Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte de Proyecto de Integración

Proceso de Minería sobre la calidad de aire en la Ciudad de México

David Venegas Martínez 209204019

Asesora:

Silvia Beatriz González Brambila

Profesora Titular

Departamento de Sistemas

Trimestre 14P

15 de Julio de 2014

Yo, Silvia Beatriz González Brambila, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma

Yo, David Venegas Martínez doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repsitorio Institucional de UAM Azcapotzalco.

Firma

1. Resumen

Este proyecto se centra en el estudio de un análisis de los 6 contaminantes que afectan de una manera importante la vida en el Distrito Federal, estos contaminantes son: Monóxido de carbono (CO), Dióxido de nitrógeno (NO2), Óxido de nitrógeno (NOX), Ozono (O3), Partículas suspendidas (PM10) y Dióxido de azufre (SO2).

La contaminación en el Distrito Federal ha sido y será uno de los principales problemas que afectan a los seres vivos que radican aquí, por lo cual es importante predecir el comportamiento de los mismos en un futuro próximo.

Una vez que se localizaron los contaminantes más dañinos para la salud, se realizó una recolección de información gracias al apoyo de la Secretaría del Medio Ambiente, misma que facilitó las bases de datos que contienen información al respecto desde 1996 hasta el 2011.

Las bases de datos vienen expresadas en archivos con extensión (.xls). Para cada base de datos se realizó una depuración y verificación de los mismos, asegurando que no existiese ningún valor faltante o que las unidades en las que están expresados estos valores sean correctas, para después hacerles el análisis de suavizado exponencial o modelos ARIMA para cada contaminante que se requiera analizar.

Una vez que se realice alguno de los análisis antes mencionados, la gráfica resultante será expresada en horas, en la cual, se representa el comportamiento que tuvo el contaminante elegido, como también, su comportamiento futuro a corto plazo o predicción.

En dado caso que la predicción muestre un aumento en los niveles del contaminante, se podrán tomar cartas en el asunto haciendo un programa de contingencia ambiental.

Tabla de contenido

Tabla de Figuras

2. Introducción

Con el propósito de proteger la salud humana contra los daños provocados por la contaminación del aire, la Secretaria de Salud cuenta con las Normas Oficiales Mexicanas de Salud Ambiental a nivel federal, estas normas establecen los valores límites de los contaminantes del aire para la protección de la salud de la población; además, aplican al territorio nacional y es responsabilidad de las autoridades locales su observación y cumplimiento.

En la actualidad la calidad del aire compromete seriamente la salud y calidad de vida de los habitantes del Distrito Federal, a tal grado la Secretaría del Medio Ambiente pone diariamente alguna medida relacionada a la contaminación y cada año se publican reportes anuales sobre la calidad del aire [1][1].

Los habitantes de la ciudad lidian con los efectos de la contaminación del aire, generada en su mayoría por el parque vehicular de más de 5 millones de vehículos. Solo a principios del 2014 se rebasó durante varios días el límite de contaminantes, lo que llevó a las autoridades a limitar el uso de vehículos con una inédita aplicación del programa "*Hoy no circula*"[2], por lo cual el tránsito vehicular se mantiene como la fuente más importante de emisiones de contaminantes en la Ciudad, fenómeno asociado con el comportamiento de los contaminantes en el aire. Las dimensiones actuales de la Ciudad de México rebasan los límites geográficos del Distrito Federal y su área metropolitana. Partiendo de dicha situación, se necesitó tener una forma de automatizar la interpretación de la información recolectada. Automatizando la lectura de información representada en los archivos brindados por la Secretaría del Medio Ambiente, se generó un análisis especializado en el cual se muestre el comportamiento que han tenido los contaminantes en lapsos de tiempo considerables partiendo del tamaño de los datos que tengamos para un óptimo funcionamiento.

Gracias a los análisis de resultados, se puede observar la tendencia que se fue siguiendo, además de una predicción para saber qué es lo que nos espera en días futuros.

En la Secretaría del Medio Ambiente las bases de datos están expresadas en archivos con extensión (.xls) y existen en total 96 archivos con los que se trabajó, aclarando que se pueden ir sumando carpetas referentes a años posteriores a los cargados al sistema sin ningún problema, cada carpeta deberá contener los 6 contaminantes antes mencionados.

Este proyecto de integración se plantea como un estudio que puede otorgar pauta para realizar versiones más grandes del mismo, mostrando gráficas y tablas como resultado ya no solo del DF sino de todo México, intentando predecir lo que pasará en un futuro cercano.

Los módulos a desarrollar este proyecto son los que se muestran en la figura 2.1.

[1] En el 2011 la Ciudad de México fue la ciudad de más de 5 millones de habitantes más contaminada del planeta.*http://www.buzzecolo.com/15312/top-10-villes-plus-polluees-monde/*

[2] El programa *"Hoy no circula"* a partir del 2014 tuvo modificaciones considerables para autos con más de 15 años de antigüedad, ya que no pueden circular ningún sábado y un día de la semana laboral.

Figura 2.1 Funcionamiento general

El módulo de lectura de información se realizó al momento de cargar todos los archivos de la base de datos al sistema, haciendo una limpieza y depuración de datos, esto es, escribir un 0 en cada espacio donde falten datos.

Para el Pre procesamiento de información se tuvo que realizar el cambio de partes por billón (ppb) a partes por millón (ppm) de todos los datos que van de julio del 2011 en adelante, así como la eliminación de valores no deseados, esto se realizó gracias a la técnica de imputación múltiple, de la cual se hablará más adelante.

Antes de pasar al análisis de series de tiempo, el usuario previamente realizó una búsqueda en el sistema, y al resultado de esa búsqueda se aplicó la técnica de datos faltantes (imputación múltiple) con la cual a todos los valores que se les puso 0 les asigna un valor dependiendo del comportamiento que van teniendo con los datos de alrededor. Una vez hecho esto pasamos al proceso de minería de datos que son los análisis de series de tiempo (suavizado exponencial y modelos ARIMA). En cualquier análisis que se elija se necesita tener el promedio de cada fila del resultado de la búsqueda, para tener una serie de tiempo con la cual se trabajará en los análisis.

En el módulo "Mostrar Resultados" se observan tablas en archivos con formato .xls y una gráfica por cada uno de los contaminantes que el usuario definió previamente.

El CD que se entregará contendrá:

- Documento de reporte final
- Código fuente

Recursos

El hardware que se utilizó es una computadora portátil con las siguientes características

- Procesador Intel core i5® @ 2.30 GHz.
- 6.00 GB de memoria RAM.
- Disco duro 1 TB/ 5800 rpm.
- Sistema Operativo Microsoft Windows 8 Professional.

El software que se utilizó para este proyecto se menciona a continuación:

- R 3.1.0 x64

Los paquetes utilizados son:

- Forecast 5.4
- Sqldf 0.4-7.1
- XLConnect 0.2-7
- Mice 2.22
- gWidgets 0.0-52
- tcltk2 1.2-10

3. Justificación

La Secretaría del Medio Ambiente genera archivos anuales sobre los contaminantes más comunes que se presentan en la Ciudad de México, estos reportes son de difícil lectura ya que es muy compleja su interpretación, además de ser demasiados archivos con un tamaño significativo de datos. Crear herramientas que permitan obtener patrones y tendencias de los principales contaminantes serían de gran apoyo para entender esta información de una manera rápida y sencilla.

Teniendo en cuenta que existen alteraciones en el ambiente gracias a estos contaminantes, es necesario predecir y encontrar tendencias para anticipar golpes al ambiente, esto es, lapsos en los cuales se contamine más, aplicando así, por ejemplo un programa de contingencia ambiental con una anticipación tal que permita a los ciudadanos estar alerta para tener una mejor calidad del aire.

Este proyecto está enfocado a analizar los datos de RAMA (Red Atmosférica del Medio Ambiente) de tal forma que se obtengan dos tipos de análisis que faciliten su interpretación y que se generen fácilmente, además, forma parte de una propuesta más general de análisis de información sobre la explotación y extracción de información relacionada con la contaminación del aire en la Ciudad de México [8].

Las bases datos con la que se trabajó son publicadas por la Secretaria del Medio Ambiente y se encuentran en formato (.xls).

4. Trabajos relacionados

En este capítulo se mostrarán los trabajos relacionados internos como externos, de los cuales, se extrajo la mayor información para la elaboración de este proyecto. Estos trabajos sirvieron principalmente para ambientarse en la minería de datos, así como una base para saber por dónde comenzar los trabajos de este proyecto.

4.1 Trabajos internos

Proyectos Terminales

1. Lenguaje de manipulación y minería de datos [1].Crea un lenguaje adaptado de SQL para incluir algoritmos existentes y otros novedosos de investigación que no son necesariamente usados por programas comerciales, en este proyecto se usó una herramienta diferente para aplicar minería de datos, además de que el enfoque de este proyecto es la calidad del aire de la Ciudad de México.

2. Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información [2]. Aplica limpieza de datos pero faltan mecanismos de depuración y enriquecimiento de información; además, en este trabajo se analizan análisis de series de tiempo para toda la información recabada.

4.2 Trabajos Externos

Tesis

3. Descubrimiento de patrones secuenciales utilizando razonamiento lógico temporal [3]. Esta tesis hace uso de minería de datos para el descubrimiento de patrones temporales. En este proyecto terminal no se desarrolló ningún método, sólo se aplicaron métodos existentes sobre datos relacionados con la calidad del aire.

4. Desarrollo de un modelo basado en técnicas de Minería de Datos para clasificar zonas climatológicamente similares en el Estado de Michoacán [4]. En esta tesis se describe a detalle un proceso particular para regiones del estado de Michoacán empleando series de datos temporales. Se hace énfasis en obtener un balance de agua subterránea que refleje la situación de los acuíferos por sobreexplotación y no se centró en la calidad del aire que tenemos desde 1994.

Investigación

5. Análisis de las series temporales de los precios del mercado eléctronico mediante técnicas de clustering [5]. Se basa en algoritmos de agrupamiento pero no toca el análisis de series de tiempo.

6. Prediction of Air Pollution of Boushehr City Using Data Mining [6]. En este artículo se habla acerca de la contaminación del aire en la mayoría de los países. Toma en cuenta

técnicas de agrupamiento como K-means y los datos presentados van desde 1951 al 2003. A pesar de ser un buen artículo acerca de este tema, maneja pocos datos en comparación de este proyecto.

7. When Urban Air Quality Inference Meets Big Data [7]. En este artículo se nombran dos de los seis contaminantes que se analizaron. Es una propuesta más detallada y a la vez más grande del proyecto desarrollado en este documento, por lo cual, podría existir la posibilidad de expandir el mismo para obtener alcances como los desglosados en este artículo.

A pesar que los trabajos citados en esta sección fueron de utilidad para la elaboración de este proyecto, no significa que fueron similares en su totalidad, por lo cual, eso hace que este proyecto sea original a los presentados.

5. Objetivos

Enlistamos los objetivos con el fin de tener un propósito para el proyecto de integración, fueron la meta inicial de la cual se partió y guiaron las acciones realizadas.

5.1 Objetivo general:

Desarrollar una aplicación que muestre la tendencia de los valores de concentración de los contaminantes de 1996 a 2011 con base en los reportes de la Secretaria del Medio Ambiente del Gobierno del Distrito Federal[3].

5.2 Objetivos específicos

- Analizarla tendencia mensual de los principales contaminantes atmosféricos de la Red Automática de Monitoreo Atmosférico (RAMA).

  Se cumplió al momento que se realiza una búsqueda asignándole el rango de fechas ej. "Febrero - Marzo" o "Enero 1996 – Agosto 2000"[4].

- Analizar la tendencia semanal de los principales contaminantes atmosféricos de la RAMA.

  Escogiendo una rango de semanas en el mismo año, por ejemplo. "semana 5 – semana 9" o "semana 5 a semana 5"[2].

- Analizar la tendencia diaria de los principales contaminantes atmosféricos de la RAMA.

  Se cumplieron al momento que se realiza una búsqueda y se le asigna el rango de fechas ya sea por mes ej. "5 Febrero – 3 Marzo" o "14 Enero – 31 Diciembre". No importa el rango de fechas, mientras la fecha final sea mayor que la fecha inicial[2].

- Analizar la tendencia en días feriados de los principales contaminantes atmosféricos de la RAMA.

  Como se ha mencionado antes, se realiza una búsqueda por fecha, no por semana para recolectar los datos del lapso que se desea analizar, una vez que se recolectaron se envían a el análisis que el usuario elija[2].

Para que todo un proyecto tenga una forma profesional y dedicada, se deben cumplir los objetivos o gran parte de ellos, esto podría suceder por falta de tiempo o falta de información necesaria para el desarrollo del mismo. Para eso siempre es bueno tener un asesor en el cual tener como base para guiarte en los momentos complicados.

[3] Todos los resultados de los análisis sin excepción se tomaron por horas.
[4] Los análisis descritos se realizaron siempre en base de las horas, ya que entre más datos existan, habrá más fiabilidad para las predicciones a realizar.

6. Marco Teórico

Se mostrarán todos los conocimientos necesarios para sustentar las bases en las que se basó el desarrollo de este proyecto. Igual abarcaremos algunos tipos de análisis de series de tiempo, que aunque no se usaron, es bueno tenerlos en mente para algún análisis posterior.

6.1 Series de tiempo

Una serie tiempo es una secuencia de las observaciones, medidas en determinados momentos del tiempo, ordenados cronológicamente y, espaciadas entre sí de manera uniforme, así los datos usualmente son dependientes entre sí. El principal objetivo de una serie de tiempo, donde $t = 1,2,\ldots n$ es su análisis para hacer pronóstico.

El análisis clásico de las series temporales se basa en la suposición de que los valores que toma la variable de observación es la consecuencia de tres componentes, cuya actuación conjunta da como resultado los valores medidos, estos componentes son [8]:

- Componente tendencia.- Se puede definir como un cambio a largo plazo que se produce en la relación al nivel medio, o el cambio a largo plazo de la media. La tendencia se identifica con un movimiento suave de la serie a largo plazo.

- Componente estacional.- Muchas series temporales presentan cierta periodicidad o dicho de otro modo, variación de cierto período (semestral, mensual, etc.). Por ejemplo las Ventas al Detalle aumentan por los meses de noviembre y diciembre por las festividades navideñas. Estos efectos son fáciles de entender y se pueden medir explícitamente o incluso se pueden eliminar de la serie de datos, a este proceso se le llama des-estacionalización de la serie.

- Componente aleatoria.- Esta componente no responde a ningún patrón de comportamiento, sino que es el resultado de factores fortuitos o aleatorios que inciden de forma aislada en una serie de tiempo.

De estos tres componentes los dos primeros son componentes determinísticos, mientras que la última es aleatoria. Así se puede denotar la serie de tiempo como la fórmula 6.1.1 [8].

$$X_t = T_t + E_t + I_t$$

Fórmula 6.1.1. Serie de tiempo

Donde:
T_t es la tendencia,
E_t es la componente estacional e
I_t es la componente aleatoria.

Clasificación de las series temporales o series de tiempo.

Las series temporales se pueden clasificar en [16]:

- Estacionarias.- Una serie es estacionaria cuando es estable a lo largo del tiempo, es decir, cuando la media y varianza son constantes en el tiempo. Esto se refleja gráficamente en que los valores de la serie tienden a oscilar alrededor de una media constante y la variabilidad con respecto a esa media también permanece constante en el tiempo.

- No estacionarias.- Son series en las cuales la tendencia y/o variabilidad cambian en el tiempo. Los cambios en la media determinan una tendencia a crecer o decrecer a largo plazo, por lo que la serie no oscila alrededor de un valor constante.

Las series temporales se definen como un caso particular de los procesos estocásticos, por lo cual cabe mencionar que un proceso estocástico, se dice que es estacionario si su media y su varianza son constantes en el tiempo y si el valor de la covarianza entre dos periodos depende solamente de la distancia o rezago entre estos dos periodos de tiempo y no del tiempo en el cual se ha calculado la covarianza [16]. Como ese muestran en la Fórmula 6.1.2.

$$Media\ E(X_t) = E(X_{t+k}) = \mu$$
$$Varianza\ V(X_t) = V(X_{t+k}) = \sigma^2$$
$$Covarianza\ \gamma_k = [E(X_t - \mu)(X_{t+k} - \mu)]$$

Fórmula 6.1.2. Medidas de series de tiempo

Donde γ_k , la covarianza (o autocovarianza) al rezago k, es la covarianza entre los valores de X_t y X_{t+k} , que están separados k periodos. Un ruido blanco es un caso simple de los procesos estocásticos, donde los valores son independientes e idénticamente distribuidos a lo largo del tiempo con media cero e igual varianza, se denota por ε_t [16]. El ruido blanco se presenta cuando existen valores mayores al 0.05% o 0.01% en los residuos de nuestra predicción.

6.2 Técnicas de datos faltantes.

Existen varias técnicas de datos faltantes. Para este proyecto de integración usamos la imputación múltiple.

La imputación múltiple es una técnica en la que los valores perdidos son sustituidos por m > 1 valores simulados. Consiste en la imputación de los casos perdidos a través de la estimación de un modelo aleatorio apropiado realizada *m* veces y, como resultado, se obtienen *m* archivos completos con los valores imputados [9].

Los métodos de imputación múltiple (MI) como su nombre lo dice realizan múltiples imputaciones para un mismo valor faltante, produciendo múltiples estimaciones de parámetros del modelo. Estas estimaciones luego se combinan de forma adecuada produciendo una estimación única de los parámetros de interés, junto con un error estándar que refleja la incertidumbre inherente de la imputación [9].

Como se puede observar en la figura 6.2.1 Para llevar a cabo la imputación múltiple de datos faltantes, se procedería del siguiente modo [9]:

- En primer lugar se seleccionan las variables que se emplearán en el modelo de imputación. Es imprescindible que todas las variables que se van a utilizar conjuntamente en posteriores análisis se incluyan en dicho modelo, también se deben incluir todas aquellas variables que puedan ayudar a estimar los valores perdidos o llamados "*missing*".

- En segundo lugar, se decide el número de imputaciones que se desea realizar. En general, entre 3 y 5 imputaciones son suficientes; en este proyecto se utilizaron 5 imputaciones por cada búsqueda realizada.

Aplicando la imputación múltiple, aseguramos que al hacer esta, contiene algún componente de imputación aleatoria. Con esta propiedad se asegura la posibilidad de obtener, para cada registro a imputar, modificaciones entre los valores imputados al completar los distintos ficheros de datos [9].

El siguiente paso será de llevar a cabo los análisis estadísticos (univariantes, bivariantes o multivariantes) necesarios para la investigación. El análisis se realizará con las matrices generadas tras la imputación [10].

Figura 6.2.1. Funcionamiento del algoritmo de imputación múltiple

A continuación se detalla el proceso del algoritmo de imputación múltiple descrito en la figura 6.2.1:

X es matriz de datos disponible compuesta por X= (X_{obs}, X_{aus})
Q es una variable aleatoria,
(X_{obs}, X_{aus}) = valor del estimador Q
U= U(Y_{obs}, Y_{aus}) el error estándar de $\hat{e}$

Se obtiene un único coeficiente que combina los m estimadores $\hat{Q}_j$ obtenidos de los j (*j =1,...m*) archivos de datos completos generados y U_j es la varianza estimada del parámetro $\hat{Q}_j$. Para la puesta en común completado resultados de los datos, un análisis completo de datos está representado por la dupla $(\hat{Q}, U)$ donde $\hat{Q}$ es un punto estimado de un parámetro de interés Q y U es la matriz de covarianza de $\hat{Q}$ [10].

Después de generar m conjuntos de datos mediante simulaciones, se tiene m estimaciones de $\hat{Q}$ y U$(\hat{Q}_1^*, U_1^*)$,…, $(\hat{Q}_m^*, U_m^*)$.Donde la dupla $(\hat{Q}, U)$ son estimado por $(\overline{Q}_m, \overline{U}_m)$ así como se expresan en la fórmula 6.2.1 [10]:

$\overline{Q}_m$ Dado por: $\overline{Q}_m = \frac{1}{m}\sum_{i=1}^{m} \hat{Q}_i^* \gamma$
$\overline{U}_m$ Dado por: $\overline{U}_m = \frac{1}{m}\sum_{i=1}^{m} U_i^*$

Fórmula 6.2.1. Estimación para $(\hat{Q}, U)$

$\overline{U}_m$ Es la varianza de cada imputación, y la varianza entre las imputaciones es como se muestra en la fórmula 6.2.2 [10]:

$$B = \frac{1}{m-1}\sum_{i=1}^{m} (Q_i^* - \overline{Q}_m)^2$$

Fórmula 6.2.2. Varianza entre imputaciones

Por tanto, la varianza total (T) se obtiene sumando las expresiones de las fórmulas 6.2.1 y 6.2.2, corrigiendo el número finito de imputaciones por el valor $\frac{m+1}{m}$, como se muestra en la fórmula 6.2.3 [10].

$$T = U + \frac{(m+1)}{m}B$$

Fórmula 6.2.3. Varianza total

$\frac{B}{U}$ Indica cuánta información corresponde a los datos faltantes, y se estima a partir de $\frac{\gamma}{1-\gamma}$, en donde representa la fracción de información que se pierde por falta de respuesta. En el caso de que se observa que $\gamma = 0$ se observa que B =0 [10].

El intervalo de confianza se obtiene por medio de la fórmula 6.2.4:

$$\overline{Q} = \pm t_{gl}\sqrt{T}$$

Fórmula 6.2.4. Intervalo de confianza

Y los grados de libertad de t se calculan como lo muestra en la fórmula 6.2.5 [10]:

$$gl = (m-1)(1+\frac{1}{r^2})$$

Con:

$$r = (1+\frac{1}{m})\frac{B}{U}$$

Fórmula 6.2.5. Grados de libertad

6.3 Suavizado exponencial

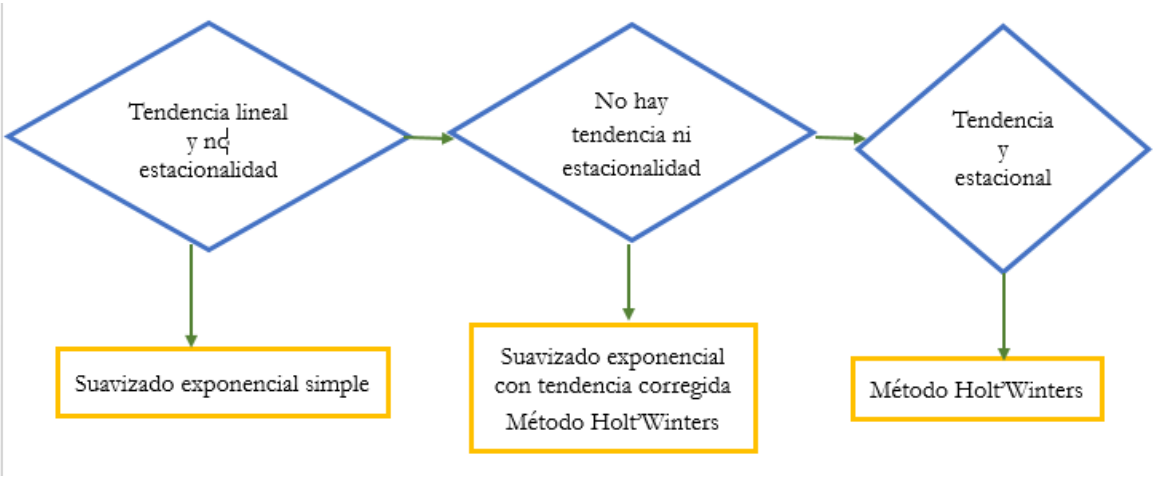


Figura 6.3.1. Casos para usar los métodos de suavizado exponencial

Como se expresa en la figura 6.3.1 existen varios análisis para modelos de este tipo, de tal forma que se puede seleccionar el que mejor se adapte a una serie de tiempo.

Métodos de suavizado exponencial dan mayor peso a las observaciones más recientes, y los pesos disminuyen exponencialmente a medida que las observaciones se hacen más distantes.
Estos métodos son más eficaces cuando los parámetros que describen las series de tiempo están cambiando lentamente con el tiempo.

