



INSTITUTO SUPERIOR MINERO METALURGICO

“Dr. Antonio Núñez Jiménez”.

Facultad de Metalurgia - Electromecánica

Moa, Holguín

Trabajo de Diploma

Trabajo de diploma para optar por el título de Ingeniería en Informática

“Desarrollo de una interfaz gráfica de usuario para módulo de cálculo de dispersión AERMOD”

Autor: Obed Matos Prats

Tutores:

Ing. Roiky Rodríguez Noa.

Lic. Liban Montes de Oca González.

Consultante:

Dr. Allan Pierre Conde.

Moa, Cuba

2009

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez" (ISMMM) y al Centro de Estudios del Medio Ambiente (CEMA) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del 2008.

Obed Matos Prats

Firma autor

Ing. Roiky Rodríguez Noa.

Firma tutor

AGRADECIMIENTOS

Quiero agradecer a todas las personas que de una forma u otra me han servido de apoyo para el desarrollo de este trabajo, en especial:

A Dios por inspirarme y estar presente en cada línea de código que escribí.

A todo el departamento de informática del CEMA.

A mis abuelos por su amor y comprensión que nunca me han faltado y su constante preocupación respecto a mi superación profesional.

A padres por inculcarme siempre el deseo de estudiar y graduarme.

A mi esposa y niño por su amor y cariño.

A mi compañero de estudios Daríel Raúl Subiros Muñoz por su ayuda y apoyo incondicional.

A Ramona Urgelles Cardoza por su colaboración y ayuda incondicional.

A todos, gracias...

DEDICATORIA

Cuando el culminar una meta deseada impone momentos de especial alegría, cuando se quiere resumir, los momentos de duda, dolor, nostalgia, se deben tener presente aquellas personas que por su dedicación y amor contribuyeron a la realidad que representa llegar hasta el final, en especial dedico este logro personal:

A Dios puesto que Él lo merece todo.

A mis abuelos que siempre han querido verme graduado.

A mis padres por enseñarme y guiarme en el estudio desde pequeño.

A mi esposa y a mi niño por ser tan especiales, por su amor y sus demostraciones de cariño.

A mi hermano por alegrarme la vida con su presencia.

A mis suegros, por acogerme como un hijo más en la familia.

En general a todas aquellas personas que no menciono, no por ser menos importantes sino porque no alcanzarían estas pocas líneas para mencionarlos.

PENSAMIENTO

“El sabio no dice todo lo que piensa, pero siempre piensa todo lo que dice.”

Aristóteles

RESUMEN

La Agencia de Protección Ambiental de los Estados Unidos de América (EUA), (EPA¹) estableció el 21 de octubre del 2005 el sistema de modelos AERMOD², como el modelo de uso recomendado para la dispersión de contaminantes a escala local [1]. La utilización de este modelo permite conocer el comportamiento de los contaminantes dispersos en el aire para tomar las medidas pertinentes.

La aplicación del sistema de modelos AERMOD resulta complicada, pues su interfase con el usuario no es idónea, debido a su limitada capacidad expresiva e interactiva, tipo de texto sin colores y con unos caracteres limitados, por lo que interpretar los resultados es difícil.

En la investigación que se presenta se propuso implementar una interfaz gráfica de usuario amigable para el módulo de cálculo de dispersión atmosférica AERMOD, que facilite la entrada de los datos e interpretación de los resultados.

Se utilizaron herramientas libres compatibles con Linux/Ubuntu y la metodología ágil XP (Extreme Programming), que facilitaron la implementación del sistema propuesto, con representación gráfica de los resultados, textos con colores e imágenes y formularios para introducción de los datos.

¹ Environmental Protection Agency.

² American Meteorology Society – E.P.A. - Regulatory **MO**Del

ABSTRACT

The Environmental Protection Agency of the United States of America (USA), (EPA) established October of 2005 the pollution modelling system AERMOD, as the one recommended to use for the dispersion of pollutants at local scale [1]. Using this modelling system allows to know the behavior of the dispersed pollutants in the air to take the pertinent measures.

The application of the system of models AERMOD is complicated, because its interface with the user is not suitable, due to its limited expressive and interactive capacity, text type without colors and with some limited characters, and because of that, it's very difficult to understand its results.

The investigation that is presented is intended to implement a friendly user's graphic interface for the module of calculation of atmospheric dispersion AERMOD, that facilitates the entrance of the data and interpretation of the results.

Free software and Linux/Ubuntu compatible tools, and the agile XP methodology for software development were used to facilitate the design and implementation of the desired system, obtaining graphic representation of results, texts with colors, images and forms for data input.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Los modelos de dispersión	5
1.3 Modelo de dispersión AERMOD.....	6
1.4 La interfaz conceptos fundamentales y evolución.....	7
1.5 Modelos de dispersión visuales Basados en AERMOD.....	9
1.6 Tendencias y Tecnologías Actuales	10
1.6.1 Política de Migración a Software libre	10
1.6.2 Lenguajes de Programación.....	10
1.6.3 Fundamentación de la selección de lenguaje a utilizar.....	15
1.7 Arquitectura.....	16
1.8 Metodologías de Desarrollo De Software.....	17
1.8.1 Proceso de desarrollo de software.....	17
1.8.2 Metodologías.....	18
1.9 Herramientas de Desarrollo.....	22
1.9.1 Netbean 6.0 (IDE).....	22
1.9.2 Eclipse 3.2 IDE	23
1.10 CONCLUSIONES.....	25
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA.....	26
2.1 Introducción.....	26
2.2 Descripción de los procesos vinculados al campo de acción.....	26
2.2.1 Flujo actual del proceso	26
2.2.2 Objeto de automatización.....	26
2.2.3 Personas relacionadas con el sistema.....	27
2.3 Planificación.....	27
2.4 Exploración.....	27
2.5 Planificación de la Entrega	34
2.5.1 Estimación de esfuerzos por historias de usuario	35
2.5.2 Plan de entregas	35
2.6 Plan de iteraciones.....	36
2.6.1 Plan de duración de las iteraciones	37
2.7 Conclusiones.....	38
CAPÍTULO 3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA.....	39
3.1 Introducción.....	39
3.2 Diseño	39
3.3 Tarjetas CRC (Cargo o clase, Responsabilidad y Colaboración).....	39
3.4 Implementación	40
3.5 Conclusiones.....	51
CAPÍTULO 4 PRUEBA	52
4.1 Introducción.....	52
4.2 Pruebas de Aceptación (PA).....	52
4.2 Conclusiones.....	53
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD	54
5.1 Introducción.....	54

5.2	Efectos Económicos	54
5.3	Beneficios y costos intangibles en el proyecto.....	56
5.4	Ficha de costo.....	57
5.5	Conclusiones.....	59
CONCLUSIONES GENERALES		60
RECOMENDACIONES.....		61
REFERENCIAS BIBLIOGRÁFICAS.....		62
BIBLIOGRAFÍA.....		64
GLOSARIO DE TÉRMINOS.....		65
ANEXOS.....		67
	Anexo 1: Tarjetas CRC	67
	Anexo 2: Plantilla de Pruebas de Aceptación.....	71
	Anexo 3: Pruebas de Aceptación	71

ÍNDICE DE TABLAS

Tabla 1: Personas relacionadas con el sistema.....	27
Tabla 2: Representación de una historia de usuario.....	28
Tabla 3: HU Gestionar Opciones de Control.....	28
Tabla 4: HU Gestionar Fuentes.....	29
Tabla 5: HU Gestionar grupo de Fuentes.....	29
Tabla 6: HU Gestionar Factores de Emisión.	30
Tabla 7: HU Gestionar rejilla.....	30
Tabla 8: HU Gestionar Receptores Cartesianos Discretos.....	31
Tabla 9: HU Gestionar la entrada de datos meteorológicos.....	31
Tabla 10: HU Gestionar las opciones de salida.	32
Tabla 11: HU Generar Gráfico.....	32
Tabla 12: HU Abrir y Guardar trabajos realizados.	33
Tabla 13: HU Crear un nuevo trabajo.....	33
Tabla 14: HU Procesar con el AERMOD el archivo de entrada.	34
Tabla 15: Estimación de esfuerzos por historias de usuario.	35
Tabla 16: Plan de entregas.	35
Tabla 17: Plan de duración de las iteraciones.....	37
Tabla 18: Cronograma de entrega.....	37
Tabla 19: Plantilla Tarjeta CRC.....	40
Tabla 20: TI Insertar datos de control.....	41
Tabla 21: TI Modificar datos de control.....	41
Tabla 22: TI Llenar datos de la fuente.....	42
Tabla 23: TI Ver y modificar datos de la fuente.....	42
Tabla 24: TI Llenar los datos de grupos de Fuentes.....	43
Tabla 25: TI Ver y modificar datos de los grupos de Fuentes.....	43
Tabla 26: TI Llenar datos de los factores de Emisión.....	44
Tabla 27: TI Ver y modificar datos de los Factores de Emisión.....	44
Tabla 28: TI Llenar datos de rejilla de Receptores.....	45
Tabla 29: TI Ver y modificar los datos de rejilla de Receptores.....	45
Tabla 30: TI Llenar los datos de los Receptores Cartesianos Discretos.....	46
Tabla 31: TI Ver y modificar los datos de los Receptores Cartesianos Discretos.....	46
Tabla 32: TI Insertar datos meteorológicos.....	47
Tabla 33: TI Modificar datos meteorológicos.....	47
Tabla 34: TI Insertar datos de las opciones de salida.....	48
Tabla 35: TI Modificar datos de las opciones de salida.....	48
Tabla 36: TI Abrir trabajos.....	49
Tabla 37: TI Guardar trabajo realizado.....	49
Tabla 38: TI Crear trabajo nuevo.....	50
Tabla 39: TI Crear Gráfico.....	50
Tabla 40: TI Mostrar Gráfico.....	51
Tabla 41: TI Procesar el archivo aermod.INP.....	51

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Metodología XP.....	20
Ilustración 2: IDE Netbeans 6.0.....	22
Ilustración 3: IDE Eclipse 3.2.....	23
Ilustración 4: Punto de Equilibrio.....	59



INTRODUCCIÓN

La contaminación del aire constituye hoy uno de los principales problemas ambientales de las ciudades del mundo, muchas actividades industriales son altamente contaminantes, afectan de manera directa la calidad del aire y la salud humana.

Esta contaminación, incluye tanto a los países desarrollados como a los que están sumidos en el mayor atraso tecnológico, en los primeros consecuencia de la diversificación de la producción industrial y al flujo intenso de vehículos automotores generalmente, mientras que en los segundos se debe al desarrollo no planificado de las escasas industrias, es decir al uso de tecnologías obsoletas en la producción, los servicios y el transporte, además se debe a la mala calidad del saneamiento básico y al crecimiento urbanístico no planificado.

Cuba no está exenta de esta problemática, que tiene su origen fundamentalmente en deficiencias en ciertas cuestiones de la planificación territorial en cuanto a la ubicación de asentamientos industriales y humanos, así como en el empleo de viejas tecnologías, entre otras razones no ha tenido la posibilidad de avanzar en los últimos 20 años en la modelación de la dispersión. Esto trae como consecuencia que los sistemas utilizados sean atrasados respecto a los recomendados por las organizaciones internacionales de protección del medio ambiente como: EPA [6].

Conocer en que proporciones se ve afectada la calidad del aire dependiendo de los factores que caracterizan la fuente contaminante, la localización de los receptores, los datos meteorológicos, la topografía y uso del terreno es imprescindible para el control de la dispersión de contaminantes.

En Moa se realiza una intensa labor minera y metalúrgica para la obtención del níquel, por lo que se deteriora de manera creciente la calidad del aire en esta región, es de vital importancia la aplicación de los modelos de dispersión más avanzados, para evaluar el impacto ambiental actual de los contaminantes en el aire en la localidad.

Con la aplicación del sistema de modelos AERMOD se pueden evaluar la dispersión de contaminantes desde fuentes continuas y corresponde a un sistema de modelación de emisiones que simula procesos atmosféricos físicos esenciales y provee estimaciones refinadas de concentración sobre un amplio rango de condiciones meteorológicas y escenarios de modelación, permitiendo la evaluación y predicción del impacto causado por la actividad industrial, el tráfico de vehículos, los incendios, o cualquier otra situación en la que



se emitan contaminantes hacia la atmósfera. Estos modelos son útiles para identificar la contribución de las fuentes a la contaminación del aire y sirven de apoyo a la hora de buscar soluciones para mitigar la contaminación atmosférica.

El sistema de modelos AERMOD es el recomendado por La Agencia de Protección Ambiental de los Estados Unidos de América (EPA) para la dispersión de contaminantes a escala local establecido el 21 de octubre del 2005. [1]. Ha sido demostrado y documentado, tanto por evidencias científicas como por estudios de validación, que el AERMOD representa un sólido y significativo avance respecto a otros modelos de dispersión. La formulación del módulo de cálculo AERMOD ha sido sometida a una revisión profunda e independiente, concluyéndose que las bases científicas del AERMOD, están al nivel del estado del arte de la ciencia [2] [3] [4] [5].

En la página Web de la (EPA) están disponibles varios simuladores libres y a código abierto, con aplicaciones diferentes entre ellos el AERMOD. Este es difícil de aplicar ya que la interfase con el usuario no es idónea debido a que se encuentra en modo texto o consola con lo que esto implica: limitada capacidad expresiva, un tipo de texto sin colores y con unos caracteres limitados, además la entrada de datos se realiza por medio de un fichero texto que usa un formato predeterminado y que el usuario debe respetar para poder obtener resultados correctos, lo que resulta engorroso, con grandes posibilidades de cometer errores y la demora en este proceso es grande, luego se le pasa este fichero al AERMOD (Módulo de cálculo), y este genera como resultado otro fichero texto que contiene los resultados, que son grandes cantidades de datos difíciles de interpretar.

