
AIRQUD Plugin para QGIS que permite realizar un modelo básico de la dispersión de contaminantes por fuentes fijas.

Sindy Dayany Quijano González
Lady Miyarlay Rodríguez Espinosa
UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
Junio 10 2016

This paper describes the creation of the tool to approach to models of dispersion of pollutants, framed under a methodology of software development, which identified the sequence to follow in the life cycle extension development, through an incremental model in which the stages of the project were identified. At each stage a series of activities that helped define inputs and outputs in each was made. According the above in the first stage the functional requirements defined and nonfunctional, then in stage two architecture and graphical interface, followed by the coding stage extension and finally the stage of performance testing and user, in order to improve or correct the functionality of the extension.

Key Words: Extensión; Qgis; GRASS; Interpolación; RST; IDW

Received...

1. Introducción

Asociado al crecimiento poblacional se encuentra la demanda de productos y servicios de acuerdo a sus necesidades, en consecuencia se genera el aumento de actividades industriales que en su mayoría emiten contaminantes a la atmósfera, generalmente por el consumo de combustibles fósiles. Estos contaminantes son nocivos para la salud, los ecosistemas y los recursos naturales. Por lo anterior se hace fundamental un control de emisiones, con el objetivo de que se genere una base legal y una mejor zonificación del suelo. Como parte del progreso de la normatividad se requiere de bases para establecer límites mínimos permisibles de emisiones, y una forma de obtener estos, es gracias a los modelos de dispersión de contaminantes que se ejecutan tanto para una como para varias fuentes puntuales de emisión, ya que estos modelos tienen como objetivo presentar una simulación de las concentraciones del contaminante a diferentes distancias de la fuente de emisión. Existen diversas herramientas para obtener estos modelos, los cuales en su mayoría requieren de un costo elevado ya sea monetario o de conocimiento experto sobre algún software en especial, es por esta razón que se propone una alternativa, que permite obtener un modelo de dispersión con los datos mínimos requeridos, como una herramienta que extiende la funcionalidad de un software libre SIG de escritorio. Los modelos de dispersión de contaminantes son de gran utilidad para establecer normas como para aquellas industrias las cuales requieren de la aprobación de nuevos procesos por parte de la autoridad ambiental competente o que desarrollan políticas de producción limpia con el objetivo de minimizar las emisiones y a su vez aportar a la sostenibilidad ambiental. La creación de la herramienta se enmarcó bajo una metodología de desarrollo de software, que permitió identificar la secuencia a seguir en el ciclo de vida del desarrollo de la extensión, por medio de un modelo incremental en el cual se identificaron las etapas del proyecto. En cada etapa se realizó una serie de actividades

que ayudaron a definir entradas y resultados en cada una de ellas. De acuerdo lo anterior en la primera etapa se definió los requerimientos funcionales y no funcionales, posteriormente en la etapa dos la arquitectura y la interfaz gráfica, seguido la etapa de codificación de la extensión y por último la etapa de pruebas de funcionamiento y de usuario, con la finalidad de mejorar o corregir la funcionalidad de la extensión. El desarrollo de la extensión *AirQud* permitió hacer una aproximación básica al comportamiento de dispersión de unas variables (contaminantes), mediante el uso de funciones de interpolación las cuales tienen como salida un archivo tipo raster (.tif), donde se observa el comportamiento de la variable en el espacio, sin embargo se deja para posteriores trabajos que se puedan incluir más variables que permitan una predicción más próxima a la dispersión respecto a condiciones meteorológicas. Este puede ser complementando la función RST con parámetros o con como ángulo de dispersión y tensión de las variables o implementado modelos gaussianos.

2. Justificación

El uso de modelos de dispersión atmosférica permite a las industrias delimitar y cuantificar las posibles zonas de impacto en torno a fuentes fijas de emisión, con beneficios como: ayudar a mejorar los procesos con la toma de decisiones para seleccionar la tecnología adecuada, diseñar programas para asegurar la calidad del aire, como ejercicio de control a largo plazo para mantener o mejorar la calidad del aire actual, investigaciones que impliquen diversas fuentes, donde se necesite identificar la causa más probable de emisiones, evaluación de riesgos y situaciones de catástrofe donde hay que evaluar minuciosamente episodios breves de contaminación elevada, entre otros aspectos importantes. Existen diversos modelos de dispersión que se pueden clasificar en: empíricos, semi-empíricos y numéricos, donde los empíricos se basan en análisis estadísticos de datos

obtenidos de calidad del aire, datos de fuentes de emisión y datos meteorológicos, los numéricos son obtenidos a partir de las características químicas y físicas, relacionadas con los fenómenos de transporte, difusión, transformación y remoción de contaminantes y, requieren de información científica experimental muy extensa, mientras que los modelos semi-empíricos, son una mezcla de los otros dos modelos y son los más utilizados actualmente, tanto por su facilidad de aplicación como por el tiempo requerido para su solución: estos modelos son del tipo gaussiano (Ruíz, 2002). La información que necesita la extensión para su funcionamiento es recolectada por el usuario y como mínimo debe tener los datos de la fuente de emisión y el modelo digital del terreno.

Con la implementación de una extensión que permita obtener un modelo de dispersión de contaminantes haciendo uso de un lenguaje de programación que no presente dificultades de interoperabilidad con diferentes clientes SIG como OSGeo, QGIS y GRASS GIS en trabajos futuros.

3. Objetivos

3.1. Objetivo General

Análisis, diseño e implementación de una extensión para un cliente SIG de escritorio que realizar un modelo básico de la dispersión de contaminantes por fuente fijas.

3.2. Objetivos Específicos

- (i) Identificar los requerimientos funcionales y no funcionales a considerar para la implementación de la extensión.
- (ii) Establecer la metodología de desarrollo a emplear durante la implementación de la extensión.
- (iii) Definir la arquitectura de la extensión que permitan cumplir con los requerimientos funcionales y no funcionales.
- (iv) Diseñar la interfaz gráfica para la extensión acorde a los requerimientos planteados.
- (v) Realizar pruebas de funcionalidad y de usuario con base en la norma de calidad de software ISO 9126-3

4. Marco Teórico

En este apartado se presenta una breve descripción de las entradas y salidas además del modelo matemático empleado en la elaboración de la extensión. En la Figura 1 se observan los datos mínimos necesarios para el cálculo del modelo de dispersión de contaminantes.