6.3.1 Suavizado exponencial simple

El método de suavizado exponencial simple se usa para pronosticar una serie de tiempo cuando no hay una tendencia o patrón estacional, pero la media (o nivel) de la serie de tiempo y_t está cambiando lentamente con el tiempo [11].

$$y_t = \beta_o + \varepsilon_t$$

Pasos:
Paso 1: Calcular la estimación inicial de la media (o nivel) de la serie en el período de tiempo $t = 0$, se muestra en la fórmula 6.3.1.1 [11].

$$\ell_0 = \bar{y} = \frac{\sum_{t=1}^{n} y_t}{n}$$

Fórmula 6.3.1.1. Estimación inicial

Paso 2: Calcular la estimación actualizada mediante el uso de la ecuación de suavizado, se muestra en la fórmula 6.3.1.2 [11].

$$\ell_T = \alpha y_T + (1-\alpha)\ell_{T-1}$$

Fórmula 6.3.1.2. Estimación actualizada

Donde α es una constante de alisamiento entre 0 y 1 [11].
Debemos tener en cuenta:

$$\begin{aligned}
\ell_T &= \alpha y_T + (1-\alpha)\ell_{T-1} \\
&= \alpha y_T + (1-\alpha)[\alpha y_{T-1} + (1-\alpha)\ell_{T-2}] \\
&= \alpha y_T + (1-\alpha)\alpha y_{T-1} + (1-\alpha)^2 \ell_{T-2} \\
&= \alpha y_T + (1-\alpha)\alpha y_{T-1} + (1-\alpha)^2 \alpha y_{T-2} + \ldots + (1-\alpha)^{T-1}\alpha y_1 + (1-\alpha)^T \ell_0
\end{aligned}$$

Los coeficientes que miden la contribución de las observaciones disminuyen exponencialmente con el tiempo.

Para crear la predicción en el momento T para y_{T+p}, se muestra en la fórmula 6.3.1.3 [11].

$$\hat{y}_{T+p}(T) = \ell_T \qquad (p = 1, 2, 3, \ldots)$$

Fórmula 6.3.1.3. Predicción

6.3.2 Suavizado exponencial con tendencia corregida

Sí una serie de tiempo se incrementa o disminuye aproximadamente a una tasa fija, entonces puede ser descrita por el modelo de tendencia lineal, se muestra en la fórmula 6.3.2.1 [11]:

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

Fórmula 6.3.2.1 tendencia lineal

Sí los valores de los parámetros β_0 y β_1 están cambiando lentamente con el tiempo, se puede aplicar a las observaciones de series de tiempo.

Nota: Cuando ni β_0 ni β_1 están cambiando con el tiempo, la regresión puede utilizarse para predecir los valores futuros de y_t [11].

Un enfoque suavizado para pronosticar una serie consta de dos constantes de suavización, denotados por α y γ.

Hay dos estimaciones ℓ_{T-1} and b_{T-1}

ℓ_{T-1} es la estimación del nivel de la serie de tiempo construido en período de tiempo T-1 (esto normalmente se denomina componente permanente [11]).

b_{T-1} es la estimación de la tasa de crecimiento de la serie histórica construida en el período de tiempo T-1 (a esto se le suele llamar el componente de tendencia).

La Estimación Nivel, se muestra en la fórmula 6.3.2.2 [11].

$$\ell_T = \alpha y_T + (1-\alpha)(\ell_{T-1} + b_{T-1})$$

Fórmula 6.3.2.2. Estimación de nivel.

La estimación de tendencia, se muestra en la fórmula 6.3.2.1 [11].

$$b_T = \gamma(\ell_T - \ell_{T-1}) + (1-\gamma)b_{T-1}$$

Fórmula 6.3.2.3. Estimación de tendencia

Donde α = constante de suavizamiento para el nivel ($0 \leq \alpha \leq 1$)
γ = constante de suavización de la tendencia ($0 \leq \gamma \leq 1$)

Para crear la predicción en el momento T para y_{T+p}

$$\hat{y}_{T+p}(T) = \ell_T + pb_T \qquad (p = 1,2,3,...)$$

Fórmula 6.3.2.4. Predicción

6.3.3 Métodos Holt-Winters

El método de Holt-Winters es un enfoque suavizado exponencial para el manejo de datos estacionales. Existen dos métodos y están diseñados para series de tiempo que muestran tendencia lineal [11]:

- Aditivo: se utiliza para series de tiempo con constantes (aditivo) variaciones estacionales.
- Multiplicativo: se utiliza para series de tiempo con el aumento (multiplicativos) variaciones estacionales.

El método de Holt-Winters multiplicativo es el más conocido de los dos métodos.

6.3.3.1 Multiplicativo

Por lo general, se considera que es el más adecuado para la previsión de series de tiempo que puede ser descrito por la ecuación que se muestra en la fórmula 6.3.3.1.1 [11]:

$$y_t = (\beta_0 + \beta_1 t) \times SN_t \times IR_t$$

Fórmula 6.3.3.1.1. Holt-Winters multiplicativo

SN_t: patrón estacional
IR_t: componente irregular

Este método es adecuado cuando una serie de tiempo tiene una tendencia lineal con un patrón estacional multiplicativo para los que el nivel ($\beta_0 + \beta_1 t$), tasa de crecimiento (β_1), y el patrón estacional (SN_t) puede estar cambiando lentamente con el tiempo [11].

Estimación del nivel se muestra en la fórmula 6.3.3.1.2

$$\ell_T = \alpha(y_T / sn_{T-L}) + (1-\alpha)(\ell_{T-1} + b_{T-1})$$

Fórmula 6.3.3.1.2. Estimación de nivel.

Estimación de la tasa de crecimiento (o tendencia) se muestra en la fórmula 6.3.3.1.3

$$b_T = \gamma(\ell_T - \ell_{T-1}) + (1-\gamma)b_{T-1}$$

Fórmula 6.3.3.1.3. Estimación de tendencia.

Estimación del factor estacional se muestra en la fórmula 6.3.3.1.4

$$sn_T = \delta(y_T / \ell_T) + (1-\delta)sn_{T-L}$$

Fórmula 6.3.3.1.4. Estimación del factor estacional.

Donde α, γ y δ está alisando constantes entre 0 y 1, *L* = número de estaciones en un año (L = 12 para datos mensuales, y L = 4 para datos trimestrales) [11].

Para crear la predicción en el momento T para y_{T+p} se muestra en la fórmula 6.3.3.1.5:

$$\hat{y}_{T+p}(T) = (\ell_T + pb_T)sn_{T+p-L} \qquad (p = 1, 2, 3, \ldots)$$

Fórmula 6.3.3.1.5. Predicción

6.3.3.2 Aditivo

Este método es adecuado cuando una serie de tiempo tiene una tendencia lineal con una constante (aditiva) patrón estacional tal que el nivel ($\beta_0 + \beta_1 t$), tasa de crecimiento (β_1), y el patrón estacional (SN_t) pueden estar cambiando lentamente con el tiempo [11].

Por lo general, se considera que es el más adecuado para la previsión de una serie de tiempo que puede ser descrito por la ecuación que se muestra en la fórmula 6.3.3.2.1:

$$y_t = (\beta_0 + \beta_1 t) + SN_t + IR_t$$

Fórmula 6.3.3.2.1. Holt-Winters aditivo.

SN_t: patrón estacional
IR_t: componente irregular

Estimación del nivel se muestra en la fórmula 6.3.3.2.2 [11].

$$\ell_T = \alpha(y_T - sn_{T-L}) + (1-\alpha)(\ell_{T-1} + b_{T-1})$$

Fórmula 6.3.3.2.2. Estimación de nivel.

Estimación de la tasa de crecimiento (o tendencia) se muestra en la fórmula 6.3.3.2.3

$$b_T = \gamma(\ell_T - \ell_{T-1}) + (1-\gamma)b_{T-1}$$

Fórmula 6.3.3.2.3. Estimación de tendencia.

Estimación del factor estacional se muestra en la fórmula 6.3.3.2.4.

$$sn_T = \delta(y_T - \ell_T) + (1-\delta)sn_{T-L}$$

Fórmula 6.3.3.2.4. Estimación de nivel.

Para crear la predicción en el momento T para y_{T+p} se muestra en la fórmula 6.3.3.2.5

$$\hat{y}_{T+p}(T) = \ell_T + pb_T + sn_{T+p-L} \qquad (p = 1, 2, 3, \ldots)$$

Fórmula 6.3.3.2.5. Estimación de nivel.

En este proyecto de integración se utilizaron los métodos Holt-Winters aditivo y multiplicativo para que el usuario, una vez hecho el análisis correspondiente, seleccione el modelo que tenga la predicción más semejante a la realidad [11].

6.4 Modelos ARIMA

Los modelos ARIMA o modelos de promedio móvil autorregresivo integrado son un tipo general de los modelos de Box-Jenkins para series de tiempo estacionarias [12].

Este grupo incluye a los modelos:

- AR sólo con términos autorregresivos,
- MA sólo con términos de promedio móvil y
- ARIMA que comprenden tanto términos autorregresivos como de promedio móvil.

Para efectuar la selección del modelo apropiado se compara la distribución de los coeficientes de autocorrelación de la serie histórica que se está ajustando, con las distribuciones teóricas para los distintos modelos [12].

6.4.1 Modelo AR

En estos modelos los coeficientes de regresión se pueden estimar mediante 2 formas:

- El método de mínimos cuadrados lineal.
- El método de mínimos cuadrados no lineal.

Por lo regular, el método de mínimos cuadrados no lineal utiliza una técnica de solución iterativa para calcular los parámetros en vez de usar un cálculo directo. Se emplean estimaciones preliminares como puntos iniciales. Luego, estas estimaciones se mejoran sistemáticamente hasta encontrar valores óptimos [12].

En ocasiones pretendemos predecir el comportamiento de una variable *y* en un momento futuro *t*, a partir del comportamiento que la variable tuvo en un momento pasado, por ejemplo, en el período anterior, y_{t-1}. [12]

Formalmente se denota por

$y_t = f(y_{t-1})$

Es decir, que el valor de la variable y en el momento t es función del valor tomado en el período t-1.

Puesto que en el comportamiento de una variable influyen más aspectos, debemos incluir en la relación anterior un término de error, e_t. Este e_t es una variable aleatoria a la que suponemos ciertas características estadísticas apropiadas [12].
Es decir:

$y_t = f(y_{t-1}, e_t)$

En general, un modelo AR(p) viene dado por la ecuación que se muestra en la fórmula 6.4.1.1

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + e_t$$

Fórmula 6.4.1.1. Modelo AR.

Donde ϕ_0 es un término independiente y ϕ_1 es un parámetro que multiplica al valor de la variable *y* en el período t-1 y así sucesivamente [12].

6.4.2 Modelo MA

Una alternativa de modelización explicar el comportamiento de una variable *y,* no en función de los valores que tomó en el pasado (modelos AR) sino a través de los *errores* al estimar el valor de la variable en los períodos anteriores. Ello da lugar a los *modelos de medias móviles* [13].

En general un modelo MA(q) viene dado por la expresión que se muestra en la fórmula 6.4.2.1:

$$y_t = \mu + e_t + \theta_1 e_{t-1} + \ldots + \theta_q e_{t-q}$$

Fórmula 6.4.2.1. Modelo MA.

Al igual que ocurría con los modelos AR, en series con componente estacional es frecuente que el retardo coincida con la periodicidad de los datos [13].

6.4.3 Modelos ARMA

Entre los modelos AR y los Modelos MA existe una relación que, bajo ciertas condiciones, es útil conocer.
Los modelos ARMA integran a los modelos AR y a los modelos MA en una única expresión.
Por tanto, la variable *y* queda explicada en función de los valores tomados por la variable en períodos anteriores y los errores cometidos en la estimación [12].

Una expresión general de un modelo ARMA (p, q) viene dada por la expresión que se muestra en la fórmula 6.4.3.1

$$y_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + e_t + \theta_1 e_{t-1} + \ldots + \theta_q e_{t-q}$$

Fórmula 6.4.3.1. Modelo ARMA.

que, es la unión de un modelo AR(p) y un modelo MA(q). Donde μ es el valor constante alrededor del cual se mueve la variable, y ha de ser estimado igualmente con los coeficientes θ.

Obviamente, los modelos AR(p) se corresponden con modelo ARMA(p,0), mientras que los modelos MA(q) se corresponden con ARMA(0,q) [12].

6.4.4 Modelo ARIMA

Para la obtención de estimaciones con propiedades estadísticas adecuadas de los parámetros de un modelo ARMA, es necesario que la serie muestral que utilizamos para la estimación sea estacionaria en media y varianza.Para efectos prácticos, el cumplimiento de esta propiedad pasa por tomar logaritmos y diferenciar adecuadamente la serie original objeto de estudio [12].

Un Modelo Autorregresivo-Integrado de Medias Móviles de orden *p, d, q* abreviadamente ARIMA (*p,d,q*), no es más que un modelo ARMA (*p,q*) aplicado a una serie integrada de orden d, es decir, una serie a la que ha sido necesario diferenciar d veces para eliminar la tendencia [12].

Por lo tanto, la expresión general de un modelo ARIMA*(p,d,q)* viene dada por la expresión que se muestra en la fórmula 6.4.4.1

$$\Delta^d y_t = \phi_1 \Delta^d y_{t-1} + \ldots + \phi_p \Delta^d y_{t-p} + e_t + \theta_1 e_{t-1} + \ldots + \theta_q e_{t-q}$$

Fórmula 6.4.4.1. Modelo ARIMA.

Donde $\Delta^d y_t$, expresa que sobre la serie original y_t, se han aplicado *d* diferencias.

En la estimación de los modelos ARIMA el problema principal parte de identificar el modelo que mejor describe el fenómeno (la serie económica) a predecir, esto es, la clave de una buena predicción necesita determinar el más adecuado de los órdenes del autorregresivo, de la media móvil, y el orden de integrabilidad [12].

6.5 Paquetes utilizados

Los paquetes que se han utilizado en esta etapa del proyecto son:

- XLConnect

  Este paquete da un control programático de los archivos de Excel con el lenguaje de programación R. Una API de alto nivel permitiendo al usuario leer una página de un documento con formato .xlsx y escribirla en un data.frame.

  Permite la manipulación directa de hojas, filas y celdas, además de utilizar una biblioteca de java del proyecto Apache. Utiliza solo un subconjunto del proyecto Apache POI, todos los archivos jar necesarios se mantienen en xlsxjars, paquete que es importado al paquete xlsx [14].

- SQLDF

  Es un paquete para ejecutar sentencias SQL sobre data frames. Para este paquete se tiene que especificar una sentencia SQL utilizando de nombre de un data frame en lugar de los nombre de tablas y bases de datos, sí se tuviese que utilizar una base de datos, SQLDF Trabaja con SQLite, H2, PostgreSQL y MySQL [15].

  Con SQLDF el usuario se libera de tener que hacer lo siguiente, todas las cuales se realiza de forma automática:

  - Configuración de base de datos
  - Escribir la instrucción "create table" que define cada tabla
  - Importación y exportación desde y hacia la base de datos
  - Coacción de las columnas devueltas a la clase apropiada en casos comunes.

- gWidgets y tcltk2

  En el lenguaje de programación R existen varios paquetes como (RGtk2, tcltk, rJava, RwxWidgets, …) que permiten al usuario interactuar con este lenguaje de programación mediante herramientas GUI (Graphical User Interface).

  Proporciona herramientas independientes para realizar una interfaz e interactuar con la misma, mostrando al programador una manera simplificada de programación de una interfaz.

  En la figura 6.5.1 se muestran los paquetes que existen en el lenguaje R para realizar una interfaz de usuario [16].

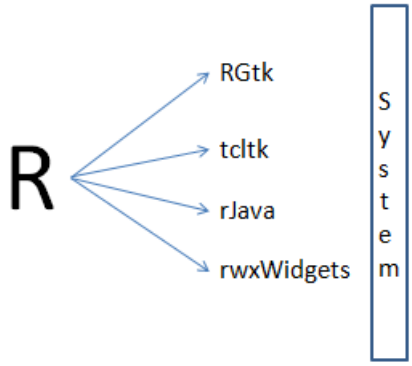


Figura 6.5.1. Paquetes existentes para la realización de una interfaz gráfica

En la Figura 6.5.2 se muestra la relación que tiene el paquete gWidgets con los demás paquetes, es el intermediario para establecer una comunicación entre los paquetes en forma de extensión y el lenguaje R

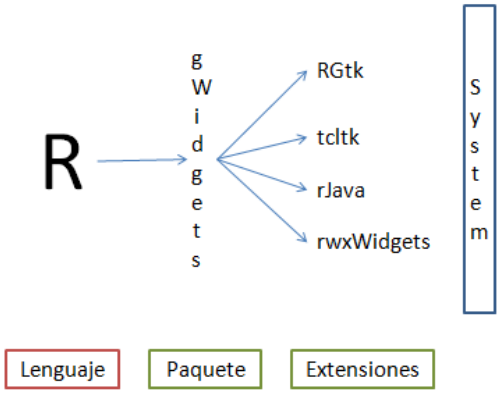


Figura 6.5.2. Relación

- Mice

Para el proceso de técnicas de datos faltantes necesitábamos una forma de asegurar la consistencia de los mismos afirmando que serán lo más parecidos a la realidad [17].

El paquete Mice cuenta con un algoritmo en el que se preserva:

- las relaciones entre datos.
- la incertidumbre acerca de estas relaciones.

- Forecast

En la elaboración de nuestro análisis de series de tiempo este paquete juega un papel fundamental, ya que existen métodos y herramientas para la visualización y el análisis de pronósticos de series de tiempo univariadas incluyendo suavizado exponencial a través de modelos de espacio de estado y la modelización automática ARIMA [18].

7. Desarrollo

En esta sección encontraremos como se elaboró cada módulo, desde la interfaz hasta su fin, como también encontraremos que paquetes se usaron en el proyecto así como las restricciones para asegurar el buen funcionamiento del mismo.

Aquí mostramos un diagrama de bloques del sistema, para tener una mayor claridad al explicarlo.

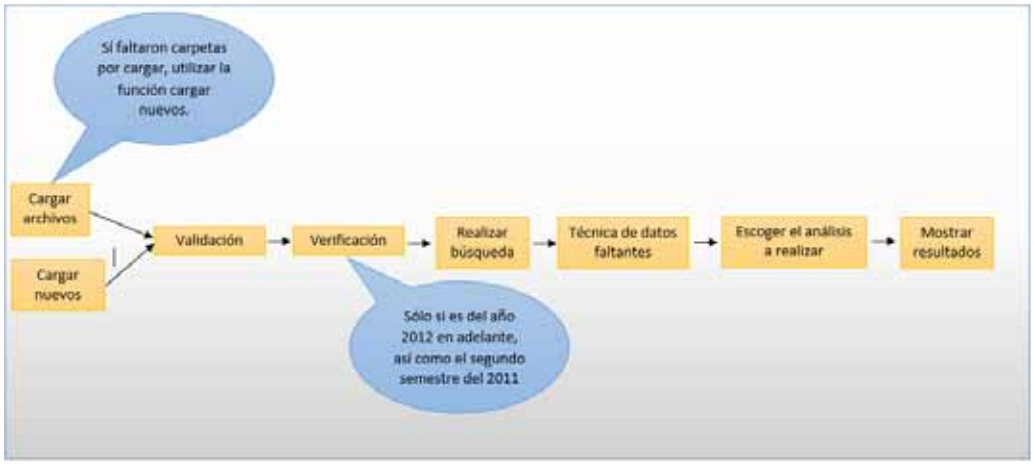


Figura 7.1. Diagrama de bloques del sistema en general

7.1 Interfaz de usuario

La interfaz de usuario (figura 7.1.1) fue diseñada teniendo en cuenta la experiencia del usuario, siendo muy intuitiva.

Figura 7.1.1. Interfaz grafica

Esta interfaz fue realizada con el paquete gWidgets.

Como podemos observar en la parte del menú tenemos 5 botones, estos son:

- Cargar todos: carga todos los archivos contenidos en todas las carpetas.
- Cargar nuevos: carga solo los archivos contenidos en la carpeta seleccionada.
- Suavizado exponencial: realiza el análisis aditivo y multiplicativo.
- Modelo ARIMA: realiza el análisis de los modelos ARIMA.
- Cerrar: cierra nuestra interfaz.

Existen 2 formas de cargar archivos:

- *Cargar todos*: Sí existiesen archivos previamente cargados, se borran y empieza a cargar todos los archivos contenidos en nuestra carpeta raíz "RedAutomaticaMonitoreoAtmosferico", extrayendo los archivos acerca de los contaminantes antes mencionados de cada una de las carpetas contenidas. Al utilizar este botón la única carpeta que no toma en cuenta es "NO_BORRAR", hablaremos después de esa carpeta.

- *Cargar nuevos*: Esta opción es especialmente para elegir que carpeta se desea ingresar al sistema. Para cargar una carpeta en específico se deberá hacer click en el botón "Cargar nuevos" donde nos muestra una interfaz amigable para el usuario para elegir la carpeta que se deseé cargar (figura 7.1.2).

  Sí se desea realizar un análisis acerca de la última búsqueda se deberá oprimir el botón del análisis que deseen llevar acabo.

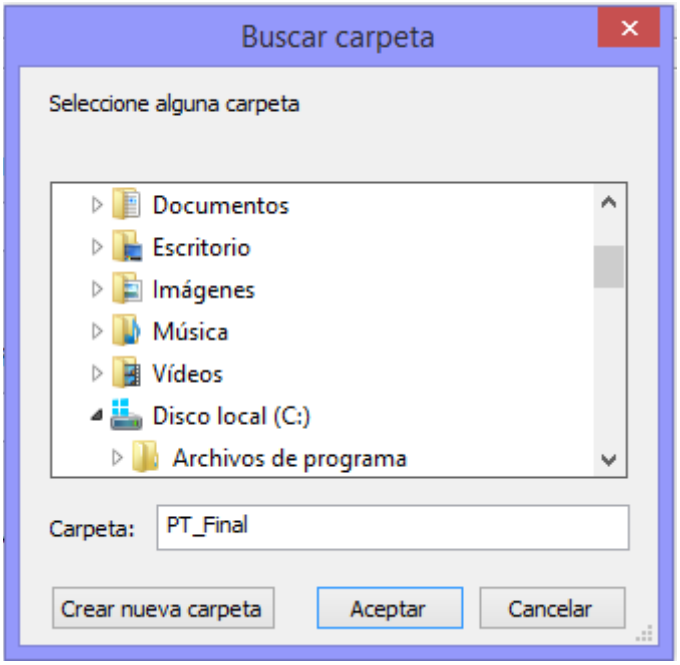


Figura 7.1.2. Selección de carpetas específicas

Siempre se debe tener una carpeta llamada "RedAutomaticaMonitoreoAtmosferico" dentro de la dirección "C:/" en donde se encuentran las subcarpetas con el nombre "RAMA" seguida de

un número que corresponde al año, ej. "RAMA14" para leer la carpeta del 2014. Dentro de éstas se encuentran seis archivos correspondientes a los contaminantes que se mencionaron con anterioridad referentes al año de la carpeta que los contiene. El nombre de cada archivo es de la siguiente manera, ej. "2000CO.xls", con lo cual, se tiene presente el año al que pertenece así como el contaminante al que los datos contenidos se refieren.

El panel llamado "Por Fecha" y "Por Semana" sirven para realizar búsquedas como su nombre lo indica (Véase figura 7.1.3).

Existen los siguientes casos a considerar:

- La carpeta raíz es "RedAutomaticaMonitoreoAtmosferico".
- La carpeta raíz deberá en la dirección "C:/".
- Sí se ingresan años que no sean continuos, el sistema no funcionara. Siempre se debe llevar un orden en los años.
- Sí se presiona el botón de algún análisis sin haber realizado una búsqueda previa el sistema marca un error.

Para realizar una búsqueda, primero se deberá cargar las carpetas con las que deseé trabajar, seguido de esto, se irán agregando años en nuestra fecha de inicio (Véase figura 7.1.3).

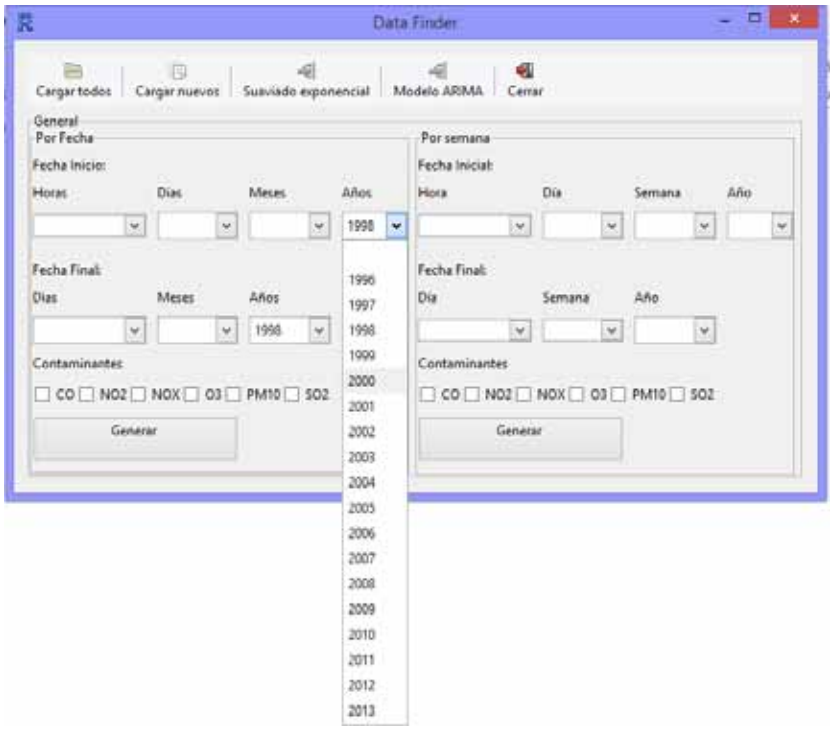


Figura 7.1.3. Años desplegables después de cargar los archivos

Una vez que ya este cargado al menos 1 año, es decir, el mismo año en cada una de nuestras variables, aparecerá el año en nuestra interfaz. Si seleccionamos un año a través de la fecha inicial se seleccionará el mismo año para la fecha final teniendo opción de elegir un año mayor (figura 7.1.4).

