

Manual Modulos AIRE

immediate

1. 192.168.1.9:/home/calidadaire/CalidadAire/AirNew.py

1.1. Lectura y procesamiento de datos con AirNew

Tabla 1: AirNew.Air()

Parametros	AirNew.Air(Variable=None,Estaciones=None,Fechai=None, Fechaf=None)
Entradas	<p>Fechai: AAA-MM-DD HH:MM Indicará la fecha inicia en la cual se desarrollarán los cálculos o procesos del modulo. Tiene por defecto el primer día del mes en el que sea corrido.</p> <p>Fechaf:AAA-MM-DD HH:MM Indicará la fecha inicia en la cual se desarrollarán los calculos o procesos del modulo. Tiene por defecto el último día del mes en el que sea corrido.</p>
salidas	<p>nombre_contaminantes: list Lista los contaminantes que son medidos por la Red de Calidad de Aire</p> <p>nombre_meteorologicas:list Lista las variables meteorológicas medida por la red meteorologica asociada a cada estación de la Red de Calidad de Aire</p> <p>codigo_meteorologicas:list Presenta el código de las estaciones que cuentan con sensores de medición meteorológicas instalados</p> <p>norma:diccionario La variable de llamada es la resolución horaria, y presenta las concentraciones máximas estipuladas por la norma (Resolución 2254 del 2017), en cada resolución</p> <p>limc:diccionario de diccionarios El primer llamado es para la resolución temporal, mientras el segundo es para la variable de interés; el objetivo del diccionario es presentar el límite inferior y superior de -</p> <p>Est: DataFrame Es el resultado de la lectura de Estaciones.txt ubicado en el file /Tools</p> <p>:</p> <p>:</p> <p>:</p>

Tabla 2: AirNew.Air.Aire_db()

Parametros	Air.Air_db(Variable=None,Estaciones=None,Flags=False,Crudos=False,Validado=False,ValidadoSiata=False,Meteorologicas=False,Manual=False,filtro=False,Vaisala=False,query=False)
Entradas	<p>ValidadoSiata: Boolean Instrucción asociada con los datos validados de la Red aire y de meteorología, para este último necesita estar acompañado de la sentencia Meteorologicas=True</p> <p>Flags: Boolean Tiene por objetivo obtener el Dataframe con los banderines asociados a cada datos, bien sea de la red de aire o de las estaciones meteorológicas</p> <p>:</p> <p>:</p>
salidas	<p>Data: DataFrame multiindex El contenido de este DataFrame dependera de la sentencia seleccionada ('=True'), tendrá por columnas las variables asociadas a la sentencia, y en ellas las estaciones de medición.</p> <p>Dataica: list Lista las variables meteorológicas medida por la red meteorologica asociada a cada estación de la Red de Calidad de Aire</p> <p>:</p> <p>:</p> <p>:</p>

1.1.1. Calidad_Thiess

Función para traducir los códigos de calificación de los datos meteorológicos usados en la red general del SIATA a la calidad de datos usada por la Red de Aire. Para adecuar la calidad con la resolución horaria de los datos de la Red de Aire, se selecciona la calidad que tenga más del 50 % de los datos y sólo se identificara como calidad buena al dato horario calculado con más del 50 % del registro minutal calificado como bueno, en otro caso sera malo o faltante según los flags asignados.

En la siguiente Tabla se presentan los calificadore básicos usados en la red de meteorologicas del SIATA, es importante tener en cuenta que la calificación completa de los datos se realiza pasado dos meses del registro; pero igualmente hay una serie de algoritmos que le dan una calificación preliminar a los datos. También se resalta la existencia de una única entrada en base de datos para la calificación de las tramas.

código	Interpretación
1	Dato confiable en tiempo real
2	Dato confiable vía importador
151	Trama de datos dudosa en tiempo real
153	Temperatura dudosa en tiempo real
154	Humedad dudosa en tiempo real
155	Presión Atmosférica dudosa en tiempo real
156	Magnitud del viento dudosa en tiempo real
1511	pulso de lluvia dudoso en tiempo real

Con estos códigos básicos se estructuran la totalidad de códigos de calificación; por ejemplo, para los códigos con

más de dos dígitos, se reemplazara el 1 inicial con un 2 para el caso de datos dudosos captados vía importar. Cada número que se anexa a los anteriores códigos de tres dígitos se asocia a más de una variable con dato dudoso, dependiendo del número se identifica la variable, la correspondencia se presenta en el tercer número de los códigos de la tabla anterior.