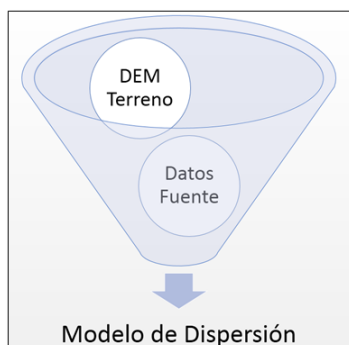


Figure 1. Datos mínimos requeridos para el modelo de dispersión. Fuente: Autoras.

Para la fuente los datos necesarios son: Tipo y concentración de contaminante (NO_x, SO₂ o Partículas). El cálculo se realizó a través de dos métodos de interpolación que se explican más adelante.

4.1. Interpolación

La interpolación de las variables del modelo de dispersión se realizó utilizando complementos de GRASS en QGIS, estas herramientas utilizadas se describen a continuación.

4.1.1. Método de interpolación por “splines con tensión”

Con el fin de identificar la dispersión de contaminantes se realiza una interpolación por método splines; con la función v.surf.rst del software GRASS; esta función genera una aproximación espacial y análisis topográfico desde un conjunto de datos puntuales en formato vectorial, cuyo resultado final es una capa en formato raster utilizando curvas splines regularizadas con tensión (Arce,2014). Las curvas spline son curvas diferenciables definidas en porciones mediante polinomios. En los problemas de interpolación, se utiliza a menudo la interpolación mediante splines porque da lugar a resultados similares requiriendo solamente el uso de polinomios de bajo grado, evitando así las oscilaciones, indeseables en la mayoría de las aplicaciones, encontradas al interpolar mediante polinomios de grado elevado.(GRASS,2016).

4.1.2. Método de interpolación IDW (distancia inversa ponderada)

Por medio de la función v.surf.idw igualmente de GRASS, se realiza la interpolación de superficies desde datos puntuales vectoriales mediante la técnica de distancia inversa ponderada. Este comando llena una matriz raster con valores interpolados generados desde un conjunto de datos puntuales mediante una técnica de aproximación numérica (Arce,2014). El valor interpolado de una celda es determinado por los valores de los datos puntuales cercanos y la distancia de la celda desde los puntos de entrada. En comparación con otros métodos, la aproximación numérica permite representaciones de superficies más complejas, restringe la influencia espacial de cualquier error, y genera la superficie interpolada desde datos puntuales (Arce,2014).

Funciones de GRASS	
IDW	v.surf.idw input=datos output=salida npoints=3 column=valor
RST	v.surf.rst input=datos elev=salida zcolumn=valor

Figure 2. Funciones de GRASS para interpolación. Fuente: Arce,A. (2014).Aprendiendo GRASS.(p. 6).

5. Metodología

La creación de la extensión se enmarcó bajo una metodología de desarrollo de software, la cual permitió identificar lo que se debía hacer a lo largo del ciclo de vida de la extensión, estableciendo así un modo sistemático de gestionar el desarrollo de la extensión. Esta sistematización dividió el proyecto en etapas, en las cuales se realizaron una serie de actividades que ayudaron a definir entradas y salidas en cada una de ellas.

5.1. Modelo incremental

Se escogió este modelo de desarrollo ya que permitió identificar el ciclo de vida de una forma incremental, donde estos incrementos están definidos por una serie de subproyectos que permitieron minimizar el riesgo. El desarrollo se descompone en pequeñas partes de trabajo, basados en riesgos, arquitectura y requerimientos. En el modelo de desarrollo incremental, los requerimientos son conocidos desde el comienzo del desarrollo.

Ofrece una serie de entregas o lanzamientos que proveen progresivamente más funcionalidad para el cliente a medida que cada incremento es entregado (TECHNOTRICE, 2013, p. 1).

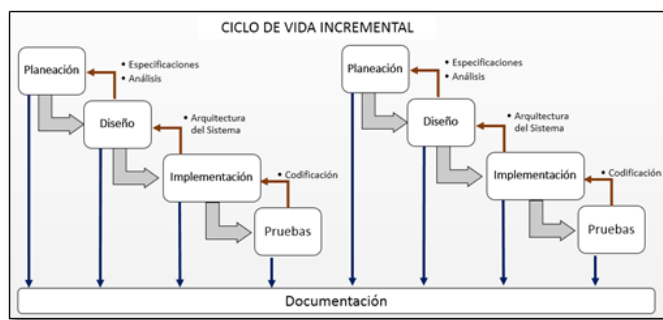


Figure 3. Ciclo de vida incremental. Fuente: Adaptado de “Desarrollo en Cascada Vs. Desarrollo Iterativo e Incremental”, 2013.

Teniendo en cuenta que este modelo se basa en secuencias lineales escalonadas, se describe a continuación las secuencias realizadas para el desarrollo del presente trabajo:

Planeación: en esta etapa se realizó la especificación de requerimientos funcionales y no funcionales. Inicialmente se realizó una revisión bibliográfica para identificar los antecedentes en cuanto a software existente que realizan modelos de dispersión, encontrando que en su mayoría son de carácter comercial. Posteriormente se realizaron encuestas a personal de compañías que manejan este tipo de software para conocer sus aplicaciones, tipos de usuarios y costos aproximados. Con esta información se optó por proponer una extensión para un cliente SIG libre de escritorio que cumpliera con los requerimientos mínimos para realizar modelos de dispersión de contaminantes. Se identificaron tanto requerimientos funcionales como no funcionales, descritos en la fase de planeación de la sección de resultados, a partir de los cuales se determinó el caso de uso, identificando los actores y sus roles. Para la generación de los casos de uso y la posterior fase de diseño se utilizaron diagramas UML; una breve descripción de esta metodología de diagramas se describe a continuación. UML (Lenguaje Unificado de Modelado) es una herramienta que permite a los desarrolladores de software generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. Este lenguaje está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, con la finalidad de presentar diversas perspectivas donde se describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema (Schmuller, 1997, p.24).

Diseño: Con el fin de identificar la organización lógica de la extensión a crear se diseñaron por medio de diagramas UML los modelados de dominio, interacción, estructural, así como también la definición de la arquitectura lógica del software usando de diagramas de clases y de paquetes y su estructura de hardware con diagrama de despliegue.

Implementación: Con el fin de realizar una programación más dinámica, fácil de escribir, entender, mantener y distribuir se utilizó lenguaje de programación Python, que a través de los bindings PyQt4, PyQGIS permiten desarrollar extensiones para el cliente SIG de escritorio denominado QGIS. Para el cálculo de interpolación se utilizaron las funciones `v.surf.rst` y `v.surf.idw` de GRASS para QGIS. Los datos base utilizados para la codificación de la extensión, corresponde a dos capas de puntos que representan fuentes fijas de emisión y un DEM (Modelo

digital de Elevación) de la zona donde están ubicadas, esto con el fin de definir un área de influencia y la interpolación de las variables del modelo de dispersión sobre esta.