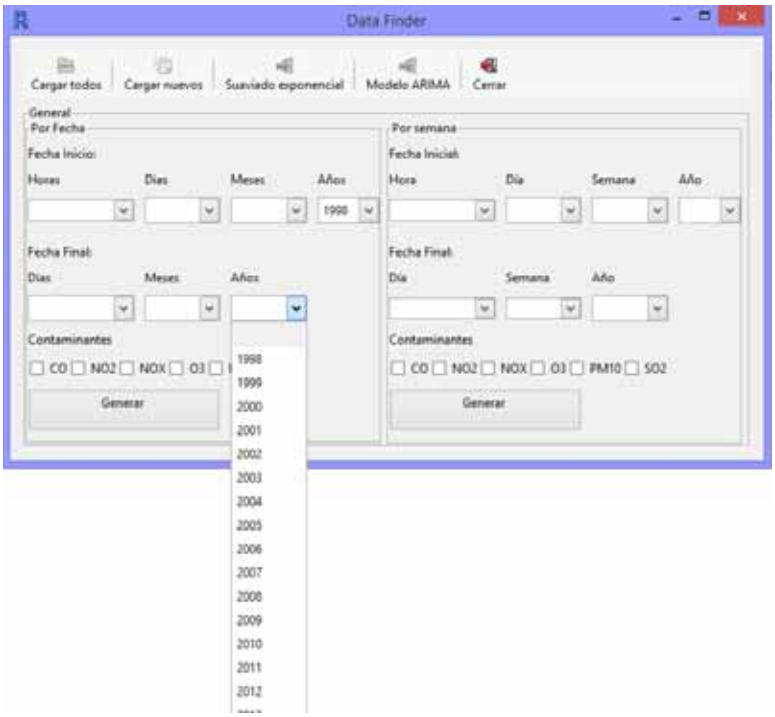


Figura7.1.4. Relación entre la fecha inicial y final

Una vez que se haya elegido el año inicial nos mostrará los 12 meses de cada año. Cuando el usuario seleccione uno, en la sección de mes final se podrá seleccionar ese mismo mes o uno mayor (figura 7.1.5).

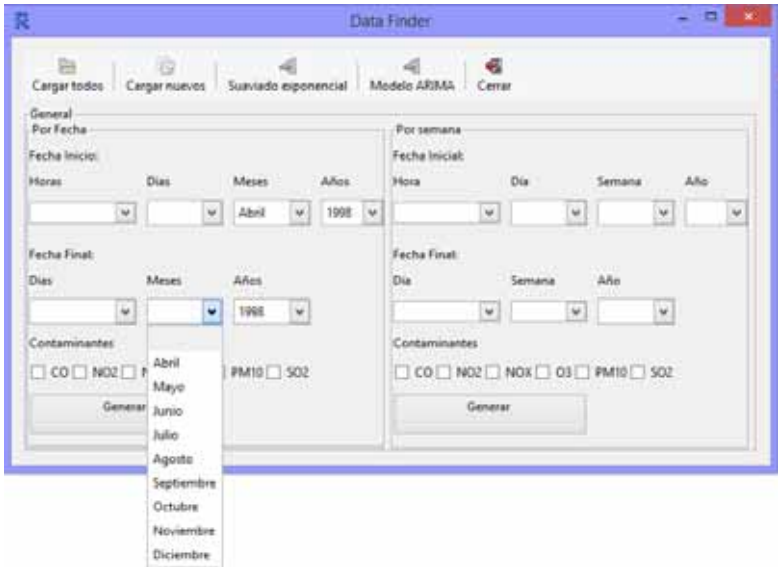


Figura 7.1.5. Relación de meses

En caso que no se especifiquen los parámetros de mes se tomará por defecto Enero como mes inicial y Diciembre como mes final.

En la opción de días se seleccionará el día inicial y el día final será igual o mayor que el día inicial que seleccionamos. Si alguno de estos parámetros se omite se seleccionará "01" para el día inicial y "31" para el día final si no se escogió un mes, en caso contrario se tomará el último día del mes seleccionado en fecha final.

Sí la búsqueda se hace por semanas, es prudente especificar que sólo se hará ésta en el mismo año desde la primera semana que tenga todos los días dentro del mismo hasta la última con la misma característica.

Lectura de Archivos

En este módulo se utilizó el paquete "XLConnect" para hacer la lectura y escritura sobre archivos en formato .xls o .csv [14].

Lo primero que se realizó en este módulo fue poner por defecto la dirección raíz en donde se encuentran las carpetas que contienen los archivos a analizar, la dirección es la siguiente: "C:\RedAutomaticaMonitoreoAtmosferico", seguido de seleccionar cada una de las carpetas existentes, sin tomar en cuenta la carpeta con nombre: "NO_BORRAR", más adelante se explicará con detalle.

Una vez que tenemos el nombre de todas las carpetas, concatenamos la dirección raíz seguida del nombre de la carpeta. Posteriormente, se extraen los dos últimos caracteres del nombre de cada carpeta, con éstos aseguramos que los archivos que están almacenados en las mismas corresponden al mismo año, en caso contrario se mostrará un error, (no se encuentra el archivo a leer), de igual manera, tienen que estar los 6 archivos de los contaminantes, en caso donde existan menos, nos mostrará un mensaje esperando que el usuario coloque el archivo a leer en la carpeta correspondiente.

La forma de los archivos se muestra en la figura 7.1.6.

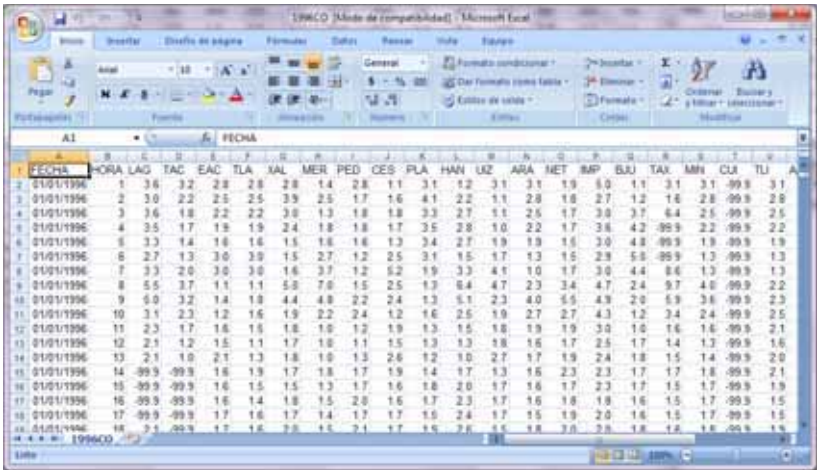


Figura 7.1.6. Archivos leídos

La carpeta "NO_BORRAR" contiene un archivo llamado "no_tocar.csv" (figura 7.1.7), el cual ayuda en la lectura, ya que los archivos en formato .xls muestran alguna dificultad al trabajar con

ellos directamente con el paquete sqldf [15], por lo cual, se tuvo la necesidad de generar el archivo "no tocar.csv" para la escritura del archivo cargado con anterioridad. Una vez realizado lo anterior se leyó éste para tener el fácil manejo de los datos con el paquete antes mencionado.

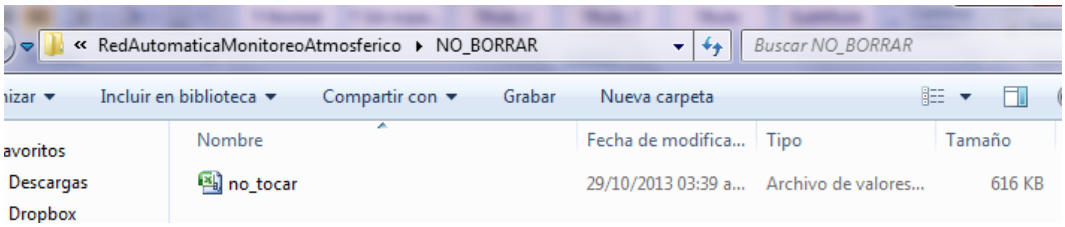


Figura 7.1.7. Archivo no modificable

Es importante mencionar que en el momento en que se carga cada archivo pasa por los módulos de preprocesamiento:

- Validación.
- Verificación.

De ellos se hablará más en el siguiente módulo.

Se realizaron las búsquedas necesarias para poder obtener valores más específicos de todos los datos recolectados, para estas búsquedas se usaron sentencias SQL gracias al paquete "sqldf".

Figura 7.1.8 Búsqueda por fecha

En las búsquedas por fechas (figura 7.1.8), al momento de seleccionar una fecha de inicio y una de fin, el programa automáticamente crea sentencias como las siguientes:

Por ejemplo:

Búsqueda en el mismo archivo:

SELECT * FROM tabla WHERE FECHA < “2000-01-01” AND FECHA >”2000-03-31”

Búsqueda de varios archivos (01-Enero-2000 al 03-Febrero-2002):

(Año 2000)
SELECT * FROM tabla WHERE FECHA > “2000-01-01”, seguida de,
(Año 2001)
SELECT * FROM tabla WHERE FECHA, seguida de,
(Año 2002)
SELECT * FROM tabla WHERE FECHA <”2002-02-03”

Existen los siguientes casos a considerar:

- Si falta algún año, el sistema marca error.
- Tiene por defecto Enero para fecha de inicio y Diciembre para fecha final
- Detecta los meses introducidos y pone por defecto en el día de inicio “01” y calcula el último día del mes final para nuestras búsquedas.

En las búsquedas por semanas, sólo existe un tipo de búsqueda y es cuando tienen el mismo año ejemplo:

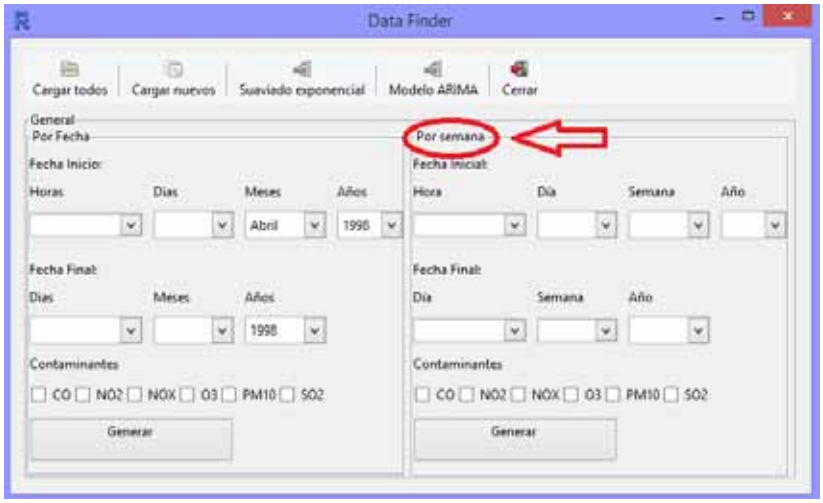


Figura 7.1.9. Búsqueda por semana

Para una búsqueda como la que se muestra del lado derecho de la figura 7.1.9 el programa calcula cual es la semana 6 del año elegido teniendo en cuenta lo siguiente:

- La semana inicia en domingo y termina en sábado.
- No se cuenta como semana sí existen días de otros años en esta.

Al especificar el día de inicio y el día final, el sistema buscará todos los días que estén contenidos en ese lapso.

En caso que no se especifiquen, el sistema por defecto tomara el día de inicio el domingo y el día final el sábado.

Una vez realizada alguna de las búsquedas anteriores pasamos al submódulo "técnicas de datos faltantes" del módulo Preprocesamiento de datos.

En la figura 7.1.10 se muestra un diagrama de cada paso a seguir para la elaboración de este proyecto, en la parte superior se muestra que representa cada símbolo.

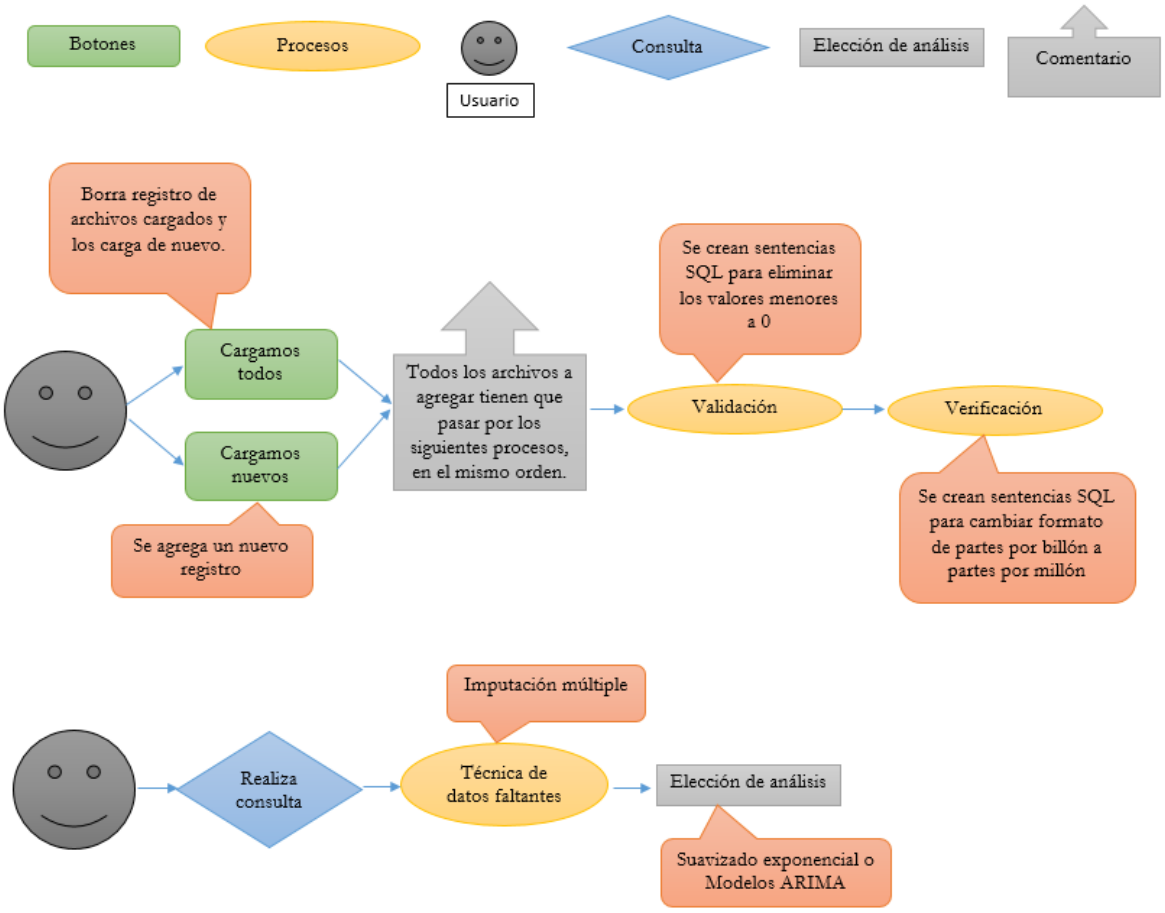
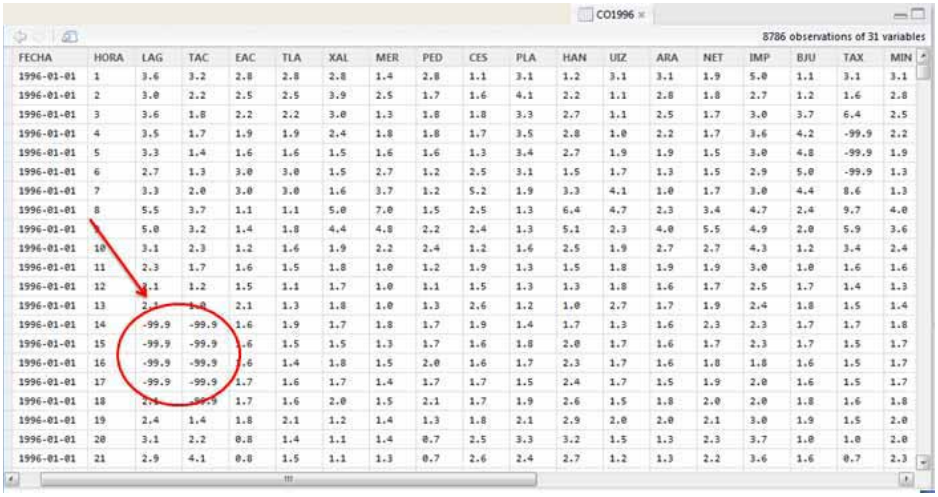


Figura 7.1.10 Diagramas

7.2 Preprocesamiento de datos.

7.2.1 Validación

Este módulo verifica la validación de los datos que se ingresaron, esto es, si existiese una celda que tuviese "NA" o un número menor a 0 (Véase figura 7.2.1.1) se tendrá que cambiar por un 0 para no alterar ningún resultado.



CO1996

8786 observations of 31 variables

FECHA	HORA	LAG	TAC	EAC	TLA	XAL	MER	PED	CES	PLA	HAN	UIZ	ARA	NET	IMP	BJU	TAX	MIN
1996-01-01	1	3.6	3.2	2.8	2.8	2.8	1.4	2.8	1.1	3.1	1.2	3.1	3.1	1.9	5.0	1.1	3.1	3.1
1996-01-01	2	3.0	2.2	2.5	2.5	3.9	2.5	1.7	1.6	4.1	2.2	1.1	2.8	1.8	2.7	1.2	1.6	2.8
1996-01-01	3	3.6	1.8	2.2	2.2	3.0	1.3	1.8	1.8	3.3	2.7	1.1	2.5	1.7	3.0	3.7	6.4	2.5
1996-01-01	4	3.5	1.7	1.9	1.9	2.4	1.8	1.8	1.7	3.5	2.8	1.0	2.2	1.7	3.6	4.2	-99.9	2.2
1996-01-01	5	3.3	1.4	1.6	1.6	1.5	1.6	1.6	1.3	3.4	2.7	1.9	1.9	1.5	3.0	4.8	-99.9	1.9
1996-01-01	6	2.7	1.3	3.0	3.0	1.5	2.7	1.2	2.5	3.1	1.5	1.7	1.3	1.5	2.9	5.0	-99.9	1.3
1996-01-01	7	3.3	2.0	3.0	3.0	1.6	3.7	1.2	5.2	1.9	3.3	4.1	1.0	1.7	3.0	4.4	8.6	1.3
1996-01-01	8	5.5	3.7	1.1	1.1	5.0	7.0	1.5	2.5	1.3	6.4	4.7	2.3	3.4	4.7	2.4	9.7	4.0
1996-01-01	9	5.0	3.2	1.4	1.8	4.4	4.8	2.2	2.4	1.3	5.1	2.3	4.0	5.5	4.9	2.0	5.9	3.6
1996-01-01	10	3.1	2.3	1.2	1.6	1.9	2.2	2.4	1.2	1.6	2.5	1.9	2.7	2.7	4.3	1.2	3.4	2.4
1996-01-01	11	2.3	1.7	1.6	1.5	1.8	1.0	1.2	1.9	1.3	1.5	1.8	1.9	1.9	3.0	1.0	1.6	1.6
1996-01-01	12	2.1	1.2	1.5	1.1	1.7	1.0	1.1	1.5	1.3	1.3	1.8	1.6	1.7	2.5	1.7	1.4	1.3
1996-01-01	13	2.1	1.0	2.1	1.3	1.8	1.0	1.3	2.6	1.2	1.0	2.7	1.7	1.9	2.4	1.8	1.5	1.4
1996-01-01	14	-99.9	-99.9	1.6	1.9	1.7	1.8	1.7	1.9	1.4	1.7	1.3	1.6	2.3	2.3	1.7	1.7	1.8
1996-01-01	15	-99.9	-99.9	1.6	1.5	1.5	1.3	1.7	1.6	1.8	2.0	1.7	1.6	1.7	2.3	1.7	1.5	1.7
1996-01-01	16	-99.9	-99.9	1.6	1.4	1.8	1.5	2.0	1.6	1.7	2.3	1.7	1.6	1.8	1.8	1.6	1.5	1.7
1996-01-01	17	-99.9	-99.9	1.7	1.6	1.7	1.4	1.7	1.7	1.5	2.4	1.7	1.5	1.9	2.0	1.6	1.5	1.7
1996-01-01	18	2.1	-99.9	1.7	1.6	2.0	1.5	2.1	1.7	1.9	2.6	1.5	1.8	2.0	2.0	1.8	1.6	1.8
1996-01-01	19	2.4	1.4	1.8	2.1	1.2	1.4	1.3	1.8	2.1	2.9	2.0	2.0	2.1	3.0	1.9	1.5	2.0
1996-01-01	20	3.1	2.2	0.8	1.4	1.1	1.4	0.7	2.5	3.3	3.2	1.5	1.3	2.3	3.7	1.0	1.0	2.0
1996-01-01	21	2.9	4.1	0.8	1.5	1.1	1.3	0.7	2.6	2.4	2.7	1.2	1.3	2.2	3.6	1.6	0.7	2.3

Figura 7.2.1.1. Datos sin tratamiento

La elaboración de este módulo se realizó por medio del paquete sqldf [15].

Este módulo se ejecuta cuando leemos cada archivo pasando como parámetros de entrada el data frame en el que se guardó lo leído del archivo.

Acto seguido se deberán extraer los nombres de todas las columnas de nuestro data frame[5]. Antes de empezar a hacer una sentencia SQL, necesitamos delimitar desde que columna se podrá empezar la limpieza de datos. Esta decisión se toma dependiendo de la columna llamada *Hora*, después de esa columna todos los archivos tienen las estaciones que contienen los datos a limpiar.

Puede llegar el caso en el cual tenga columnas que se denominaron "Basura", estas aparecen cuando existen problemas para leer un archivo, estas columnas tienen las siguientes siglas: "X, X_1, X_2,…". Nuestro módulo está diseñado para no tomar en cuenta estas columnas basura.

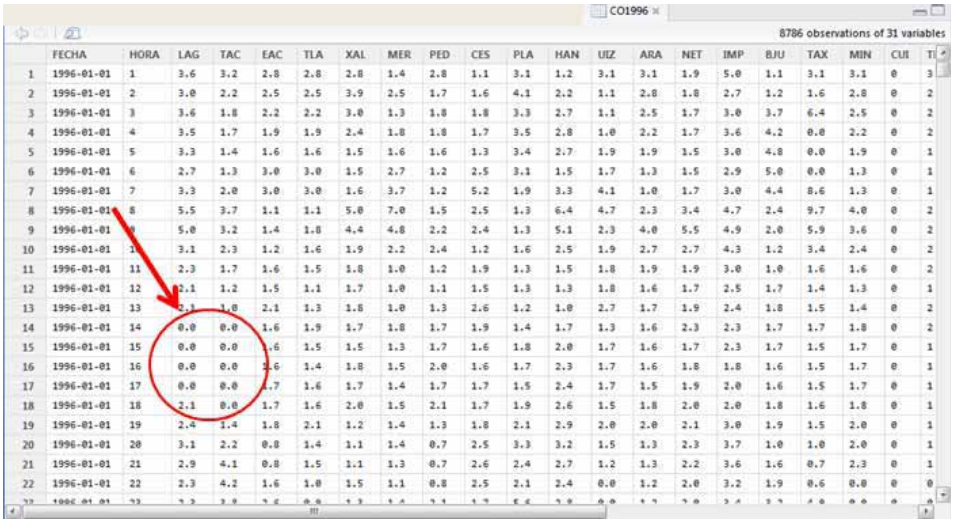
Una vez que tenemos todos los nombres de las columnas empezamos a hacer una sentencia UPDATE con el nombre de todas las columnas de tal forma que la sentencia queda de esta manera:

sqldf(c("UPDATE tabla SET columna = 0 WHERE columna < 0 ", "SELECT * FROM tabla"))

[5] "Data.frame" se utiliza para almacenar las tablas de datos y tener un mejor manejo de ellas en el lenguaje de programación R.