Existen otros sistemas de modelación visuales que tienen despliegue gráficos de los resultados y facilitan la entrada de los datos pero son costosos (CALPUFF View US\$3,495.00 AERMOD View US\$1, 599.00) y su código es privativo, por lo que sería difícil su aplicación para un país en vías de desarrollado.

Teniendo en cuenta lo antes expuesto se plantea el siguiente **problema científico**: ¿Cómo facilitar la entrada de los datos e interpretación de los resultados en el módulo de cálculo de dispersión AERMOD?

En correspondencia con el problema planteado el **objeto de estudio** es: el sistema de modelado AERMOD y el **campo de acción**: Interfaz grafica de usuario del módulo de cálculo de dispersión AERMOD.



Objetivo General:

Construir una interfaz gráfica de usuario amigable para el módulo de cálculo de dispersión AERMOD, que facilite la entrada de los datos e interpretación de los resultados.

Objetivos específicos:

1. Diseñar e Implementar una interfaz gráfica para el módulo de cálculo de dispersión AERMOD, que facilite la entrada de los datos.
2. Obtener una representación gráfica de los datos de salida del módulo de cálculo de dispersión AERMOD, que permita una mejor apreciación de los resultados.
3. Obtener el manual de usuario.

Como **Hipótesis** partimos de la idea de que: Si se desarrolla una interfaz gráfica de usuario para el módulo de cálculo de dispersión AERMOD que permita la entrada de los datos con despliegue gráfico y personalizado de los resultados facilitará la entrada y análisis de los datos.

Para cumplir los objetivos propuestos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

1. Estudio de las características fundamentales del sistema de modelado AERMOD
2. Estudio de las funcionalidades del módulo de cálculo de dispersión AERMOD.
3. Estudio del formato y orden de entrada de los datos al módulo de cálculo de dispersión AERMOD.
4. Estudio del formato de los datos de salida del módulo de cálculo de dispersión AERMOD.
5. Seleccionar las herramientas, lenguajes y metodología a utilizar.
6. Prueba y documentación del Sistema.

La metodología utilizada es fundamentalmente cualitativa, se emplearon métodos como:

Métodos empíricos

Entre los métodos empíricos usados se puede citar **la entrevista y la observación** para la recopilación de la información. La entrevista permitió determinar los principales requerimientos del sistema y funcionalidades que necesita plasmadas en las historias de usuarios. La observación fue útil para entender el comportamiento del sistema y sus especificaciones.



Métodos teóricos

Entre los métodos teóricos se puede encontrar: **análisis y síntesis, Hipotético-deductivo, Histórico y Lógico y modelación.** Mediante el análisis y síntesis de la documentación disponible se conoció el funcionamiento actual de los modelos de dispersión y en la confección del informe final. El Hipotético-deductivo en la elaboración de la Hipótesis, a partir de la cual se realizaran deducciones que arriben a la solución del problema. La modelación permitió realizar una representación de la realidad, se logró detectar problemas en la forma actual de procesar la información y encontrar las funcionalidades que debe de tener el sistema que se propone, que lo harán más completo y le brindarán satisfacción al usuario con un producto de mayor calidad. Histórico y Lógico para investigar el desarrollo que ha tenido el tema (antecedentes) y apoyar los conocimientos que sobre este existen.

El presente trabajo está estructurado en 4 capítulos.

Capítulo 1: Contiene la fundamentación teórica del tema, donde se abordan los lenguajes de programación y las tecnologías que se utilizan en el desarrollo de la aplicación. También se exploran soluciones existentes similares al campo de acción para tener una guía de las posibles automatizaciones que se pueden realizar.

Capítulo 2: Aborda los aspectos funcionales para el desarrollo del sistema, se definen los procesos fundamentales por medio de las historias de usuarios creadas por el cliente y se realiza la planificación de entrega de los diferentes módulos que componen la aplicación.

Capítulo 3: Este capítulo contiene los aspectos relacionados con el diseño e implementación del sistema. Se presenta la técnica de tarjetas CRC y además las tareas de ingeniería de cada módulo del sistema para garantizar la entrega en la planificación establecida.

Capítulo 4: Este capítulo está dedicado a las pruebas que se le realizan al funcionamiento del software, las pruebas de aceptación del cliente. Las pruebas se realizan por módulos para la aceptación de cada uno de forma independiente.

Capítulo 5: Este capítulo está dedicado al estudio de factibilidad del proyecto mediante el uso de la metodología Costo Efectividad (Beneficio) la cual plantea la conveniencia de la ejecución del proyecto.

Para concluir se muestran las Conclusiones a las que se arribaron, las Recomendaciones que se proponen, las Referencias Bibliográficas, la Bibliografía utilizada, Glosario de Términos y Anexos con información necesaria sobre el trabajo.



CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este Capítulo se muestra una visión general del escenario donde radica el objeto de estudio de la investigación, los conceptos principales asociados al dominio del problema y a la solución planteada. También se justifica la elección del lenguaje así como la metodología usada para el desarrollo de este trabajo y sus principales características.

Por otra parte se hace una valoración de las características de los softwares que han trabajado la temática, analizando las desventajas de los mismos, así como los beneficios del sistema.

1.2 Los modelos de dispersión

Los modelos de dispersión atmosférica son herramientas informáticas que permiten obtener proyecciones de un contaminante en el ambiente.

Se ingresan los datos obtenidos en el muestreo pertinente y el programa muestra los resultados requeridos. Por ende, la exactitud de la proyección dependerá de como se hallan tomado los datos y el correcto ingreso de los mismos al programa de modelación.

Para hacer sus estimaciones los modelos incorporan gran cantidad de fórmulas matemáticas, basadas en fundamentos teóricos acerca del comportamiento de los contaminantes, y su interacción con el ambiente.

Por lo que no basta con conocer el funcionamiento del modelo, sino que hay que familiarizarse con los conceptos teóricos que fundamentan las variables; para comprender, por ejemplo, el comportamiento de una pluma de emisión en una atmósfera estable o inestable.

Existen gran variedad de modelos, adecuados a las diferentes condiciones del ambiente, características del contaminante y la fuente emisora, entre otras particularidades; es decir, el mismo modelo no es siempre apropiado para todos los ambientes y/o contaminantes. Un ejemplo claro de esto, es la utilización de modelos globales para calcular las emisiones de CO₂, que es ridículo utilizarlos para proyectar la emisión de una industria, a escala local.



Por lo tanto, la elección del modelo de dispersión adecuado requerirá de buen criterio y conocimiento de los conceptos teóricos que fundamentan los algoritmos componentes del modelo, por parte de los encargados del monitoreo ambiental. [7].

1.3 Modelo de dispersión AERMOD

AERMOD es un modelo de dispersión de contaminantes atmosféricos recomendado por la EPA para la evaluación de dispersión de contaminantes desde fuentes continuas y corresponde a un sistema de modelación de emisiones que simula procesos atmosféricos físicos esenciales y provee estimaciones refinadas de concentración sobre un amplio rango de condiciones meteorológicas y escenarios de modelación.

AERMOD incluye un programa central (AERMOD) y dos pre-procesadores de datos, AERMET y AERMAP.

AERMET es un pre-procesador meteorológico, que procesa la capa límite y otros parámetros necesarios para el funcionamiento de AERMOD y acepta datos de fuentes ubicadas tanto en el sitio como fuera de este. AERMET crea dos archivos: un archivo de datos de superficie y un archivo de datos de viento cuyas extensiones son SFC y PFL respectivamente.

AERMAP es un pre-procesador de topografía que utiliza los datos del terreno para calcular una altura de influencia representativa del terreno (h_c), también referida como escala de altura del terreno. Este pre-procesador es usado también para crear grillas de receptores.

AERMOD es un Modelo de Pluma Estacionario aplicable tanto a áreas rurales como urbanas, a terrenos planos y complejos, y a múltiples tipos de fuentes (puntuales, areales y volumétricas).

En la capa límite estable, asume que la distribución de la concentración es Gaussiana tanto para el eje x como para el eje y. En la capa límite convectiva, la distribución horizontal también es asumida Gaussiana, pero la distribución vertical es descrita mediante una Función de Densidad de Probabilidad Bi-Gaussiana.

El modelo incorpora, mediante un acercamiento simple, conceptos básicos acerca de flujo y dispersión de contaminantes en terrenos complejos, donde la pluma es modelada tanto donde impacta como siguiendo el terreno. Este acercamiento ha sido diseñado para ser físicamente realista y simple de implementar, eliminando así la necesidad de distinguir entre



terrenos simples, intermedios y complejos, como es requerido para el resto de los modelos regulatorios.

AERMOD elimina la necesidad de definir regimenes de terrenos complejos, todo el terreno es manejado de manera consistente y continua.

Parámetros requeridos por el modelo:

El modelo requiere como datos de entrada los siguientes parámetros:

- Datos referentes a las características del sistema al que se pretende aplicar el modelo, es decir, la ubicación de las fuentes emisoras y receptores con sus correspondientes características y datos de emisión de las mismas.
- Datos meteorológicos del sector a modelar. Los parámetros meteorológicos mínimos que necesita el programa para operar son:
 - Mediciones superficiales horarias de: velocidad del viento, dirección del viento y temperatura ambiente.
 - Cobertura de nubes de la zona y altura de la capa nubosa.

Datos referentes a las características topográficas y características del suelo donde se realiza la modelación. En la presente modelación se construyó un archivo con la descripción de las elevaciones del terreno, el que fue usado como una grilla receptora. [8]

1.4 La interfaz conceptos fundamentales y evolución

Al término interfaz se le han atribuido varios significados:

En software, parte de un programa que permite el flujo de información entre un usuario y la aplicación, o entre la aplicación y otros programas o periféricos. Esa parte de un programa está constituida por un conjunto de comandos y métodos que permiten estas intercomunicaciones.

Interfaz también hace referencia al conjunto de métodos para lograr interactividad entre un usuario y una computadora. Una interfaz puede ser del tipo GUI, o línea de comandos, etc. También puede ser a partir de un hardware, por ejemplo, el monitor, el teclado y el mouse, son interfaces entre el usuario y el ordenador.

En electrónica, un interfaz es el puerto por el cual se envían o reciben señales desde un sistema hacia otros. Por ejemplo, el interfaz USB, interfaz SCSI, interfaz IDE, interfaz puerto paralelo o serial.



En **sentido amplio**, puede definirse **interfaz** como el conjunto de comandos y/o métodos que permiten la intercomunicación del programa con cualquier otro programa o entre partes (módulos) del propio programa o elemento interno o externo. De hecho, los periféricos son controlados por interfaces. [9]

En el contexto del proceso de interacción persona - ordenador, la interfaz gráfica de usuario (GUI) es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático. [10] Es de vital importancia en la actividad diaria pues permite al usuario interactuar de manera visual con el software y hardware. La GUI, no fue siempre tan vistosa o amigable como lo es en la actualidad, existen muchas dependiendo del sistema operativo.

La primera GUI fue desarrollada por investigadores en Xerox Palo Alto Research Center (PARC) en los años '70. El primer computador personal que usó una interfaz moderna fue Xerox Alto, desarrollado en 1973. [11]

En 1984, System 1.0 fue la primer GUI de Sistema Operativo desarrollado por Macintosh. Tuvo muchas características de un moderno sistema operativo usando ventanas e íconos.

En 1985 Microsoft desarrolló su primera GUI de sistema operativo, con íconos de 32x32 pixeles, así han ido evolucionado las GUI en los distintos sistemas operativos.

Java es un lenguaje de programación multiplataforma que permite desarrollar software que no dependen del sistema operativo. Este contiene las librerías AWT y SWING que permiten desarrollar interfaces gráficas de usuario con varias apariencias llamadas Look and Feel, que tienen grandes capacidades expresivas como: colores, diferentes tipos de texto, botones e imágenes.

Ventajas y desventajas de las GUI

Analizar las ventajas y limitaciones de las interfaces GUI resulta interesante, no necesariamente son las más eficientes y en muchos casos complican los procesos y distraen al usuario.

En segundo lugar, las interfaces constituyen una pantalla que se interpone entre el usuario y las funciones del software, y por lo tanto constituyen un marco operativo que limita las opciones del usuario.

En tercer lugar la estructura de la interfaz y su navegabilidad constituye una estructura que define el proceso de diseño del usuario.



En este sentido la interfaz es una componente delicada porque puede facilitar o inhibir la creatividad de los usuarios. La interfaz debe posibilitar el acceso a todo el potencial del software, es decir, el usuario debe ser aquel que diseña los procesos en lugar de utilizar procesos ya diseñados.

Trabajando con gramáticas, el usuario debe tener una herramienta que facilite el diseño, la implementación y las pruebas de estas gramáticas, editarlas con facilidad, guardarlas en el disco, analizarlas, entre otras.

Estas opciones son esencialmente de interfaz, ya que el código correspondiente no presenta ninguna dificultad especial, eso es, no cuesta mucho implementarlas.

1.5 Modelos de dispersión visuales Basados en AERMOD

Aquí se intentará dar una visión de varios modelos de dispersión visuales basados en AERMOD desde el punto de vista de las características principales, áreas de aplicación, requisitos técnicos.

❖ AERMOD View

AERMOD VIEW (antes conocido como ISC-AERMOD View) es un simulador de modelos de dispersión de aire que analiza de forma intuitiva, robusta y estable la concentración y deposición de la contaminación atmosférica originada por diversas fuentes. Ofrece en una sola interfaz los modelos de referencia internacional: ISCST3, ISC-PRIME y AERMOD.

