, el cual por el momento se encuentra en versión de prototipo y en etapa de entrenamiento; ya que para obtener mejores resultados de predicción se requiere de una gran cantidad de datos de aproximadamente 5 años de preferencia para que las predicciones sean más exactas que las que se generan al momento. El lenguaje utilizado para la programación del algoritmo es R y se desarrolló en la plataforma Jupyter.
 - e) Se desarrolló una aplicación de fácil acceso y uso para que las personas puedan consultar las condiciones ambientales de PM 2.5, PM 10 y Temperatura en cualquiera de las estaciones situadas en el país. Dicha aplicación está integrada con base de datos Big Data y Laravel como framework de desarrollo en lenguaje PHP.

UTEC

- f) Por parte de la Universidad Tecnológica de El Salvador se realizaron los estudios de requerimientos y factibilidades en la selección de ubicaciones donde se ubicarían las estaciones de monitoreo. Logrando instalar 6 estaciones en: ITCA-FEPADE Sede Central Santa Tecla, Universidad Tecnológica de El Salvador (Edificio de Vicerrectoría de Investigación y Proyección Social), Alcaldía Municipal de Santa Ana, Instituto Nacional de Antiguo Cuscatlán, Colegio Adventista de Chalatenango y en la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador (San Salvador).
- g) Se impartieron capacitaciones a todo el equipo de trabajo de ambas IES sobre Machine Learning, diseño, ensamblado y mantenimiento de las estaciones de monitoreo de calidad del aire, así como su programación requerida en lenguaje Python.

9. RECOMENDACIONES

Las recomendaciones que se plantean basado en la experiencia obtenida en el desarrollo del proyecto de investigación son las siguientes:

1. Se recomienda profundizar en la generación de dispositivos de bajo costo y programación de código abierto para brindar soluciones más acordes a la situación actual de las instituciones tanto públicas como privadas; para lograr mejores resultados en diferentes actividades en beneficio de la población en general.
2. Se recomienda al Ministerio de Medio Ambiente y Recursos Naturales invertir en el desarrollo de más estaciones con el fin de lograr una cobertura a nivel nacional y lograr medir con más detalle la calidad del aire que se respira en El Salvador, obteniendo los beneficios de poder mejorar la toma de decisiones respecto al combate de las enfermedades respiratorias y bajar los costos adheridos a dicha acción, pudiendo invertir en otras necesidades de la población.
3. Se recomienda a las instituciones académicas que incluyan en sus planes de estudio contenidos con temáticas de Internet de las cosas IoT y electrónica básica para el manejo de open hardware, con el cual se pretende inculcar en los estudiantes ideas o motivaciones por contribuir en la ciencia, investigación e innovación con el uso de esta tecnología.

10. GLOSARIO

B

Bases de datos de tipo relacional

Es un tipo de base de datos que cumple con el modelo relacional, que es el modelo más utilizado actualmente para implementar las Bases de Datos ya planificadas.

Big Data

Es un concepto que hace referencia al almacenamiento de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos.

Big Table

Es un sistema de gestión de base de datos creado por Google con las características de ser: distribuido, de alta eficiencia y propietario.

C

Cassandra

Es una base de datos NoSQL distribuida y basada en un modelo de almacenamiento de «clave-valor», de código abierto que está escrita en Java.

CouchDB

Es un gestor de bases de datos de código abierto, cuyo foco está puesto en la facilidad de su uso y en ser "una base de datos que asume la web de manera completa".

CSS3

Siglas en inglés de cascading style sheets. Es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.

E

Emision

Es el volcado de determinadas sustancias a la atmósfera.

G

GraphDB

Es una base de datos orientada a grafos (BDOG) representa la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos.

H

HBase

Es un modelo de datos que es similar a Google BigTable, diseñada para permitir el rápido acceso aleatorio a enormes cantidades de datos estructurados.

HTML5

(HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

HyperTable

Es un sistema de almacenamiento de datos distribuido y de alto desempeño, ideal para aplicaciones que necesitan manejar datos que evolucionan rápidamente y diseñado para soportar una gran demanda de datos en tiempo real.

I

Inmisión atmosférica

Es la concentración de contaminantes a nivel del suelo que puede afectar a personas, animales, vegetación o materiales.

K

KeyValue

Los datos se almacenan de forma similar a los maps o diccionarios de datos, donde se accede al dato a partir de una clave única.

M

Material particulado

Conocido como PM, es la mezcla de partículas líquidas y sólidas, de sustancias orgánicas e inorgánicas, que se encuentran en suspensión en el aire. El material particulado forma parte de la contaminación del aire.

MongoDB

Es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

N

NoSQL

Not Only SQL y son sistemas de almacenamiento que no cumplen con el esquema entidad-relación. Proveen un sistema de almacenamiento mucho más flexible y concurrente y permiten manipular grandes cantidades de información de manera mucho más rápida que las bases de datos relacionales.

O

Open hardware

Conocido como hardware libre, de código abierto, electrónica libre o máquinas libres a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago, o de forma gratuita.

P

Portal Web

Es un sitio web que ofrece al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema.