Esta sentencia se repetirá para cada columna que se encuentre en el archivo.

En la figura 7.2.1.2 se muestra el resultado después de la validación.



	FECHA	HORA	LAG	TAC	EAC	TLA	XAL	MER	PED	CES	PLA	HAN	UIZ	ARA	NET	IMP	BJU	TAX	MIN	CUI	TI
1	1996-01-01	1	3.6	3.2	2.8	2.8	2.8	1.4	2.8	1.1	3.1	1.2	3.1	3.1	1.9	5.0	1.1	3.1	3.1	0	3
2	1996-01-01	2	3.0	2.2	2.5	2.5	3.9	2.5	1.7	1.6	4.1	2.2	1.1	2.8	1.8	2.7	1.2	1.6	2.8	0	2
3	1996-01-01	3	3.6	1.8	2.2	2.2	3.0	1.3	1.8	1.8	3.3	2.7	1.1	2.5	1.7	3.0	3.7	6.4	2.5	0	2
4	1996-01-01	4	3.5	1.7	1.9	1.9	2.4	1.8	1.8	1.7	3.5	2.8	1.0	2.2	1.7	3.6	4.2	0.0	2.2	0	2
5	1996-01-01	5	3.3	1.4	1.6	1.6	1.5	1.6	1.6	1.3	3.4	2.7	1.9	1.9	1.5	3.0	4.8	0.0	1.9	0	1
6	1996-01-01	6	2.7	1.3	3.0	3.0	1.5	2.7	1.2	2.5	3.1	1.5	1.7	1.3	1.5	2.9	5.0	0.0	1.3	0	1
7	1996-01-01	7	3.3	2.0	3.0	3.0	1.6	3.7	1.2	5.2	1.9	3.3	4.1	1.0	1.7	3.0	4.4	8.6	1.3	0	1
8	1996-01-01	8	5.5	3.7	1.1	1.1	5.0	7.0	1.5	2.5	1.3	6.4	4.7	2.3	3.4	4.7	2.4	9.7	4.0	0	2
9	1996-01-01	9	5.0	3.2	1.4	1.8	4.4	4.8	2.2	2.4	1.3	5.1	2.3	4.0	5.5	4.9	2.0	5.9	3.6	0	2
10	1996-01-01	10	3.1	2.3	1.2	1.6	1.9	2.2	2.4	1.2	1.6	2.5	1.9	2.7	2.7	4.3	1.2	3.4	2.4	0	2
11	1996-01-01	11	2.3	1.7	1.6	1.5	1.8	1.0	1.2	1.9	1.3	1.5	1.8	1.9	1.9	3.0	1.0	1.6	1.6	0	2
12	1996-01-01	12	2.1	1.2	1.5	1.1	1.7	1.0	1.1	1.5	1.3	1.3	1.8	1.6	1.7	2.5	1.7	1.4	1.3	0	1
13	1996-01-01	13	2.1	1.8	2.1	1.3	1.8	1.0	1.3	2.6	1.2	1.0	2.7	1.7	1.9	2.4	1.8	1.5	1.4	0	2
14	1996-01-01	14	0.0	0.0	1.6	1.9	1.7	1.8	1.7	1.9	1.4	1.7	1.3	1.6	2.3	2.3	1.7	1.7	1.8	0	2
15	1996-01-01	15	0.0	0.0	1.6	1.5	1.5	1.3	1.7	1.6	1.8	2.0	1.7	1.6	1.7	2.3	1.7	1.5	1.7	0	1
16	1996-01-01	16	0.0	0.0	1.6	1.4	1.8	1.5	2.0	1.6	1.7	2.3	1.7	1.6	1.8	1.8	1.6	1.5	1.7	0	1
17	1996-01-01	17	0.0	0.0	1.7	1.6	1.7	1.4	1.7	1.7	1.5	2.4	1.7	1.5	1.9	2.0	1.6	1.5	1.7	0	1
18	1996-01-01	18	2.1	0.0	1.7	1.6	2.0	1.5	2.1	1.7	1.9	2.6	1.5	1.8	2.0	2.0	1.8	1.6	1.8	0	1
19	1996-01-01	19	2.4	1.4	1.8	2.1	1.2	1.4	1.3	1.8	2.1	2.9	2.0	2.0	2.1	3.0	1.9	1.5	2.0	0	1
20	1996-01-01	20	3.1	2.2	0.8	1.4	1.1	1.4	0.7	2.5	3.3	3.2	1.5	1.3	2.3	3.7	1.0	1.0	2.0	0	1
21	1996-01-01	21	2.9	4.1	0.8	1.5	1.1	1.3	0.7	2.6	2.4	2.7	1.2	1.3	2.2	3.6	1.6	0.7	2.3	0	1
22	1996-01-01	22	2.3	4.2	1.6	1.0	1.5	1.1	0.8	2.5	2.1	2.4	0.0	1.2	2.0	3.2	1.9	0.6	0.0	0	0

Figura 7.2.1.2. Datos tratados para técnicas de datos faltantes.

7.2.2 Verificación

En esta parte del módulo de preprocesamiento se verificó que los datos ingresados a partir de julio del 2011 sean puestos es partes por millón, ya que están expresados por partes por billón (figura 7.2.2.1).

La conversión que se realizó en este módulo es la siguiente:

$1ppm^{6} = 1/10^{6} = 10^{-6}$
$1ppb^{7} = 1/10^{9} = 10^{-9}$
Por lo cual
1ppm = 1000ppb

Para este módulo se necesita mandar un data frame y el nombre del mismo, éstos se consiguieron con anterioridad en la función donde concatenamos el nombre de cada archivo para obtener una cadena de toda la dirección en donde se encuentra.

También concatenamos una sentencia SQL ahora será una sentencia SELECT, ésta será de siguiente manera:

sqldf("SELECT columna/1000, columna/1000, … FROM tabla")

[6] ppm (partes por millón).
[7] ppb (partes por billón)

Esta sentencia fue ejecutada gracias al paquete sqldf.

Se hizo un filtrado de información para comprobar la existencia de columnas basura, sí se llegase a encontrar alguna, la aplicación no las tomará en cuenta.

En la figura 7.2.2.1 se muestran los datos una vez realizada la verificación.

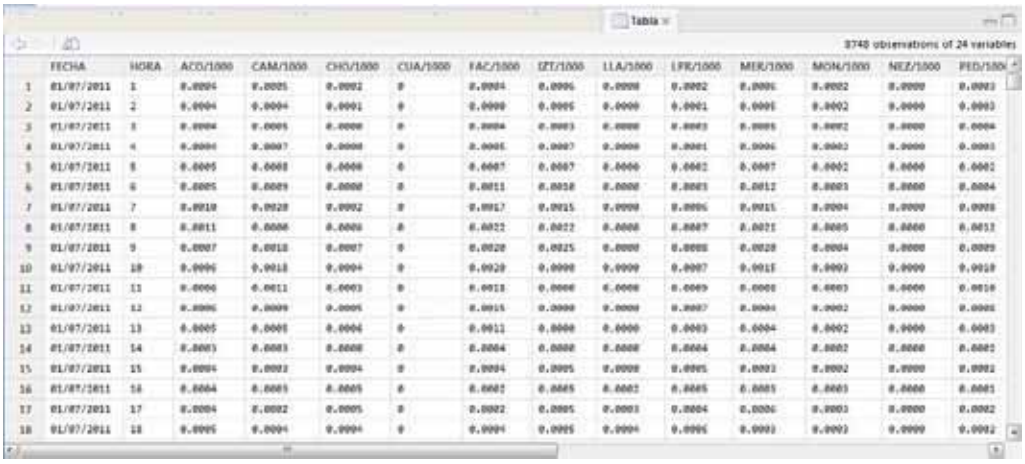


Figura 7.2.2.1. Datos expresados en partes por millón

7.2.3 Técnicas de datos faltantes

En este submódulo se utilizó la técnica de imputación múltiple, este proceso se lleva a cabo una vez que se ejecutó una búsqueda por el usuario. Partiendo de ésto, se establecieron como "NA" (valor nulo en el lenguaje de programación R) a todos los valores a los que se le asignó 0 al momento de la lectura de información.

Cuando empieza el completado de datos faltantes, primero se realiza la suma de cuantas celdas tienen valores "NA" en cada columna, después seleccionamos

El modelo de imputación múltiple deberá contener tantas celdas que no tengan datos faltantes como sea posible, ya que cuanto mayor sea el número de estas celdas, mayor es la cantidad de información a partir de la cual se hacen estimaciones.

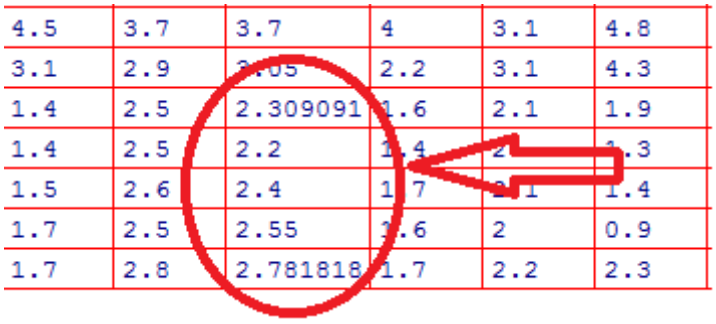
En este proceso se crean 5 imputaciones para cada valor faltante, esto es porque a pesar de ser un método de varias imputaciones tiene presente la coincidencia media predicha, por lo cual, a los valores de estas 5 imputaciones se les sacará la media y se pondrá en su lugar correspondiente donde no existía un dato. Si se eligiesen más imputaciones nuestro valor resultante no variará demasiado.

iter	imp	variable					
1	1	XAL	CES	BJU	TLI	SAG	AZC
1	2	XAL	CES	BJU	TLI	SAG	AZC
1	3	XAL	CES	BJU	TLI	SAG	AZC
1	4	XAL	CES	BJU	TLI	SAG	AZC
1	5	XAL	CES	BJU	TLI	SAG	AZC
2	1	XAL	CES	BJU	TLI	SAG	AZC
2	2	XAL	CES	BJU	TLI	SAG	AZC
2	3	XAL	CES	BJU	TLI	SAG	AZC
2	4	XAL	CES	BJU	TLI	SAG	AZC
2	5	XAL	CES	BJU	TLI	SAG	AZC

Figura 7.2.3.1. Columnas que se trataron con imputación múltiple

En la figura 7.2.3.1 se muestra un ejemplo de las iteraciones e imputaciones que se hace a las estaciones de medición de contaminantes representadas en las columnas de cada archivo, en total se realizaron cinco iteraciones de 5 imputaciones cada una.

Mientras que existiesen datos en la misma columna se predijeron bajo esta técnica, para los datos que no se pudieron predecir se decidió asignarles la media de su propia fila, no alterando así su valor más bajo y más alto. Véase la figura 7.2.3.2.



4.5	3.7	3.7	4	3.1	4.8
3.1	2.9	3.05	2.2	3.1	4.3
1.4	2.5	2.309091	1.6	2.1	1.9
1.4	2.5	2.2	1.4	2	1.3
1.5	2.6	2.4	1.7	2.1	1.4
1.7	2.5	2.55	1.6	2	0.9
1.7	2.8	2.781818	1.7	2.2	2.3

Figura7.2.3.2. Promedio para valores no calculados

Este proceso se desarrolla al momento en que el usuario realice una búsqueda.

7.3 Proceso de minería de datos

La frecuencia es muy importante en las series de tiempo, para predecir el comportamiento de la contaminación en un lapso determinado, se desarrolló la fórmula que se muestra en la figura 7.3.1, ya que los métodos de análisis de tiempo siempre son mejores a corto plazo, así, por cada medio mes se predijo un día.

```
diferencia = fecha_inicial-fecha_final
diferencia = diferencia/15
if(diferencia %% 1 != 0){
    diferencia = diferencia -(diferencia %% 1)
}
diferencia = diferencia*24
diferencia
```

Figura 7.3.1. Fórmula de frecuencia

Antes de pasar a los análisis que se utilizaron para este proyecto es importante mencionar que una vez teniendo los datos del rango que se escogió para la búsqueda, se convirtieron en series de tiempo, ya que si no se elaboraba esto, nuestros análisis no se hubieran concretado.

Se convirtieron en series de tiempo sacando el promedio de cada una de las filas resultantes y almacenando todo en un solo vector. También se experimentó utilizando el valor máximo y mínimo por fila.

7.3.1 Suavizado exponencial

Como se puede ver en la figura 7.3.1 primero se obtuvo la diferencia entre las fechas ingresadas, después se dividió entre 15, ya que es la media promedio. A este resultado le quitamos todo lo que esté a la derecha del punto decimal y lo multiplicamos por 24, que son las horas que tenemos por cada día.

Para este análisis se necesitó sacar el mínimo, máximo y el promedio de cada una de las filas del archivo resultante, por defecto se trabajó con el valor mínimo de cada fila, esto es, para hacer series de tiempo y poder manejar los datos de una mejor manera.

Una vez que tenemos un solo vector, aplicamos el método Holt-Winters, este método es utilizado cuando se quiere un análisis con estacionalidad aditiva o multiplicativa [19].

En este tipo de análisis se intenta recrear los valores desde el comienzo de la serie de tiempo, por lo cual a pesar de tener los valores "reales" nos muestra los valores que se van prediciendo para que así como frecuentemente sucede el usuario final decida si la estimación se apega a la realidad y decidir el mejor método de estimación.

Para no errar en este modelo, se decidió que genere los 2 tipos de suavizado exponencial (Aditivo y Multiplicativo), con la finalidad que el usuario pueda observar las diferencias y distinguir con facilidad, qué modelo es el más adecuado a los datos previamente obtenidos.

Teniendo en mente que el tipo multiplicativo se presenta cuando la magnitud del patrón estacional se incrementa conforme los valores aumentan y decrece cuando los valores de los datos disminuyen, y el aditivo es mejor cuando el patrón estacional en los datos no depende del valor de los datos o el patrón estacional no cambia conforme la serie se incrementa o disminuye, los resultados casi siempre saldrán mejor estimados en el tipo aditivo, a menos que existan cambios muy significativos en la tendencia por lo que los valores de la serie de tiempo subirían o bajarían [20].

El paquete con el que se realizó este modelo es "forecast" [18], la función Holt-Winters es la intérprete para este tipo de análisis. Estimando los valores alfa, beta y gamma hace que el resultado sea una estimación más precisa [21]. Además de esta función existen otras igual de poderosas como lo es ("SES y ETS"), pero la función "SES" solo es para suavizado exponencial simple y la función ETS hace lo mismo que la función Holt-Winters sólo que un poco más alejada a la realidad [18].

Una vez que ya tenemos la estimación nos dedicamos a hacer la predicción de los siguientes valores a los tomados. Esta predicción se realizó para días posteriores de nuestro rango que determinamos como frecuencia [22].

7.3.2 Modelos ARIMA

En este modelo se tienen que hacer varias pruebas para saber qué modelo es el mejor.

Para saber los valores de *(p, d, q)* tuvimos que probar diferentes combinaciones siempre buscando tener parámetros AIC (Akaike, Criterio de Información); sobre los que se deberán ajustar los modelos, siempre eligiendo el modelo con el AIC más pequeño [23].

El paquete que se utilizó fue "forecast", la función auto.arima () en R utiliza una variación del algoritmo de Hyndman y Khandakar, para el modelado automático ARIMA [24], en este algoritmo el número de diferencias *d* se determina mediante pruebas KPSS[8] repetidas. Con esta función se calcula cualquiera de los modelos ARIMA (AR, MA, ARMA y ARIMA) [25].

Los valores de *p* y *q* son entonces elegidos por reducir al mínimo el AIC después de diferenciación de los tiempos de los datos d. En lugar de considerar cada combinación posible de p y q, el algoritmo utiliza una búsqueda paso a paso para atravesar el espacio modelo [19].

(A) El mejor modelo (con AIC más pequeño) se selecciona casi siempre entre los cuatro siguientes:

ARIMA (2, *d*, 2),
ARIMA (0, *d*, 0),
ARIMA (1, *d*, 0),
ARIMA (0, *d*, 1).

El rango por default de *p* y *q* es $0 \leq p \leq 5$ and $0 \leq q \leq 5$.

La función auto.arima() automatiza la inclusión de una constante. De forma predeterminada, para $d = 0$ o $d = 1$, una constante se incluirá si mejora el valor de AIC; para $d > 1$ la constante siempre se omite, un ejemplo de esta se muestra en la figura 7.3.2.1.

[8] Pruebas KPSS se utilizan para probar una hipótesis nula de que una serie de tiempo observable es estacionaria.

```
> auto.arima(x)
Series: x
ARIMA(2,1,2)

Call: auto.arima(x = x)

Coefficients:
         ar1      ar2      ma1     ma2
      0.0451  -0.9183  -0.0066  0.6178
s.e.  0.0249   0.0252   0.0496  0.0517

sigma^2 estimated as 0.9877:  log likelihood = -708.45
AIC = 1426.9   AICc = 1427.02   BIC = 1447.98
```

Figura7.3.2.1. Análisis del modelo ARIMA

También nos podremos dar cuenta que sugiere una buena estimación, cuando revisamos los residuos después del análisis y encontramos ruido blanco.

Sí probásemos con otros modelos, esto es escogiendo diferentes *(p, d, q)* los valores resultantes en (AIC) serían más altos, por lo que no tendríamos un modelo lo mejor estimado posible.

7.4 Mostrar resultados

Aunque más adelante se hablará acerca de los resultados obtenidos, aquí mostramos el proceso para llevarlo a cabo.

Una vez realizado alguno de los análisis, se elaboraron gráficas para cada uno de los contaminantes seleccionados previamente a la búsqueda, además de archivos en formato .xls.

Esta gráfica y archivo .xls sirven para la interpretación de nuestro modelo a seguir ya que con la gráfica se puede observar si tiene tendencia, sí la predicción realizada fue buena, regular o mala dependiendo de qué tanto se adecua a nuestra serie de tiempo y demás. Sí quedase alguna duda, en el archivo .xls de cada contaminante viene la forma en que se comportó numéricamente.

7.5 Restricciones

Las restricciones que se deben tener en cuenta para el correcto funcionamiento de esta aplicación son:

- Carpeta llamada "RedAutomaticaMonitoreoAtmosferico" en el directorio "C:" por lo cual la dirección total sería de esta manera: "C:\RedAutomaticaMonitoreoAtmosferico".

- Dentro de la carpeta “RedAutomaticaMonitoreoAtmosferico” debe haber carpetas con nombre compuesto de la siguiente manera: “RAMA” y el año al que pertenece como se muestra a continuación: RAMA98, RAMA99, RAMA00, etc.

- Dentro de las carpetas descritas en el punto anterior se deben encontrar todos los archivos con extensión .xls, estos archivos deben tener la siguiente estructura año completo y el contaminante, por ejemplo: 2000CO.xls.

Existe una carpeta llamada “NO_BORRAR” que como su nombre lo indica, no deberán borrar por ningún motivo, en caso de hacerlo, deberán crear una carpeta con el mismo nombre manualmente dentro de la carpeta raíz.

8. Resultados y su análisis

Una vez realizado el análisis elegido por el usuario, en nuestra carpeta de “NO_TOCAR” (figura 8.1) aparecerá una carpeta nueva con un número, esté número es para saber cuál búsqueda fue la última.

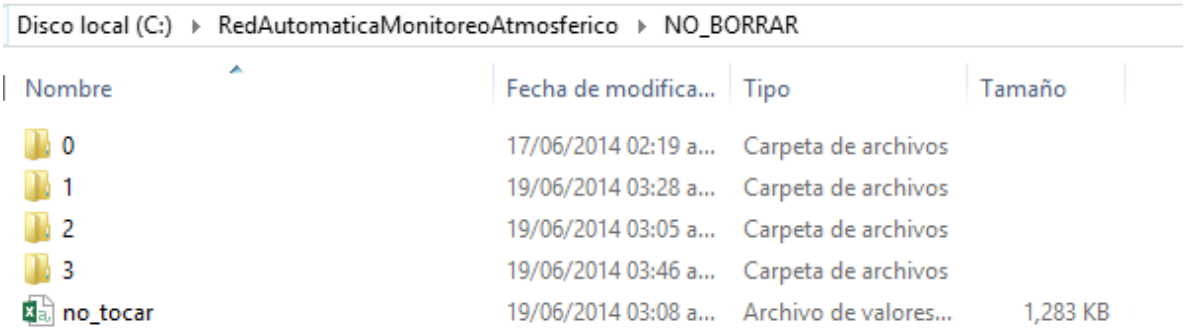


Figura 8.1. Numeración de carpetas resultantes

Dentro de cada una de las carpetas encontramos una gráfica por cada contaminante, como también un documento con formato .xls, el cual, contendrá información relevante acerca del análisis hecho. Cada una de estas gráficas que contiene la predicción realizada, muestra también la forma en la cual la serie de tiempo se fue comportando (figura 8.2).

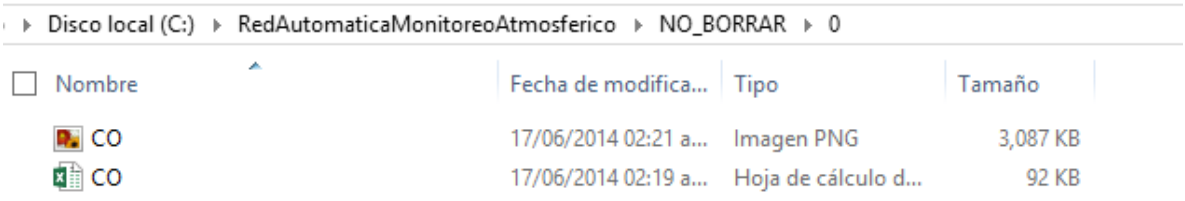


Figura 8.2. Composición de carpetas resultantes

Al realizar todas las búsquedas por rango semanal, por días o por meses siempre se tomaron en cuenta todos los valores de cada archivo, dando así más fiabilidad para que los resultados sean más exactos.

8.1 Suavizado exponencial

Como resultado de una búsqueda de enero a febrero del año 2000 se obtuvo la gráfica de la figura 8.1.1.

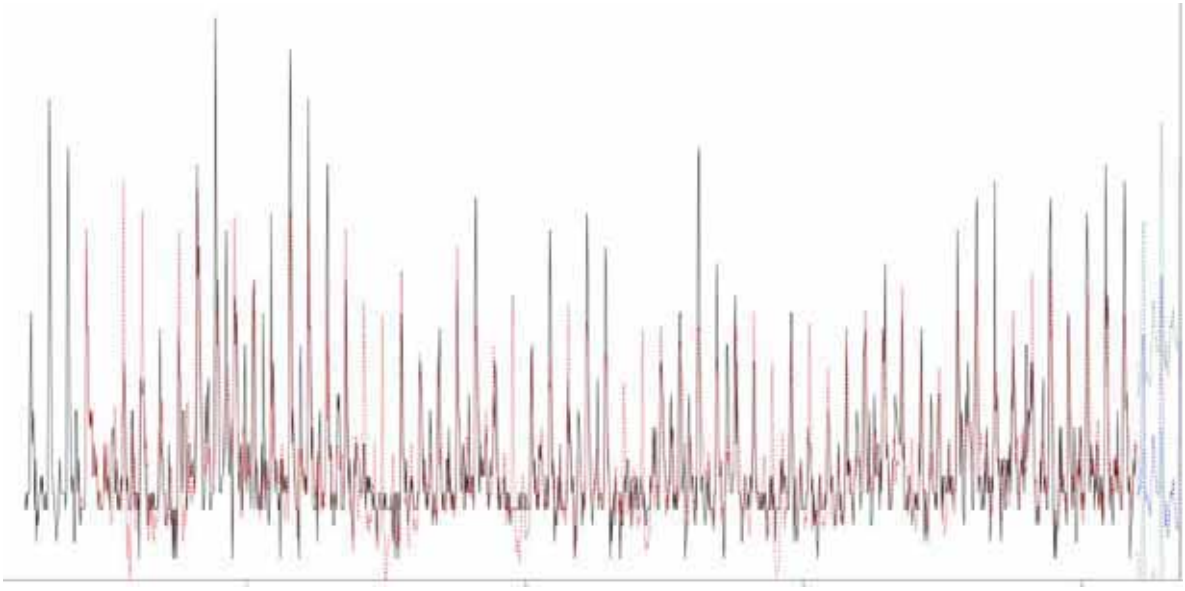


Figura 8.1.1. Gráfica resultante del método suavizado exponencial

Lo que se puede observar de esta gráfica es:

- La línea negra son los valores como se fueron presentando en la serie de tiempo.
- La línea roja son los valores predichos, como se puede observar, se va comportando parecido a la gráfica normal.
- La línea azul nos indica el valor predicho para el futuro.
- Las líneas verdes nos indican el mayor y menor valor predicho en cada uno de los puntos, como se puede ver, la línea verde inferior está por debajo de la gráfica en general, por lo cual se presentaron valores negativos que no pueden ser ciertos.