❖ CALPUFF View

La interfaz más potente para la evaluación de la dispersión tipo Puff (Soplo) CALPUFF View es una interfaz basada en GIS diseñada para el modelo de referencia CALPUFF aprobado por la US EPA. Se compone de tres módulos:

- **CALMET:** Un modelo de diagnóstico meteorológico en 3-D que acepta campos de viento preprocesados
- **CALPUFF:** Es uno de los Modelos de dispersión de aire más avanzados que existen en la actualidad. Un modelo tipo soplo (PUFF) que ofrece información completa hora a hora de la variación espacial del viento y su estabilidad. Contiene algoritmos adicionales que le permiten emular al potente AERMOD en distancias cortas donde los modelos PUFF son generalmente menos fidedignos. Es razonablemente preciso en distancias no superiores a los 300 Km



- **CALPOST:** Programa de Postprocesado que compila resultados de CALPUFF.

1.6 Tendencias y Tecnologías Actuales

1.6.1 Política de Migración a Software libre

Se denomina software libre a todo aquel que permita a los usuarios ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. A menudo es confundido con el software gratuito, sin embargo no se trata de una cuestión de precio sino de libertad. Precisamente, las cuatro libertades que se definen son:

1. La libertad de ejecutar el programa para cualquier propósito.
2. La libertad de estudiar cómo trabaja el programa y adaptarlo a sus necesidades (El acceso al código fuente es una condición necesaria).
3. La libertad de redistribuir copias para que pueda ayudar al vecino.
4. La libertad de mejorar el programa y publicar sus mejoras y versiones modificadas en general para que se beneficie toda la comunidad (El acceso al código fuente es una condición necesaria). [12]

Las ventajas especialmente económicas que aportan las soluciones libres y las aportaciones de la comunidad de desarrollo han permitido un constante crecimiento del software libre hasta superar en ocasiones al mercado propietario. Estas ventajas hacen que el país siga una política de migración hacia el software libre y como parte de este proceso se decide para el desarrollo de la aplicación la utilización de herramientas y tecnologías pertenecientes al software libre.

1.6.2 Lenguajes de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. [13] A continuación se realiza un estudio de algunos de los distintos lenguajes de programación más conocidos para desarrollar aplicaciones de escritorio.



C++.

Es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

El nombre C++ fue propuesto por Rick Masciatti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases".

Se puede decir que abarca tres paradigmas de la programación:

1. La programación estructurada.
2. La programación genérica.
3. La programación orientada a objetos.

Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel:

- Posibilidad de redefinir los operadores (sobrecarga de operadores).
- Identificación de tipos en tiempo de ejecución.

Está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo obliga a hacerlo casi todo manualmente al igual que C lo que "dificulta" mucho su aprendizaje.

Delphi.

Es un entorno de desarrollo de software (IDE) diseñado para la programación de propósito general con énfasis en la programación visual. El Delphi utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal. Es producido comercialmente por la empresa estadounidense CodeGear. En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET.

Está basado en una versión moderna de Pascal, denominada Object Pascal. Borland en los últimos años defendía que el nombre correcto del lenguaje es también Delphi, posiblemente debido a pretensiones de marca, aunque en sus mismos manuales el nombre del lenguaje aparecía como Object Pascal, por lo que la comunidad de programadores no ha adoptado



mayoritariamente este cambio (supuesta aclaración, según Borland). Object Pascal expande las funcionalidades del Pascal estándar:

- Soporte para la programación orientada a objetos (habitualmente llamada POO) también existente desde Turbo Pascal 5.5, pero más evolucionada en cuanto a:
 - Encapsulación: declarando partes privadas, protegidas, públicas y publicadas de las clases
 - Propiedades: concepto nuevo que luego han adaptado muchos otros lenguajes. Las propiedades permiten usar la sintaxis de asignación para setters y getters.
 - Simplificación de la sintaxis de referencias a clases y punteros.
- Soporte para manejo estructurado de excepciones, mejorando sensiblemente el control de errores de usuario y del sistema.
- Programación activada por eventos (event-driven), posible gracias a la técnica de delegación de eventos. Esta técnica permite asignar el método de un objeto para responder a un evento lanzado sobre otro objeto. Fue adoptada por Niklaus Wirth, autor del Pascal Original, e incorporada a otros de sus lenguajes como Component Pascal.

Java.

Es un lenguaje de programación desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software



libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

Las características principales que ofrece Java son:

Simple

Ofrece toda la funcionalidad de un lenguaje potente. Debido a que C y C++ son los lenguajes más difundidos, Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Elimina muchas de las características de otros lenguajes para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.

Orientado a objetos

Con el objetivo de mantener la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, necesitan ser construidas y destruidas en espacios de memoria.

Incorpora funcionalidades como por ejemplo, la resolución dinámica de métodos mediante una interfaz específica llamada RTTI (RunTime Type Identification) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

Distribuido

Se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.



Robusto

Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

Arquitectura neutral

Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows NT, Linux, Irix, Aix, Mac, Apple y probablemente haya grupos de desarrollo trabajando en el porting a otras plataformas.

Seguro

La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el casting se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador.

El código Java pasa muchos tests antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto-.

El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.



Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

Multihebra

Soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

1.6.3 Fundamentación de la selección de lenguaje a utilizar

Teniendo en cuenta las características de los lenguajes antes mencionados se ha seleccionado Java para el desarrollo de la aplicación por contar con las siguientes características:

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- Aritmética de punteros.
- No existen referencias.
- Registros (struct).
- Definición de tipos (typedef).
- Macros (#define).
- Necesidad de liberar memoria (free).
- Aunque, en realidad, lo que hace es eliminar las palabras reservadas (struct, typedef), ya que las clases son algo parecido.
- Además, el intérprete completo de Java que hay en este momento es muy pequeño, solamente ocupa 215 Kb de RAM.

Con respecto a delphi tiene ventajas en cuanto a seguridad ya que al tener la característica de ser interpretado el código Java pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso



sobre objetos o intenta cambiar el tipo o clase de un objeto. El Cargador de Clases, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.

1.7 Arquitectura

La Arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". Se adoptó los patrones arquitectónicos MVC y Singleton para el desarrollo de la propuesta de solución.

Modelo Vista Controlador (MVC). Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Presenta el modelo en un formato adecuado.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) logrando un mantenimiento más rápido y sencillo de las aplicaciones.

Ventajas del Modelo Vista Controlador

- La separación del Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación en caso de ser requerido.

Patrón de diseño singleton (instancia única). Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de



acceso global a ella. Se implementa creando en la clase un método que crea una instancia del objeto sólo si todavía no existe alguna.

Ventajas de singleton

- El patrón singleton provee una única instancia global.
- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no sea instanciable directamente.

1.8 Metodologías de Desarrollo De Software.

1.8.1 Proceso de desarrollo de software

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora de los procesos de software.

El **proceso de desarrollo de software** "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". [14]

Las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas, proyecto, producto y proceso. Estos términos son conocidos como las cuatro «P» en el desarrollo de software. El resultado final de un proyecto software es un producto, donde intervienen personas a través de un proceso de desarrollo de software que guía los esfuerzos de las personas implicadas en el proyecto.



1.8.2 Metodologías

Hoy en día, llevar a cabo el desarrollo de un buen software depende de un gran número de actividades y etapas donde elegir la mejor metodología para el equipo influye directamente en el futuro éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos. Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. Teniendo en cuenta ambos enfoques se da paso al análisis de dos de las metodologías más usadas actualmente.

El Proceso Unificado de Desarrollo (RUP).

Es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está **dirigido por casos de uso**: que orientan al proyecto a la importancia para el usuario y lo que se quiere, está **centrado en la arquitectura**: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es **iterativo e incremental**: donde divide el proyecto en mini-proyectos donde los casos de uso y al arquitectura cumplen sus objetivos de manera depurada. RUP propone cuatro etapas para el desarrollo de un producto: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros. El Proceso



Unificado de Desarrollo tiene 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce *release* del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (*Unified Model Language*), para describir todo el proceso basándose en componentes. Este lenguaje es estándar, con él se puede visualizar, especificar, construir y documentar los artefactos de un sistema.

XP (*Extreme Programming*)

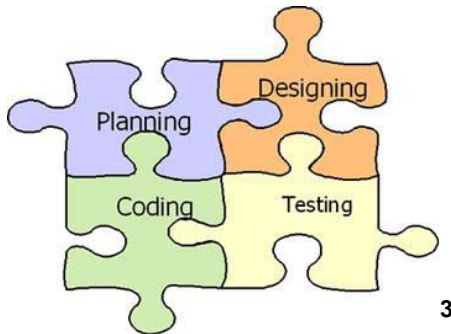


Ilustración 1: Metodología XP

Es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los programadores que la practican consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Es utilizada para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Las características fundamentales son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera- el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.

³ http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html



- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que en más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Ventajas

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, ya conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- Permite definir en cada iteración cuales son los objetivos de la siguiente.
- Permite la retroalimentación.
- La presión esta a lo largo de todo el proyecto y no en una entrega final.

Desventajas

- Delimitar el alcance del proyecto con el cliente.

Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia.



Fundamentación de la selección de la metodología de desarrollo de software.

XP y RUP son dos grandes metodologías que después de analizar sus principales características y los aspectos preponderantes de cada una de ellas, se ha determinado la implantación de XP, una metodología ligera, con menos requerimientos de documentación y planificación para el desarrollo de la aplicación. Si se escoge RUP traería grandes dificultades como son:

Multitud de artefactos: El hecho de realizar varios artefacto y mantenerlos actualizados consume mucho tiempo.

El poco personal de desarrollo: Al ser solo dos personas a cargo del desarrollo de la aplicación, estas tomarían varios roles en cada etapa y sería muy complejo cumplir con las actividades de cada uno de ellos.

Requisitos cambiantes: Los cambios en un proceso de desarrollo son inevitables, al aparecer un nuevo requisito hace que se tenga que comenzar una nueva iteración para dar cumplimiento a su funcionalidad.

Planificación inexistente: La planificación que se realiza en las fases iniciales está sujeta a muchas variaciones en dependencia de los cambios que se experimenten en los requisitos. Por tanto se hace muy difícil planificar actividades específicas. Los inconvenientes planteados pueden ser eliminados con la utilización de la metodología XP.

1.9 Herramientas de Desarrollo

1.9.1 Netbean 6.0 (IDE)



Ilustración 2: IDE Netbeans 6.0

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.



La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

El IDE NetBeans 6.0 es una herramienta para programadores, pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, JavaEE, web, EJB y aplicaciones móviles). Entre sus características se encuentran un sistema de proyectos basado en Ant, control de versiones y refactoring.

El IDE NetBeans IDE 6.0 corre en diferentes sistemas operativos entre los que se pueden mencionar linux, Mac OS X, Solaris y Windows. Como se puede observar se ha lanzado con soporte para la mayoría de las plataformas. Como requerimiento se necesita tener previamente instalado el JDK 5.0 o 6.0.

1.9.2 Eclipse 3.2 IDE

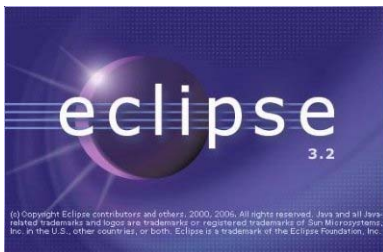


Ilustración 3: IDE Eclipse 3.2

Eclipse comenzó como un proyecto de IBM Canadá. Fue desarrollado por OTI (Object Technology International) como reemplazo de VisualAge también desarrollado por OTI. En noviembre del 2001, se formó un consorcio para el desarrollo futuro de Eclipse como código abierto. En 2003, la fundación independiente de IBM fue creada.

Eclipse es multiplataforma, es decir que puedes encontrar instaladores para instalar el programa en Windows, Mac o Linux. Eclipse empezó siendo de IBM pero ya es independiente.

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés *plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de



otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadata en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

Fundamentación de la herramienta de desarrollo escogida.

Puedes instalar Netbeans en casi cualquier sistema con el mismo instalador, solo se requiere Java en el sistema. Mientras que Eclipse solo corre en otras plataformas si consigues el instalador de respectiva plataforma que lo hace algo frustrante, y decir que Eclipse es más pesado en memoria y viene con menos cosas incluidas.

En NetBeans los módulos son obtenidos por un asistente que se conecta a un servidor universal que contiene módulos generales para el público, y la opción de conectarte a servidores terceros, y si así lo prefieres obtener el módulo en extensión .nbm e instalarlo en NetBeans, mientras que para conseguir módulos (extensiones) en Eclipse debes recurrir a un servidor que tenga módulos disponibles a descargar, es algo molesto cuando buscas un módulo en específico, debes primero saber la dirección URL del servidor donde está dicho módulo.

NetBeans 6.0 ha mejorado de su antecesor en rendimiento, en versiones anteriores al escribir código el desarrollador se sentía que tenía que esperar al programa mientras uno escribía código. NetBeans es más rápido al compilar código, y adaptarse a otros entornos de desarrollo, en eso para mi NetBeans se ha vuelto superior.



1.10 CONCLUSIONES

Los modelos de dispersión atmosférica son herramientas informáticas que permiten obtener proyecciones de un contaminante en el ambiente.

AERMOD es un modelo de dispersión de contaminantes atmosféricos recomendado por la EPA para la evaluación de dispersión de contaminantes desde fuentes continuas.

Existen varios modelos de dispersión visuales: AERMOD VIEW y CALPUFF VIEW.

En el contexto del proceso de interacción persona - ordenador, la interfaz gráfica de usuario (GUI) es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

La interfaz es una componente delicada porque puede facilitar o inhibir la creatividad de los usuarios.

Existen múltiples lenguajes, metodologías y tecnologías para el desarrollo de aplicaciones. Después de realizar un estudio de las más utilizadas se escogieron las que presentan mayores ventajas. Entre ellas se seleccionó el lenguaje de programación Java. Se elige el IDE de programación Netbean 6.0. Se decide realizar la aplicación sobre la base de la metodología ágil XP.

Optar por una base tecnológica basada en software libre permite estar a tono con la estrategia nacional respecto a este tema. Escoger una metodología de desarrollo ágil como XP permite obtener resultados visibles a corto plazo.



CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

No existe una metodología universal para hacer frente, con éxito, a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo y tipo de sistema). La metodología seleccionada para desarrollar el software fue XP motivo por el cual cada uno de los artefactos generados se encuentran condicionados al uso de la misma.

Se presentan de la primera fase de la metodología las historias de usuarios realizadas por el cliente, así como toda la planificación de entrega para la implementación de las mismas.

2.2 Descripción de los procesos vinculados al campo de acción

2.2.1 Flujo actual del proceso

La información que se entra al AERMOD se realiza mediante ficheros de texto con el formato determinado para cada palabra clave y ruta (PATHWAY) que determinan la salida de los resultados en otro fichero texto.

La información se entra en el fichero distribuido en: Opciones de Control, Opciones de Fuentes, Red de Receptores, Entrada Meteorológica y Opciones de Salida respectivamente. Estas rutas contienen las palabras claves que determinan la información que se dará como resultado en la salida.

2.2.2 Objeto de automatización

Los usuarios del AERMOD tienen que gestionar información en grandes cantidades, lo que se hace difícil y tedioso a la hora de insertar esta en el fichero texto ya que lo tienen que hacer manualmente aumentando así la posibilidad de cometer errores y el tiempo que se emplea en ello, además se dificulta la interpretación de los resultados.

La realización de una interfaz gráfica de usuario que facilite la entrada de la información al AERMOD y la interpretación de los resultados es la propuesta que tiene este trabajo. El mismo facilitaría la entrada de la información al AERMOD, también facilitaría la interpretación de los resultados mediante gráficos.



2.2.3 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo.

Personas relacionadas con el sistema	Justificación
Especialista	Es la persona que tiene conocimientos en la parte de dispersión de contaminantes, y se encarga de entrar los datos y de interpretar los resultados.

Tabla 1: Personas relacionadas con el sistema.

2.3 Planificación

Planificación es la primera fase de la metodología XP, en la misma quedan definidos los procesos que el cliente quiere automatizar y el tiempo en que quedará concluida la aplicación.

2.4 Exploración

En esta etapa el cliente crea las historias de usuarios que son de interés para la entrega del producto. Además el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto.

Las historias de usuarios se utilizan para especificar las funcionales del software. Cada una de estas es suficientemente sencilla y comprensible para que el programador pueda implementarla en un corto período de trabajo. Se caracterizan por prioridad, riesgo y esfuerzo, además son la base de las pruebas de aceptación, pues mediante estas se verifica si se cumplen las expectativas del cliente. Para definir las historias de usuario se emplea la siguiente plantilla.



Historia de Usuario	
Número: (Número de la historia de usuario incremental en el tiempo)	Usuario: (sistema que protagoniza la historia)
Nombre historia: (El nombre de la historia de usuario para identificarlas mejor entre los desarrolladores y el cliente)	
Prioridad en negocio: (La importancia que tiene para el cliente) (Alta / Media / Baja)	Riesgo en desarrollo: (Qué tan difícil es para el desarrollador) (Alto / Medio / Bajo)
Puntos estimados: Un punto, equivale a una semana ideal de programación	Iteración asignada: (La iteración a la que corresponde)
Programador responsable: Nombre(s) de los programadores encargados (Paco Valero – David Ferrer)	
Descripción: (Se especifican las operaciones del usuario y las respuestas que dará el sistema)	
Observaciones: (Algunas observaciones de interés, como glosario, información sobre usuarios, etc.)	

Tabla 2: Representación de una historia de usuario.

A continuación se presentan las historias de usuarios que fueron creadas por el cliente.

Historia de Usuario	
Número: 1	Usuario: Especialista
Nombre historia: Gestionar Opciones de Control	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 1/2	Iteración asignada: 1
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de las opciones de control. Debe guardar (los títulos, opciones de modelado, períodos de promedio, contaminante, decaimiento exponencial, coeficiente de decaimiento)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar. Decaimiento exponencial y coeficiente de decaimiento son opcionales. Se pueden insertar uno o dos títulos.	

Tabla 3: HU Gestionar Opciones de Control.



Historia de Usuario	
Número: 2	Usuario: Especialista
Nombre historia: Gestionar Fuentes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de las Fuentes. Debe guardar (el identificador, tipo de fuente, localización, parámetros, etc.)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar. Se debe insertar por cada identificador de fuente su respectivo tipo localización y parámetro.	

Tabla 4: HU Gestionar Fuentes.

Historia de Usuario	
Número: 3	Usuario: Especialista
Nombre historia: Gestionar grupo de Fuentes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de los Grupos de Fuentes. Debe guardar (el identificador del grupo con sus fuentes correspondientes)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar. También se pueden agrupar todas las fuentes en un grupo (ALL).	

Tabla 5: HU Gestionar grupo de Fuentes.



Historia de Usuario	
Número: 4	Usuario: Especialista
Nombre historia: Gestionar Factores de Emisión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de los Factores de emisión. Debe guardar (el identificador del factor de emisión con sus fuentes y tasa de flujo correspondientes)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar. Cada Factor de emisión debe tener al menos una fuente	

Tabla 6: HU Gestionar Factores de Emisión.

Historia de Usuario	
Número: 5	Usuario: Especialista
Nombre historia: Gestionar rejilla	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de las rejillas. Debe guardar (tipo de rejilla, identificador de la rejilla, origen polar (x , y) o identificador de la fuente, rejilla polar, cartesiana, cartesiana generada o polar generada, elevación de la rejilla(Por direcciones y distancias entre anillos) y altura de banderilla de la rejilla)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar.	

Tabla 7: HU Gestionar rejilla.



Historia de Usuario	
Número: 6	Usuario: Especialista
Nombre historia: Gestionar Receptores Cartesianos Discretos	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de los receptores cartesianos discretos. Debe guardar (Los valores para las x en dirección este y para las y en dirección norte, además de los valores de elevación y banderilla)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar. Los valores de elevación y banderilla, son opcionales, dependen de su selección en las opciones de control.	

Tabla 8: HU Gestionar Receptores Cartesianos Discretos.

Historia de Usuario	
Número: 7	Usuario: Especialista
Nombre historia: Gestionar la entrada de datos meteorológicos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1/2	Iteración asignada: 2
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de los datos meteorológicos. Debe guardar (el nombre de archivo y estructura para los archivos de datos meteorológicos de perfil y superficie, la información sobre la superficie, información sobre el aire superior, elevación base sobre MSL (Nivel medio del mar) para el perfil de temperatura potencial)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar.	

Tabla 9: HU Gestionar la entrada de datos meteorológicos.



Historia de Usuario	
Número: 8	Usuario: Especialista
Nombre historia: Gestionar las opciones de salida	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de gestionar la información de la información de salida. Debe guardar (la especificación de los valores más altos por receptor y la selección de los valores máximos de la tabla de salida global)	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar.	

Tabla 10: HU Gestionar las opciones de salida.

Historia de Usuario	
Número: 9	Usuario: Especialista
Nombre historia: Generar Gráfico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de graficar la concentración de un determinado contaminante en una dirección. Debe graficar los valores de concentración en una dirección seleccionada para las distancias determinadas.	
Observaciones:	

Tabla 11: HU Generar Gráfico.



Historia de Usuario	
Número: 10	Usuario: Especialista
Nombre historia: Abrir y Guardar trabajos realizados	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de guardar y abrir trabajos realizados. Debe de poder abrir y mostrar trabajos guardados con los valores de opciones de control, fuentes, receptores, datos meteorológicos, y opciones de salida.	
Observaciones:	

Tabla 12: HU Abrir y Guardar trabajos realizados.

Historia de Usuario	
Número: 11	Usuario: Especialista
Nombre historia: Crear un nuevo trabajo	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de crear un trabajo nuevo. Debe de poder crear un trabajo nuevo comenzando con todos los registros vacíos.	
Observaciones:	

Tabla 13: HU Crear un nuevo trabajo.



Historia de Usuario	
Número: 12	Usuario: Especialista
Nombre historia: Procesar con el AERMOD el archivo de entrada	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Obed Matos Prats	
Descripción: El Especialista debe de tener la posibilidad de procesar con el aermod el archivo de entrada generado por la aplicación que se realizó. Debe de poder pasarle al aermod el archivo de entrada generado por la aplicación que realizamos y obtener del aermod un archivo con los resultados.	
Observaciones: Se le pasa al AERMOD el archivo .inp generado por la aplicación que realizamos y el aermod genera un archivo de salida .out con los resultados.	

Tabla 14: HU Procesar con el AERMOD el archivo de entrada.

2.5 Planificación de la Entrega

El cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Para la estimación se utiliza como medida el punto, el mismo equivale a una semana (6 días) ideal de programación. Generalmente las historias valen de 1 a 3 puntos.



2.5.1 Estimación de esfuerzos por historias de usuario

Historia de Usuario	Número	Puntos de Estimación
Gestionar Opciones de Control	1	1/2
Gestionar Fuentes	2	2
Gestionar grupo de Fuentes	3	1
Gestionar Factores de Emisión	4	2
Gestionar rejilla	5	2
Gestionar Receptores Cartesianos Discretos	6	1
Gestionar la entrada de datos meteorológicos	7	1/2
Gestionar las opciones de salida	8	1
Generar Gráfico	9	1
Abrir y Guardar trabajos realizados	10	2
Crear un nuevo trabajo	11	1
Procesar con el AERMOD el archivo de entrada	12	1

Tabla 15: Estimación de esfuerzos por historias de usuario.

El plan de entregas se realiza teniendo en cuenta las unidades funcionales que se quieren entregar y cada uno de estos módulos abarca un número de historias de usuarios a implementar para dar cumplimiento al funcionamiento del mismo.

2.5.2 Plan de entregas

Módulos	Historias de usuario que abarca
Control	1
Fuentes	2,3,4
Receptores	5,6
Meteorología	7
Salida	8
Archivo	10,11
Compilar Aermod	12
Gráficos	9

Tabla 16: Plan de entregas.



2.6 Plan de iteraciones

Para el desarrollo del sistema en cuestión se han planificado 4 iteraciones teniendo en cuenta las características de las historias de usuario necesarias para su implementación, la distribución de las historias de usuario por cada iteración queda de la siguiente manera:

Iteración 1: En esta iteración se les dará solución a las historias de usuarios correspondientes a los módulos **Control y Fuentes**. Las opciones de Fuentes son uno de los procesos de alta prioridad para el cliente. Se considera que algunas de las historias correspondientes a esta iteración no necesitan una semana completa de implementación, pues las mismas son sencillas y así se puede dedicar el tiempo restante a las demás que tienen un grado mayor de complejidad.

Iteración 2: A esta iteración le corresponden las historias correspondientes a los módulos **Receptores y Meteorología** que son también de alta prioridad.

Iteración 3: Para esta iteración se implementarán las historias de usuarios corresponden a los módulos **Salida y Archivo**.

Iteración 4: Para esta iteración se implementarán las historias de usuarios que no se pueden realizar sin antes haber implementado las demás ya que se corresponden generalmente con la generación de gráficos. Son las correspondientes a los módulos **Compilar Aermod y Gráficos**.



2.6.1 Plan de duración de las iteraciones

Iteraciones	Orden de las historias usuario a implementar	Duración total de las iteraciones en semanas
1	<ul style="list-style-type: none"> • Gestionar Opciones de Control • Gestionar Fuentes • Gestionar grupo de Fuentes • Gestionar Factores de Emisión 	5 y 1/2
2	<ul style="list-style-type: none"> • Gestionar rejilla • Gestionar Receptores Cartesianos Discretos • Gestionar la entrada de datos meteorológicos 	3 y 1/2
3	<ul style="list-style-type: none"> • Gestionar las opciones de salida • Abrir y Guardar trabajos realizados • Crear un nuevo trabajo 	4
4	<ul style="list-style-type: none"> • Procesar con el AERMOD el archivo de entrada • Generar Gráfico 	2

Tabla 17: Plan de duración de las iteraciones.

Como producto del plan de entregas y el plan de iteraciones se harán *releases* al sistema en las fechas que se muestran a continuación.

Módulos	Final 1ra iteración	Final 2da iteración	Final 3ra iteración	Final 4ta iteración
Control	11-03-2009	-	-	-
Fuentes	11-03-2009	-	-	-
Receptores	-	4- 4 - 2009	-	-
Meteorología	-	4- 4 - 2009	-	-
Salida	-	-	2-5-2009	-
Archivo	-	-	2-5-2009	-
Compilar Aermod	-	-	-	16-5-2009
Gráficos	-	-	-	16-5-2009

Tabla 18: Cronograma de entrega.



2.7 Conclusiones

La entrada de la información al AERMOD y la interpretación de los resultados junto con las historias de usuario creadas por el cliente es el objetivo de automatización del presente trabajo. A partir de estas historias se realiza toda la planificación de entrega del software, la misma se realiza conjuntamente con el cliente que es el actor principal en la etapa de planificación de la metodología seleccionada. Se decide realizar el software en cuatro iteraciones para garantizar la entrega al cliente en el tiempo establecido.



CAPÍTULO 3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1 Introducción

En el presente capítulo se abordan las fases diseño e implementación de la metodología XP. Uno de los artefactos fundamentales es la creación de las tarjetas CRC (Clase-Responsabilidades-Colaboración) las cuales permiten brindar un mayor enfoque orientado a objetos. Por otra parte se describen cada una de las tareas planificadas para llevar a cabo el desarrollo de cada una de las historias de usuario detectadas.

3.2 Diseño

Modelo-Vista-Controlador (MVC) en Java Swing [15]

Java tiene como características fundamentales su portabilidad a un gran número de plataformas (Java es en sí una plataforma), su simplicidad y su extenso conjunto de librerías. Su arquitectura está profundamente basada en MVC, lo que proporciona un alto grado de extensibilidad y de personalización de los componentes de la librería.