Pruebas: Para evaluar la extensión generada se realizaron pruebas de funcionalidad, de usuario y de usabilidad, estas últimas se aplicaron entrevistando a diferentes personas que utilizaron la extensión y respondieron sobre la usabilidad de la misma, igualmente se tuvieron cuenta los criterios de la norma ISO/IEC TR 9126-3:2003, donde permite evaluar la usabilidad que según la norma se evalúa el entendimiento (capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación), aprendizaje (Es la capacidad del producto de software para permitir al usuario aprender su aplicación), operabilidad (capacidad del producto de software para permitir al usuario operarlo y controlarlo) y atractivo (capacidad del producto de software de ser atractivo al usuario. Esto se refiere a las cualidades del software para hacer el software más atractivo al usuario, tal como el uso de color y la naturaleza del diseño gráfico). Para las pruebas funcionales del plugin se utilizaron tres capas de datos, donde la primera solo tiene una fuente, otra tiene dos fuentes y una con más de dos fuentes de emisión que corresponde a estaciones de bombeo en zona de Caucasia con el respectivo DEM de la zona.

6. Resultados

6.1. Planeación

Durante esta fase se obtuvo como resultado una descripción de los servicios que ofrecería la extensión y las restricciones asociadas a su funcionamiento, con el fin de establecer una primera aproximación abstracta.

6.1.1. Requerimientos

A continuación se describen los requerimientos identificados: el usuario podrá seleccionar el contaminante para el cual quiere calcular el modelo de dispersión, los datos del punto de muestreo que deberán estar almacenados en una capa de información espacial que será previamente adicionada al entorno de trabajo de la herramienta SIG elegida, el DEM o modelo digital de elevación, seleccionar la columna que contiene los datos para el modelo y la función a usar para la interpolación, después permitirá al usuario iniciar el proceso del cálculo del modelo, durante el cálculo se facilitarán mensajes que permitan retroalimentar al usuario respecto al tiempo que tome la ejecución del cálculo, al finalizar el mismo, los resultados obtenidos se envían al área de visualización de datos espaciales de la herramienta SIG elegida, la cual debe usar como lenguaje programación Python, la interfaz debía permitir a los usuarios identificar qué se debe asignar en cada uno de los campos con mensajes que aparecen al pasar el cursor del mouse, permitiendo además ser muy intuitiva y de fácil comprensión, como se trata de una extensión de código abierto es necesaria una adecuada documentación dentro del código fuente para mantenerlo y posteriormente mejorar la extensión.

6.1.2. Casos de uso

A partir de los requerimientos funcionales se determinó la existencia un único caso de uso, como se observa en la Figura 4, que permite condensar los requerimientos funcionales y no funcionales planteados, con el escenario ideal de funcionamiento. Para esta extensión se identificó un único actor “usuario”, que hace referencia a cualquier persona que esté interesada en realizar

modelos de dispersión, quien podrá hacer uso de la herramienta construida.

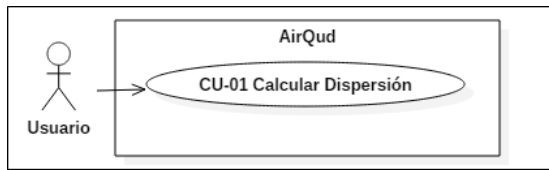


Figure 4. Diagrama de casos de uso. Fuente: Autoras.

La serie de eventos asociados al anterior caso de uso permite identificar los campos para obtener los datos necesarios para el cálculo, es decir se deben tener en cuenta por lo menos tres campos donde cada uno corresponderá a los datos: tipo de contaminante, meteorológicos y del punto de muestreo, además de las características que debe garantizar el usuario que deben tener cada uno de los datos para el adecuado funcionamiento de la extensión, finalmente después de cargar todos los datos se activará un botón que permita iniciar el modelo de dispersión, durante el proceso la extensión muestra mensajes que permitan conocer el tiempo restante para finalizar el proceso y cuando finaliza.

6.2. Diseño

6.2.1. Diagrama de secuencias

Este diagrama hace parte del modelo de iteración de la extensión y tiene como propósito presentar la comunicación entre los objetos, que se representan con rectángulos, mensajes representados por líneas continuas con punta de flecha con sus operaciones y el tiempo representado como una progresión vertical. En la Figura 5, se observan mensajes sincrónicos representados como una flecha con punta rellena donde se espera respuesta antes de continuar y asíncrónicos representados por las otras flechas donde se retorna aunque no acabe la operación que tenga en ejecución el objeto.

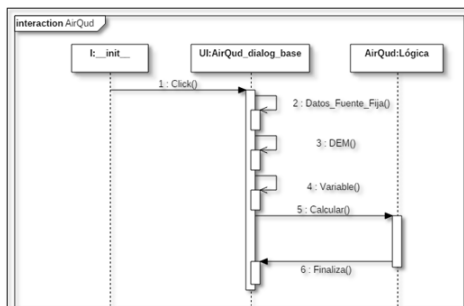


Figure 5. Diagrama de secuencias. Fuente: Autoras.

6.2.2. Diagrama de clases

Este diagrama forma parte del modelo estructural y representa un modelo estático de la extensión, sirve para mostrar la visibilidad entre clases, que trata de la habilidad de un objeto de tener una referencia hacia otro objeto. En la Figura 6 se muestra el diagrama de clases de la extensión donde se observan cada una de los atributos, operaciones y la visibilidad que tiene cada clase (flechas), donde este último si la flecha tiene dos puntas se denomina bidireccional y representa que las clases se pueden ver entre sí, mientras que la si la flecha tiene una sola punta se denomina direccional y solo la clase que apunta a la otra puede ver a esta última.

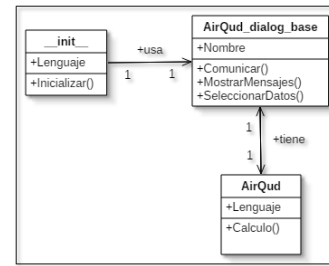


Figure 6. Diagrama de Clases. Fuente: Autoras.