Tabla 3: AirNew.Air.Calidad_Thiess()

Parametros	AirNew.Air.Calidad_Thiess(thies=True,Variable=None)
Entradas	<p>thies: Boolean Opción para seleccionar el tipo de estación de origen de los datos, las cuales pueden ser thies o vaisala (thies=False da la opción de vaisalas automáticamente).</p> <p>Variable: string Las variables meteorológicas son nombradas tal y como se hace en la base de datos de la Red de Aire. Donde, P es presión atmosférica, PPT es precipitación, T es temperatura del Aire, RG es radiación solar global, HR representa humedad Relativa, VV es la velocidad del viento, y DV es la dirección del viento.</p>
salidas	<p>data: DataFrame DataFrame cuyas columnas representan cada una de las estaciones meteorológicas del tipo seleccionado, con índice de fecha. En este los datos son el Flag de cada dato con resolución horario.</p>

Tabla 4: AirNew.Air.Air_db_time()

Parametros	Air.Air_db_time(Variable=None,filtró=True)
Entradas	<p>Variable: 'PM2.5','PM2.5M','PM10','PM10M','SO2','CO','NO2' En esta opción se indica la variable de interés, es importante tener en cuenta que la manuales en el tiempo de lectura pueden no tener datos</p> <p>filtró: Boolean -</p>
salidas	<p>Data: DataFrame multiindex El contenido de este DataFrame dependera de la sentencia seleccionada ('=True'), tendrá por columnas las variables asociadas a la sentencia, y en ellas las estaciones de medición.</p> <p>Dataica: list Lista las variables meteorológicas medida por la red meteorologica asociada a cada estación de la Red de Calidad de Aire</p> <p>:</p> <p>:</p> <p>:</p>

Tabla 5: AirNew.Air.Plot_CD_HR_T_RG()

Parametros	Air.Plot_CD_HR_T_RG()
Entradas	<p>ValidadoSiata: Boolean Instrucción asociada con los datos validados de la Red aire y de meteorología, para este último necesita estar acompañado de la sentencia Meteorologicas=True</p>

Tabla 5: AirNew.Air.Plot_CD_HR_T_RG()

Parametros	Air.Plot_CD_HR_T_RG()
	<p>Flags: Boolean Tiene por objetivo obtener el Dataframe con los banderines asociados a cada datos, bien sea de la red de aire o de las estaciones meteorológicas</p> <p>:</p> <p>:</p>
salidas	<p>Data: DataFrame multiindex El contenido de este DataFrame dependera de la sentencia seleccionada ('=True'), tendrá por columnas las variables asociadas a la sentencia, y en ellas las estaciones de medición.</p> <p>Dataica: list Lista las variables meteorológicas medida por la red meteorologica asociada a cada estación de la Red de Calidad de Aire</p> <p>:</p> <p>:</p> <p>:</p>

1.2. Figuras con AirNew

A continuación se explica la forma en la que se crean las figuras tradicionales haciendo uso de las funciones en AirNew. Como soporte adicional se tienen un notebook en jupyter con ejemplos (https://siata.gov.co:19999/notebooks/mariapvg/Plots_AirNew.ipynb).

1.2.1. Plot_ICA

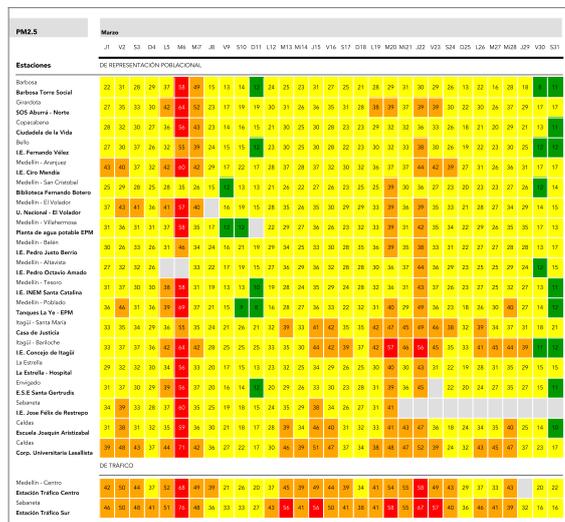


Figura 1: Ejemplo gráfica a partir de Plot_ICA

Esta función tiene pre-requisitos del estado de los datos. Primero hay que asegurarse de leer los datos de estaciones automáticas. Hay dos formas de leer los datos, una es la forma tradicional que depende de Estaciones.txt, y otra depende únicamente de base de datos y presenta todas las estaciones al público que tienen datos para las fechas entre las que se está corriendo el módulo, además esta última opción tiene la posibilidad de seleccionar la o las variables de interés. A continuación se presentan las formas de leer base de datos:

- Forma Tradicional
Air.Read_db(ValidadoSiata=True,filtro=True)
- Forma Tradicional
Air.Read_db(ValidadoSiata=True,Manual=True) - de manuales estaciones manuales únicamente
- De base de datos
Air.Read_db(Red=True)
- De base de datos
Air.Read_db(Red=True,Manual=True) - con estaciones manuales
- De base de datos
Air.Read_db(Red=True,Variables=['PM2.5'])

En segundo lugar, como la gráfica presenta los datos de del ICA, es necesario pasar el dato de concentración al índice de interés.