Permite “conectar” y “desconectar” estilos de interfaz de usuario (llamados “look and feels”) que modifican la forma en que se muestra y se comporta toda la interfaz de usuario, así, la misma aplicación puede verse como una aplicación Windows o como una aplicación Motif7 simplemente conmutando el look and feel en tiempo de ejecución.

Permite la creación de interfaces de usuario de una manera sencilla y rápida, permitiendo el manejo del patrón MVC pero ocultando los detalles de su implementación. El mecanismo de eventos de Java se adapta perfectamente al mecanismo de notificaciones de MVC. Al estar los modelos separados de la vista, las posibilidades de extensión de la librería y de personalización de componentes ya existentes son enormes. Permite al usuario crear sus propias estructuras de datos y adaptar la interfaz de usuario a ellas y no a la inversa, como sucede con librerías ya implementadas.

3.3 Tarjetas CRC (Cargo o clase, Responsabilidad y Colaboración).

XP propone realizar diseños simples y sencillos, hacerlo todo lo menos complejo posible para lograr que sea entendible e implementable. Realizar una correcta especificación de los nombres de métodos y clases ayuda a comprender mejor lo diseñado y facilita las



posteriores ampliaciones y la reutilización del código. Nunca se debe añadir funcionalidades extras al software aunque se piense que serán factibles en el futuro.

El uso de la técnica de las tarjetas CRC para diseñar con la máxima simplicidad posible permite al programador centrarse en el desarrollo orientado a objetos. Estas tarjetas representan objetos, la clase a la que pertenece el objeto se escribe en la parte superior de la tarjeta, en una columna a la izquierda se escriben las responsabilidades u objetivos que debe cumplir el objeto y a la derecha las clases que colaboran con cada responsabilidad [16] como se muestra en la siguiente plantilla:

Clase	
Responsabilidades	Colaboraciones

Tabla 19: Plantilla Tarjeta CRC.

Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar.

Las tarjetas determinan el comportamiento de cada actividad. Ver [anexo1](#).

3.4 Implementación

El cliente es una parte más del equipo de desarrollo, su presencia es indispensable en las distintas fases de X.P. En el momento de codificar una historia de usuario su presencia es aún más necesaria. Los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que esta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada. La optimización del código siempre se debe dejar para el final.

De las historias de usuarios creadas por el cliente se generan tareas de ingeniería (TI) o tareas de programación como también se les conoce, algunas historias no se van a dividir en tareas pues se les dará solución conjuntamente como propone XP en sus principios, la programación en parejas para una mejor optimización del código.



El desarrollo del software se planificó en cuatro iteraciones de trabajo. En la primera de estas se implementan los módulos **Control y Fuentes**. Las historias de usuarios a las que se les da solución en estos módulos se encuentran en Tabla #16: Plan de entrega.

HU #1 Gestionar Opciones de Control

Tarea	
Número tarea: 1	Número historia: 1
Nombre tarea: Insertar datos de control	
Tipo de tarea : Desarrollo	Puntos estimados: 0.25
Fecha inicio: 1/02/09	Fecha fin: 2/02/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra las opciones para que se pueda insertar la información	

Tabla 20: TI Insertar datos de control

Tarea	
Número tarea: 2	Número historia: 1
Nombre tarea: Modificar datos de control	
Tipo de tarea : Desarrollo	Puntos estimados: 0.25
Fecha inicio: 2/02/09	Fecha fin: 3/02/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra información previamente insertada y permite hacer cambios en la información mostrada.	

Tabla 21: TI Modificar datos de control



HU #2 Gestionar Fuentes

Tarea	
Número tarea: 3	Número historia: 2
Nombre tarea: Llenar datos de la fuente	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 3/02/09	Fecha fin: 10/02/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos de la fuente y da la opción de guardarlos.	

Tabla 22: TI Llenar datos de la fuente

Tarea	
Número tarea: 4	Número historia: 2
Nombre tarea: Ver y modificar datos de la fuente	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 11/02/09	Fecha fin: 18/02/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema brinda la opción de listar los datos de la Fuente. Además si desea modificarlo se muestra la forma de modificar, además da la opción de guardar los cambios.	

Tabla 23: TI Ver y modificar datos de la fuente



HU #3 Gestionar grupo de Fuentes

Tarea	
Número tarea: 5	Número historia: 3
Nombre tarea: Llenar los datos de grupos de Fuentes	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 18/02/09	Fecha fin: 21/02/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos y da la opción de guardarlos.	

Tabla 24: TI Llenar los datos de grupos de Fuentes

Tarea	
Número tarea: 6	Número historia: 3
Nombre tarea: Ver y modificar datos de los grupos de Fuentes	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 21/02/09	Fecha fin: 24/02/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema brinda la opción de listar los datos de los grupos de Fuentes. Además si desea modificarlo se muestra el formulario correspondiente y da la opción de guardar los cambios.	

Tabla 25: TI Ver y modificar datos de los grupos de Fuentes



HU #4 Gestionar Factores de Emisión

Tarea	
Número tarea: 7	Número historia: 4
Nombre tarea: Llenar datos de los factores de Emisión	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 25/02/09	Fecha fin: 4/03/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos y da la opción de guardarlos.	

Tabla 26: TI Llenar datos de los factores de Emisión

Tarea	
Número tarea: 8	Número historia: 4
Nombre tarea: Ver y modificar datos de los Factores de Emisión.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 4/03/09	Fecha fin: 11/03/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema brinda la opción de listar los datos de los Factores de Emisión. Además si desea modificarlo se muestra el formulario correspondiente y da la opción de guardar los cambios.	

Tabla 27: TI Ver y modificar datos de los Factores de Emisión.

La **iteración 2** se dedica a al módulo **Receptores y Meteorología**. Las historias de usuarios que abarca se pueden encontrar en la Tabla #16: Plan de entrega.



HU #5 Gestionar rejilla

Tarea	
Número tarea: 9	Número historia: 5
Nombre tarea: Llenar datos de rejilla de Receptores.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 12/03/09	Fecha fin: 19/03/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos y da la opción de guardarlos.	

Tabla 28: TI Llenar datos de rejilla de Receptores.

Tarea	
Número tarea: 10	Número historia: 5
Nombre tarea: Ver y modificar los datos de rejilla de Receptores.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 19/03/09	Fecha fin: 26/03/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema brinda la opción de listar los datos de las rejillas de receptores. Además si desea modificarlo se muestra el formulario correspondiente y da la opción de guardar los cambios.	

Tabla 29: TI Ver y modificar los datos de rejilla de Receptores.



HU #6 Gestionar Receptores Cartesianos Discretos

Tarea	
Número tarea: 11	Número historia: 6
Nombre tarea: Llenar los datos de los Receptores Cartesianos Discretos.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 26/03/09	Fecha fin: 29/03/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos y da la opción de guardarlos.	

Tabla 30: TI Llenar los datos de los Receptores Cartesianos Discretos.

Tarea	
Número tarea: 12	Número historia: 6
Nombre tarea: Ver y modificar los datos de los Receptores Cartesianos Discretos.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 29/03/09	Fecha fin: 1/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema brinda la opción de listar los datos de los receptores cartesianos discretos. Además si desea modificarlo se muestra el formulario correspondiente y da la opción de guardar los cambios.	

Tabla 31: TI Ver y modificar los datos de los Receptores Cartesianos Discretos.



HU #7 Gestionar la entrada de datos meteorológicos

Tarea	
Número tarea: 13	Número historia: 7
Nombre tarea: Insertar datos meteorológicos.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.25
Fecha inicio: 2/04/09	Fecha fin: 3/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos	

Tabla 32: TI Insertar datos meteorológicos.

Tarea	
Número tarea: 14	Número historia: 7
Nombre tarea: Modificar datos meteorológicos.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.25
Fecha inicio: 3/04/09	Fecha fin: 4/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra información previamente insertada y permite hacer cambios en la información mostrada.	

Tabla 33: TI Modificar datos meteorológicos.

La iteración 3 se dedica a al módulo Salida y Archivo



HU #8 Gestionar las opciones de salida

Tarea	
Número tarea: 15	Número historia: 8
Nombre tarea: Insertar datos de las opciones de salida.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5/04/09	Fecha fin: 8/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos de las opciones de salida	

Tabla 34: TI Insertar datos de las opciones de salida.

Tarea	
Número tarea: 16	Número historia: 8
Nombre tarea: Modificar datos de las opciones de salida.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 8/04/09	Fecha fin: 11/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra información previamente insertada y permite hacer cambios en la información mostrada.	

Tabla 35: TI Modificar datos de las opciones de salida.



HU #10 Abrir y Guardar trabajos realizados

Tarea	
Número tarea: 17	Número historia: 10
Nombre tarea: Abrir trabajos.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 11/04/09	Fecha fin: 18/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra información guardada en un fichero de salvas.	

Tabla 36: TI Abrir trabajos.

Tarea	
Número tarea: 18	Número historia: 10
Nombre tarea: Guardar trabajo realizado.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 18/04/09	Fecha fin: 25/04/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema guarda la información contenida en los formularios y variables en un fichero de salvas.	

Tabla 37: TI Guardar trabajo realizado.



HU #11 Crear un nuevo trabajo

Tarea	
Número tarea: 18	Número historia: 11
Nombre tarea: Crear trabajo nuevo.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 25/04/09	Fecha fin: 2/05/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema inicializa todas las variables y formularios en blanco para iniciar un nuevo trabajo.	

Tabla 38: TI Crear trabajo nuevo.

La iteración 4 se dedica a al módulo **Compilar Aermod y Gráficos**

HU #9 Generar Gráfico

Tarea	
Número tarea: 19	Número historia: 9
Nombre tarea: Crear Gráfico.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 2/05/09	Fecha fin: 5/05/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema lee el fichero Aermod.out generado por el Aermod.exe sacando los valores de concentración en una determinada dirección dada por el usuario saca los valores y crea el gráfico en una imagen JPG.	

Tabla 39: TI Crear Gráfico.



Tarea	
Número tarea: 20	Número historia: 9
Nombre tarea: Mostrar Gráfico.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5/05/09	Fecha fin: 8/05/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema muestra la imagen del gráfico creado en un panel.	

Tabla 40: TI Mostrar Gráfico.

HU #12 Procesar con el AERMOD el archivo de entrada

Tarea	
Número tarea: 21	Número historia: 12
Nombre tarea: Procesar el archivo aermod.INP	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 8/05/09	Fecha fin: 15/05/09
Programador responsable: Obed Matos Prats	
Descripción: El sistema le pasa al aermod.exe el archivo texto aermod.inp y obtiene el archivo texto aermod.out con los valores de concentración que serán graficados.	

Tabla 41: TI Procesar el archivo aermod.INP.

3.5 Conclusiones

El empleo de la técnica de las tarjetas CRC en la fase de diseño de la metodología XP, es favorable para alcanzar en el equipo de desarrollo un mayor enfoque en la programación orientada a objetos garantizando así una mejor calidad del código en la fase de implementación. Dividir algunas historias de usuario en tareas de ingeniería a las cuales se le asignaron tiempos de implementación, permite optimizar el tiempo de desarrollo de la aplicación.



CAPÍTULO 4 PRUEBA

4.1 Introducción

En el presente capítulo se muestran las pruebas de aceptación creadas por el cliente para verificar el buen funcionamiento de la aplicación. Las mismas fueron verificadas en cada entrega que se realiza del producto en la planificación establecida.

4.2 Pruebas de Aceptación (PA)

Uno de los pilares de la metodología XP es el proceso de pruebas. La metodología propone probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones.

XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente. [17]

A la hora de codificar no se sigue la regla de XP que aconseja crear pruebas unitarias con entornos de desarrollo antes de programar. Las pruebas se obtienen de la descripción de requisitos plasmados en las historias de usuarios, y estas especifican las barreras que deben pasar las distintas funcionalidades del programa, procurando codificar teniendo en cuenta las pruebas que se deben vencer. Para realizar las pruebas de aceptación el cliente utiliza la siguiente plantilla. Ver [anexo2](#).

Las pruebas realizadas para verificar el funcionamiento de cada módulo en las entregas que se le hacen al cliente cumpliendo con lo establecido en el cronograma de entregas se muestran en el [anexo 3](#).



4.2 Conclusiones

Con la realización de las pruebas de aceptación el cliente verifica el funcionamiento del software, probando individualmente cada módulo y asignándole la evaluación correspondiente. Todas las pruebas realizadas fueron aceptadas por el cliente cumpliendo entonces la aplicación con las historias de usuarios definidas inicialmente.



CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

5.1 Introducción

Para el estudio de factibilidad de este proyecto se utilizará la **Metodología Costo Efectividad (Beneficio)**, la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo, que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados
- La efectividad, que se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo).

5.2 Efectos Económicos

- ✓ Efectos directos
- ✓ Efectos indirectos
- ✓ Efectos externos
- ✓ Intangibles

Efectos directos:

- ✓ POSITIVOS:
 - Se facilitará la entrada de los datos al módulo de cálculo de dispersión AERMOD.
 - Se facilitará la interpretación de los resultados mediante gráficas.
 - Se ganará en velocidad al automatizar la entrada de los datos al módulo de cálculo de dispersión AERMOD.
- ✓ NEGATIVOS
 - Para el empleo de la aplicación es imprescindible el uso de un ordenador, aparejado a los gastos que este trae de consumo de corriente eléctrica y mantenimiento.



Efecto indirecto:

- Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

Externalidades:

- Se contará con una herramienta disponible que permitirá a los especialistas en materia de medio ambiente modelar la dispersión de contaminantes en el aire.

Intangibles:

- En la valoración económica siempre hay elementos perceptibles por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.

A fin de medir con precisión los efectos, deberán considerarse tres situaciones:

✓ **SITUACIÓN SIN PROYECTO**

La información se entra en el fichero en las rutas (CO, SO, RE, ME, OU), que son: Opciones de Control, Opciones de Fuentes, Red de Receptores, Entrada Meteorológica y Opciones de Salida respectivamente. Estas rutas contienen las palabras claves que determinan la información que se dará como resultado en la salida.