6.2.3. Diagrama de paquetes

Este diagrama forma parte del modelo estructural de la extensión y representa las dependencias entre los paquetes que componen un modelo. Es decir, muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones. En la Figura 7 se presenta el diagrama de paquetes de la extensión donde se organizan los elementos en grupos, permitiendo decidir: Por dónde empezar? y Qué se puede hacer en paralelo?. Para el desarrollo de la extensión se puede realizar en paralelo el diseño de la interfaz junto con el desarrollo de la funcionalidad del mismo.

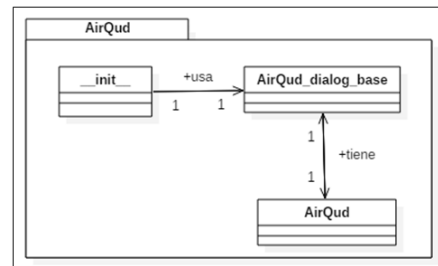


Figure 7. Diagrama de Paquetes. Fuente: Autoras.

6.2.4. Vista de alto nivel

El diagrama de alto nivel describe al usuario de una manera sencilla la arquitectura del software, en el cual se puede identificar que se desarrolló la interfaz usando PyQt mientras que la lógica o modelo de negocio se desarrolló usando PyQgis, y el software GRASS como se muestra en la Figura 8.

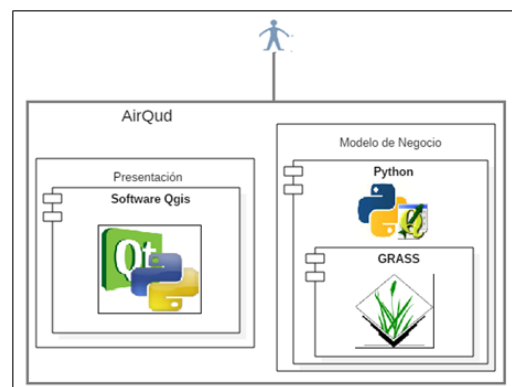


Figure 8. Vista de alto nivel. Fuente: Autoras.

6.2.5. Vista de despliegue

En la Figura 9 se muestra el diagrama de despliegue que representa un modelo monolítico de la arquitectura de la extensión, en el cual todos los componentes operan en un mismo computador usando como entorno de ejecución un software libre de escritorio.

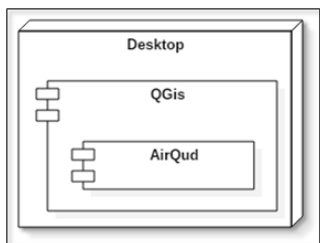


Figure 9. Vista de Despliegue.Fuente:Autoras.

6.2.6. Vista de implementación

El diagrama de implementación de la Figura 10 tiene como objetivo capturar las decisiones que se tomaron entorno a la arquitectura para la implementación, sirve como base para asignar actividades durante la codificación y determinar la cantidad de código a desarrollar. Para la extensión solo se necesita codificar la parte lógica y gracias a QGIS Qt Designer es fácil diseñar la interfaz.

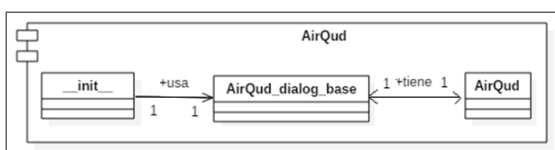


Figure 10. Vista de Implementación.Fuente:Autoras.

6.3. Implementación

Con el fin de crear el código de la extensión se utilizaron dos capas de información espacial: datos de chimeneas ubicadas y monitoreadas en el Municipio de Cauca; así como también el DEM de esta zona, el cual se descargó de la página web del Servicio Geológico de los Estados Unidos. Partiendo de una capa vector de puntos o fuentes fijas se generó un área de influencia entre éstas sobre la capa raster o DEM de la zona. Luego de tener definida el área de influencia se codificaron las funciones v.surf.rst y v.surf.idw de GRASS para realizar la interpolación de las variables y/o contaminantes El área de influencia se generó a partir del siguiente código:

```
## Usa el método de correr algoritmo, algoritmo polygonfromlayer...
processing.runalg('qgis:polygonfromlayerextent',cLayer.dataSourceUri(),\
"false", output_vector_path)

## área de influencia de capa vectorial
area_inf = QgsVectorLayer("D:\\bbox.shp", 'area_influencia', 'ogr')

## se adiciona al panel de capas la capa de area_inf creada
QgsMapLayerRegistry.instance().addMapLayer(area_inf)
## Selección de todos los objetos de la capa area_inf
area_inf.selectAll()
## Se instancia un objeto con la selección de la capa area_inf
fets=area_inf.selectedFeatures()
## Rectángulo método boundingbox borde de la caja
## Se adiciona un margen de 1 En el rectángulo creado
rect=fets[0].geometry().boundingBox()
xmin = rect.xMinimum() - 0.01
ymax = rect.yMaximum() + 0.01
xmax = rect.xMaximum() + 0.01
ymin = rect.yMinimum() - 0.01
print xmin, ymax, xmax, ymin

## Definir el extent con string parámetro de entrada cada uno de las esquinas
extent = "projwin " + str(xmin) + " " + str(ymax) + " " + str(xmax) + " " + str(ymin) + " "
```

Figure 11. Código área de influencia Vector.Fuente:Autoras.

Posteriormente sobre el DEM (mapa digital de elevación) de la zona en la que están ubicadas las fuentes, se plasmó el área de influencia generada, con el siguiente código:

```
## Trae el proveedor de la capa raster
cRaster = QgsMapLayerRegistry.instance().mapLayersByName(
self.dlg.comboBoxRaster.currentText())[0].dataProvider()
## donde está almacenada la capa
## donde sale el recorte del raster
output_raster_path = "D:\\salida.tif"
## cambia el input por el output raster
cmd = "gdal_translate " + extent + cRaster.dataSourceUri() + " " + output_raster_path
print os.system(cmd) # como validador donde 0 es ok 1 indica error

## Cargar la capa Raster creada con el drive gdal
area_inf_dem = QgsRasterLayer("D:\\salida.tif", 'area_influencia_dem', 'gdal')
```

Figure 12. Código área de influencia Raster.Fuente:Autoras.

En la Figura 13 corresponden al área de influencia que se crea ya se para una fuente, dos fuentes y más de dos fuentes. Si el usuario selecciona una capa con una sola fuente la extensión calculará un área de influencia alrededor de esta de aproximadamente 2 km2.