- Air.Calcula_ICA()

Finalmente se debe seleccionar una variable de la cual se pretende obtener el gráfico ('PM2.5','PM2.5M','PM10','PM10M','SO2','CO','NO2').

- Air.var = 'PM2.5','PM2.5M','PM10', 'PM10M','SO2','CO','NO2'

Tabla 6: Air.Plot_ICA()

Parametros	Air.Plot_ICA(Medellin=False, Valle=False,Red=False,Resolucion='D')
Entradas	<p>Medellin: Boolean Opción para seleccionar las estaciones ubicadas en la ciudad de Medellín.</p> <p>Valle: Boolean Opción para seleccionar las estaciones ubicadas en la ciudad del Valle diferentes a Medellín.</p> <p>Antiguas: Boolean Este atributo permite seleccionar las estaciones antigua, siempre y cuando tengan datos en el periodo seleccionadas.</p> <p>Manual: Boolean Si el objetivo es graficar datos desde mediciones con estaciones manuales, esta opción debe ser True.</p> <p>Red: Boolean Esta opción esta asociado de manera intrínseca con la forma de leer los datos, la cual preferiblemente debe ser Air.Read_db(Red=True), de acuerdo a los datos desarrollara la gráfica, si esta opción es False la gráfica se genera con unas estaciones predeterminadas en el .py.</p> <p>Resolución: 'D','8h','1h', Resolución temporal, se selecciona según la norma (horaria, diurna y octohoraria)</p>
salidas	<p>Plot: gráfico de cuadritos Se obtiene figura de cuadritos</p>

1.2.2. Plot_ICA_Resumen

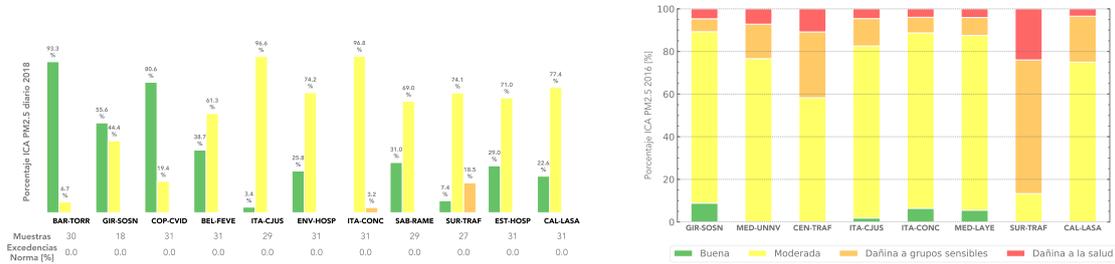


Figura 2: Ejemplo gráfica a partir de Plot_ICA_Resumen - derecha stacked=True-

La obtención de este gráfico presenta iguales pre-requisitos que las de la Figura 1 (Plot_ICA).

Tabla 7: Air.Plot_ICA_Resumen()

Parametros	Air.Plot_ICA_Resumen (resolucion='D', Medellin=True, filtro_Anual=False, fontsize=12, stacked=True, numeroD=False)
Entradas	<p>Medellin: Boolean Opción para 'PM2.5' de seleccionar las estaciones ubicadas en la ciudad de Medellín y fuera de ella.</p> <p>stacked: Boolean Opción para graficar las proporciones de cada indice de calidad en forma de barras, bien sea apilada(stacked=True), o por separado.</p> <p>resol: 'D','8h','1h', Resolución temporal, se selecciona segun la norma (horari, diurna y octohoraria)</p> <p>numeroD: Boolean , opción que especifica el número de días en cada nivel ICA.</p> <p>Antigua: Boolean , Con esta opción se seleccionan las estaciones antiguas, siempre y cuando tengan datos. Toma utilidad a partir del año 2017.</p>
salidas	<p>Plot: gráfico de barras Se guarda gráfica en la carpeta ModuloAire/Figuras y var/www/CalidadAire/Figuras como 'variable'/ICA_Resumen_'variable'_'resolucion'.pdf (stacked=True), o 'variable'/ICA_Histograma_'variable'_'resolucion'.pdf (stacked=False)</p>