Para la entrada de los datos al AERMOD sin el proyecto se debe seguir los siguientes pasos.

1. Introducción de las palabras claves de la Ruta CO
 - a) insertar las opciones de control.
2. Introducción de las palabras claves de la Ruta SO
 - a) Insertar fuentes.
 - b) Insertar grupo de fuentes.
 - c) Insertar factores de emisión.
3. Introducción de las palabras claves de la Ruta RE
 - a) Insertar rejilla de receptores
 - b) Insertar receptores cartesianos
4. Introducción de las palabras claves de la Ruta ME
 - a) Insertar datos meteorológicos
5. Introducción de las palabras claves de la Ruta OU



a) Insertar las opciones de salida

6. Interpretar los resultados.

Nota: Inserción de los datos se hace más trabajosa en dependencia de la cantidad de receptores y fuentes.

✓ **SITUACIÓN CON PROYECTO**

Para la entrada de los datos al AERMOD con el proyecto se debe seguir los siguientes pasos.

1. Introducción de las opciones de control

Pulsar el botón “opciones de control” y seleccionar las opciones de control.

2. Introducción de las fuentes

Pulsar el botón “opciones de fuentes” e insertar los datos de las fuentes.

3. Introducción de los receptores

Pulsar el botón “Opciones de Receptores” e insertar los datos de los receptores

4. Introducción de las opciones meteorológicas.

Pulsar el botón “Opciones Meteorológicas” e insertar los datos meteorológicos.

5. Introducción de las opciones de salida.

Pulsar el botón “Opciones de salida” y seleccionar las opciones de salida.

6. Graficar los resultados

Pulsar el botón “Graficar” y seleccionar la dirección en la que se desea graficar las concentraciones de dispersión del contaminante.

5.3 Beneficios y costos intangibles en el proyecto

COSTOS:

- ✓ Resistencia al cambio.

BENEFICIOS:

- ✓ Mayor comodidad para los usuarios.
- ✓ Mejora en la calidad de la información.
- ✓ Menor tiempo empleado en la introducción de los datos.
- ✓ Facilidad a la hora de interpretar los datos de concentración.



5.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizó el procedimiento para elaborar Una Ficha De Costo de un Producto Informático [Dra. Ana Ma. Gracia Pérez, UCLV].

Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costos en Moneda Librementemente Convertible:

✓ Costos Directos.

1. Compra de equipos de cómputo: No procede.
2. Alquiler de equipos de cómputo: No procede.
3. Compra de licencia de Software: No procede.
4. Depreciación de equipos: \$ 60.78.
5. Materiales directos: No procede.

Total: \$ 60.78.

✓ Costos Indirectos.

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Total: \$0.00.

✓ Gastos de distribución y venta.

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00.

Costos en Moneda Nacional:

✓ Costos Directos.

1. Salario del personal que laborará en el proyecto: \$100.00.
2. El 12% del total de gastos por salarios se dedica a la seguridad social: No procede.



3. El 0.09% de salario total, por concepto de vacaciones a acumular: No procede.
4. Gasto por consumo de energía eléctrica: \$ 5.94.
5. Gastos en llamadas telefónicas: No procede. 6.
6. Gastos administrativos: No procede.

✓ **Costos Indirectos.**

1. Know How: \$ 108,75.

Total: \$ 214.69.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, se tomará como costo el tiempo en horas empleado para introducir los datos al modelo de dispersión AERMOD y la variable sería la complejidad del problema para lo cual se tiene cinco valores.

Valores de la variable (Solución manual):

- 1 Modelo de hasta 20 fuentes y un receptor. (45 min.)
- 1 Modelo de hasta 40 fuentes y un receptor. (85 min.)
- 1 Modelo de hasta 60 fuentes y un receptor. (125 min.)
- 1 Modelo de hasta 80 fuentes y un receptor. (165 min.)
- 1 Modelo de hasta 100 fuentes y un receptor. (205 min.)

Valores de la variable (Solución con el software):

- 1 Modelo de hasta 20 fuentes y un receptor. (1 min.)
- 1 Modelo de hasta 40 fuentes y un receptor. (2 min.)
- 1 Modelo de hasta 60 fuentes y un receptor. (3 min.)
- 1 Modelo de hasta 80 fuentes y un receptor. (4 min.)



- 1 Modelo de hasta 100 fuentes y un receptor. (5 min.)

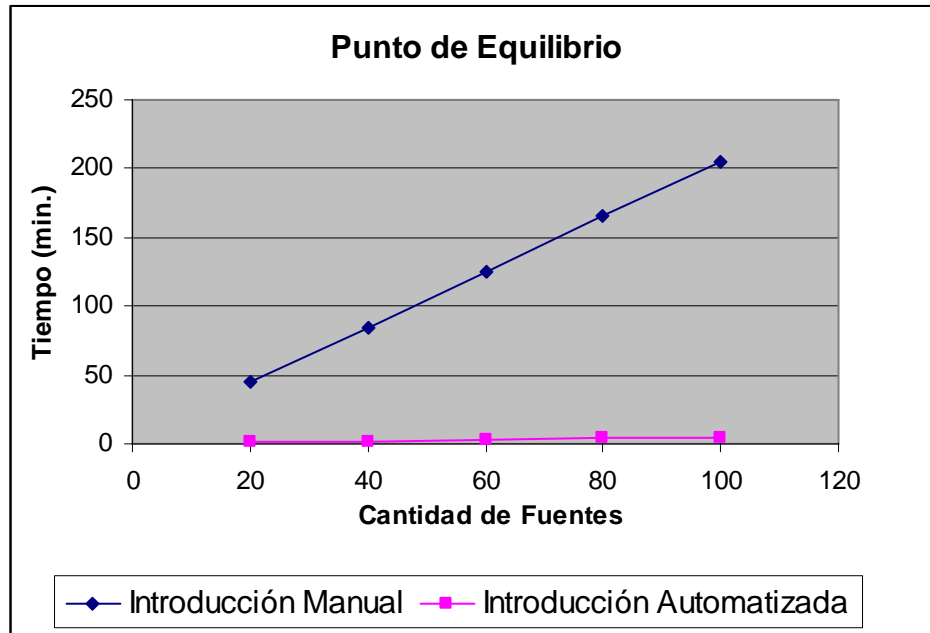


Ilustración 4: Punto de Equilibrio

Teniendo en cuenta los resultados reflejados en la gráfica en cuanto al Punto de Equilibrio queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de fuentes y receptores) y el tiempo que demora la introducción de los datos de forma manual y automatizada.

5.5 Conclusiones

Con la realización del estudio de factibilidad, mediante La Metodología Costo Efectividad (Beneficio), el análisis de los efectos económicos, beneficios y costos intangibles, se calculó el costo de ejecución del proyecto arrojando como resultado \$ 60.78 CUC. y \$ 214.69 MN demostrándose la conveniencia de la elaboración del sistema. También se llegó a la conclusión de que dicha aplicación resulta factible, ya que los costos son los mínimos y los beneficios considerables.



CONCLUSIONES GENERALES

El módulo de calculo AERMOD carece de una interfaz gráfica de usuario, esto hace difícil la introducción de los datos e interpretación de los resultados. Para solucionar estas dificultades se desarrolló una aplicación que facilita la interacción con el usuario.

Para el desarrollo del sistema se efectuaron las siguientes actividades:

- Se hizo un análisis de los sistemas similares existentes de los cuales se obtienen ejemplos de cómo podrían solucionarse algunas funcionalidades que se requieren para la aplicación.
- Se llevó a cabo un estudio de las principales metodologías, lenguajes y herramientas que se consideraron factibles para el desarrollo del sistema.
- Se realizó todo el proceso de desarrollo del software siguiendo las fases de la metodología XP, lo cual queda plasmado en el presente documento.

Como resultado de la investigación se logra el desarrollo de un software en el que se da cumplimiento a las especificidades de los objetivos propuestos. Con el valor fundamental de simplificar la demora que produce el procesamiento manual de la información, de mejorar la interacción del usuario con la interfaz, y además para contribuir a elevar la calidad del desarrollo del trabajo.



RECOMENDACIONES

- Aplicar el software en el departamento del Centro de Estudios del Medio Ambiente (CEMA) para obtener un mayor rendimiento del mismo.
- Extender las funcionalidades acorde a nuevos requisitos que surjan en el departamento del CEMA.
- Agregar nuevas funcionalidades.
- Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en nuevas versiones del software.



REFERENCIAS BIBLIOGRÁFICAS

- [1]. TURTÓS CARBONELL, L. et al. *Sistema de Modelos AERMOD para dispersión local de contaminantes atmosféricos*. 2007. [En línea]. <http://www.cubaenergia.cu> (Consultado: 2008).
- [2]. U.S. EPA, 2002. Compendium of Reports from the Peer Review Process for AERMOD. Febrero 2002, Disponibles en www.epa.gov/scram001
- [3]. Cimorelli, A. et al., 2005. AERMOD: A Dispersion Model for Industrial Source Applications. Part I: General Model Formulation and Boundary Layer Characterization. *Journal of Applied Meteorology*, 44(5): 682–693.
- [4]. Perry, S. et al., 2005. AERMOD: A Dispersion Model for Industrial Source Applications. Part II: Model Performance against 17 Field Study Databases. *Journal of Applied Meteorology*, 44(5): 694–708.
- [5]. Paine R. J. et al., 1998. Evaluation Results for AERMOD, Draft Report. Docket No. A-99-05; II-A-05. Disponible en: www.epa.gov/scram001/
- [6]. Turtós Carbonell, L. et al., “EXTERNALIDADES AMBIENTALES ATMOSFÉRICAS DE LA GENERACIÓN ELÉCTRICA”. 2004. [En línea] 2009. http://www.cubaenergia.cu/WWWExternalidades/00613055_2004_2.pdf
- [7]. Algunas palabras sobre el ambiente. *Análisis comunicacional*. [En línea] Mayo 30, 2005. <http://juanlpino.wordpress.com/2008/06/24/analisis-comunicacional-del-%E2%80%9Cno-a-las-papeleras%E2%80%9D/>.
- [8]. ESTRATEGIAS SUSTENTABLES PARA SU NEGOCIO. Anexo nº5: Modelación de dispersión de material particulado proyecto “Ampliación planta Santa Margarita”, Noviembre 2007. [En línea] 2008. <http://www.sga-ltda.cl>
- [9]. PUERTO PARALELO. *Descripción general*. Evolución, tipos, programación. [En línea] 2003. <http://www.globu.net/pp>
- [10]. Wikipedia. *Interfaz gráfica de usuario*. [En línea] julio 2009. http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario.
- [11]. Textualmente. *La evolución de la interfaz gráfica de usuario*. [En línea] julio 2009. [http://www.textualmenteactivo.com/La evolución de la interfaz gráfica de usuario — Textualmenteactivo.com.htm](http://www.textualmenteactivo.com/La%20evoluci%C3%B3n%20de%20la%20interfaz%20gr%C3%A1fica%20de%20usuario%20-%20Textualmenteactivo.com.htm)



- [12]. GNU Operating System. *La Definición de Software Libre*. [En línea] marzo de 2009.
<http://www.gnu.org/philosophy/free-sw.es.html>
- [13]. Gutierrez, Jorge A. Saavedra. El Mundo Informático. *Software Libre*. [En línea] 2007.
<http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>
- [14]. Mauricio, Ruben. *Proceso de Desarrollo de Software RUP*. [En línea] 2007.
<http://pvgdata.com/Mauricio/index.php/archives/46>
- [15]. John Deacon. Model-View-Controller (MVC) Architecture. [En línea] 2007
<http://www.jdl.co.uk/briefings/mvc.pdf>.
- [16]. Fuentes, Isidro. edib. *Tarjetas CRC*. [En línea] 28 de octubre de 2008.
http://informatica.escuelaedib.com/index.php?option=com_myblog&show=Tarjetas-CRC.html&Itemid=59
- [17]. J.J Gutierrez, M.J Escalona, M. Mejias, J.Torres. *Pruebas del sistema en programación extrema*. [En línea] 2008.
http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf



BIBLIOGRAFÍA

1. Addlink. *CALPUFF View (CALPUFF - CALMET - CALPOST)*. [En línea] 2009.
<http://www.productos.asp.html>
2. Beck, K.. “Extreme Programming Explained. Embrace Change”, Pearson Education, 1999. Traducido al español como: “Una explicación de la programación extrema. Aceptar el cambio”, Addison Wesley, 2000.
3. FERRER, J. Metodologías Ágiles. Páginaweb BECK, K. et. al. Manifiesto for Agile Software Development <http://agilemanifesto.org>, 2001.
4. JEFFRIES, R., Anderson, A., Hendrickson, C. “Extreme Programming Installed”. Addison-Wesley. 2001
5. JEFFRIES, R. *What is Extreme Programming?*. [En línea] 2002.
<http://www.xprogramming.com/xpmag/whatisxp.htm>
6. Wake, W.C. “Extreme Programming Explored”. Addison-Wesley. 2002.



GLOSARIO DE TÉRMINOS

IDE: *Integrated Development Environment* / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Patrón: Es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

Singleton: Patrón de diseño que representa una clase que posee una sola instancia.

GPL: *General Public License* / Licencia Pública General. Orientada principalmente a proteger la libre distribución, modificación y uso de software.

GUI: *Graphical User Interface* / Interfaz Gráfica de Usuario. Programa software que gestiona la interacción con el usuario de manera gráfica mediante el uso de íconos, menú, mouse, etc.

AERMOD: El código AERMOD (**A**merican **M**eteorology **S**ociety – **E.P.A.** – **R**egulatory **M**ODEl)

Software Libre: es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente, aunque conserve su carácter de libre, puede ser vendido comercialmente.