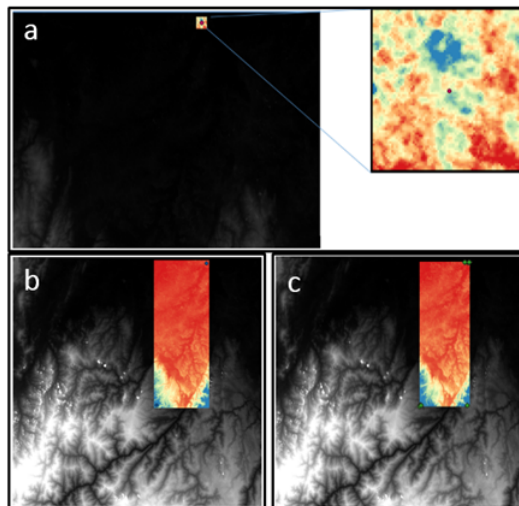


Figure 13. Área de influencia. a) Una fuente, b) dos fuentes y c) más de dos fuentes.

Posterior a la definición del área de influencia se definió la programación de la interpolación por dos métodos, Splines o Distancia Inversa Ponderada, por medio de las funciones de GRASS v.surf.rst y v.surf.idw respectivamente. En la Figura 14 se muestra la parte del código con la cual se programaron los métodos de la extensión.

```
var = self.dlg.comboBoxContami.currentText()
print var
if (self.dlg.comboBoxFunc.currentText()=="IDW"):
processing.runalg('grass7:v.surf.idw', Layer, 2,\
2, str(var), "false",\
str(xmin) + " " + str(xmax) + " " + str(ymin) + " " + str(ymax),\
0.000278, -1, 0.0001,\
"D:\\interpolo_IDW.tif")
inter = QgsRasterLayer("D:\\interpolo_IDW.tif", 'interpolacion_IDW', 'gdal')
QgsMapLayerRegistry.instance().addMapLayer(inter)
elif (self.dlg.comboBoxFunc.currentText()=="RST"):
processing.runalg('grass7:v.surf.rst', Layer, "",\
area_inf_dem, str(var), 40,40,300,0.000100,\
2.500,1.00,0,0, "false", "false", \
str(xmin)+","+str(xmax)+","+str(ymin)+","+str(ymax),\
0.000278,-1,0.0001,\
"D:\\interpolo_RST.tif",\
None, None, None, None)
inter = QgsRasterLayer("D:\\interpolo_RST.tif", 'interpolacion_RST', 'gdal')
QgsMapLayerRegistry.instance().addMapLayer(inter)
```

Figure 14. Programación de los métodos de interpolación.

Posterior a codificación del área de influencia y la interpolación se procedió con la generación de la extensión AirQud, por medio de la herramienta Plugin Builder que permite la generación de extensiones por medio de un formulario. El diseño de la Interfaz se realizó con la herramienta Qt Designer de QGIS, de tal forma que el usuario pueda seleccionar la capa con los datos de la o las fuentes de emisión, seleccionar el mapa de elevación y seleccionar entre los dos métodos de interpolación programados.

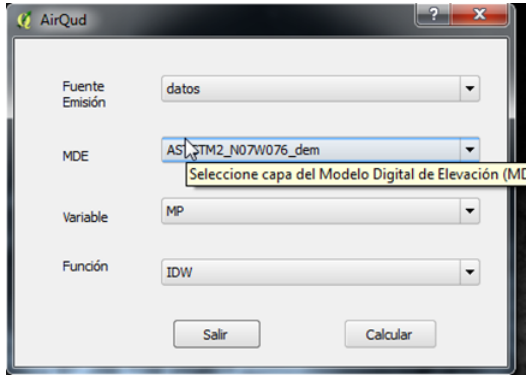


Figure 15. Interfaz gráfica de usuario de la extensión AirQud.

6.4. Pruebas

6.4.1. Funcionalidad

Las pruebas de funcionalidad de la extensión se realizaron con tres capas tipo vector, la primera contiene datos de una fuente, la segunda con datos de fuentes fijas y la tercera con datos de 10 fuentes fijas de emisión de contaminantes, cuyos atributos se relacionan en la tabla 1.

Columna	Descripción
Puntos	Identificación de la fuente fija
X	Coordenada X
Y	Coordenada Y
altura_fuente	Altura de la fuente en metros
diame_ducto	Diámetro del ducto de la chimenea en metros
veloc_flujo	Velocidad del flujo de los gases en m/s
temperat_K	Temperatura promedio de los gases en Kelvin
caudal_gas	Caudal del gas en la chimenea en m ³ /min
mp	Material Particulado en mg/m ³
SO ₂	Dióxido de azufre SO ₂ en mg/m ³
NO _x	Óxido de nitrógeno NO _x en mg/m ³

Table 1. Variables capas vectoriales

Adicionalmente una capa raster que corresponde al DEM (modelo digital de elevación) del municipio de Caucaasia ubicado al nororiente del departamento de Antioquia (Figura 16).

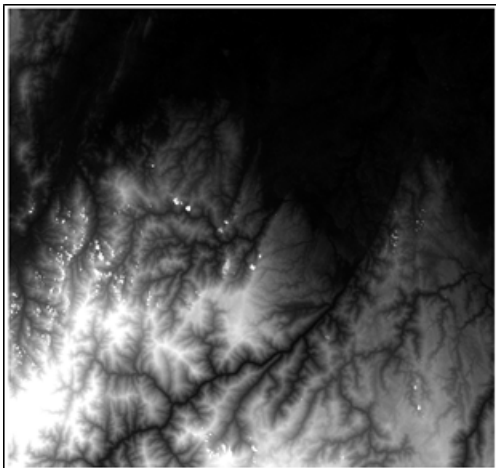


Figure 16. DEM Municipio de Caucaasia.

La extensión presenta cuatro campos los cuales permiten desplegar una serie de datos, que el usuario debe modificar

para luego dar clic en el botón calcular que permite ejecutar la funcionalidad. Se realizaron pruebas de funcionalidad para cada una de las funciones implementadas y para tres casos de datos de fuentes de emisión puntuales o chimeneas, cuando se tiene una chimenea, dos chimeneas o más. En la Figura 17 se muestra el resultado de la extensión para la capa de “uno” la cual contiene información para una sola fuente de emisión; para el parámetro de material particulado “PM” para las funciones a) RST y b) IDW, donde la extensión genera un recorte sobre el DEM de 2km cuadrados aproximadamente sobre el cual se calcula la interpolación, sin embargo la función RST no genera resultados (nan), ya que esta función necesita más puntos para generar el cálculo, mientras que la función IDW si lo genera debido al parámetro de distancia que maneja.