En los documentos con formato .xls que se presenten encontraremos los valores (figura 8.1.2):

- Alfa: parámetro del filtro de Holt-Winters Filtro.
- Beta: parámetro del filtro de Holt-Winters Filtro. Si se establece en FALSE, la función hará suavizado exponencial.
- Gamma: parámetro utilizado para el componente estacional. Si se establece en FALSE, un modelo no estacional se cabe.

Estos valores fueron calculados automáticamente para una mayor precisión.

matrix(c("	V1
Alpha	0.219448798331426
Betha	0
Gamma	0.311037527092593
Seasonal	additive
SEE	120.363248855771

Figura 8.1.2. Valores expresados en el archivo .xls

Para la gráfica pasada, los valores observados fueron los que se muestran en la figura 8.1.2. El valor de alfa (0,21) es relativamente baja, lo que indica que la estimación del nivel en el punto de tiempo actual se basa en dos o tal vez cuatro observaciones recientes y algunas observaciones en el pasado más lejano. El valor de beta es 0.00, lo que indica que la estimación de la pendiente *b* de la componente de tendencia no se actualiza sobre la serie de tiempo, y en su lugar se establece igual a su valor inicial. Esto tiene sentido, ya que el nivel cambia bastante en la serie temporal, pero la pendiente *b* de la componente tendencia sigue siendo más o menos lo mismo, que sería 0 ya que la pendiente es casi totalmente horizontal. El valor de gamma (0.31) es relativamente bajo, lo que indica que la estimación de la componente estacional en el punto de tiempo actual se basa sólo en observaciones poco recientes.

Sí existiesen más dudas acerca del análisis antes dicho podemos corroborar la decisión de asignar beta = 0 en la pestaña de análisis aditivo y análisis multiplicativo, ya que ahí se puede notar que la tendencia siempre es la misma.

Las estacionalidades, por su parte, intentan reflejar movimientos ligados a las influencias específicas de las horas, que corresponde a elementos. La estacionalidad que se muestra en ese mismo análisis es muy baja respecto cada uno de los valores, ya que no varía mucho del 0.

La columna "xhat" muestra los valores que fueron tomados para mostrarlos como una predicción a valores ya establecidos para corroborar si se adapta al modelo real. Sí los valores de esta columna al momento de graficar son muy parecidos a lo que podría ser o siempre siguen la tendencia de la gráfica, significa que se ha realizado un modelo que puede predecir bien los siguientes datos.

Otros datos importantes son los valores que se fueron presentando, como es la tendencia llamada "trend" que muestra si aumenta o desciende el contaminante a lo largo del tiempo, la temporada llamada "season", el nivel llamado "level" y los datos que se muestran en la gráfica anterior con línea roja llamados "xhat", estos datos son presentados al usuario para poder ver con base en números como se fue comportando el modelo (figura 8.1.3).

1	xhat	level	trend	season
2	0.399986	0.760056	-0.00113	-0.35894
3	0.476915	0.736985	-0.00113	-0.25894
4	0.459601	0.718977	-0.00113	-0.25825
5	0.444698	0.704768	-0.00113	-0.25894
6	0.428899	0.69383	-0.00113	-0.2638
7	0.413789	0.686359	-0.00113	-0.27144
8	0.5013	0.682204	-0.00113	-0.17977
9	1.092247	0.680789	-0.00113	0.412587
10	2.087263	0.681361	-0.00113	1.407031
11	1.488235	0.683027	-0.00113	0.806337
12	1.491771	0.68448	-0.00113	0.80842

Figura8.1.3. Análisis de datos mostrados en la gráfica

Aparte se muestran los datos predichos en otra hoja del documento de Excel donde se expresa el valor mínimo como "lwr", máximo como "upr" y el predicho como "fit", como se muestra en la figura 8.1.4.

1	fit	upr	lwr
2	0.708455	1.289964	0.126946
3	0.705883	1.301229	0.110537
4	0.467794	1.076663	-0.14108
5	0.511827	1.133925	-0.11027
6	0.526283	1.161335	-0.10877
7	0.519609	1.167356	-0.12814
8	0.897268	1.557465	0.23707
9	1.33934	2.011758	0.666923
10	1.439154	2.123574	0.754735

Figura8.1.4. Parámetros de predicción

8.2 Modelo ARIMA

Como resultado de una búsqueda de la semana 3 a la semana 5 del año 2000 se obtuvo la gráfica de la figura 8.2.1.

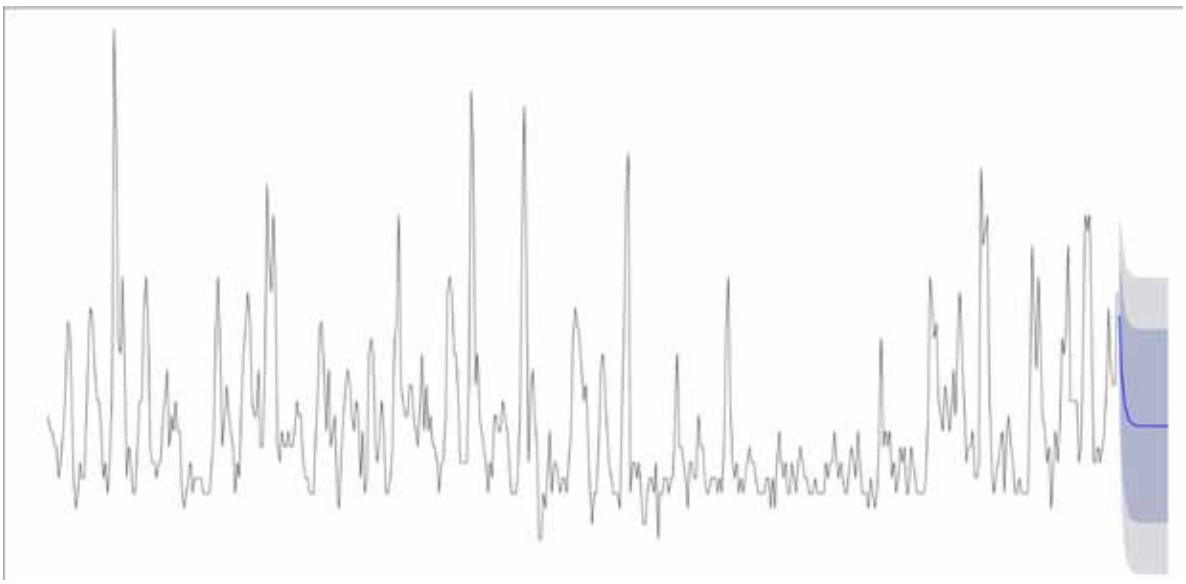


Figura8.2.1. Grafica de comportamiento y predicción

- La línea negra dice como se comportó la línea de tiempo.
- La línea azul muestra la predicción de la tendencia que seguirá en nuestro siguiente plazo de tiempo.
- La línea gris fuerte nos muestra que hay un 95% de fiabilidad que la serie de tiempo se comporte en ese rango y un 80% en la el área gris claro.

Gracias a la figura 8.2.1, podemos observar que la tendencia casi siempre mantiene una constante por lo cual podemos ver que existe cierta estacionalidad. Para cualquier grafica que mantenga un punto medio en el eje Y como su centro, significa que existe por lo general estacionalidad, sí encontramos una gráfica que tenga tendencia, o sea, que con forme pasa el tiempo aumente o disminuye su valor entonces es una gráfica no estacionaria.

Las predicciones que se realizan en este modelo eventualmente convergerán con la media o promedio y luego se quedarán ahí u oscilarán en valores muy cercanos a ella.

En la realización de este modelo se encontró la problemática de buscar un modelo que se ajustara lo mejor posible a los datos obtenidos de las bases de datos. Cuando se desarrolló el algoritmo, para saber si era un buen modelo el que se escogió, los datos arrojados no debieron ser demasiado grandes ya que esto implicaba buscar posibles soluciones a este problema, siendo la

mejor el paquete en R llamado "Forecast", en el que se encuentra una función denominada auto.arima, la cual nos ayuda a encontrar un mejor modelo que analiza la serie de tiempo.

Al momento de hacer pruebas con diferentes funciones Arima como son: "Arima, arima, auto.arima" se observó que todos los valores eran prácticamente iguales, por lo cual se asegura que el modelo escogido es verídico pero no hay manera de saber si es el mejor. En dado caso que hubiera una variación significativa se tendría que volver a escoger un nuevo modelo. Sí utilizamos en vez del método auto.arima, otro método que se mencionó en este mismo párrafo, la respuesta sería más rápida, aunque sí no se sabe el orden los resultados variarán considerablemente.

Acerca de cuantos valores se hicieron para la predicción, se escogieron por la fórmula de la figura 7.3.1.

En el documento de Excel se muestra las variables Log likelihood, Aic, AICc. Estas variables nos ayudan para saber cuál es el mejor modelo ARIMA ya que mientras más pequeño es el valor Aic, es un mejor modelo.

1	variable	V1
2	Log likelihood	-166.0375033
3	AIC	340.0750066
4	AICc	340.1551669

Figura 8.2.2. Variables a considerar

De igual manera mostramos los datos de la predicción hecha para el rango de 80% y 95% de fiabilidad con el valor promedio.

1	80% menor	95% menor	estimado	80% mayor	95% mayor
2	1.113942338	0.88591986	1.544687061	1.975431784	2.203454267
3	0.643767622	0.33887027	1.219732426	1.79569723	2.100594579
4	0.431360739	0.10730765	1.043511592	1.655662444	1.979715531
5	0.325555823	-0.00391877	0.947948153	1.570340484	1.899815081
6	0.270752453	-0.06029962	0.896124708	1.521496963	1.852549037
7	0.241775282	-0.08973927	0.868021182	1.494267083	1.825781637
8	0.226278227	-0.10537221	0.852780819	1.479283411	1.81093385
9	0.217938006	-0.11375238	0.844516067	1.471094128	1.802784517
10	0.213433892	-0.11826825	0.840034145	1.466634397	1.798336535

Figura8.2.3. Predicciones del modelo ARIMA

Además mostramos los residuos después de la predicción ya que si encontramos "ruido blanco" es altamente probable que la predicción sea buena, en la figura 8.2.4 podemos ver como se presenta el ruido blanco.

	residuals
2	0.0448721604548783
3	-0.0783488799886814
4	0.010316156664624
5	-0.119393776266107
6	-0.0204459067246674
7	-0.254601559940435
8	0.0347243755225126
9	0.080572820240729
10	0.156279691450961

Figura8.2.4. Ruido Blanco

8.3 General

Otro ejemplo más ilustrativo es el análisis de algunos los contaminantes en el 2010 que se adaptaron al modelo ARIMA, mostramos las gráficas a continuación.

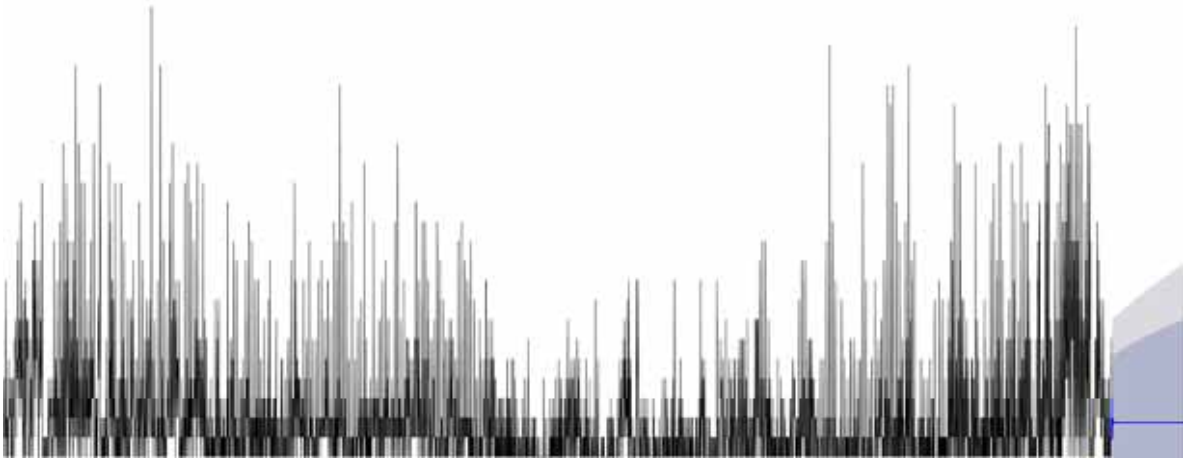


Figura 8.3.1 Gráfica de CO del año 2010

Como se puede apreciar en la figura 8.3.1 existe una ligera tendencia creciente como decreciente en el inicio de la predicción, después se estabiliza en un punto. Si queremos corroborar esto vamos al archivo con nombre CO figura 8.3.2 y verificamos lo observado en la gráfica anterior.

	80% menor	95% menor	promedio	80% mayor	95% mayor
2	0.174355	0.055333	0.399191	0.624028	0.743049
3	0.013395	-0.14264	0.308146	0.602897	0.758929
4	-0.08813	-0.26122	0.238858	0.565843	0.738939
5	-0.14118	-0.32042	0.197415	0.536009	0.71525
6	-0.15438	-0.33484	0.186511	0.527404	0.707861
7	-0.14331	-0.32377	0.197584	0.538479	0.718938
8	-0.12085	-0.30165	0.220673	0.562201	0.742994
9	-0.0958	-0.27717	0.246814	0.589427	0.770795
10	-0.07386	-0.2556	0.269468	0.612795	0.794541
11	-0.05848	-0.24031	0.285006	0.628492	0.810322
12	-0.05096	-0.2328	0.292535	0.636034	0.817871
13	-0.05066	-0.23268	0.293163	0.636991	0.819003
14	-0.05553	-0.23796	0.289092	0.633711	0.816141
15	-0.06294	-0.24595	0.282773	0.628487	0.811497
16	-0.07054	-0.25415	0.276306	0.623154	0.806764
17	-0.07671	-0.26084	0.271114	0.618937	0.803063
18	-0.0807	-0.26522	0.267867	0.616436	0.800958
19	-0.08251	-0.26732	0.266596	0.615706	0.800514
20	-0.08261	-0.26763	0.266903	0.616411	0.80143

Figura 8.3.2 Predicción de CO

Este promedio se estabiliza después de varios resultados, mostrando que en inicios del 2011 podría haber una alteración con este contaminante. Acto seguido, se verifican los residuos para encontrar ruido blanco (figura 8.3.3), y se encuentran valores negativos. Cuando se encuentren valores negativos en nuestro ruido blanco, se tomarán como positivos, recordando un poco lo

mostrado con anterioridad, sí se encuentra ruido blanco en los residuos es altamente probable que el modelo utilizado sea correcto.

	residuals
2	0.000899999448493496
3	-0.0903297766470973
4	0.00373683609787618
5	-0.457606070282476
6	0.0976701738634336
7	-0.13319588379267
8	-0.156872488569414
9	-0.084297952267221
10	-0.200048710405625

Figura 8.3.3 Residuos de CO

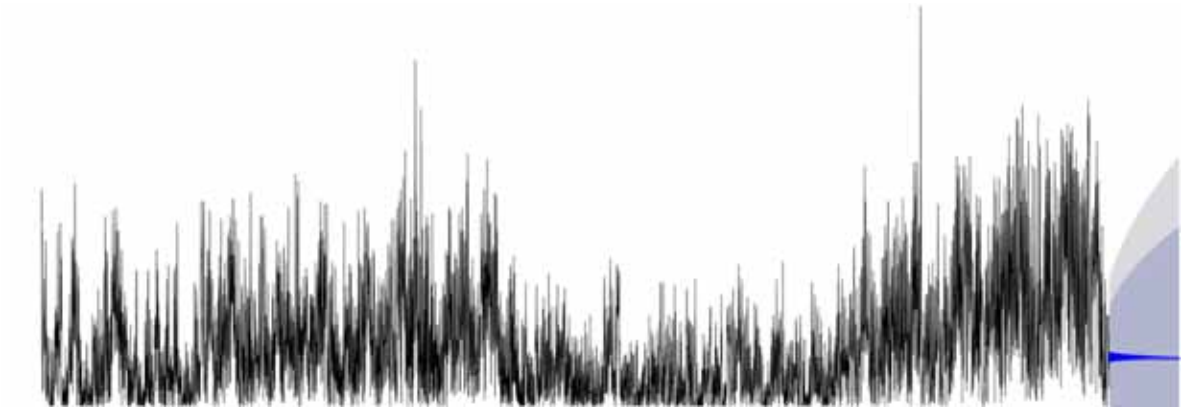


Figura 8.3.4 Gráfica de PM10 del año 2010

En la figura 8.3.4 se muestra la gráfica de las partículas suspendidas. Como se puede apreciar existe la probabilidad que crezca de una manera si no alarmante si de cuidado. La tendencia es casi 0, aunque el promedio varía al principio de la predicción, esto indica que podría existir un aumento peligroso en los niveles. Ya que como se muestra antes de la predicción existió una baja abrupta del contaminante. Verificamos la predicción realizada en la figura 8.3.5.

	80% menor	95% menor	promedio	80% mayor	95% mayor
2	1.895256	-5.0976	15.10507	28.31488	35.30773
3	-0.38044	-8.69756	15.33097	31.04238	39.3595
4	-0.90829	-9.75927	15.81163	32.53154	41.38253
5	-0.6563	-9.75235	16.52654	33.70939	42.80543
6	-0.062	-9.27668	17.34493	34.75187	43.96654
7	0.563024	-8.71126	18.08256	35.60209	44.87637
8	0.981154	-8.32617	18.5631	36.14505	45.45237
9	1.045826	-8.28478	18.67175	36.29768	45.62828
10	0.71916	-8.63467	18.38896	36.05876	45.41258
11	0.069602	-9.31467	17.7969	35.52421	44.90848
12	-0.7519	-10.1798	17.05788	34.86766	44.29559
13	-1.55151	-11.0391	16.371	34.2935	43.78111
14	-2.14077	-11.7011	15.91917	33.97911	43.53947
15	-2.38506	-12.0226	15.82077	34.0266	43.66419
16	-2.24117	-11.9498	16.09879	34.43876	44.14736
17	-1.77179	-11.5373	16.67565	35.12309	44.88857
18	-1.1293	-10.9354	17.39479	35.91888	45.72494
19	-0.5123	-10.3456	18.06317	36.63865	46.47191
20	-0.1087	-9.96114	18.503	37.11469	46.96713

Figura 8.3.5 Predicción PM10

Se observa que a inicios del año 2011 hay una gran probabilidad de un aumento de este contaminante en la Ciudad de México, Acto seguido, se verifican los residuos para encontrar ruido blanco (figura 8.3.6).

	residuals
2	0.0620007861835454
3	9.89619166328007
4	-0.168235037087436
5	-46.8818284158006
6	-11.5011239580686
7	32.5460518067903
8	-36.2430157880184
9	-10.6578435812127
10	-11.3276998309972

Figura 8.3.6 Residuos de PM10

Observamos que si se encuentra ruido blanco por lo cual, nuestro pronóstico es altamente probable que sea verídico.

Los resultados de cada análisis que se realizó, muestran una tendencia casi nula por cada año, con la cual, hace pensar que siempre seguirá la misma trayectoria, lo cual es falso. Las tendencias a pesar que suben o bajan mínimamente, por lo general se compensan en el año siguiente, o no muestran una tendencia clara. Se puede decir que la contaminación a pesar de ser un problema al cual se dedica tiempo y recursos, siempre se mantiene constante, aunque con tendencia a subir, esto se puede deber por diversos factores en la ciudad.

Existieron contaminantes que no se adecuaron al modelo ARIMA como son Óxido de nitrógeno NOX y Ozono O3. Como no se acoplaron al modelo mencionado, se analizaron por el método de suavizado exponencial. En la figura 8.3.7, 8.3.8 y 8.3.9 se muestran las gráficas aditiva, multiplicativa y el modelo ARIMA del contaminante NOX. En las figuras En las 8.3.10, 8.3.11 y 8.3.12 se muestran las gráficas del comportamiento del contaminante llamado ozono O3.

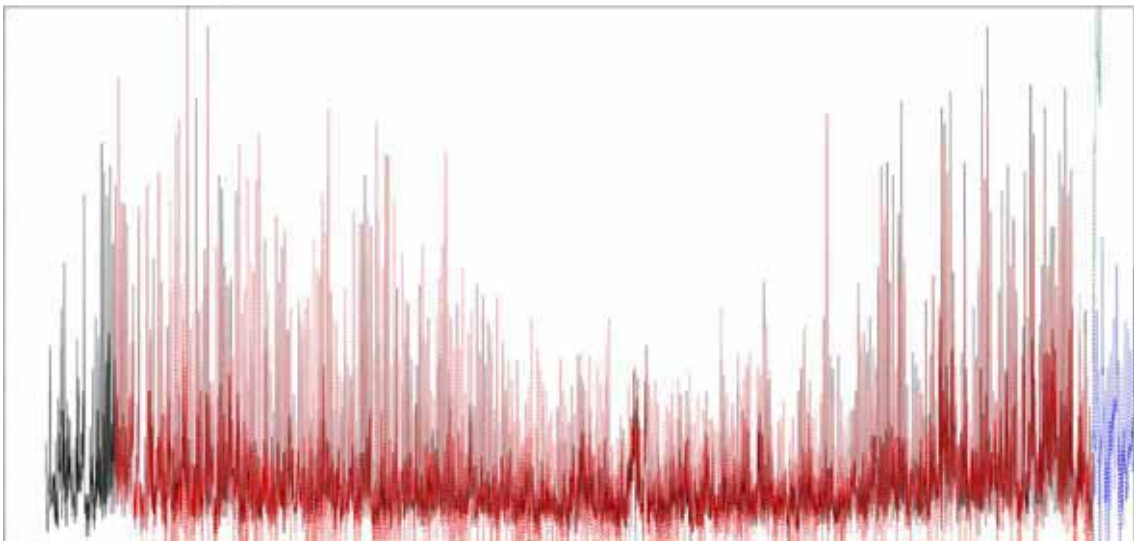


Figura 8.3.7 Gráfica aditiva de NOX

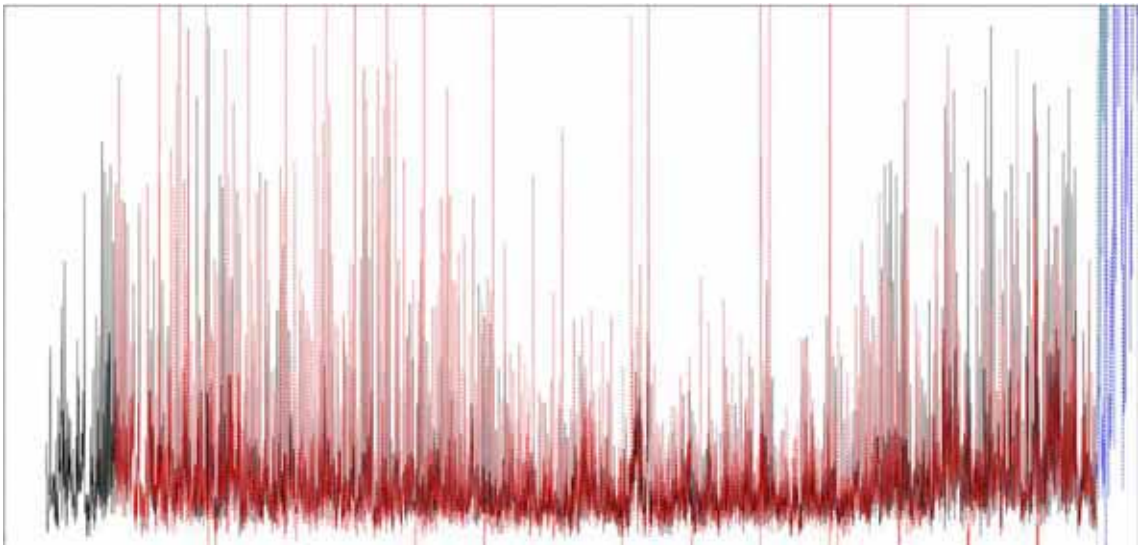


Figura 8.3.8 Gráfica multiplicativa de NOX

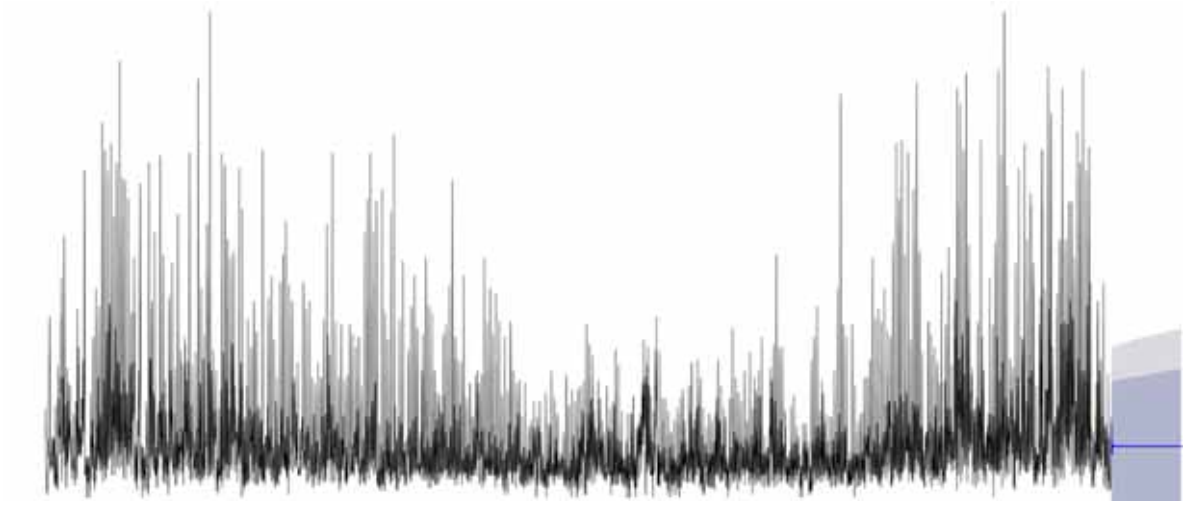


Figura 8.3.9 Gráfica sobre el modelo ARIMA de NOX

Como se muestra en la figura 8.3.7 el modelo aditivo predijo mejor el comportamiento del contaminante en comparación del método multiplicativo figura 8.3.8 y el modelo ARIMA 8.3.9, ya que se logró adecuar de una mejor manera a la gráfica.