Linux: Sistema Operativo abierto y libre creado por el finlandés Linus Torvalds.



Informática: Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales. También es definida como el procesamiento de la información en forma automática.

Interfaz: Término informático que permite la circulación correcta y sencilla de información entre varias aplicaciones y entre el propio programa y el mono usuario.



ANEXOS

Anexo 1: Tarjetas CRC

CO	
Responsabilidades	Colaboraciones
Mostrar Opciones de Control	Control_opcion
Modificar Opciones de Control	Select_Pollutant_Names1
Insertar contaminante	
Modificar contaminante	File_class
Eliminar contaminante	File_IO
Seleccionar Contaminante	
Guardar las opciones de Control en el fichero .INP	
Salvar los cambios realizados en las opciones de Control en un fichero	

Tarjeta CRC de CO.

File_class	
Responsabilidades	Colaboraciones
Guardar en el fichero .INP	

Tarjeta CRC de File_class.



SO	
Responsabilidades	Colaboraciones
Insertar Fuente	File_class
Modificar Fuente	File_IO
Eliminar Fuente	Source_Pthway
Insertar Tipo de Fuente	Edit_group
Modificar Tipo de Fuente	Edit_Factor_sets
Salvar los cambios realizados en las Fuentes	Edit_Sources
Mostrar datos de la fuente	
Guardar Fuentes en el fichero .INP	
Crear Grupos de fuentes	
Modificar Grupos de Fuentes	
Eliminar Grupos de Fuentes	
Mostrar Grupos de Fuentes	
Salvar cambios realizados en los Grupos de Fuentes	
Guardar grupos de Fuentes en el fichero .INP	
Crear Factores de Emisión	
Modificar Factores de Emisión	
Eliminar Factores de Emisión	
Mostrar Factores de Emisión	
Salvar los cambios realizados en los factores de emisión	
Guardar Factores de emisión en el fichero .INP	

Tarjeta CRC de SO.

File_IO	
Responsabilidades	Colaboraciones
Guardar Salva del Proyecto	

Tarjeta CRC de File_IO.



RE	
Responsabilidades	Colaboraciones
Insertar Rejilla de Receptores polares	File_class
Modificar Rejilla de Receptores polares	File_IO
Eliminar Rejilla de Receptores polares	Receptor_Option
Insertar Rejilla de Receptores cartesianos	Elevation_Grid
Modificar Rejilla de Receptores cartesianos	Flagpole_heights
Eliminar Rejilla de Receptores cartesianos	Gen_Cart_grid
Insertar Rejilla de Receptores polares generados	Gen_pol_grid
Modificar Rejilla de Receptores polares generados	PolarGrid
Eliminar Rejilla de Receptores polares generados	Cartesian_Grid
Insertar Rejilla de Receptores cartesianos generados	
Modificar Rejilla de Receptores cartesianos generados	
Eliminar Rejilla de Receptores cartesianos generados	
Mostrar Rejilla de Receptores	
Salvar cambios realizados en las Rejillas de Receptores	
Guardar Rejilla de Receptores en el fichero .INP	
Insertar Receptores cartesianos discretos	
Modificar Receptores cartesianos discretos	
Eliminar Receptores cartesianos discretos	
Mostrar Receptores cartesianos discretos	
Salvar cambios realizados en los Receptores cartesianos discretos	
Guardar Receptores cartesianos discretos en el fichero .INP	

Tarjeta CRC de RE.



ME	
Responsabilidades	Colaboraciones
Mostrar opciones meteorológicas	File_class
Modificar opciones meteorológicas	File_IO
Guardar las opciones meteorológicas en el fichero .INP	Met_Option
Salvar los cambios realizados en las opciones meteorológicas en un fichero	

Tarjeta CRC de ME.

OU	
Responsabilidades	Colaboraciones
Mostrar opciones de Salida	File_class
Modificar opciones de Salida	File_IO
Guardar opciones de Salida en el fichero .INP	OutPut_Option
Salvar los cambios realizados en opciones de Salida en un fichero	

Tarjeta CRC de OU.

GR	
Responsabilidades	Colaboraciones
Crear gráfica	Source_Pthway OU Graficas CO RE SO

Tarjeta CRC de GR.



BarChart	
Responsabilidades	Colaboraciones
Mostrar Gráfico	

Tarjeta CRC de BarChart.

Anexo 2: Plantilla de Pruebas de Aceptación

Prueba de aceptación
HU: Nombre de la historia de usuario que va a comprobar su funcionamiento.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.
Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar.
Entrada/Pasos de ejecución: Pasos para probar la funcionalidad.
Resultado esperado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o Denegada.

Plantilla Prueba de Aceptación.

Anexo 3: Pruebas de Aceptación

Control

Prueba de aceptación
HU: Gestionar Opciones de Control.
Nombre: Seleccionar opciones de control.
Descripción: El especialista puede seleccionar las opciones de control.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta seleccionar las opciones de control.
Resultado esperado: Se selecciona la opción de control
Evaluación de la prueba: Satisfactoria.

PA Seleccionar opciones de control.



Prueba de aceptación
HU: Gestionar Opciones de Control.
Nombre: Modificar opción de control.
Descripción: El especialista puede modificar las opciones de control.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta modificar los datos de las opciones de control.
Resultado esperado: Se modifica la opción de control
Evaluación de la prueba: Satisfactoria.

PA Modificar opción de control.

Fuentes

Prueba de aceptación
HU: Gestionar Fuentes.
Nombre: Adicionar Fuente.
Descripción: El especialista puede adicionar una fuente.
Condiciones de ejecución: El identificador de la fuente que se desea adicionar no se ha insertado antes
Entrada/Pasos de ejecución: Se intenta adicionar una nueva fuente.
Resultado esperado: Se adiciona la fuente.
Evaluación de la prueba: Satisfactoria.

PA Adicionar Fuente.



Prueba de aceptación
HU: Gestionar Fuentes.
Nombre: Modificar Fuente.
Descripción: El especialista puede modificar los datos de una fuente.
Condiciones de ejecución: Si se modifica el identificador de la fuente tiene que ser distinto a los insertados antes
Entrada/Pasos de ejecución: Se modificar los datos de la fuente.
Resultado esperado: Se actualizan los datos de la fuente.
Evaluación de la prueba: Satisfactoria.

PA Modificar Fuente.

Prueba de aceptación
HU: Gestionar Fuentes.
Nombre: Eliminar Fuente.
Descripción: El especialista puede Eliminar una fuente.
Condiciones de ejecución: Debe existir al menos una fuente.
Entrada/Pasos de ejecución: Se intenta eliminar una fuente.
Resultado esperado: Se elimina de la lista de fuentes existentes.
Evaluación de la prueba: Satisfactoria.

PA Eliminar Fuente.



Prueba de aceptación
HU: Gestionar Fuentes.
Nombre: Mostrar Fuentes.
Descripción: El especialista puede ver los datos de una fuente.
Condiciones de ejecución: Debe existir al menos una fuente.
Entrada/Pasos de ejecución: Se selecciona la opción de mostrar los datos de las fuentes.
Resultado esperado: Se muestran los datos de la fuente.
Evaluación de la prueba: Satisfactoria.

PA Mostrar Fuentes.

Prueba de aceptación
HU: Gestionar grupo de Fuentes.
Nombre: Adicionar grupo de Fuentes.
Descripción: El especialista puede adicionar un grupo de fuentes.
Condiciones de ejecución: El identificador del grupo fuentes que se desea adicionar no se ha insertado antes
Entrada/Pasos de ejecución: Se intenta adicionar un nuevo grupo de fuentes
Resultado esperado: Se adiciona el grupo de fuentes.
Evaluación de la prueba: Satisfactoria.

PA Adicionar grupo de Fuentes.



Prueba de aceptación
HU: Gestionar grupo de Fuentes.
Nombre: Modificar grupo de Fuentes.
Descripción: El especialista puede Modificar los datos de un grupo de fuentes.
Condiciones de ejecución: Si se modifica el identificador del grupo fuentes debe ser distinto a los insertados antes
Entrada/Pasos de ejecución: Se intenta adicionar un nuevo grupo de fuentes
Resultado esperado: Se adiciona el grupo de fuentes.
Evaluación de la prueba: Satisfactoria.

PA Modificar grupo de Fuentes.

Prueba de aceptación
HU: Gestionar grupo de Fuentes.
Nombre: Eliminar grupo de Fuentes.
Descripción: El especialista puede eliminar un grupo de fuentes.
Condiciones de ejecución: Debe existir al menos un grupo de fuentes
Entrada/Pasos de ejecución: Se intenta eliminar un grupo de fuentes
Resultado esperado: Se elimina de la lista fe grupos de fuentes existentes.
Evaluación de la prueba: Satisfactoria.

PA Eliminar grupo de Fuentes.



Prueba de aceptación
HU: Gestionar grupo de Fuentes.
Nombre: Mostrar grupo de Fuentes.
Descripción: El especialista puede ver los datos de un grupo de fuentes.
Condiciones de ejecución: Debe existir al menos un grupo de fuentes
Entrada/Pasos de ejecución: Se selecciona la opción de mostrar los datos de un grupo de fuentes
Resultado esperado: Se muestran los datos.
Evaluación de la prueba: Satisfactoria.

PA Mostrar grupo de Fuentes.

Prueba de aceptación
HU: Gestionar Factores de Emisión.
Nombre: Adicionar factor de emisión.
Descripción: El especialista puede adicionar un factor de emisión.
Condiciones de ejecución: El identificador del factor de emisión que se desea adicionar no se ha insertado antes
Entrada/Pasos de ejecución: Se intenta adicionar un nuevo factor de emisión.
Resultado esperado: Se adiciona el factor de emisión.
Evaluación de la prueba: Satisfactoria.

PA Adicionar factor de emisión.



Prueba de aceptación
HU: Gestionar Factores de Emisión.
Nombre: Modificar factor de emisión.
Descripción: El especialista puede modificar un factor de emisión.
Condiciones de ejecución: Debe existir al menos un factor de emisión.
Entrada/Pasos de ejecución: Se intenta modificar los datos de un factor de emisión.
Resultado esperado: Se actualizan los datos del factor de emisión.
Evaluación de la prueba: Satisfactoria.

PA Modificar factor de emisión.

Prueba de aceptación
HU: Gestionar Factores de Emisión.
Nombre: Eliminar un factor de emisión.
Descripción: El especialista puede eliminar un factor de emisión.
Condiciones de ejecución: Debe existir al menos un factor de emisión.
Entrada/Pasos de ejecución: Se intenta eliminar un factor de emisión.
Resultado esperado: Se elimina de la lista de factores de emisión.
Evaluación de la prueba: Satisfactoria.

PA Eliminar un factor de emisión.



Prueba de aceptación
HU: Gestionar Factores de Emisión.
Nombre: Mostrar un factor de emisión.
Descripción: El especialista puede ver los datos de un factor de emisión.
Condiciones de ejecución: Debe existir al menos un factor de emisión.
Entrada/Pasos de ejecución: Se selecciona la opción de mostrar los datos.
Resultado esperado: Se muestran los datos.
Evaluación de la prueba: Satisfactoria.

PA Mostrar un factor de emisión.

Receptores

Prueba de aceptación
HU: Gestionar rejilla.
Nombre: Crear rejilla de receptores.
Descripción: El especialista puede crear rejillas de receptores.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Crea rejilla de receptores.
Resultado esperado: Se crea la rejilla de receptores
Evaluación de la prueba: Satisfactoria.

PA Crear rejilla de receptores.



Prueba de aceptación
HU: Gestionar rejilla.
Nombre: Crear rejilla de receptores.
Descripción: El especialista puede crear rejillas de receptores.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Crea rejilla de receptores.
Resultado esperado: Se crea la rejilla de receptores
Evaluación de la prueba: Satisfactoria.

PA Crear rejilla de receptores.

Prueba de aceptación
HU: Gestionar rejilla.
Nombre: Modificar rejilla de receptores.
Descripción: El especialista puede modificar los datos de la rejilla de receptores.
Condiciones de ejecución: Debe existir al menos una rejilla de receptores.
Entrada/Pasos de ejecución: Se intenta modificar los datos de una rejilla de receptores.
Resultado esperado: Se actualizan los datos de la rejilla de receptores.
Evaluación de la prueba: Satisfactoria.

PA Modificar rejilla de receptores.



Prueba de aceptación
HU: Gestionar rejilla.
Nombre: Eliminar rejilla de receptores.
Descripción: El especialista puede eliminar una rejilla de receptores.
Condiciones de ejecución: Debe existir al menos una rejilla de receptores.
Entrada/Pasos de ejecución: Se intenta eliminar una rejilla de receptores.
Resultado esperado: Se elimina la rejilla de receptores.
Evaluación de la prueba: Satisfactoria.

PA Eliminar rejilla de receptores.

Prueba de aceptación
HU: Gestionar Receptores Cartesianos Discretos.
Nombre: Adicionar receptor
Descripción: El especialista puede adicionar un receptor.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta adicionar un nuevo receptor.
Resultado esperado: Se adiciona un receptor.
Evaluación de la prueba: Satisfactoria.

PA Adicionar receptor.



Prueba de aceptación
HU: Gestionar Receptores Cartesianos Discretos.
Nombre: Modificar receptor
Descripción: El especialista puede modificar los datos de un receptor.
Condiciones de ejecución: Debe existir al menos un receptor
Entrada/Pasos de ejecución: Se intenta modificar los datos de un receptor.
Resultado esperado: Se actualizan los datos del receptor.
Evaluación de la prueba: Satisfactoria.

PA Modificar receptor.

Prueba de aceptación
HU: Gestionar Receptores Cartesianos Discretos.
Nombre: Eliminar receptor
Descripción: El especialista puede eliminar un receptor.
Condiciones de ejecución: Debe existir al menos un receptor
Entrada/Pasos de ejecución: Se intenta eliminar un receptor.
Resultado esperado: Se elimina el receptor.
Evaluación de la prueba: Satisfactoria.