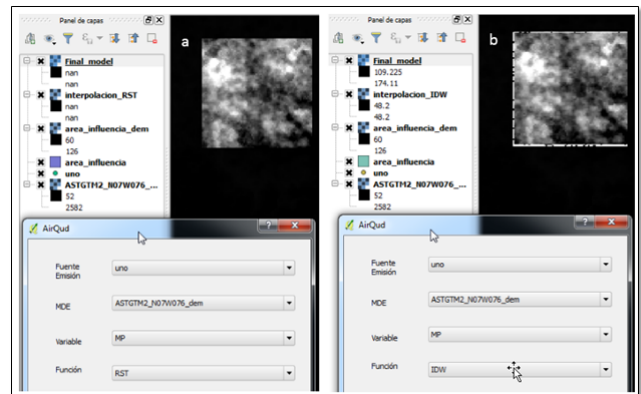


Figure 17. Interfaz con ejemplo para una fuente de emisión. a) RST y b) IDW.

Para el caso en que se tengan dos o más fuentes o chimeneas, el resultado se presentará como se muestra en las Figuras 18 y 19, en las cuales las pruebas de funcionalidad de la extensión con los parámetros de Dióxido de azufre SO₂ y óxido de nitrógeno NO_x respectivamente arroja los resultados del cálculo de interpolación sin tener en cuenta el DEM con el nombre de InterpolaciónIDW y InterpolaciónRST, de acuerdo a lo que decida el usuario sobre qué función usar y FinalModel el nombre con el cual se presenta el resultado de la extensión para cualquiera de las dos funciones teniendo en cuenta el DEM. En el caso de la interpolación por RST el DEM se utiliza como una máscara dentro de los parámetros de entrada a la función.

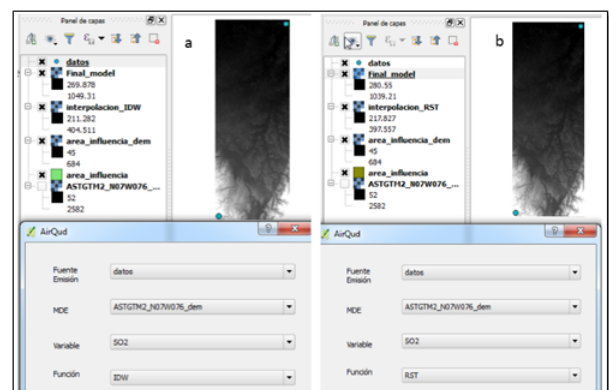


Figure 18. Interfaz con ejemplo para dos fuentes de emisión. a) IDW y b) RST.

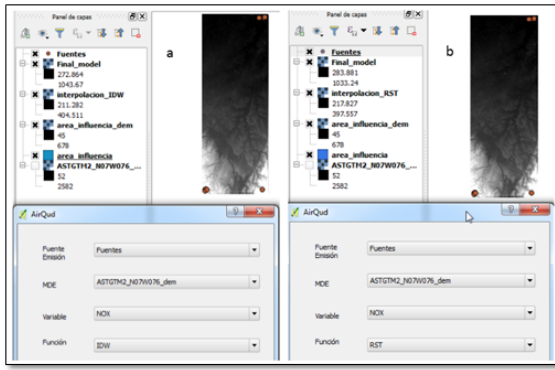


Figure 19. Interfaz con ejemplo para más de dos fuentes de emisión. a) IDW y b) RST.

6.5. Usabilidad

La usabilidad de la extensión se evaluó de acuerdo a las métricas de calidad de software indicadas en la ISO 9126-3 como se observa en la Tabla 2. El indicador se evalúa de 0 a 100%, donde dependiendo de la métrica a evaluar, se dice si es o no satisfactorio, las métricas evaluadas para la extensión cuentan el número de variables que cumplen con lo establecido en cada una de ellas y se comparan con el total de funcionalidades o actividades de la extensión, teniendo entonces lo que se muestra en la Figura 20.

Métricas de Usabilidad	Entendimiento	Aprendizaje	Operabilidad	Atractivo
Actividades	Total funciones	Variables evidentes al usuario	Variables con retroalimentación	Cantidad de problemas a realizar tarea
Interfaz	3	3	3	0
Carga de datos	4	4	4	0
Calculo	2	2	2	1
Total	9	9	9	1
Resultado del indicador (%)		100	100	11.1

Figure 20. Evaluación de usabilidad ISO 9126-3

La usabilidad de la extensión arroja de forma general unos resultados positivos para cada una de las métricas evaluadas. En cuanto a entendimiento, los resultados evidencian altos estándares de calidad para cada uno de los grupos, ya que el usuario puede identificar con claridad que se debe hacer en cada uno de los campos y para qué sirve cada uno de los botones. En cuanto al aprendizaje, el sistema presenta una adecuada retroalimentación de las funciones presentes en la extensión ya que se muestran mensajes para cada uno de los campos que permitió al usuario saber para qué sirve y como usarlo, como se observa en la Figura 21 se entiende que los botones aunque no tienen mensajes, es claro el nombre que tiene cada uno con respecto a lo que realiza.

En la métrica de operabilidad el indicador debería ser 0 ya que no debe presentar fallas, pero se encontró que para el cálculo de la interpolación por medio de la función RST cuando se tiene una sola fuente de emisión o chimenea no se obtienen resultados.

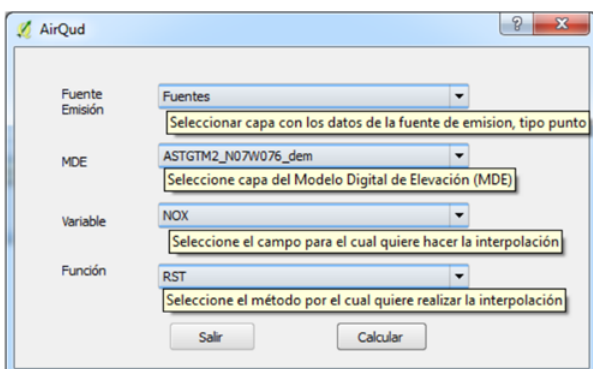


Figure 21. Mensajes para el uso de la extensión.

Finalmente, en cuanto al atractivo de la extensión, el nivel de calidad se ubica en el punto óptimo. Las etiquetas que acompañan cada campo permiten al usuario inferir su objetivo, y la disposición de los elementos evita posibles confusiones.