1	matrix(c(" V1	
2	Alpha	0.751781392775665
3	Betha	0
4	Gamma	1
5	Seasonal	additive
6	SEE	1.20424819992563

Figura 8.3.10 Tabla del modelo aditivo de NOX

La tabla de la figura 8.3.10 referente a la figura 8.3.6 muestra que el valor de alfa (0.75) es relativamente alta, lo que indica que la estimación del nivel en el punto de tiempo actual se basa varias observaciones recientes. El valor de beta es 0.00, lo que indica que la estimación de la pendiente *b* de la componente de tendencia no se actualiza sobre la serie de tiempo, y en su lugar se establece igual a su valor inicial. Esto tiene sentido, ya que el nivel cambia bastante en la serie temporal, pero la pendiente *b* de la componente tendencia sigue siendo más o menos lo mismo, que sería 0 ya que la pendiente es casi totalmente horizontal. El valor de gamma (1) es relativamente alto, lo que indica que la estimación de la componente estacional en el punto de tiempo actual se basa sólo en observaciones recientes.

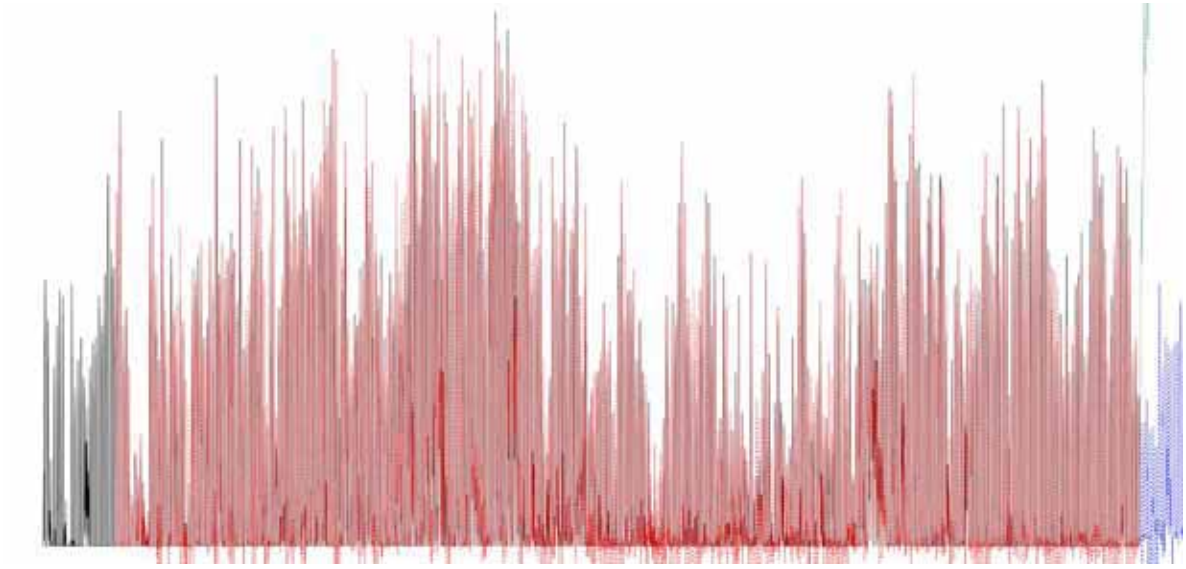


Figura 8.3.11 Gráfica aditiva de O3

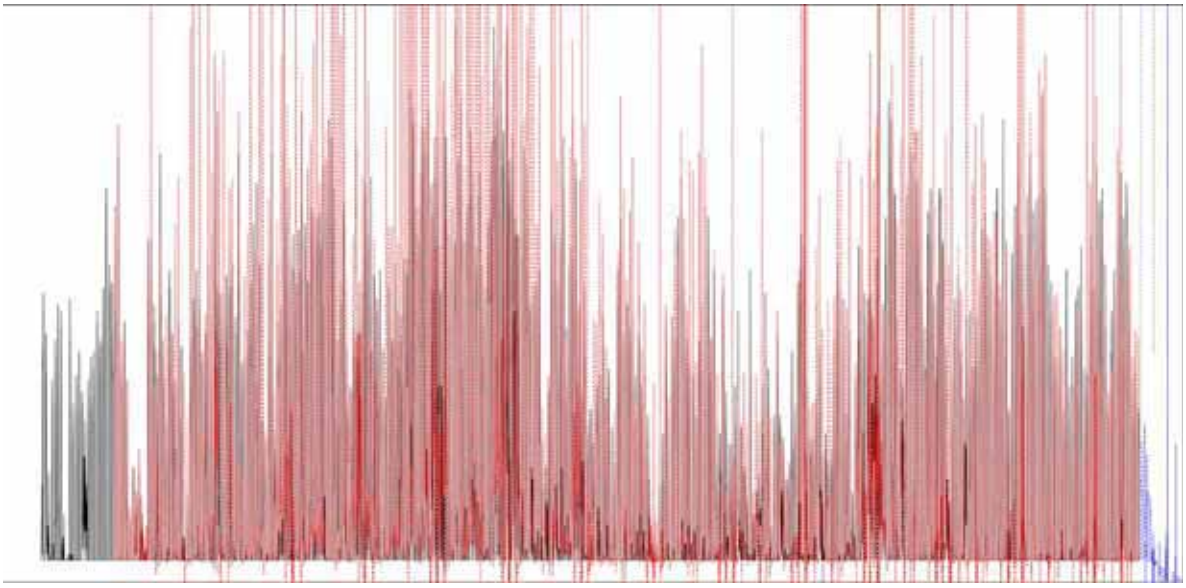


Figura 8.3.12 Gráfica multiplicativa de O3

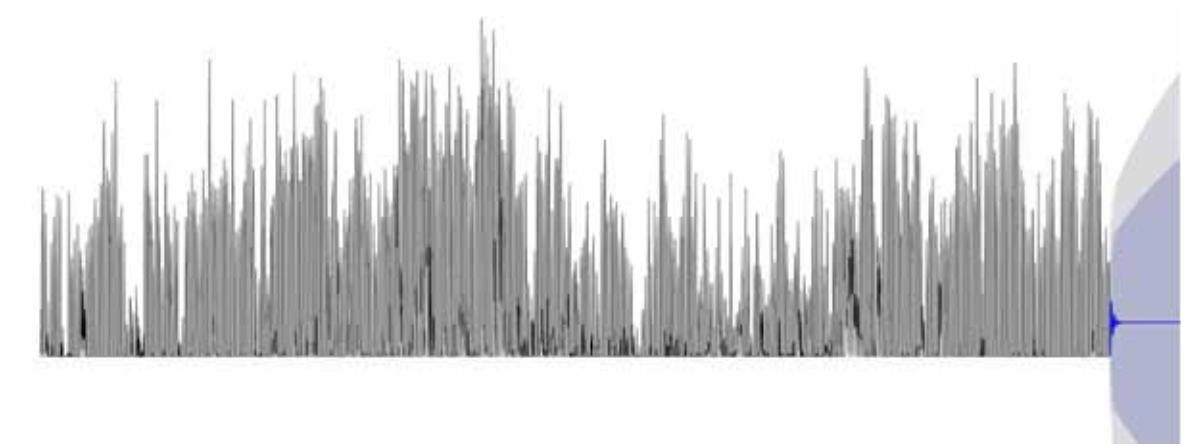


Figura 8.3.13 Gráfica sobre el modelo ARIMA del O3

1	matrix(c("	V1
2	Alpha	0.897959544991007
3	Betha	0
4	Gamma	1
5	Seasonal	additive
6	SEE	0.374525777466457

Figura 8.3.14 Tabla del modelo aditivo del O3

Como se muestra en la figura 8.3.11 el modelo aditivo predijo mejor el comportamiento del contaminante en comparación del método multiplicativo figura 8.3.12 y el modelo ARIMA 8.3.13, ya que se logró adecuar de una mejor manera a la gráfica.

La tabla de la figura 8.3.14 referente a la figura 8.3.11 muestra que el valor de alfa (0.89) es relativamente alta, lo que indica que la estimación del nivel en el punto de tiempo actual se basa varias observaciones recientes. El valor de beta es 0.00, lo que indica que la estimación de la pendiente *b* de la componente de tendencia no se actualiza sobre la serie de tiempo, y en su lugar se establece igual a su valor inicial. Esto tiene sentido, ya que el nivel cambia bastante en la serie temporal, pero la pendiente *b* de la componente tendencia sigue siendo más o menos lo mismo, que sería 0 ya que la pendiente es casi totalmente horizontal. El valor de gamma (1) es alto, lo que indica que la estimación de la componente estacional en el punto de tiempo actual se basa sólo en observaciones recientes. Los contaminantes no se comportan de la misma manera, el contaminante con mayor presencia en la Ciudad de México es el CO es el Monóxido de Carbono, este como se muestra en las gráficas pasadas se encuentra siempre en una cantidad descomunal, seguido por las partículas suspendidas y el ozono. Observando las gráficas se aprecian los pronósticos que son similares, se debe tener especial atención en estos.

Al cierre de este capítulo se ha demostrado que la contaminación no muestra una tendencia clara, casi siempre es constante, todos los valores recolectados en las series de tiempo oscilan alrededor de la tendencia. En el pronóstico que se realizó con cualquiera de los análisis sigue este mismo patrón, por lo cual, se observa que en los próximos días no habrá cambios significativos, en dado caso que los hubiera, se deberán tomar las acciones necesarias para prevenirlas.

9. Conclusiones

Las situaciones presentadas en la vida real no siempre responden a los modelos matemáticos, este es un caso en el cual no sabremos si en realidad respondieron como se esperaba, ya que en el Distrito Federal, no se lleva un orden adecuado a los registros de las estaciones de medición además de no contar con los todos los valores reales de los datos que se analizaron y al estimarlos siempre hay oportunidad a que se presenten errores.

Si bien hacer análisis de series de tiempo no es imposible tampoco es sencillo, pero gracias a herramientas como el lenguaje de programación R, es posible realizarlos de una manera en la cual se tenga un grado de certeza mayor en los resultados obtenidos.

Usando técnicas de datos faltantes lo más probable es que nos alejemos de la realidad al no saber con certeza cuanta fiabilidad tienen al estimar valores. Para evitar este dilema, a pesar que se usó una técnica de datos faltantes, se prefirió tomar el valor máximo, el promedio y el mínimo por fila para su análisis, y los resultados fueron interesantes.

El tiempo crece de manera exponencial al hacer esta técnica de datos faltantes cuando se tienen demasiados datos, por ejemplo: de 2 años para los 6 contaminantes, es capaz de llevarse hasta 1 día de procesamiento. Por lo cual, una recomendación es no utilizarlo para datos de más de 1 año. Aunque aseguramos su funcionamiento con un rango de hasta 6 años tomando en cuenta que su procesamiento será tardado.

Al utilizar el valor máximo la gráfica solo se desplazó hacia arriba sin alterar demasiado la predicción, la mayor parte del tiempo obteniendo valores positivos. Al utilizar el valor promedio la gráfica no tuvo ninguna alteración significante, sólo se desplazó un poco hacia el eje de las X. al tomar el mínimo, llego a dar valores negativos, esto no se puede permitir, porque la contaminación jamás será menor que cero.

Para un modelo estacionario con tendencia constante, el modelo suavizado exponencial no es el mejor, aunque arrojó resultados buenos ya que siempre mantiene la tendencia ya que es relativamente 0, prediciendo en buena parte los cambios drásticos que se presentaban en las series de tiempo.

Los modelos ARIMA se adecuaron de una mejor manera ya que están diseñados para series de tiempo con tendencia y sin tendencia, siempre teniendo en cuenta un rango de probabilidad en donde la predicción podría caer. Estos rangos fueron del 80% y 95%. Gracias a esto, la certeza de los resultados de la predicción son muy estrictos, además, sí se observa después del análisis que hay una probabilidad de un aumento desmedido, tener la precaución de poner un plan en el cual se pueda prever dicha situación.

Si bien es sabido por la mayoría de las personas que la contaminación del aire no se acabará de la noche a la mañana, ni a mediano plazo, es de vital importancia empezar a hacer conciencia de los resultados que tienen la presencia de estos contaminantes en esta ciudad. Haciendo varias pruebas en diferentes lapsos, no se nota que cambie la tendencia, pero si lo va haciendo muy lentamente, por lo cual, es indispensable tomar conciencia ya que si no se hace se estarán

levantando programas para regular la contaminación como se hizo recientemente con el "Hoy no circula".

Para todo lo descrito en este documento fue de vital importancia utilizar una herramienta tan poderosa como lo es el lenguaje R, gracias a él, se obtienen resultados más precisos que en otros lenguajes y otras formas de hacerlo, ya que esta optimizado para análisis estadísticos de esta magnitud. El inconveniente más grande que tiene el lenguaje, es que la documentación a veces es muy borrosa, por lo que no se explica bien como hace cada paso.

Al realizar un proyecto así, en un lenguaje que es poco conocido entre programadores, hace pensar en la idea que no siempre lo más usado es lo mejor, ni lo menos usado es lo peor, a pesar de ser difícil tener que familiarizarse con otro lenguaje de programación, siempre es bueno tener un amplio criterio de qué lenguaje aportará más a tu proyecto, no solo dejarse llevar por el lenguaje que más se domina o se conoce.

10. Bibliografía

[1] A. D. J. Cristina, «Lenguaje de manipulación y mineria de datos,» Mexico, 2011.

[2] N. G. González, «Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información,» México D.F., 2011.

[3] B. J. S. Rojas, «Descubrimiento de patrones secuenciales utilizando razonamiento logico temporal,» Costa Rica, 2011.

[4] G. G. M. Castillo, «Desarrollo de un modelo basado en técnicas de Minería de Datos para clasificar zonas climatológicamente similares en el estado de Michoacán,» México, D.F., 2008.

[5] F. M. Álvarez, «Análisis de las series temporales de los precios del mercado eléctronico mediante técnicas de clustering,» España.

[6] A. E. Sahafizadeh Ebrahim, «Prediction of Air Pollution of Boushehr City Using Data Mining,» de *Second International Conference on Environmental and Computer Science*, 2009.

[7] L. F. H. H.-P. Zheng Yu, «U-Air: When Urban Air Quality Inference Meets Big Data,» Beijing China , 2003.

[8] J. Villavicencio, «Introducción a Series de Tiempo,» [En línea]. Available: http://www.estadisticas.gobierno.pr/iepr/LinkClick.aspx?fileticket=4_BxecUaZmg%3D&tabid=100. [Último acceso: 07 07 2014].

[9] P. G. Aitor, «Imputación basada en arboles de clasificación,» 06 2002. [En línea]. Available: http://www.eustat.es/documentos/datos/ct_04_c.pdf. [Último acceso: 02 02 2014].

[10] D. P. G. D. D. F. G. S. D. M. X. R. Á. Cadarso Suárez Carmen, «Técnicas estadísticas aplicadas al valor pronostico de un viomarcador en la supervivencia de pacientes sometidos a implantación de una válvula cardíaca,» Enero 2001. [En línea]. Available: http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_765.pdf. [Último acceso: 7 Julio 2014].

[11] «Exponential Smoothing Methods,» [En línea]. Available: http://personal.cb.cityu.edu.hk/msawan/teaching/ms6215/Exponential%20Smoothing%20Methods.ppt. [Último acceso: 7 Julio 2014].

[12] G. Leandro, «Principios de Econometría,» Costa Rica, 09 Junio 2008.

[13] UAM, «Curso de Predicción Económica y Empresarial,» [En línea]. Available: http://www.uam.es/docencia/predysim/prediccion_unidad3/3_2_ficha.htm.

[14] M. S. GmbH, «Package XLConnect,» [En línea]. Available: http://cran.r-project.org/web/packages/XLConnect/XLConnect.pdf. [Último acceso: 03 Enero 2014].

[15] Grothendieck, «sqldf,» [En línea]. Available: https://code.google.com/p/sqldf/. [Último acceso: 13 noviembre 2012].

[16] J. Verzani, gWidgets: a Toolkit-Independent API for Building GUIs in R, The College of Staten Island, 2007.

[17] G.-O. K. Van Buuren Stef, «Package 'mice',» 10 Junio 2014. [En línea]. Available: http://cran.r-project.org/web/packages/mice/mice.pdf. [Último acceso: 7 Julio 2014].

[18] R. J. Hyndman, «Package forecast,» [En línea]. Available: http://cran.r-project.org/web/packages/forecast/forecast.pdf. [Último acceso: 16 Mayo 2014].

[19] G. G. Norman, Notas de clase, Series de Tiempo con R, Universidad Nacional de Colombia.

[20] «exponential smoothing,» [En línea]. Available: https://www.otexts.org/fpp/7. [Último acceso: 17 Junio 2014].

[21] J. H. Rob, Time series and forecasting in R, Monash University, 2008.

[22] N. Zucchini Walter, Time Series Analysis with R Part I.

[23] «Arima models,» [En línea]. Available: https://www.otexts.org/fpp/8. [Último acceso: 16 Junio 2014].

[24] C. Avril, A Little Book of R for Time Series, 2014.

[25] K. Y. J. Hyndman Rob, «Automatic Time Series Forecasting: The forecast Package for R,» *Journal of Statistical Software,* vol. 27, nº 3, p. 22, Julio 2008.

[26] C. a. Yar, Holt-Winters forecasting: some practical issues, The Statistician, 1988.

[27] A. E. Sahafizadeh Ebrahim, «Prediction of Air Pollution of Boushehr City Using Data Mining,» *Second International Conference on Environmental and Computer Science,* 2009.