1.2.3. Plot_Imshow

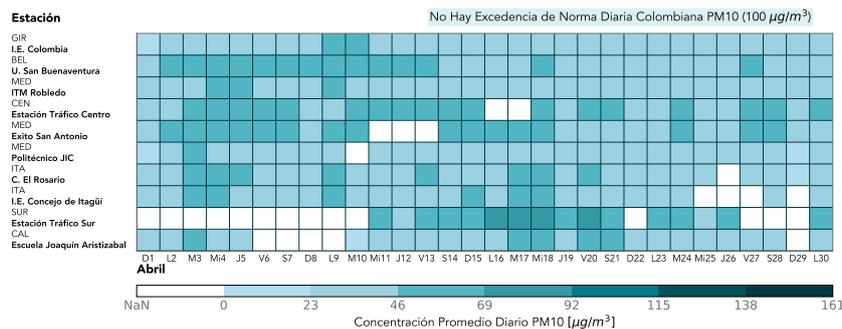


Figura 3: Ejemplo gráfica a partir de Plot_Imshow

Esta función tiene como pre-requisito la lectura de los datos, y solo está programada para estaciones automáticas. A continuación se presenta la forma de hacerlo:

- `Air.Read_db(ValidadoSiata=True,filtro=True)`
- `Air.Read_db(Red=True)`
- `Air.Read_db(Red=True,Variables=['PM2.5'])`

Tabla 8: `Air.Plot_Imshow()`

Parametros	<code>Air.Plot_Imshow (valores=False, resol='D', resolucion='Diaria', fontsize=12)</code>
Entradas	<p>valores: Boolean Esta opción permite visibilizar o no las concentraciones de cada cuadro</p> <p>resolucion: 'Diaria','Octohoraria','Horaria' Su selección no repercute en la graficación, pero sí en cómo se guarde la figura.</p> <p>resolución: 'D','8h','1h', Resolución temporal, se selecciona según la norma (horari, diaria y octohoraria)</p>
salidas	<p>Plot: gráfico de cuadritos Se guarda gráfica en la carpeta <code>ModuloAire/Figuras</code> y <code>var/www/CalidadAire/Figuras</code> como <code>'variable'/'Resolucion'_'variable'.pdf</code>.</p>

1.2.4. `Air.Plot_ICA_Anual()`

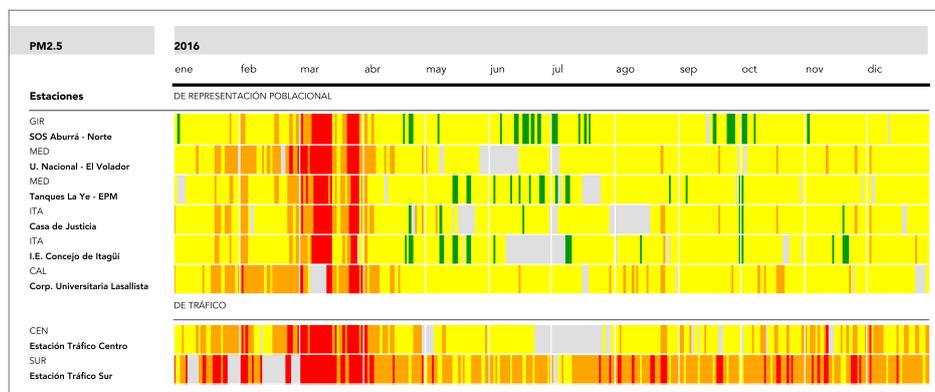


Figura 4: Ejemplo gráfica a partir de `Plot_ICA_Anual`

Esta función tiene pre-requisitos del estado de los datos. Primero hay que asegurarse de leer los datos de estaciones automáticas. Hay dos formas de leer los datos, una es la forma tradicional que depende de `Estaciones.txt`, y otra depende únicamente de base de datos y presenta todas las estaciones al público que tienen datos para las fechas entre las que se está corriendo el módulo.

- Forma Tradicional
`Air.Read_db(ValidadoSiata=True,filtro=True)`
- De base de datos
`Air.Read_db(Red=True)`

En segundo lugar, como la gráfica presenta los datos de del ICA, es necesario pasar el dato de concentración al índice de interés.

- `Air.Calcula_ICA()`

Finalmente se debe seleccionar una variable de la cual se pretende obtener el gráfico ('PM2.5', 'PM10', 'SO2', 'CO', 'NO2').