7. Conclusiones

De acuerdo a las especificaciones de requerimientos y el diseño propuesto en el presente proyecto se implementó una extensión en el software de escritorio QGIS que permite realizar una modelación básica de dispersión de contaminantes por medio dos modelos uno con base curvas Spline y otro por medio de Distancia Inversa Ponderada que pueden ser ejecutados para una o varias fuentes de emisión. La extensión aplica variables mínimas con las cuales se puede estimar la posible dispersión de los contaminantes Dióxido de azufre, Óxido de nitrógeno y Material particulado emitidos por fuentes fijas.

Por medio de la revisión de antecedentes en cuanto a costos, modelos matemáticos, licenciamiento de software de modelos de dispersión y encuestas a usuarios se identificaron los requerimientos funcionales y no funcionales para la extensión propuesta, de tal forma que la extensión brindará una aproximación al usuario de un modelo de dispersión con las variables mínimas para el mismo. El usuario podrá elegir el contaminante de interés, los datos del punto de muestreo, y el mapa de elevación con los cuales desea realizar el cálculo de interpolación ya sea con IDW o RST.

La metodología incremental definida para la creación de la extensión permitió plantear cuatro etapas, las cuales corresponden a la planeación, el diseño, la implementación y las pruebas de funcionalidad del ciclo de vida de la extensión, de tal manera que para cada una se generará un entregable que retroalimenta la siguiente etapa. De la etapa de planeación se derivaron los requerimientos funcionales y no funcionales que sirvieron de insumo para las etapas de diseño e implementación; una vez implementada la extensión se realizaron pruebas de funcionalidad y usabilidad, está última evaluada por medio de la aplicación de los criterios de la norma ISO 9126.

El diseño de la arquitectura permitió el cumplimiento de los requerimientos funcionales y no funcionales establecidos durante la etapas de planeación, siendo esta una arquitectura basada en capas que comprenden el desarrollo de una interfaz usando PyQt y una lógica de negocio usando PyQgis y GRASS, utilizando el lenguaje Python que permitió una codificación dinámica y multiplataforma, este último presenta vez ventajas en cuanto a interoperabilidad ya que esta extensión podrá ser usada en otros clientes SIG como OSGeo y GRASS GIS.

La interfaz diseñada a partir de la herramienta de Qt Designer de QGIS cumple con los requerimientos establecidos; permitiendo al usuario seleccionar el tipo de contaminantes para el cual realizará el modelo, la capa con los datos de la(s) fuente(s), la capa raster con la información de elevación y por último seleccionar el tipo de modelo de cálculo de interpolación, ya sea por el modelo de Splines o Distancia Inversa Ponderada. Igualmente la interfaz cuenta con los mensajes de retroalimentación necesarios para hacer de la extensión una herramienta intuitiva y fácil para el usuario.

La extensión permite obtener una salida gráfica como un archivo tipo raster (.tif) la cual es una es una simulación de un modelo de dispersión de contaminantes básica ya que las funciones utilizadas no permiten la integración de datos meteorológicos, aunque la función RST tiene un parámetro el cual recibe un ángulo que puede tomarse como la dirección del viento predominante en la zona para el momento de la tomas de los datos y un parámetro de tensión para darle relevancia a la variable.

El estándar ISO 9126-3 permitió establecer algunos criterios para

cuantificar los aspectos de calidad en cuanto a usabilidad para la extensión AirQud, donde la carga de datos en la extensión presenta mensajes en cada uno de los campos establecidos, con un adecuado nivel de aprendizaje y que permite una fácil operación de sus funciones con una organización que evita posibles confusiones durante su uso.

8. Recomendaciones

Para futuros trabajos se recomienda incluir a la extensión variables como características del terreno, características de la fuente de emisión, presencia de edificios cercanos a la fuente de emisión, coeficiente de dispersión de acuerdo a la zona predominante alrededor de la fuente (rural o urbana), entre otros sugeridos por la EPA. Los modelos presentados permiten de manera básica la simulación de dispersión de contaminantes, por lo cual se recomienda para futuros trabajos complementar la extensión con un método como el gaussiano que es más completo. Complementar la función RST con el fin de que genera una aproximación más cercana de la dispersión de los contaminantes.

8.1. Bibliografía

- Technotrice. (2013). What Is Incremental Model In Software Engineering Its Advantages and Disadvantages. Estados Unidos. Recuperado de <http://aprendiendo-grass.readthedocs.io/es/latest/index.html>
- Ruiz, José. (2002). Simulación de la contaminación atmosférica generada por fuentes móviles en Bogotá. Bogotá D.C. Facultad de Ciencias. Universidad Nacional de Colombia.
- Arce, Armando. (2014). Aprendiendo GRASS. Recuperado de <http://aprendiendo-grass.readthedocs.io/es/latest/index.html>
- Hofierka, Jaroslav, Blazek, Radim (s.f.). v.surf.rst. University of Presov. Recuperado de <https://grass.osgeo.org/grass64/manuals/v.surf.rst.html>.
- Shapiro, Michael, Blazek, Radim (s.f.). v.surf.idw. U.S. Army Construction Engineering Research Laboratory. Recuperado de <https://grass.osgeo.org/grass64/manuals/v.surf.idw.html>
- U.S. (2002) Geological Survey.. (s.f.). Estados Unidos.: Recuperado de <https://www.usgs.gov/>
- ISO IEC TR 9126 3:2003 Software engineering Product quality Part 3: Internal metrics
- Mitasova, H., Mitas, L. and Harmon, R.S., 2005, Simultaneous spline approximation and topographic analysis for lidar elevation data in open source GIS, IEEE GRSL 2 (4), 375- 379.
- Hofierka, J., 2005, Interpolation of Radioactivity Data Using Regularized Spline with Tension. Applied GIS, Vol. 1, No. 2, pp. 16-01 to 16-13. DOI: 10.2104/ag050016
- Hofierka J., Parajka J., Mitasova H., Mitas L., 2002, Multivariate Interpolation of Precipitation Using Regularized Spline with Tension. Transactions in GIS 6(2), pp. 135-150.
- H. Mitasova, L. Mitas, B.M. Brown, D.P. Gerdes, I. Kosinovsky, 1995, Modeling spatially and temporally distributed phenomena: New methods and tools for GRASS GIS. International Journal of GIS, 9 (4), special issue on Integrating GIS and Environmental modeling, 433-446.
- Mitasova, H. and Mitas, L., 1993: Interpolation by Regularized Spline with Tension: I. Theory and Implementation, Mathematical Geology ,25, 641-655.
- Mitasova, H. and Hofierka, J., 1993: Interpolation by Regularized Spline with Tension: II. Application to Terrain Modeling and Surface Geometry Analysis, Mathematical Geology 25, 657-667.
- Mitas, L., and Mitasova H., 1988, General variational approach to the approximation problem, Computers and Mathematics with Applications, v.16, p. 983-992.
- Neteler, M. and Mitasova, H., 2008, Open Source GIS: A GRASS GIS Approach, 3rd Edition, Springer, New York, 406 pages.
- Talmi, A. and Gilat, G., 1977 : Method for Smooth Approximation of Data, Journal of Computational Physics, 23, p.93-123.
- Wahba, G., 1990, : Spline Models for Observational Data, CNMS-NSF Regional Conference series in applied mathematics, 59, SIAM, Philadelphia, Pennsylvania.
- Shapiro, Michael , U.S. Army Construction Engineering Research Laboratory Improved algorithm (indexes points according to cell and ignores points outside current region) by Paul Kelly.
- Koerper, Greg . Global Climate Research Project. U.S. EPA Environmental Research Laboratory.
- Grohmann, Carlos.(2004). Morphometric analysis in geographic information systems: applications of free software GRASS and R
- Fang Huang, Dingsheng Liub, Xicheng Tanc, Jian Wangb, Yunping Chena, Binbin Hea. (2010). Explorations of the implementation of a parallel IDW interpolation algorithm in a Linux cluster-based parallel GIS.
- Jaroslav, Hofierka, Michal, Gally, Ján, Kaňuk.(). Spatial Interpolation of Airborne Laser Scanning Data with Variable Data Density.
- Armstrong and Marciano, (1994). Inverse distance weighted spatial interpolation using a parallel supercomputer Photogrammetric Engineering and Remote Sensing pp. 1097-1103.
- Armstrong and Marciano, (1996). Local interpolation using a distributed parallel supercomputer International Journal of Geographical Information Systems pp. 713-729.