Anexos

A continuación adjuntamos el código fuente con el que se realizó este proyecto.

```
#####################################################################
#####################################################################
###################################
#####################################################################
#####################################################################
###################################
##
##
##                                        Librerías
##
##
##
#####################################################################
#####################################################################
###################################
#####################################################################
#####################################################################
###################################


#Setear en la carpeta donde se alojará este código
setwd("C:/PT_Final")

#Librerías

library(mice)
library(sqldf)
library(XLConnect)
library(gwidgets)
library(forecast)
library(tcltk2)

#####################################################################
#####################################################################
###################################
#####################################################################
#####################################################################
###################################

if(file.exists("busquedas.R")){
  source("busquedas.R")
}else{
```

```
  message("No existe el archivo busquedas.R")
  gmessage(paste("No existe el archivo busquedas.R, ¡No correr el sistema!"), title="Error")
}


if(file.exists("validaciones.R")){
  source("validaciones.R")
}else{
  gmessage(paste("No existe el archivo validaciones.R, ¡No correr el sistema!"), title="Error")
}

#Variables globales
lista_variables <- c("CO","NO2","NOX","O3","PM10","SO2")
anos <-list("")
#cuenta_rama=1;
#cuenta_arch =1;
#n <- 1
lista_CO <-list("CO")
lista_NO2 <- list("NO2")
lista_NOX <- list("NOX")
lista_O3 <- list("O3")
lista_PM10 <- list("PM10")
lista_SO2 <- list("SO2")
lista_resultados_busquedas<-list("resultados")

datos_contam<-NA
datos_mesin<-NA
datos_anoin<-NA
datos_diain<-NA
datos_mesfin<-NA
datos_anofin<-NA
datos_diafin<-NA
sem_dia<-NA
fecha_inicial<-NA
fecha_final <- NA
setwd("C:/RedAutomaticaMonitoreoAtmosferico")
ramas <- list.files()
anos <-c("")




#############################################################
###################################
#CONCATENA CADENAS

concat = function(v) {
  res = ""
```

```
  for (i in 1:length(v))
    res = paste(res,v[i],sep="")

  res
}

#################################################################
#################################
#agarrar los ultimos caracteres de una cadena
substrRight <- function(x, n){
  substr(x, nchar(x)-n+1, nchar(x))
}

#################################################################
####################################
anocambio<- function(ano){
  if(ano <= 99 && ano >=95){
    ano <- concat(c("19",ano))
  }
  else if(ano >= 0 && ano <= 9){
    ano <- concat(c("20",ano))
  }
  else{
    ano<- concat (c("20",ano))
  }
  ano
}


#################################################################
#################################################################
#################################
#################################################################
#################################################################
#################################
##
##
##                                  Submódulo de validación
##
##
##
#################################################################
#################################################################
#################################
#################################################################
#################################################################
#################################
```

```
#############################################################
################################
#CAMBIAMOS TODOS LOS VALORES -99.9 A 0 POR MIENTRAS EN LO QUE SE
HACE EL CAMBIO DE PPB A PPM


cambio_a_NA <- function(Tabla){
  require("sqldf")
  library("sqldf")
  Nombres_tabla<- names(Tabla)
  DF <- data.frame(Tabla)

  update_first <- "update DF set "
  where <- " = 0 where "
  is <-" <= 0"
  select <- "select * from DF"
  num_x <- 0
  x <- "NA."
  la_x <-"NA.."
  i <- 3
  #cambio_ppb_ppm(Tabla,"Tabla",1)
  #print(select_first)

  while(i <= length(Nombres_tabla)){
    print(concat(c(la_x,num_x)))
    print(concat(c(update_first, Nombres_tabla[i],where, Nombres_tabla[i],is)))
   if(grepl("\\<.\\>", Nombres_tabla[i] ) || grepl("\\<_\\>", Nombres_tabla[i] )||
(Nombres_tabla[i] =="X") || (Nombres_tabla[i] =="NA")  ){
      print("entro ")
      break
    }else{
      print("paso")
      String <- concat(c(update_first, Nombres_tabla[i],where, Nombres_tabla[i],is))
      print(String)
      DF <- sqldf(c(String, select))
    }
    i <- i+1
  }
  Tabla <- assign("Tabla",DF,envir = .GlobalEnv)
}


#############################################################
#############################################################
################################
```

```
#####################################################
#####################################################
##################################
##
##
##                                  Cambio de ppb a ppm
##
##
##
#####################################################
#####################################################
##################################
#####################################################
#####################################################
##################################

#####################################################
#############################
#Cambio de ppb a ppm
#funciona para sacar una sentencia SELECT
#band = 1 solo funciona para sacar la sentencia SELECT
#band = 0 es para hacer el Cambio de ppb a ppm

cambio_ppb_ppm <- function(Tabla,nombre_t){
  require("sqldf")
  #Strings
  select_first <- "SELECT "
  select <- "select * from main.DF"
  #Enteros
  num_x <- 0
  num_na <- 1
  na <- "NA_"
  la_na <-"NA__"
  x <- "X"
  la_x <-"X."
  i <- 3
  cont_concat <- 1
  Nombres_tabla<- names(Tabla)
  while(cont_concat <= length(Nombres_tabla)){
    if(cont_concat >=4 && (Nombres_tabla[cont_concat]!="NA_")){
      select_first <-concat(c(select_first,","))
    }
    #print(concat(c(la_x,num_x)))
    if(Nombres_tabla[cont_concat]=="X" || Nombres_tabla[cont_concat]==
concat(c(la_x,num_x))){
      num_x <- num_x + 1
    }
```

```
   else if(Nombres_tabla[cont_concat]=="NA_" || Nombres_tabla[cont_concat]==
concat(c(la_na,num_na))){
     num_na <- num_na+1
     break
   }
   else{
     if(cont_concat <=2){
       select_first <-concat(c(select_first,Nombres_tabla[cont_concat],","))

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++")

     }
     if(cont_concat >= 3){
       #select_first <- concat(c(select_first, Nombres_tabla[cont_concat],"/1000"))
       select_first <-paste(select_first, Nombres_tabla[cont_concat],"/1000 as
'",Nombres_tabla[cont_concat],"'",sep="")
       #select_first <-concat(c(select_first,"as "))

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++")

       #DF <- sqldf(c(String, select))
     }

   }
   cont_concat <- cont_concat+1
 }
 select_first <- concat(c(select_first," FROM ",nombre_t))
 print(select_first)
 Tabla <- sqldf(select_first)
 print("pasamos df")
 Tabla <- assign("Tabla",Tabla, envir = .GlobalEnv)
 assign("select_first",select_first, envir = .GlobalEnv)
 print(paste("esto es select first --->",select_first))

 Tabla

}


##############################################################
##############################################################
#################################
##############################################################
##############################################################
#################################
```

```
##
##
##                                  Lectura de archivos
##
##
##
#############################################################
#############################################################
#################################
#############################################################
#############################################################
#################################


#############################################################
#######
#Validacion de archivos que coincidan con la carpeta que les toca



validacion_arch <- function(Dir_total){
  out <- tryCatch(
{
  message("Estoy leyendo")
  library(XLConnect)              # load XLConnect package
  wk = loadWorkbook(Dir_total)
  tabla_apoyo = readWorksheet(wk, sheet=1)
  #tabla_apoyo <- read.xlsx(Dir_total, 1 , stringsAsFactors=F)
  print("acabo de leer")
  tabla_apoyo <- assign("tabla_apoyo",tabla_apoyo,envir = .GlobalEnv)
  print("acabo de asignar")
  write.csv(tabla_apoyo, file
="C:/RedAutomaticaMonitoreoAtmosferico/NO_BORRAR/no_tocar.csv", na="")
  tabla_apoyo <-
read.csv("C:/RedAutomaticaMonitoreoAtmosferico/NO_BORRAR/no_tocar.csv",
header=T)
  tabla_apoyo <- assign("tabla_apoyo",cambio_a_NA(tabla_apoyo),envir = .GlobalEnv)
  out <- tabla_apoyo
},
error=function(cond) {
  message("error")
  gmessage(paste("No existe el sig archivo: ",Dir_total), title="Hubo un problema")
  tex <- gconfirm("?Volvemos a intentar?", title="Intentar")
  if(tex == TRUE){
    validacion_arch(Dir_total)
  }else{
    gmessage(paste("Se ha formateado la lectura de datos. Favor de cargarlos otra vez."),
title="Lo sentimos")
```

```
    lista_CO <-list("")
    lista_NO2 <-list("")
    lista_NOX <-list("")
    lista_O3 <-list("")
    lista_PM10 <-list("")
    lista_SO2 <-list("")
    lista_resultados_busquedas<- list("")
    stop("dummy error", call. = FALSE)
    options(error = NULL)  # restore to default
    quito <- substr(Dir_total,38,43)
    print(paste("esto es quito ",quito))
    # ramas1 <- [! ramas1 %in% quito ]
  }
  message(cond)
  # Choose a return value in case of error
  return(NA)
},
warning=function(cond) {
  message("warning")
  message(cond)
  # Choose a return value in case of warning
  return(out)
},
finally={
}
 )
  return(out)
}




#############################################################
#######
#ayudo a integrar nombres de archivos

#bandera = 1 es distinto de 2011
#bandera = 0 es igual a 2011

ingresa_arch <- function(direccion,ano,bandera){
  lista_variables <- c("CO","NO2","NOX","O3","PM10","SO2")
  extension_xls <-".xls"
  extension_csv <- ".csv"
  la_n <- "n"
  cuenta <- 1
  if(length(list.files())!=0){
    if(bandera == 1){ # menor a 2011
      while(cuenta <= length(lista_variables)){
```

```
    dir <- concat(c(direccion,ano,lista_variables[cuenta],extension_xls))
    print(paste("direccion es ---->", dir))
    #out <- validacion_arch(dir)
    out <- assign("out",validacion_arch(dir),envir = .GlobalEnv)
     if(ano > 2011){
           tabla_apoyo_der <- out[c(-1,-2,-3)]
            tabla_apoyo_izq <- out[c(1,2,3)]
            tabla_apoyo_der <- tabla_apoyo_der /1000
            out <- cbind(tabla_apoyo_izq,tabla_apoyo_der)
       }
    if(lista_variables[cuenta]== "CO"){
      print("guardo")
      assign("lista_CO",c(lista_CO,list(ano,out)),envir = .GlobalEnv)
    }else if(lista_variables[cuenta]== "NO2"){
      assign("lista_NO2",c(lista_NO2,list(ano,out)),envir = .GlobalEnv)
    }else if(lista_variables[cuenta]== "NOX"){
      assign("lista_NOX",c(lista_NOX,list(ano,out)),envir = .GlobalEnv)
    }else if(lista_variables[cuenta]== "O3"){
      assign("lista_O3",c(lista_O3,list(ano,out)),envir = .GlobalEnv)
    }else if(lista_variables[cuenta]== "PM10"){
      assign("lista_PM10",c(lista_PM10,list(ano,out)),envir = .GlobalEnv)
    }else if(lista_variables[cuenta]== "SO2"){
      assign("lista_SO2",c(lista_SO2,list(ano,out)),envir = .GlobalEnv)
    }else{
      print("tenemos un problemon de aquellos")

    }
    print("guarde")
    cuenta <- cuenta +1
  }

}
else{
  print("llegue????????")
  setwd(direccion)
  cont <- 1
  while(cont <= length( list.files() ) ){
    dir <- concat(c(direccion,list.files()[cont]))
    print(paste("esto es dir ---->",dir))
    if(ano > 2011){
      print(direccion)
      setwd(direccion)
      }else{
        setwd(dir)
      }
    #setwd(dir)
    cuenta_var <- 1
    while(cuenta_var <= length( list.files() ) ){
```

```
        if(cont ==2){
          la_n = "n"
          ayudo <- concat(c(dir,"/",ano,lista_variables[cuenta_var],la_n,extension_xls))
          print(paste("direccion en 2011
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWW         es ----> ", ayudo))
          tabla_apoyo<- validacion_arch(ayudo)
          #Tabla<-cambio_ppb_ppm(tabla_apoyo,"tabla_apoyo")
          tabla_apoyo_der <- tabla_apoyo[c(-1,-2,-3)]
          tabla_apoyo_izq <- tabla_apoyo[c(1,2,3)]
          tabla_apoyo_der <- tabla_apoyo_der /1000
          tabla_apoyo <- cbind(tabla_apoyo_izq,tabla_apoyo_der)
          if(lista_variables[cuenta_var]== "CO"){
            #zz <- as.data.frame(lista_CO[length(lista_CO)])
            t1 <- merge(as.data.frame(lista_CO[length(lista_CO)]), tabla_apoyo, all = TRUE)
            t1 <- sqldf("select * from t1 order by FECHA")
            lista_CO[length(lista_CO)] <- NULL
            assign("lista_CO",c(lista_CO,list(t1)),envir = .GlobalEnv)
          }else if(lista_variables[cuenta_var]== "NO2"){
            t1 <- merge(as.data.frame(lista_NO2[length(lista_NO2)]), tabla_apoyo, all = TRUE)
            t1 <- sqldf("select * from t1 order by FECHA")
            lista_NO2[length(lista_NO2)] <- NULL
            assign("lista_NO2",c(lista_NO2,list(t1)),envir = .GlobalEnv)
          }else if(lista_variables[cuenta_var]== "NOX"){
            t1 <- merge(as.data.frame(lista_NOX[length(lista_NOX)]), tabla_apoyo, all = TRUE)
            t1 <- sqldf("select * from t1 order by FECHA")
            lista_NOX[length(lista_NOX)] <- NULL
            assign("lista_NOX",c(lista_NOX,list(t1)),envir = .GlobalEnv)
            #assign("lista_NOX",c(lista_NOX,list(concat(c(ano,"n")),Tabla)),envir =
.GlobalEnv)
          }else if(lista_variables[cuenta_var]== "O3"){
            t1 <- merge(as.data.frame(lista_O3[length(lista_O3)]), tabla_apoyo, all = TRUE)
            t1 <- sqldf("select * from t1 order by FECHA")
            lista_O3[length(lista_O3)] <- NULL
            assign("lista_O3",c(lista_O3,list(t1)),envir = .GlobalEnv)
            #assign("lista_O3",c(lista_O3,list(concat(c(ano,"n")),Tabla)),envir = .GlobalEnv)
          }else if(lista_variables[cuenta_var]== "PM10"){
            t1 <- merge(as.data.frame(lista_PM10[length(lista_PM10)]), tabla_apoyo, all =
TRUE)
            t1 <- sqldf("select * from t1 order by FECHA")
            lista_PM10[length(lista_PM10)] <- NULL
            assign("lista_PM10",c(lista_PM10,list(t1)),envir = .GlobalEnv)
            #assign("lista_PM10",c(lista_PM10,list(concat(c(ano,"n")),Tabla)),envir =
.GlobalEnv)
          }else if(lista_variables[cuenta_var]== "SO2"){
            t1 <- merge(as.data.frame(lista_SO2[length(lista_SO2)]), tabla_apoyo, all = TRUE)
            t1 <- sqldf("select * from t1 order by FECHA")
            lista_SO2[length(lista_SO2)] <- NULL
```

```
            assign("lista_SO2",c(lista_SO2,list(t1)),envir = .GlobalEnv)
            #assign("lista_SO2",c(lista_SO2,list(concat(c(ano,"n")),Tabla)),envir = .GlobalEnv)
          }else{
            print("")
          }
          cuenta_var <- cuenta_var + 1
        }else{
            ayudo <- concat(c(dir,"/",ano,lista_variables[cuenta_var],extension_xls))
            print(paste("direccion en 2011 es ----> ", ayudo))
            Tabla<- validacion_arch(ayudo)

            if(ano > 2011){
              Tabla<-cambio_ppb_ppm(Tabla,"Tabla")
            }
            if(lista_variables[cuenta_var]== "CO"){
              assign("lista_CO",c(lista_CO,list(ano,Tabla)),envir = .GlobalEnv)
            }else if(lista_variables[cuenta_var]== "NO2"){
              assign("lista_NO2",c(lista_NO2,list(ano,Tabla)),envir = .GlobalEnv)
            }else if(lista_variables[cuenta_var]== "NOX"){
              assign("lista_NOX",c(lista_NOX,list(ano,Tabla)),envir = .GlobalEnv)
            }else if(lista_variables[cuenta_var]== "O3"){
              print("3")
              assign("lista_O3",c(lista_O3,list(ano,Tabla)),envir = .GlobalEnv)
            }else if(lista_variables[cuenta_var]== "PM10"){
              assign("lista_PM10",c(lista_PM10,list(ano,Tabla)),envir = .GlobalEnv)
            }else if(lista_variables[cuenta_var]== "SO2"){
              assign("lista_SO2",c(lista_SO2,list(ano,Tabla)),envir = .GlobalEnv)
            }else{
              print("")
            }
            cuenta_var <- cuenta_var + 1
        }

        #cuenta_var <- cuenta_var + 1
      }
      cont <- cont +1
      setwd(direccion)
    }
  }
 }else{
  gmessage("¡Uups!, No existen archivos ", title="Hubo un problema")
 }
}
```

```
#########################################################
######
#Selección

seleccion<-function(){

  setwd("C:/RedAutomaticaMonitoreoAtmosferico")
  string <- getwd()
  cuenta_carpetas <- 1
  cuenta_archivos <- 1
  ano <- NA

  #verifico(ramaºs1)

  if(length(list.files()) > 0){

    while(cuenta_carpetas <= length(list.files())){

      if(list.files()[cuenta_carpetas]!= "NO_BORRAR"){

        ano <- substrRight(list.files()[cuenta_carpetas],2)
        print(paste("año es ", list.files()[cuenta_carpetas]))

        dir1 <- concat(c(string,"/",list.files()[cuenta_carpetas],"/"))
        print(dir1)

        if(ano != 11){
          setwd(dir1)
        }

        ano <- anocambio(ano)

        print(paste("a?o es ------> ",ano))

        #mandamos el a?no
        if(ano == 2011){

          ingresa_arch(dir1,ano,0)
        }else{

          ingresa_arch(dir1,ano,1)
        }
        anos <- append(anos, ano)
        assign("anos",anos,envir=.GlobalEnv)
        vAnoI[]<- anos
        vSano[]<-anos
      }
```

```
    cuenta_carpetas <- cuenta_carpetas+1
    setwd(string)


   }

 }else{
  gmessage("¡Uups!, No existen carpetas ", title="Hubo un problema")
 }

 ramas <- list.files()

}


##########################################################
##########################################################
###################################
##########################################################
##########################################################
###################################
##
##
##                                  Modelado de sentencias
##
##
##
##########################################################
##########################################################
###################################
##########################################################
##########################################################
###################################

ejecucion_busqueda<-function(lista,vSano,select){
 i <- 2
 library("sqldf")

 print("inicie ejecucion_busqueda")
 print(paste("esto es i = ", i," esto es lista[i]"))

 vSano <- as.character(vSano)
 while(i <= length(lista)){

  print(paste("esto es i = ", i,"esto es vSano = ",vSano," esto es lista[i] ",lista[i]))
  if(vSano == lista[i]){
   print("si entre a asignar resultado")
   print(paste('esto es vSano = ',lista[i]))
```

```
    i<-i+1
    resultado <- as.data.frame(lista[i])
    resultado <- assign("resultado",resultado,envir = .GlobalEnv)
    i<-i-1
    print(paste("esta es la i despues de resultado ----------> ",i))
    print(select)
    resultado <-sqldf(select)
    resultado <- assign("resultado",resultado,envir = .GlobalEnv)


    print("pase")
   }
   i<- i+2
  }
  print("acabe ejecucion_busqueda")
  resultado
}




busca_lista<-function(vScontaminante,vSano,select){
  print(paste("++++++++++++++++++++++++++",
vScontaminante,"++++++++++++++++++++++++++++", sep=""))
  if(vScontaminante == "CO"){
    print("se escogio CO")
    #Tabla1 <- ejecucion_busqueda(lista_CO,vSano,select)
    assign("Tabla1",ejecucion_busqueda(lista_CO,vSano,select),envir = .GlobalEnv)
  }else if(vScontaminante == "NO2"){
    #Tabla1 <- ejecucion_busqueda(lista_NO2,vSano,select)
    assign("Tabla1",ejecucion_busqueda(lista_NO2,vSano,select),envir = .GlobalEnv)
  }else if(vScontaminante == "NOX"){
    #Tabla1 <- ejecucion_busqueda(lista_NOX,vSano, select)
    assign("Tabla1",ejecucion_busqueda(lista_NOX,vSano,select),envir = .GlobalEnv)
  }else if(vScontaminante == "O3"){
    #Tabla1 <- ejecucion_busqueda(lista_O3,vSano, select)
    assign("Tabla1",ejecucion_busqueda(lista_O3,vSano,select),envir = .GlobalEnv)
  }else if(vScontaminante == "PM10"){
    #Tabla1 <-  ejecucion_busqueda(lista_PM10,vSano, select)
    assign("Tabla1",ejecucion_busqueda(lista_PM10,vSano,select),envir = .GlobalEnv)
  }else if(vScontaminante == "SO2"){
    #Tabla1 <- ejecucion_busqueda(lista_SO2,vSano,select)
    assign("Tabla1",ejecucion_busqueda(lista_SO2,vSano,select),envir = .GlobalEnv)
  }
  Tabla1

}
```

```
crear_sentencia <-
function(vShora,vSdianombre,vSsemanas,vSano,vSanof,vScontaminante,buscar){

  #select <- paste('SELECT * FROM ', vScontaminante, 'WHERE ')
  vSano <- substrRight(vSano,4)
  vSanof <- substrRight(vSanof,4)
  vSsemanas<- substrRight(vSsemanas,1)
  vShora <- as.character(vShora)
  vShora <- sub(".*X", "", vShora)
  vSano<- as.numeric(vSano)
  vSanof <- as.numeric(vSanof)

  print(vSdianombre)
  while(vSano <= vSanof){
    print("entre una
##########################################################
#####################################")
    select <- paste('SELECT * FROM ', 'resultado ', 'WHERE ')
    if(vSsemanas != "X"){
      if(vSdianombre != "X"){
        dia <- validar_dias(vSdianombre)
        fecha_exacta <- as.POSIXlt(paste(vSano," ",vSsemanas," ", dia), format = "%Y %U
%u")
        select <- paste(select," FECHA='",fecha_exacta,"'", sep="")
        #select <- concat(select," FECHA = '",fecha_exacta,"'")
        print(select)
      }else{
        rango_dom <-as.POSIXlt(paste(vSano," ",vSsemanas," 7"), format = "%Y %U %u")
        rango_sab <-as.POSIXlt(paste(vSano," ",vSsemanas," 6"), format = "%Y %U %u")
        select <- paste(select," FECHA >= '", rango_dom,"' and FECHA <= '",
rango_sab,"'",sep="")
        #select <- concat(select,"FECHA >= '", rango_dom,"' and FECHA <= '",
rango_sab,"'")

      }
    }
    print(paste(vShora, " esto es vShoras"))
    if(vShora != "X" && vShora != ""){
      select <- paste(select," and HORA=",vShora," ")
    }
    print(select)


    Tabla1<-busca_lista(vScontaminante,vSano,select)
    assign("lista_resultados_busquedas",c(lista_resultados_busquedas,list(vSano,Tabla1)),envir =
.GlobalEnv)
```

```
    if(vSano == "2011"){
      Tabla1<-busca_lista(vScontaminante,paste(vSano,"n",sep=""),select)
      assign("lista_resultados_busquedas",c(lista_resultados_busquedas,list(vSano,Tabla1)),envir
= .GlobalEnv)
    }


    vSano <- vSano+1
  }

  buscar

}


numerodedias <- function(date) {
  m <- format(date, format="%m")

  while (format(date, format="%m") == m) {
    date <- date + 1
  }

  return(as.integer(format(date - 1, format="%d")))
}


crea_fecha <- function(vAno,vMes,vDia, ini_fin,semana){

    string <- NA

    if(vAno != ""){
      #exito
      string <- paste(vAno, sep = "")
    }else{
      #error
      gmessage("error en el año")
    }

    if(semana == "no"){

      if(vMes != 0){
        string <- paste(string,"-",sprintf("%02d",vMes), sep = "")
      }else{
        if(ini_fin == 0){
          string <- paste(string,"-01", sep = "")
        }else{
          string <- paste(string,"-12", sep = "")
        }
```

```
    }
    if(vDia != ""){
      string <- paste(string,"-",sprintf("%02d",as.numeric(vDia)), sep="")
    }else{
      if(ini_fin == 0){
        string <- paste(string,"-01", sep = "")
      }else{
        print("----------------------")
        print(string)
        uno <- paste(string,"-01", sep="")

        string <- paste(string,"-",numerodedias(as.Date(paste(string,"-01", sep=""))), sep = "")
      }
    }

  }else{
    #print("aqui")
    if(vMes != ""){
      if(vDia != 0){
        string <- as.POSIXlt(paste(vAno," ",vMes," ", vDia), format = "%Y %U %u")
      }else{
        if(ini_fin == 0){
          print("aqui")
          string <- as.POSIXlt(paste(vAno," ",vMes," ", 07), format = "%Y %U %u")
        }
        else{
          print("y aca")
          string <- as.POSIXlt(paste(vAno," ",vMes," ", 06), format = "%Y %U %u")
        }
      }
    }else{
      if(vDia != 0){
        string <- as.POSIXlt(paste(vAno," ",01," ", vDia), format = "%Y %U %u")
      }else{
        if(ini_fin == 0){
          print("si llegue muajajaja")
          string <- as.POSIXlt(paste(vAno," ",01," ", 07), format = "%Y %U %u")
        }
        else{
          string <- as.POSIXlt(paste(vAno," ",ultima_semana(vAno)-1," ", 06), format = "%Y
%U %u")
        }
      }
    }


  }
   string
```

```
}

ultima_semana<- function(vAno){

     days <- seq(as.Date(paste(vAno,"/01/01", sep="")), as.Date(paste(vAno,"/12/31",
sep="")), "days")
     regreso <- range(format(days, "%W"))
     regreso <- todas_las_semanas(1,as.numeric(regreso[2])-1)
     regreso

}

todas_las_semanas<- function(i,regreso){
  sem<-""
  while(i<=regreso){
     sem <- append(sem,as.character(i))
     i<- i +1
  }
  print(sem)
  sem
}