- Air.var = 'PM2.5', 'PM10','SO2','CO','NO2'

Tabla 9: Air.Plot_ICA_Anual()

Parametros	Air.Plot_ICA_Anual(var=None,Red=False ,resol='D',textsave="")
Entradas	<p>var: 'PM2.5', 'PM10','SO2','CO','NO2' La función tiene la opción para introducir la variable, pero de igual forma reconoce el var del objeto.</p> <p>Red: Boolean Esta opción esta asociado de manera intrínseca con la forma de leer los datos, la cual preferiblemente debe ser Air.Read_db(Red=True), de acuerdo a los datos desarrollara la gráfica, si esta opción es False la gráfica se genera con unas estaciones predeterminadas en el .py.</p> <p>Antiguas: Boolean Este atributo es útil únicamente si Red=True, con este se seleccionan las estaciones antigua, siempre y cuando tengan datos en el periodo seleccionadas.</p> <p>textsave: string La función de esta opción es enteramente asociada al nombre con que se guardará la imagen.</p> <p>resol: 'D','8h','1h', Resolución temporal, se selecciona segun la norma (horari, diurna y octohoraria)</p>
salidas	<p>Plot: gráfico de cuadritos Se obtiene figura de cuadritos. En la terminal aparece el nombre con el que se guarda y copia en las carpeta ModuloAire/Figuras y var/www/CalidadAire/Figuras, respectivamente.</p>