- Bartier and Keller,(1996). Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (IDW) Computers and Geosciences pp. 795–799.
- Chang et al.,(2006). The parameter optimization in the inverse distance method by genetic algorithm for estimating precipitation Environmental Monitoring and Assessment pp. 145–155.
- GDF Hannover,(2005).GDF Hannover, 2005. An Introduction to the Practical Use of the Free Geographical Information System GRASS 6.0 (Version 1.2), 156pp. http://www.gdf-hannover.de/lit_html/grass60_v1.2_en/index.html
- Guan and Wu,(2008).Parallel optimization of IDW interpolation algorithm on multicore platform. In: Proceedings of Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Advanced Spatial Data Models and Analyses, Guangzhou, China, pp. 7146Y-1–7146Y-9.
- Weng Q,(2006) An evaluation of spatial interpolation accuracy of elevation data Progress in Spatial Data Handling.Springer-Verlag, Berlin: 805-824.
- Nevtipilova,Veronika,Pastwa,Justyna, Singh,Mukesh and Vozenilek, Vit.(2014).Testing Artificial Neural Network (ANN) for Spatial Interpolation.
- Summerfield, Mark. (2007). Rapid GUI Programming with Python and Qt.
- Dobias, Martin. (2010).PyQGIS Documentation.
- Hinz, Matthias. Nüst,Daniel. Proß, Benjamin. Pebesma,Edzer.(2013).Spatial Statistics on the Geospatial Web.
- Quantum GIS Development Team. (2013) Quantum GIS. Geographic Information System. <http://qgis.osgeo.org/>.
- GDAL Development Team. (2013) GDAL - Geospatial. Data Abstraction Library, Version 1.9.2. <http://www.gdal.org>.
- Cataldi, Lage. Pessacq, y García Martínez. Ingenieria De Software Educativo.. <http://www.iidia.com.ar/rgm/comunicaciones/c-icie99-ingenieriasoftwareeducativo.pdf>.
- Anaya, Raquel.(2006). Una visión de la enseñanza de la Ingeniería de Software como apoyo al mejoramiento de las empresas de software.
- Ramirez, Salomón.(2016). Ingeniería De Software Para SIG . Introducción a la Ingeniería de Software.
- Ramirez, Salomón.(2016). Ingeniería De Software Para SIG . Procesos, metodologías y flujos de trabajo de software.
- Ramirez, Salomón.(2016). Ingeniería De Software Para SIG . Ingeniería de Requerimientos.
- Ramirez, Salomón.(2016). Ingeniería De Software Para SIG . Diseño de software y arquitectura.
- Alan J. Cimorellia, Steven G. Perryb, Akula Venkatramc, Jeffrey C. Weild, Robert J. Paine, Robert B. Wilsonf, Russell F. Leeg, Warren D. Petersh, and Roger W. Brodei.AERMOD: A Dispersion Model for Industrial Source Applications. Part I: General Model Formulation and Boundary Layer Characterization.
- Levya,Jonathan.Spenglera,John.Hlinkab, Dennis David Sullivanb.Moonc,Dennis (2002). Using CALPUFF to evaluate the impacts of power plant emissions in Illinois: model sensitivity and implications.
- Fonseca,Mariam.Mayor,Yandy.Fernández,Ana y Cuesta,Osvaldo.(2015).Análisis de sensibilidad del campo de viento producido por el modelo de diagnóstico CALMET.
- Martín,Fernando. Palomino, Inmaculada.García, Marta.(2002).Aplicación De Un Modelo De Dispersión Para La Evaluacion De La Calidad Del Aire En España.
- U.S. Enviromental Protection Agency- https://www3.epa.gov/scram001/dispersion_prefrec.htm#blp.
- Vergara, E.P.Corral, M. Pernía, A. González, A. Estudio De Dispersión De Contaminantes En La Atmósfera: Una Propuesta Metodológica.
- Carrari, Carolina L.Aparicio, Lucrecia V.Bandoni†, José A.Tonelli, Stella M. (2004).Utilizacion De Modelos De Dispersion Atmosferica Para La Estimacion De Dosis De ExposicionUtilizacion De Modelos De Dispersion Atmosferica Para La Estimacion De Dosis De Exposicion.
- Mitsova,H. Mitsova,L.(1993). Interpolation by regularized spline with tension: I theory and implementation. Mathematical Geology, 25 , pp. 641–655.
- Mitsova,H. Mitsova,L.(1993). Interpolation by regularized spline with tension: II theory and implementation. Mathematical Geology, 25 , pp. 641–655.
- López-Grao, Juan Pablo L.Mersegue,José.Campos,Javier.(2004). From UML activity diagrams to Stochastic Petri nets: application to software performance engineering.
- Schmler,Joseph.Aprendiendo UML en 24 horas.