diseno_sentencia <-
function(vHoraI,vDiaI,vDiaF,vMesI,vMesF,vAnoI,vAnoF,contaminantes, semana){

   iteracion <- 1
   string <- NA
   t1 <- NA
   if(semana == "no"){
        vAnoI <- as.numeric(vAnoI)
        vAnoF <- as.numeric(vAnoF)
        vMesI <- as.numeric(validar_mes(vMesI))
        vMesF <- as.numeric(validar_mes(vMesF))
   }else{
        vAnoI <- as.numeric(vAnoI)
        vAnoF <- as.numeric(vAnoF)
        vDiaI <- as.numeric(validar_dias(vDiaI))
        vDiaF <- as.numeric(validar_dias(vDiaF))
   }

   print(vAnoI)
   print(vAnoF)
   print(vDiaI)
   print(vDiaF)
   print(vMesI)
   print(vMesF)
   if(vAnoI < vAnoF){
```

```
        while(vAnoI <= vAnoF){
          print(paste("esto es iteracion -> ", iteracion))


          if(iteracion == 1){
             string <- paste("select * from resultado where FECHA >=
'",crea_fecha(vAnoI,vMesI,vDiaI,0, semana),"'",sep = "")
          }else{
            string <- "select * from resultado"
          }
           print(paste("esto es vAnoI -> ", vAnoI))
          if(vAnoI == vAnoF){
            string <- paste("select * from resultado where FECHA <=
'",crea_fecha(vAnoF,vMesF,vDiaF,1,semana),"'",sep = "")
          }

          if(vHoraI != ""){
            string <- paste(string,"and HORA = ",vHoraI)
          }
          print(string)
          Tabla1<-busca_lista(contaminantes,vAnoI,string)
          Tabla1 <- Tabla1[, !(colnames(Tabla1) %in% c("X"))]
          t1 <- merge(t1, Tabla1, all = TRUE, sort = TRUE)
          t1 <- assign("t1", t1,envir = .GlobalEnv)
          #le movi esto
          semana <- semana
          iteracion <- iteracion+1
          vAnoI <- vAnoI +1
        }
    }
    else{
       string <- paste("select * from resultado where FECHA >=
'",crea_fecha(vAnoI,vMesI,vDiaI,0, semana),"' and FECHA <=
'",crea_fecha(vAnoF,vMesF,vDiaF,1, semana),"'",sep = "")
       if(vHoraI != ""){
         string <- paste(string,"and HORA = ",vHoraI)
       }
       Tabla1<-busca_lista(contaminantes,vAnoI,string)
       #Tabla1 <- Tabla1[, !(colnames(Tabla1) %in% c("X"))]
       t1 <- merge(t1, Tabla1, all = TRUE, sort = TRUE)
       t1 <- assign("t1", t1,envir = .GlobalEnv)
       print(string)
    }
    t1
}
```

```
#####################################################
#####################################################
#################################
#####################################################
#####################################################
#################################
##
##
##                                  Tecnica de datos faltantes
##
##
##
#####################################################
#####################################################
#################################
#####################################################
#####################################################
#################################
llenado_res <- function(res_busqueda){
    res_busqueda[res_busqueda == 0] <- NA
    library(mice)
    res_busqueda <- mice(res_busqueda)
    res_busqueda <- complete(res_busqueda)
    res_busqueda_der <- res_busqueda[c(-1,-2,-3)]
    res_busqueda_izq <- res_busqueda[c(1,2,3)]
    ind <- which(is.na(res_busqueda_der), arr.ind=TRUE)
    res_busqueda_der[ind] <- rowMeans(res_busqueda_der,  na.rm = TRUE)[ind[,1]]
    res_busqueda <- cbind(res_busqueda_izq,res_busqueda_der)
    View(res_busqueda)
    res_busqueda
}

#####################################################
#####################################################
#################################
#####################################################
#####################################################
#################################
##
##
##                                  Tratamiento de datos
##
##
##
#####################################################
#####################################################
#################################
```

```
#########################################################
#########################################################
#################################

busca_folder<-function(){
  setwd("C:/RedAutomaticaMonitoreoAtmosferico/NO_BORRAR")
  folder<-dir()[file.info(dir())$isdir]
  if(length(folder)==1){
    url<-file.path(getwd(), "1")
    dir.create(url, showWarnings = FALSE)
  }else{
    url<-file.path(getwd(), as.character(length(folder)) )
    dir.create(url, showWarnings = FALSE)
  }
  setwd(url)
}


arima_model<-function(frecuencia){

  library(forecast)

  i<-2

  busca_folder()

  while(i < length(lista_resultados_busquedas)){

    #sacamos el minimo
      res <- as.data.frame(lista_resultados_busquedas[i+1])
      res_der <-res[c(-1,-2,-3,-4)]
      res_izq <- res[c(1,2,3,4)]
      minimo <-apply(res_der, 1, FUN = function(res_der) {min(res_der[res_der > 0])})

    #hacemos la serie de tiempo con una frecuencia
      serie_t <- ts(minimo, frequency= frecuencia)
      print("Estimando modelo ARIMA")
    #estimamos el mejor modelo arima
      au = auto.arima(serie_t,approximation=FALSE, ic=c("aic"),trace=FALSE)
#seasonal=FALSE,
      print("Prediciendo modelo ARIMA")
    #estimamos la predicción del modelo
      au.pred=forecast(au,frecuencia)

    #metemos valores significativos a una matriz
      data_frame<-as.data.frame(rbind(au$loglik,au$aic))
      data_frame<-as.data.frame(rbind(as.matrix(data_frame),au$aicc))
```

```
    v<-matrix(c("Log likelihood","AIC","AICc"), ncol=1)
    colnames(v)<-"variable"
    data_frame <- cbind(v,data_frame)

  #pegamos valores de la probabilidad del minimo,promedio y maximo valor.
    prob <-cbind(au.pred$lower,au.pred$mean)
    prob <-cbind(prob,au.pred$upper)
    colnames(prob)<- c("80% menor","95% menor","promedio","80% mayor","95% mayor")

  #mandamos tablas a archivo de excel
    print("Escribiendo modelo ARIMA")
      # valores log,aic,aicc
    writeWorksheetToFile(file
=paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data
=as.data.frame(data_frame) , sheet = "Valores")
      #au[8] = residuos
    #writeWorksheetToFile(file =
paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data = as.data.frame(au[8]),
sheet = "Residuos")
      #predicción al 80% - 95%
    writeWorksheetToFile(file
=paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data = prob, sheet =
"Predicción")
      #residuos predicción
    writeWorksheetToFile(file =
paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data =
as.data.frame(au.pred[10]), sheet = "Residuos predicción")


    print("Graficando modelo ARIMA")
  #mandamos graficar
    importar_arima(au.pred,lista_resultados_busquedas[i],frecuencia)
    print("acabe")


  i<-i+2
 }

 setwd("C:/RedAutomaticaMonitoreoAtmosferico/NO_BORRAR")
}



exponential_smoothing <- function(frecuencia){
 #numeros pares de I son los contaminantes
 #nones contenido de data.frame

 library(forecast)
```

```
library(XLConnect)

i<-2

busca_folder()
print("llegue 1")
while(i < length(lista_resultados_busquedas)){
   print("llegue 2")
   res <- as.data.frame(lista_resultados_busquedas[i+1])
   res_der <-res[c(-1,-2,-3,-4)]
   res_izq <- res[c(1,2,3,4)]
   minimo <-apply(res_der, 1, FUN = function(res_der) {min(res_der[res_der > 0])})
   #se crea la serie de tiempo con el minimo de cada fila
   serie_t <- ts(minimo, frequency= frecuencia)
   #se calcula automatico los mejores valores para ALPHA, GAMMA y BETA
   HW1 <- HoltWinters(serie_t, seasonal="additive")
   HW2 <- HoltWinters(serie_t, seasonal="multiplicative")
   #predecimos lo que pasará en los siguientes días
   HW1.pred <- predict(HW1, frecuencia*2, prediction.interval=TRUE)
   HW2.pred <- predict(HW2, frecuencia*2, prediction.interval=TRUE)

   #unimos resultados de la estimación

   data_frame<-as.data.frame(rbind(as.matrix(HW1[3]),as.matrix(HW1[4])))
   data_frame<-as.data.frame(rbind(as.matrix(data_frame),as.matrix(HW1[5])))
   data_frame<-as.data.frame(rbind(as.matrix(data_frame),as.matrix(HW1[7])))
   data_frame<-as.data.frame(rbind(data_frame,as.matrix(HW1[8])))
   data_frame <- cbind(matrix(c("Alpha","Betha","Gamma","Seasonal","SEE"),
ncol=1),data_frame)

   data_framem<-as.data.frame(rbind(as.matrix(HW2[3]),as.matrix(HW2[4])))
   data_framem<-as.data.frame(rbind(as.matrix(data_framem),as.matrix(HW2[5])))
   data_framem<-as.data.frame(rbind(as.matrix(data_framem),as.matrix(HW2[7])))
   data_framem<-as.data.frame(rbind(data_framem,as.matrix(HW2[8])))
   data_framem <- cbind(matrix(c("Alpha","Betha","Gamma","Seasonal","SEE"),
ncol=1),data_framem)

   #metemos datos en excel

   writeWorksheetToFile(file =
paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data =
as.data.frame(data_frame), sheet = "Variables aditivas",header = TRUE, rownames=TRUE)
   print("llegue 3")
   writeWorksheetToFile(file
=paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data = HW1$fitted, sheet =
"Análisis aditivo")
   print("llegue 4")
```

```
    writeWorksheetToFile(file
=paste(getwd(),"/",lista_resultados_busquedas[i],".xlsx",sep=""), data = HW1.pred, sheet =
"Predicción aditiva")
    print("llegue 5")
    writeWorksheetToFile(file =
paste(getwd(),"/",lista_resultados_busquedas[i],"m.xlsx",sep=""), data =
as.data.frame(data_framem), sheet = "Variables multiplicativas",header = TRUE,
rownames=TRUE)
    print("llegue 6")
    View(HW2$fitted)
    writeWorksheetToFile(file
=paste(getwd(),"/",lista_resultados_busquedas[i],"m.xlsx",sep=""), data = HW2$fitted, sheet
= "Análisis multiplicativo")
    print("llegue 7")
    writeWorksheetToFile(file
=paste(getwd(),"/",lista_resultados_busquedas[i],"m.xlsx",sep=""), data = HW2.pred, sheet =
"Predicción multiplicativa")
    print("llegue 8")


    #graficamos en un .PNG

    importar_exponential(HW1,HW1.pred,serie_t,paste(lista_resultados_busquedas[i],"a",
sep=""))
    importar_exponential(HW2,HW2.pred,serie_t,paste(lista_resultados_busquedas[i],"m",
sep=""))

    i<-i+2
  }
  setwd("C:/RedAutomaticaMonitoreoAtmosferico/NO_BORRAR")

}




##############################################################
##############################################################
##################################
##############################################################
##############################################################
##################################
##
##
##                                  Graficación de series de tiempo
##
```

```
##
##
#############################################################
#############################################################
#################################
#############################################################
#############################################################
#################################
importar_exponential<-function(HW1,HW1.pred,serie_t,nombre){

    png(
      filename=paste(getwd(),"/",nombre,".png",sep=""),
      width     = 20.25,
      height    = 10.25,
      units     = "in",
      res       = 1200,
      pointsize = 4
    )
    plot.ts(serie_t)
    lines(HW1$fitted[,1],lty=2,col="red")
    lines(HW1.pred[,1],lty=2,col="blue")
    lines(HW1.pred[,2],lty=2,col="seagreen")
    lines(HW1.pred[,3],lty=2,col="seagreen")
    dev.off()
}

importar_arima<-function(auto_arima, nombre, frecuencia){

    png(
      filename=paste(getwd(),"/",nombre,".png",sep=""),
      width     = 20.25,
      height    = 10.25,
      units     = "in",
      res       = 1200,
      pointsize = 4
    )
    plot(auto_arima)
    dev.off()
}



lista_variables <- c("CO","NO2","NOX","O3","PM10","SO2")
horas <-
c("","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20"
,"21","22","23","24")
```

```
dias <-
c("","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20"
,"21","22","23","24","25","26","27","28","29","30","31")
meses <-
c("","Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octu
bre","Noviembre","Diciembre")
#anos <-
c("","1996","1997","1998","1999","2000","2001","2002","2003","2004","2005","2006","2007",
"2008","2009","2010","2011","2012","2013")
ndia <- c("","Lunes", "Martes", "Miercoles","Jueves","Viernes","Sabado","Domingo")
semanas <-
c("","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20"
,"21","22","23","24","25","26","27","28","29","30","31","32","33","34","35","36","37","38","3
9","40","41","42","43","44","45","46","47","48","49","50","51","52","53")
regreso <-""
ndia1 <- ""
an <- c("")



##########################################################
##########################################################
###################################
##########################################################
##########################################################
###################################
##
##
##                                    Interfaz
##
##
##
##########################################################
##########################################################
###################################
##########################################################
##########################################################
###################################

library("gWidgets")
#Ventana padre
win = gwindow("Data Finder")
#ventana hijo

gp = ggroup(cont=win,horizontal=FALSE)

#pestañas superiores
```

```
## Not run:
 mbl <- list()
 mbl$Ayuda$Creador$handler = function(h,...) gmessage(paste("David Venegas Martínez"),
title="Autor")

mb <- gmenu(mbl, container=win)

#simbolos de la barra de arriba
#se pone el simbolo de salir
aTodos <-  gaction(label="Cargar todos",   icon="open",handler=function(h, ...){
  seleccion()
  assign("anos",sort(anos, decreasing = FALSE),envir = .GlobalEnv)
  vAnoI[]<- anos
  vSano[]<-anos
  } )
aCargar <- gaction(label="Cargar nuevos",  icon="copy", handler = function(h,...){
        dir<-tk_choose.dir(getwd(), "Seleccione alguna carpeta")
        if(dir != NA){
          anoh <- as.numeric(substrRight(dir,2))
          count_dir <- nchar(dir)
          rama <- substr(dir, start=1, stop=count_dir-2)
          true_o_false<-sapply(anoh, is.numeric)
          print("llegue")
          if(true_o_false == TRUE && rama
=="C:/RedAutomaticaMonitoreoAtmosferico/RAMA"){
              if(anoh == 2011){
                ingresa_arch(paste(dir,"/", sep = ""),2011,0)
              }else{
                ingresa_arch(paste(dir,"/", sep = ""),anocambio(anoh),1)
              }
          }else{
            gmessage(paste("No coincide la carpeta en nombre o lugar"), title="Hubo un
problema")
          }
        }

} )#, handler = verificar)

SE <-  gaction(label="Suaviado exponencial",   icon="connect",handler=function(h, ...){
    if(length(lista_resultados_busquedas) != 1){

          sum <- as.numeric(as.Date(fecha_final) - as.Date(fecha_inicial))
          frecuencia <- asign_frec(sum)
          exponential_smoothing(frecuencia)

    }else{
      gmessage(paste("No hiciste ninguna busqueda",""), title="Upps!")
    }
```

```
} )
MA <-  gaction(label="Modelo ARIMA",   icon="connect",handler=function(h, ...){
   if(length(lista_resultados_busquedas) != 1){

        sum <- as.numeric(as.Date(fecha_final) - as.Date(fecha_inicial))
        frecuencia <- asign_frec(sum)
        arima_model(frecuencia)

   }else{
     gmessage(paste("No hiciste ninguna busqueda",""), title="Upps!")
   }
} )

aCerrar <- gaction(label="Cerrar",  icon="quit", handler = function(h,...) dispose (win) )

tbl1 = list(
  Todos=aTodos,
  sep = list(separator = TRUE), # must be named component
  Cargar = aCargar,
  sep = list(separator = TRUE), # must be named component
  SuaExp = SE,
  sep = list(separator = TRUE), # must be named component
  ModAR=MA,
  sep = list(separator = TRUE), # must be named component
  Cerrar = aCerrar
)

tb= gtoolbar(tbl1, cont=gp)


#se muestra la barra horizontal en la parte de arriba de la pantalla
#tb = gtoolbar(tbl1, cont=gp)
#nb = gnotebook(cont=gp)#se muestra las pestañas de nuestra ventana
general = gframe("General", cont=gp,horizontal=TRUE)
#panel izquierrod de la pantalla
gw3 <- gframe("Por Fecha",cont = general, horizontal=TRUE)
#panel derecho de la pantalla
gw4 <- gframe("Por semana",cont = general, horizontal=TRUE)
tbl1 = glayout(cont=general)
tbl = glayout(cont=gw3)
puntos = glayout(cont=gw4)


tbl[2,1:2]<- glabel("Fecha Inicio: ")
tbl[3,1]<- glabel("Horas")
tbl[3,2]<- glabel("Dias")
tbl[3,3]<- glabel("Meses")
tbl[3,4]<- glabel("Años")
```

```
tbl[4,1]<- vHoraI <- gdroplist(horas,cont=tbl)
tbl[4,2]<- vDiaI <- gdroplist(list(""),cont=tbl,editable=TRUE,handler= function(h,...){
          if(svalue(vMesF)!="" && svalue(vAnoF)!=""){
            vDiaI1 <- numerodedias(as.Date(paste(as.numeric(svalue(vAnoF)),"-
",sprintf("%02d", validar_mes(svalue(vMesF)) ),"-",sprintf("%02d", 1 ), sep="")))


            if(svalue(vAnoF)==svalue(vAnoI) && svalue(vMesI)== svalue(vMesF)){
              vDiaF[] <- todas_las_semanas(as.numeric(svalue(vDiaI)),num_dias_inicial)
            }else{
              vDiaI1 <- numerodedias(as.Date(paste(as.numeric(svalue(vAnoF)),"-
",sprintf("%02d", validar_mes(svalue(vMesF)) ),"-",sprintf("%02d", 1 ), sep="")))
              vDiaF[] <- todas_las_semanas(1,vDiaI1)
            }
          }

  })
tbl[4,3]<- vMesI <- gdroplist(list(""),selected=0, cont=tbl,editable=TRUE,handler=
function(h,...){
       if(svalue(vAnoI) != ""){
          mes_num <- validar_mes(svalue(vMesI))
          messs <- ""
           if(mes_num < 12){
            while(mes_num <=12){
              messs<-append(messs,validar_mes_letra(mes_num))
              mes_num <- mes_num +1
            }
          }else if (mes_num =="") {
            mes_num <- ""
          }
          vMesF[] <- messs
          vDiaI1 <- numerodedias(as.Date(paste(as.numeric(svalue(vAnoI)),"-",sprintf("%02d",
validar_mes(svalue(vMesI)) ),"-",sprintf("%02d", 1 ), sep="")))
          assign("num_dias_inicial",vDiaI1,envir = .GlobalEnv)

          aux<- todas_las_semanas(1,as.numeric(vDiaI1))
          aux= aux[-which(aux=="")]
          vDiaI[]<- aux

       }else{
         messs <-""
         vMesF[] <- messs
       }

  })
tbl[4,4]<- vAnoI <- gdroplist(list(""), cont=tbl,editable=TRUE,handler= function(h,...){
  #svalue(vAnoF) <- svalue(vAnoI)
  print(svalue(vAnoI))
```

```
        if(svalue(vAnoI) ==""){
          print("entre")
            vAnoF[] <-svalue(vAnoI)
            svalue(vAnoF) <- svalue(vAnoI)
            svalue(vMesI) <- svalue(vAnoI)
            vMesI[] <-list("")
            svalue(vMesF) <- svalue(vAnoI)
            vMesF[] <-list("")
            svalue(vDiaF) <- svalue(vAnoI)
            vDiaF[] <-list("")
             svalue(vDiaI) <- svalue(vAnoI)
            vDiaI[] <- list("")
            svalue(vHoraI) <- svalue(vAnoI)
            vHoraI[]<- list("")
        }else{
            svalue(vAnoF) <- svalue(vAnoI)
            anos_finales <- todas_las_semanas(as.numeric(svalue(vAnoI)), anos[length(anos)])

            print(anos_finales)
            vAnoF[] <- anos_finales
            vMesI[] <- meses
            messs <-""
            vMesF[] <- messs
            vHoraI <-horas
        }
  })
#tbl[5,1:4]<- solo_fecha <- gradio(items, cont=tbl,horizontal=TRUE)
tbl[6,1:2]<- glabel("Fecha Final: ")
tbl[7,1]<- glabel("Dias")
tbl[7,2]<- glabel("Meses")
tbl[7,3]<- glabel("Años")

tbl[8,1]<- vDiaF <- gdroplist(list(""),cont=tbl,editable=TRUE)
tbl[8,2]<- vMesF <- gdroplist(list(""), cont=tbl,editable=TRUE)
tbl[8,3]<- vAnoF <- gdroplist(list(""), cont=tbl,editable=TRUE)

tbl[9,1:2] <- glabel("Contaminantes")
tbl[10,1:3] <- contaminantes <-gcheckboxgroup(lista_variables,horizontal=TRUE)
tbl[11,1:2] <- gbutton("Generar \n", cont=tbl, handler = function(h, ...){

      #lista_resultados_busquedas<-list("resultados")
      contaminantes <- make.names(svalue(contaminantes))
      #solo_fecha <- make.names(svalue(solo_fecha))
      vHoraI <- sub(".*X", "", make.names(svalue(vHoraI)))
      vDiaI <- sub(".*X", "", make.names(svalue(vDiaI)))
      vDiaF <- sub(".*X", "", make.names(svalue(vDiaF)))
      vMesI <- sub(".*X", "", make.names(svalue(vMesI)))
      vMesF <- sub(".*X", "", make.names(svalue(vMesF)))
```

```
      vAnoI <- sub(".*X", "", make.names(svalue(vAnoI)))
      vAnoF <- sub(".*X", "", make.names(svalue(vAnoF)))


      #validacion <-
validacion_busqueda(vHoraI,vDiaI,vDiaF,vMesI,vMesF,vAnoI,vAnoF,contaminantes,solo_fe
cha)
      assign("datos_contam",contaminantes,envir = .GlobalEnv)
      i<-1
      print(contaminantes)
      #if(validacion != 0){
        #paso
        if(vAnoI == "" || vAnoF==""){
           gmessage(paste("No se especificó el año inicial o final "), title="Hubo un problema")
         }else{
           while(i<= length(contaminantes)){


             print(contaminantes[i])
             res_busqueda <-
diseno_sentencia(vHoraI,vDiaI,vDiaF,vMesI,vMesF,vAnoI,vAnoF,contaminantes[i], "no")
             res_busqueda <- llenado_res(res_busqueda)

#assign("lista_resultados_busquedas",c(lista_resultados_busquedas,list(contaminantes[i],res_bu
squeda)), envir = .GlobalEnv)

assign("lista_resultados_busquedas",c(lista_resultados_busquedas,list(contaminantes[i],res_bus
queda)),envir = .GlobalEnv)
             print(length(lista_resultados_busquedas))
             print(i)
             i<- i+1
           }
           vAnoI <- as.numeric(vAnoI)
           vAnoF <- as.numeric(vAnoF)
           vMesI <- as.numeric(validar_mes(vMesI))
           vMesF <- as.numeric(validar_mes(vMesF))
           assign("fecha_inicial",crea_fecha(vAnoI,vMesI,vDiaI,0,"no"),envir = .GlobalEnv)
           assign("fecha_final",crea_fecha(vAnoF,vMesF,vDiaF,1,"no"),envir = .GlobalEnv)
         }

      })
tbl[12,1:4] <- gseparator(cont=tbl,expand = TRUE)
#puntos[1,1:2] <- glabel("Por semana: ")
puntos[2,1:2]<- glabel("Fecha Inicial: ")
puntos[3,1]<- glabel("Hora")
puntos[3,2]<- glabel("Día")
puntos[3,3]<- glabel("Semana")
puntos[3,4]<- glabel("Año")
```

```
puntos[4,1]<- vShora <- gdroplist(horas,cont=puntos)
puntos[4,2]<- vSdianombre <-
gdroplist(list(""),cont=puntos,editable=TRUE,handler=function(h, ...){
        dia_num <- validar_dias(svalue(vSdianombre))
        dia <- ""
        print(dia_num)
        if(dia_num < 7){
          while(dia_num <=7){
            dia<-append(dia,validar_dias_letra(dia_num))
            dia_num <- dia_num +1
          }
        }else if (dia_num =="") {
          dia <- ""
        }
        print(dia)
        vSdianombref[] <- dia
  })

puntos[4,3]<- vSsemanas <- gdroplist(list(""), cont=puntos,
editable=TRUE,handler=function(h, ...){
        regreso<-
todas_las_semanas(as.numeric(svalue(vSsemanas)),length(as.numeric(regreso)))
        assign("ndia",ndia,envir = .GlobalEnv)
        vSdianombre[]<- ndia
        vSsemanasf[] <- regreso[-length(regreso)]

  })
puntos[4,4]<- vSano <- gdroplist(list(""), cont=puntos,editable=TRUE,handler=function(h,
...){
        #vSanof [] <- vSano

        svalue(vSanof) <- svalue(vSano)
        assign("regreso",ultima_semana(as.numeric(svalue(vSano))),envir = .GlobalEnv)
        vSsemanas[] <- regreso
        #lapply(regreso, svalue)

  })
puntos[6,1:2]<- glabel("Fecha Final: ")
#puntos[7,1]<- glabel("Hora")
puntos[7,1]<- glabel("Día")
puntos[7,2]<- glabel("Semana")
puntos[7,3]<- glabel("Año")
#tbl[19,1]<- vShoraf <- gdroplist(horas,cont=tbl)
puntos[8,1]<- vSdianombref <- gdroplist(list(""),cont=puntos)
puntos[8,2]<- vSsemanasf <- gdroplist(list(""), cont=puntos)
puntos[8,3]<- vSanof <- gdroplist(list(""), cont=puntos, editable=TRUE)
puntos[9,1:2] <- glabel("Contaminantes")
puntos[10,1:3] <- vScontaminantes <-gcheckboxgroup(lista_variables, horizontal=TRUE)
```

```
puntos[11,1:2] <- gbutton("Generar \n", cont=puntos, handler = function(h, ...){

            lista_resultados_busquedas<-list("resultados")
            datos_contam<-vScontaminantes <- make.names(svalue(vScontaminantes))
            vShora <- sub(".*X", "", make.names(svalue(vShora)))
            datos_diafin<-vSdianombre <- sub(".*X", "", make.names(svalue(vSdianombre)))
            datos_diafin<-vSdianombref <- sub(".*X", "", make.names(svalue(vSdianombref)))
            datos_mesin<-vSsemanas <- sub(".*X", "", make.names(svalue(vSsemanas)))
            datos_mesfin<-vSsemanasf <- sub(".*X", "", make.names(svalue(vSsemanasf)))
            datos_anoin<-vSano <- sub(".*X", "", make.names(svalue(vSano)))
            datos_anofin<-vSanof <- sub(".*X", "", make.names(svalue(vSanof)))
            assign("sem_dia","sem",envir = .GlobalEnv)
            i<- 1
              while(i<= length(vScontaminantes)){

                print(vScontaminantes[i])
                res_busqueda <-
diseno_sentencia(vShora,vSdianombre,vSdianombref,vSsemanas,vSsemanasf,vSano,vSanof,vS
contaminantes[i], "si")
                res_busqueda<-llenado_res(res_busqueda)

assign("lista_resultados_busquedas",c(lista_resultados_busquedas,list(contaminantes[i],res_bus
queda)),envir = .GlobalEnv)
                View(res_busqueda)

                i <- i+1

              }
            vAnoI <- as.numeric(vSano)
            vAnoF <- as.numeric(vSanof)
            vMesI <- as.numeric(validar_mes(vSsemanas))
            vMesF <- as.numeric(validar_mes(vSsemanasf))

assign("fecha_inicial",crea_fecha(vSano,vSsemanas,validar_dias(vSdianombre),0,"si"),envir =
.GlobalEnv)

assign("fecha_final",crea_fecha(vSanof,vSsemanasf,validar_dias(vSdianombref),1,"si"),envir =
.GlobalEnv)

            #}
          })


visible(tbl) <- TRUE

lista_resultados_busquedas<-list("resultados")
```