1.2.5. Plot_Flags

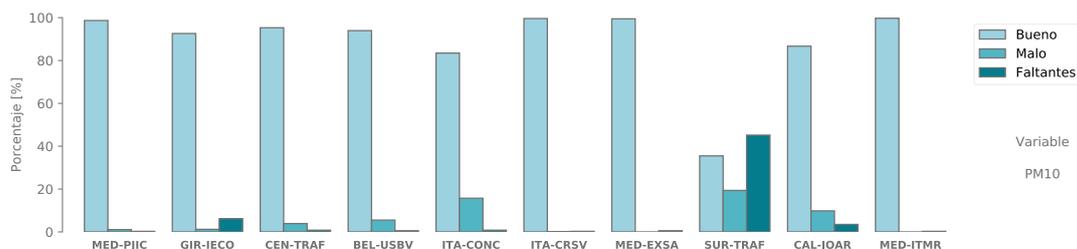


Figura 5: Ejemplo gráfica a partir de Plot_Flags

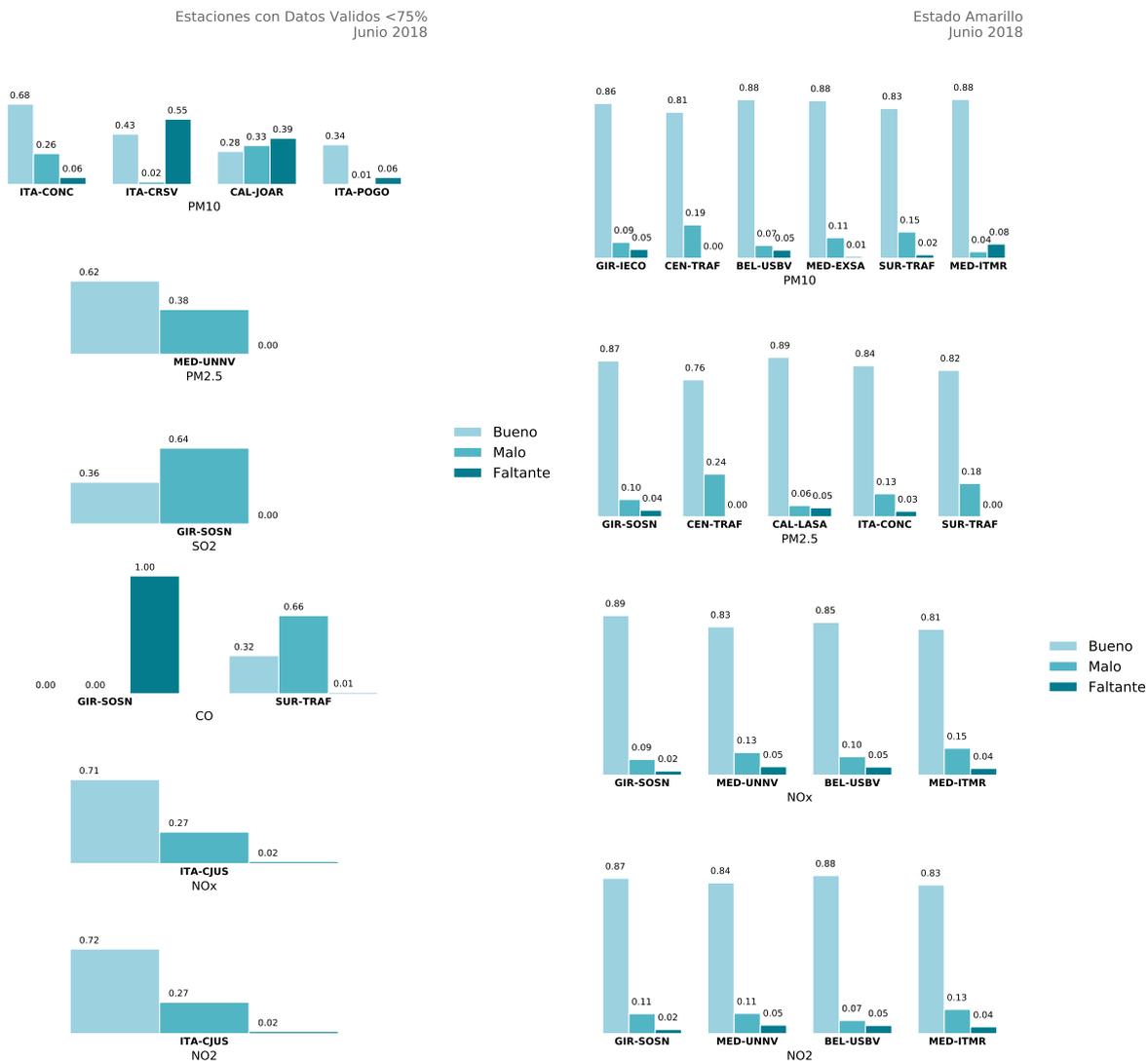


Figura 5: Ejemplo gráfica a partir de Plot_Flags - Continuación

Esta función puede usarse con y sin pre-requisito, eso dependerá de que gráfica se desea obtener. Si el objetivo es realizar las gráficas por conjuntos, no será necesario hacer lectura anticipada de los datos.

- `Air.Read_db(ValidadoSiata=True,filtro=True)`
- `Air.Read_db(Red=True)`
- `Air.Read_db(Red=True,Variables=['PM2.5'])`

Tabla 10: `Air.Plot_Flags()`

Parametros	<code>Air.Plot_Flags(df , var , One=True, Amarillo=True, Meteorologicas=True)</code>
Entradas	valores: Boolean Esta opción permite visibilizar o no las concentraciones de cada cuadro resolucion: 'Diaria','Octohoraria','Horaria' Su seleccion no repercute en la graficación, pero si en como se guarde la figura. resolución: 'D','8h','1h', Resolución temporal, se selecciona segun la norma (horari, diurna y octohoraria)
salidas	Plot: gráfico de cuadritos

Tabla 10: Air.Plot_Flags()

Parametros	Air.Plot_Flags(df , var , One=True, Amarillo=True, Meteorologicas=True)
	Se guarda gráfica en la carpeta ModuloAire/Figuras y var/www/CalidadAire/Figuras como 'variable'/'Resolucion'_'variable'.pdf.

1.2.6. Plot_Serie

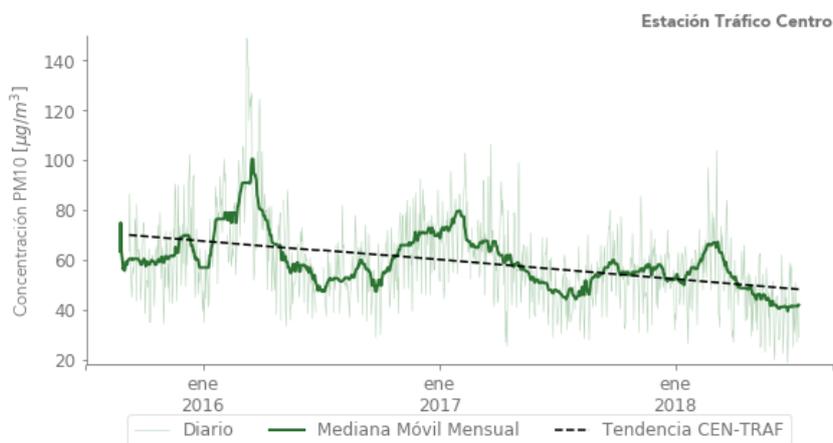


Figura 5: Ejemplo gráfica a partir de Plot_Serie

Tabla 11: Air.Plot_Serie()

Parametros	Air.Plot_Serie(df=Air.Contaminante[Air.var], textsave="", estaciones=None, legend=True,resolucion='D',Tendencia=False)
Entradas	<p>df: DataFrame Serie de datos que se desean plotear con un indice temporal asociado</p> <p>estaciones: nombre corto de las estaciones Dependerá de la serie que se desea graficar, la utilidad de especificar las estaciones de interés.</p> <p>resolucion: 'D','8h','1h', Resolución temporal, se selecciona según la norma (horari, diurna y octohoraria)</p> <p>Tendencia: Boolean atributo que permite graficar la línea de tendencia de la serie.</p>
salidas	<p>Plot: gráfico convencional</p> <p>Se guarda gráfica en la carpeta ModuloAire/Figuras y var/www/CalidadAire/Figuras como 'variable'/Serie_'variable'_'nombre corto'_'textsave'.pdf.</p>