S

SCRUM

Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

X

XML

Siglas en inglés de eXtensible Markup Language ("lenguaje de marcas Extensible"), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

11. REFERENCIAS BIBLIOGRÁFICAS

- [1] F. C. Omar Otoniel, C. R. Ronny Adalberto, y R. U. Verónica Idalia, *Sistema telemático de monitoreo de calidad del aire en zonas remotas, utilizando técnicas IoT y big data : fase I. Estación IoT automatizada para el monitoreo de calidad del aire por contaminantes PM2.5 y Pm10*. Universidad Tecnológica de El Salvador, 2019.
- [2] “¿Es seguro respirar en San Salvador?”, *elsalvador.com*, 2016. [En línea]. Disponible en: <http://especiales.elsalvador.com/2016/Es-seguro-respirar-en-San-Salvador/>. [Consultado: 28-ene-2019].
- [3] “Definición de inteligencia artificial - Qué es, Significado y Concepto”, 2014. [En línea]. Disponible en: <https://definicion.de/inteligencia-artificial/>. [Consultado: 05-feb-2019].
- [4] “Qué es Machine Learning, cómo funciona y a qué se aplica”, *APD España*, 04-mar-2019. [En línea]. Disponible en: <https://www.apd.es/que-es-machine-learning/>. [Consultado: 22-ene-2020].
- [5] V. Roman, “Introducción al Machine Learning: Una Guía Desde Cero”, *Medium*, 15-abr-2019. [En línea]. Disponible en: <https://medium.com/datos-y-ciencia/introduccion-al-machine-learning-una-gu%C3%ADa-desde-cero-b696a2ead359>. [Consultado: 23-ene-2020].
- [6] F. C. Sampieri Roberto y Baptista Pilar, *Metodología de la Investigación*. México: Mc Graw Hill, 2003.
- [7] “Cómo instalar en Ubuntu 18.04 la pila LAMP — Linux, Apache, MySQL y PHP”, *DigitalOcean*. [En línea]. Disponible en: <https://www.digitalocean.com/community/tutorials/como-instalar-en-ubuntu-18-04-la-pila-lamp-linux-apache-mysql-y-php-es>. [Consultado: 24-feb-2020].
- [8] J. L. G. Martínez, “MongoDB - Cómo instalar MongoDB en Ubuntu 18.04”, *JoLuGaMa Blog*, 24-nov-2018. [En línea]. Disponible en: <https://jologama.com/blog/2018/11/24/mongodb-Instalacion/>. [Consultado: 24-feb-2020].
- [9] “Download RStudio Server for Debian & Ubuntu”. [En línea]. Disponible en: <https://rstudio.com/products/rstudio/download-server/debian-ubuntu/>. [Consultado: 24-feb-2020].
- [10] “Entornos de trabajo para ciencia de datos, instalando R-Studio y Anaconda en Ubuntu Linux 18.04”, *Open Source Academic Initiative*. [En línea]. Disponible en: <https://opensai.org/bitacora/entornos-de-trabajo-para-ciencia-de-datos-instalando-r-studio-y-anaconda-en-ubuntu-linux-18-04>. [Consultado: 24-feb-2020].
- [11] “How to Install Laravel on Ubuntu 18.04”, 15-dic-2018. [En línea]. Disponible en: </post/how-to-install-laravel-on-ubuntu-18-04/>. [Consultado: 24-feb-2020].
- [12] “PHP: MongoDB - Manual”. [En línea]. Disponible en: <https://www.php.net/manual/en/set.mongodb.php>. [Consultado: 24-feb-2020].

12. ANEXOS

ANEXO 1



*Universidad Tecnológica
de El Salvador*



Proyecto: Sistema telemático de monitoreo de calidad del aire en zonas remotas. Desarrollo de una plataforma para el análisis de datos y generación de modelos por medio de técnicas de Big Data y Machine Learning

Propuesta de perfil Informático

Estudios	Ingeniería en sistemas y computación, Ingeniería en Desarrollo de Software, Técnico en ingeniería de sistemas informáticos.
Conocimientos, habilidades y aptitudes requeridas:	<ul style="list-style-type: none">• Conocimiento de lenguajes de programación (PHP5, HTML5, CSS3, JQuery, JQuery Mobile, JSON y Ajax).• Desarrollo de aplicaciones multiplataforma.• Conocimientos de bases de datos Big Data.• Instalación y configuración de servidor web (Apache, PHP, MongoDB).• Deseable con conocimientos básicos de Laravel PHP.• Conocimiento básico de hardware y software de PC, así como sistemas operativos Windows, Linux y Redes.• Conocimiento básico en programas de diseño y edición de imágenes.• Alto grado de responsabilidad.• Alto grado de respeto y compromiso institucional.• Sensible a temas ambientales y culturales.• Compromiso con los objetivos e imagen de la institución.• Pro actividad e iniciativa• Buena capacidad de reacción ante situaciones críticas• Dirigir y participar en grupos de trabajo multi e interdisciplinarios.• Participar en proyectos de investigación.• Liderar y formular propuestas de proyectos de medio ambiente de acuerdo a las necesidades del medio.

SEDE CENTRAL Y CENTROS REGIONALES



La Escuela Especializada en Ingeniería ITCA-FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro centros regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.

1 SEDE CENTRAL SANTA TECLA

Km. 11.5 carretera a Santa Tecla, La libertad.
Tel.: (503) 2132-7400

2 CENTRO REGIONAL SANTA ANA

Final 10a. Av. Sur, Finca Procavia.
Tel.: (503) 2440-4348

3 CENTRO REGIONAL LA UNIÓN

Calle Sta. María, Col. Belén, atrás del Instituto Nacional de La Unión.
Tel.: (503) 2668-4700

4 CENTRO REGIONAL ZACATECOLUCA

Km. 64.5, desvío Hacienda El Nilo sobre autopista a Zacatecoluca.
Tel.: (503) 2334-0763 y
(503) 2334-0768

5 CENTRO REGIONAL SAN MIGUEL

Km. 140 carretera a Santa Rosa de Lima.
Tel.: (503) 2669-2298