1.2.7. Plot_Mapa3

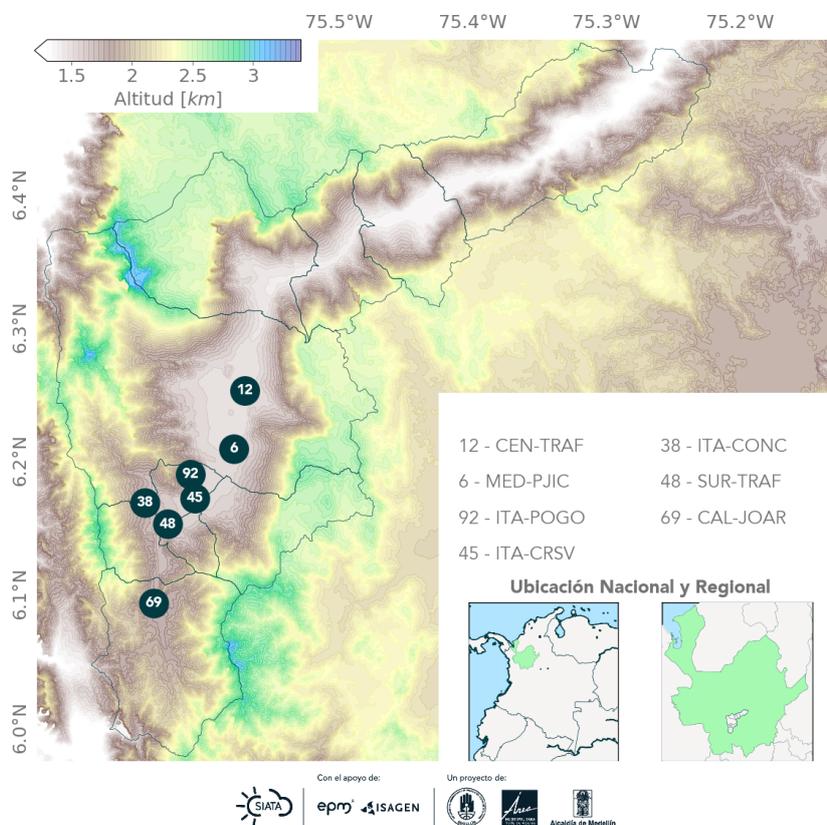


Figura 5: Ejemplo gráfica a partir de Plot_Mapa3

Si se desea crear el mapa de las estaciones de monitoreo de un contaminante en específico, basta con anteceder el comando 'Air.var=#Variable'.

Tabla 12: Air.Plot_Mapa3()

Parametros	Air.Plot_Mapa3(estaciones=None, textsave="" ,path='nroldan@siata.gov.co:/var/www/CalidadAire/Figuras/')
Entradas	<p>estaciones: list Si se quieren ubicar estaciones que no se encuentran agrupadas por variable, se puede incluir una lista con los nombres cortos.</p> <p>textsave:str Texto anexo que se incluire en el nombre con que la figura se guardara.</p> <p>path: str link con usuario y IP para hacer una copia de la figura.</p>
salidas	<p>Plot: gráfico convencional Se guarda gráfica en la carpeta ModuloAire/Figuras y var/www/CalidadAire/Figuras como 'Mapa_'textsave'.pdf.</p>

2. 192.168.1.9:/home/calidadaire/CalidadRuido/Ruido.py

Tabla 13: Ruido.Ruido.Aire_db()

Parametros	Ruido.Aire_db(Meteorologicas=False)
Entradas	Meteorologicas: Boolean Opción de lectura de las estaciones meteorológicas asociadas a la red (propias de la estación y/o thies).
salidas	Meteo: DataFrame-MultiIndex Obtenido solo si Meteorologicas=True. Meteo tiene como columnas las variables meteorológicas (velocidad_viento, acumulado_ppt), que se guardan en BD de las estaciones de ruido ambiental.

3. 192.168.1.9:/home/calidadaire/ModuloRuido/RuidoTabla.py

Para llamar la clase se importa el modulo 'import RuidoTabla' y se asigna la clase a una variable para trabajar más fácilmente con el 'self=RuidoTabla.Ruido(Fechai,Fechaf)'; Fechai y Fechaf son las fechas inicial y final del periodo de análisis, cómo default serán el primer día del año y el último día del mes anterior al tiempo de lectura.

Tabla 14: RuidoTabla: Tablas de resumen de Ruido Ambiental

Parametros	RuidoTabla.Ruido.Read_bd(Variable='LRAeqD')
Entradas	Fechai: str 'Y%-m %- %D H:MM:SS' La fecha de inicio tiene por default el primero de enero del año de corrida y debe ser asignafo a la clase directamente. Fechaf: str 'Y%-m %- %D HH:MM:SS' La fecha final tiene por default el mes anterior al que aquí se indique, también debe ser designado en el momento de llamar la clase. Variable: string Las Variables pueden ser 'LRAeqD':nivel de presión sonora continuo equivalente diurno , 'LRAeqN':nivel de presión sonora continuo equivalente nocturna y 'LRAeqDN:nivel de presión sonora continuo equivalente diurno nocturno'
salidas	Ruido.Data: DataFrame Lectura de los datos en forma de DataFrame, donde las columnas son las estaciones registradas para el periodo de tiempo seleccionado y el índice las fechas de los datos.

Tabla 15: RuidoTabla: Tablas de resumen de Ruido Ambiental

Parametros	RuidoTabla.Ruido.Tabla_Year(save=True)
Entradas	Fechai: str 'Y%-m %- %D H:MM:SS' La fecha de inicio tiene por default el primero de enero del año de corrida y debe ser asignafo a la clase directamente. Fechaf: str 'Y%-m %- %D HH:MM:SS' La fecha final tiene por default el mes anterior al que aquí se indique, también debe ser designado en el momento de llamar la clase. save: Boolean
salidas	Tabla_Ruido.pdf: documento pdf En este se anexa el cálculo e identificación del valor promedio, mínimo, máximo y las excedencias del ruido ambiental en niveles de equivalencia para el periodo de tiempo consultado. Tabla_Ruido.tex: Documento editable en latex

Tabla 15: RuidoTabla: Tablas de resumen de Ruido Ambiental

Parametros	RuidoTabla.Ruido.Tabla_Year(save=True)
	Ambos documentos se guardan en la carpeta de origen del .py

4. 192.168.1.9:/home/calidadaire/ModuloAire/Tools/TablaValidado.py

Para llamar la clase se importa el modulo 'import TablaValidado' y se asigna la clase a una variable para trabajar más fácilmente con él 'self=TablaValidado.Aire(Fechai='Y%-m%- %D HH:MM:SS',Fechaf='Y%-m%- %D HH:MM:SS')'; Fechai y Fechaf son las fechas inicial y final del periodo de análisis, cómo default serán el primer día y el último día del mes anterior al tiempo de lectura.

Tabla 16: TablaValidado-Tabla Datos Validos

Parametros	TablaValidado.Tabla(contaminantes=True,Meteorologicas=False,Historia=False)
Entradas	<p>Contaminantes: Bool ean Hace lectura de los Flags asociados a la base de datos de los contaminantes criterio, y de acuerdo a ello hace la tabla de datos validos.</p> <p>Meteorologicas: Boolean Hace lectura de los Flags pertenecientes a la red meteorológica asociada a las estaciones en medición de los contaminantes criterio, y de acuerdo a ello hace la tabla de datos validos.</p> <p>Historia: Boolean Tiene como Pre-requisito la lectura de los contaminantes (Contaminantes=True), y usa únicamente la Fecha Final como fecha de límite superior, el límite inferior lo darán los datos en cada estación; la fecha inicial será desde que haya registro valido.</p>
salidas	<p>Tabla_Proc.pdf: documento pdf Este es un documnto en donde se anexá el cálculo e identificacion del valor promedio, minimo, maximo y las excedencias de los contaminades seleccionados para los meses considerados</p> <p>Tabla_Proc.tex: Documento editable en latex</p>

5. 192.168.1.10:/home/calidadaire/Informe_Procuraduria/AnexPro.py

Tabla 17: AnexPro-Tablas de resumen para Procuraduría

Parametros	AnexPro.Tabla_Year(Fechai,Fechaf,contaminantes,save=False)
Entradas	<p>Fechai: str 'Y%-m%- %D H:MM:SS' La fecha de inicio tiene por default el primero de enero del año de corrida.</p> <p>Fechaf: str 'Y%-m%- %D HH:MM:SS' La fecha final tiene por default el mes de corrida; sin embargo todos las tablas se calculan para el mes anterior al que aquí se indique.</p> <p>contaminantes: list ['PM25','PM25M','PM10M','pm10','CO','Ozono','SO2','NO2'] Contiene todos los contaminantes asociados a la Red de Calidad de Aire; sin embargo pueden seleccionarse algunos en especifico.</p> <p>save: Boolean Esta opción permite guardar en http://siata.gov.co/CalidadAire/Informe_Procuraduria/ el pdf generado, bien sea como Anexo_ResumenAnualYYYY.pdf o como Anexo_Procuraduria_mes.pdf.</p>

Tabla 17: AnexPro-Tablas de resumen para Procuraduría

Parametros	AnexPro.Tabla_Year(Fechai,Fechaf,contaminantes,save=False)
salidas	<p>Tabla_Proc.pdf: documento pdf Este es un documnto en donde se anexá el cálculo e identificacion del valor promedio, minimo, maximo y las excedencias de los contaminades seleccionados para los meses considerados</p> <p>Tabla_Proc.txt:Documento editable en latex</p>