PA Eliminar receptor.



Meteorología

Prueba de aceptación
HU: Gestionar la entrada de datos meteorológicos.
Nombre: Seleccionar datos meteorológicos.
Descripción: El especialista puede seleccionar datos meteorológicos.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta seleccionar los datos meteorológicos.
Resultado esperado: Se seleccionan los datos meteorológicos.
Evaluación de la prueba: Satisfactoria.

PA Seleccionar datos meteorológicos.

Prueba de aceptación
HU: Gestionar la entrada de datos meteorológicos.
Nombre: Modificar datos meteorológicos.
Descripción: El especialista puede modificar datos meteorológicos.
Condiciones de ejecución: Deben de existir datos meteorológicos seleccionados.
Entrada/Pasos de ejecución: Se intenta modificar los datos meteorológicos.
Resultado esperado: Se modifican los datos meteorológicos.
Evaluación de la prueba: Satisfactoria.

PA Modificar datos meteorológicos.



Salida

Prueba de aceptación
HU: Gestionar las opciones de salida.
Nombre: Seleccionar opciones de salida.
Descripción: El especialista puede seleccionar las opciones de salida.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta seleccionar las opciones de salida.
Resultado esperado: Se seleccionan las opciones de salida.
Evaluación de la prueba: Satisfactoria.

PA Seleccionar opciones de salida.

Prueba de aceptación
HU: Gestionar las opciones de salida.
Nombre: Modificar opciones de salida.
Descripción: El especialista puede modificar las opciones de salida.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta modificar las opciones de salida.
Resultado esperado: Se modifican las opciones de salida.
Evaluación de la prueba: Satisfactoria.

PA Modificar opciones de salida.



Archivo

Prueba de aceptación
HU: Abrir y Guardar trabajos realizados.
Nombre: Guardar trabajo.
Descripción: El especialista puede salvar el trabajo realizado.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta salvar el trabajo realizado.
Resultado esperado: Se guarda el trabajo realizado.
Evaluación de la prueba: Satisfactoria.

PA Guardar trabajo.

Prueba de aceptación
HU: Abrir y Guardar trabajos realizados.
Nombre: Abrir trabajo.
Descripción: El especialista puede abrir un trabajo guardado.
Condiciones de ejecución: Debe existir algún trabajo guardado
Entrada/Pasos de ejecución: Se intenta abrir un trabajo guardado.
Resultado esperado: Se abre el trabajo guardado y se muestran los datos.
Evaluación de la prueba: Satisfactoria.

PA Abrir trabajo.



Prueba de aceptación
HU: Crear un nuevo trabajo.
Nombre: Crear nuevo trabajo.
Descripción: El especialista puede crear un trabajo nuevo.
Condiciones de ejecución:
Entrada/Pasos de ejecución: Se intenta crear un trabajo nuevo.
Resultado esperado: Se crea un trabajo nuevo.
Evaluación de la prueba: Satisfactoria.

PA Crear nuevo trabajo.

Compilar aermod

Prueba de aceptación
HU: Procesar con el AERMOD el archivo de entrada
Nombre: Crear archivo de entrada .inp.
Descripción: El especialista puede crear el archivo de entrada .inp.
Condiciones de ejecución: Debe existir la información para llenar el archivo .inp
Entrada/Pasos de ejecución: Se intenta crear el archivo .inp.
Resultado esperado: Se crea el archivo .inp.
Evaluación de la prueba: Satisfactoria.

PA Crear archivo de entrada .inp.



Prueba de aceptación
HU: Procesar con el AERMOD el archivo de entrada
Nombre: Pasar al aermod el archivo de entrada .inp.
Descripción: El especialista le puede pasar al aermod el archivo de entrada .inp.
Condiciones de ejecución: Debe existir un archivo .inp
Entrada/Pasos de ejecución: Se le intenta pasar el archivo .inp al aermod.
Resultado esperado: Se le pasa el archivo .inp al aermod y este genera el archivo .out.
Evaluación de la prueba: Satisfactoria.

PA Pasar al aermod el archivo de entrada .inp.

Gráficos

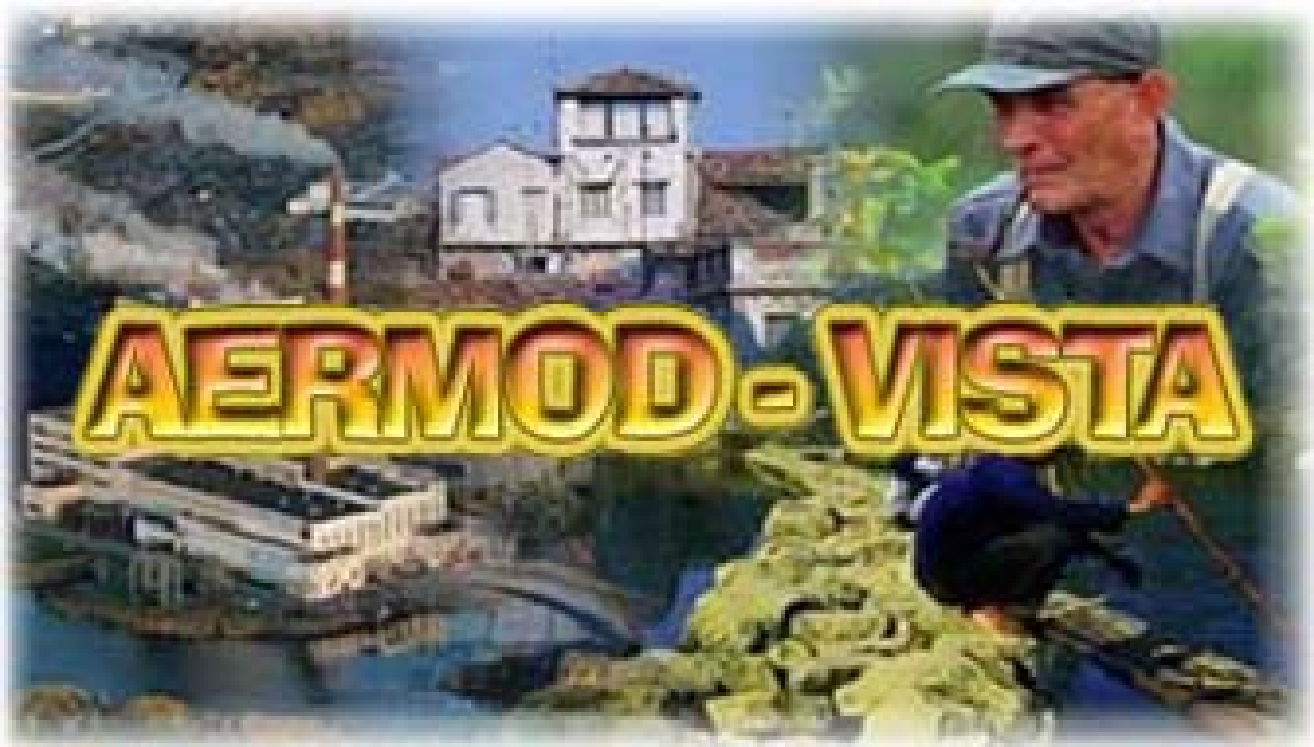
Prueba de aceptación
HU: Generar Gráfico
Nombre: Leer fichero .out y sacar los datos
Descripción: El sistema puede leer el fichero .out y sacar los datos
Condiciones de ejecución: Debe existir un archivo .out
Entrada/Pasos de ejecución: Se intenta leer el fichero .out y extraer los datos
Resultado esperado: Se lee archivo .out y se extraen los datos
Evaluación de la prueba: Satisfactoria.

PA Leer fichero .out y sacar los datos.



Prueba de aceptación
HU: Generar Gráfico
Nombre: Crear gráfico
Descripción: El sistema puede crear el gráfico.
Condiciones de ejecución: Deben existir los datos necesarios para graficar.
Entrada/Pasos de ejecución: Se intenta graficar.
Resultado esperado: Se grafican los datos.
Evaluación de la prueba: Satisfactoria.

PA Crear gráfico.



MANUAL DE USUARIO

VERSIÓN 1.0

5 de Julio 2009

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
1.1 Propósito	1
1.2 Alcance.....	1
1.3 Referencias	1
2 DESCRPCIÓN BREVE	1
3 Barra de Herramientas de AERMOD-VISTA	3
4 Módulos	4
4.1 Módulo 01: Control	5
4.1.1 Descripción.....	5
4.1.2 Funcionalidades	5
4.2 Módulo 02: Fuentes.....	10
4.2.1 Descripción.....	10
4.2.2 Funcionalidades	11
4.3 Módulo 03: Receptores	15
4.3.1 Descripción.....	15
4.3.2 Funcionalidades	15
4.4 Módulo 04: Meteorología.....	18
4.4.1 Descripción.....	18
4.4.2 Funcionalidades	18
4.5 Módulo 05: Salida.....	19
4.5.1 Descripción.....	19
4.5.2 Funcionalidades	19
4.6 Módulo 06: Archivo.....	20
4.6.1 Descripción.....	20
4.6.2 Funcionalidades	20
4.7 Módulo 07: Correr Aermod	21
4.7.1 Descripción.....	21
4.7.2 Funcionalidades	21
4.8 Módulo 08: Gráficos	21
4.8.1 Descripción.....	21
4.8.2 Funcionalidades	22



INTRODUCCIÓN

AERMOD-VISTA es una interfaz diseñada para el módulo de cálculo de dispersión AERMOD para analizar de forma intuitiva, robusta y estable la concentración y deposición de la contaminación atmosférica originada por diversas fuentes.

1.1 Propósito

El Manual de Usuario de **AERMOD-VISTA** persigue esclarecer algunos detalles básicos para el uso efectivo por parte de los Clientes (Especialistas en materia de medio ambiente) del Software de referencia, dada la necesidad de su aplicación funcional.

1.2 Alcance

Su funcionalidad servirá para ser aplicado en el Centro de estudio del Medio Ambiente (CEMA).

1.3 Referencias

Este documento hace referencia a:

NOMBRE DOCUMENTO	FUENTE	RESPONSABLE
AERMOD-VISTA Manual de usuario		

2 DESCRPCIÓN BREVE

AERMOD-VISTA permite aplicar mejor el modelo de dispersión de contaminantes AERMOD para modelar los contaminantes dispersos en el aire, provee una interfaz gráfica de usuario con suficiente capacidad expresiva, amigable y capaz de facilitar la entrada de los datos, disminuyendo el tiempo empleado en este proceso y las posibilidades de cometer errores, también hace más fácil la interpretación de los resultados con gráficos. Debido a su desarrollo con herramientas libres da la facilidad de prescindir de pagos y licencias.

Principales funcionalidades

- **Permitir que el especialista pueda:**
 - **Gestionar Opciones de Control**
 - Seleccionar opciones de control
 - Modificar opciones de control
 - **Gestionar Fuentes**
 - Insertar fuente
 - Ver y modificar datos de la fuente
 - Eliminar fuente
 - **Gestionar grupo de Fuentes**
 - Crear grupo de Fuentes
 - Ver y modificar datos de los grupos de Fuentes
 - Eliminar grupo de Fuentes
 - **Gestionar Factores de Emisión**
 - Insertar factor de Emisión
 - Ver y modificar datos de los Factores de Emisión.
 - Eliminar Factor de Emisión.
 - **Gestionar rejilla**
 - Insertar de rejilla de Receptores
 - Ver y modificar los datos de rejilla de Receptores.
 - Eliminar de rejilla de Receptores.
 - **Gestionar Receptores Cartesianos Discretos**
 - Llenar los datos de los Receptores Cartesianos Discretos.
 - Ver y modificar los datos de los Receptores Cartesianos Discretos.
 - Eliminar Receptor Cartesiano Discreto.
 - **Gestionar la entrada de datos meteorológicos**
 - Insertar datos meteorológicos.
 - Modificar datos meteorológicos.

AERMOD-VISTA

- **Gestionar las opciones de salida**
 - Seleccionar opciones de salida.
 - Modificar datos de las opciones de salida.
- **Generar Gráfico**
 - Crear Gráfico
 - Mostrar Gráfico
- **Abrir y Guardar trabajos realizados**
 - Abrir trabajo
 - Guardar trabajo realizado
- **Crear un nuevo trabajo**
 - Crear trabajo nuevo
- **Procesar con el AERMOD el archivo de entrada**
 - Procesar el archivo aermod.INP

3 Barra de Herramientas de AERMOD-VISTA



Nueva sesión de AERMOD-VISTA, restaura todos los valores por defecto, usar con precaución, puede borrar cualquier información actual en el AERMOD-VISTA.



Abre una sesión en AERMOD-VISTA, abre un fichero existente, este es el fichero “maestro” que contiene todos tus datos del modelo y opciones.



Salva una sesión de AERMOD-VISTA, salva toda la información actual al archivo “maestro”, este botón guarda todos los datos entrados previamente, incluyendo datos que no son guardados en el archivo de entrada del modelo.



Abre un formulario para editar las opciones de control.

AERMOD-VISTA



Abre un formulario para editar las opciones de fuente.



Abre un formulario para editar las opciones de receptor.



Abre un formulario para editar las opciones de datos meteorológicos.



Abre un formulario para editar las opciones de salida.



Ver archivo de entrada (.INP).



Construye el archivo (.INP) y corre el modelo AERMOD, entonces si el archivo fue creado sin errores, puede opcionalmente leer y correr el modelo AERMOD.



Grafica las concentraciones que están en el fichero (.OUT) generado por el aermod después que se halla creado correctamente el archivo (.INP) y corrido el modelo AERMOD satisfactoriamente.



Contiene la ayuda.


Nota: Estas acciones también se pueden realizar mediante el menú.

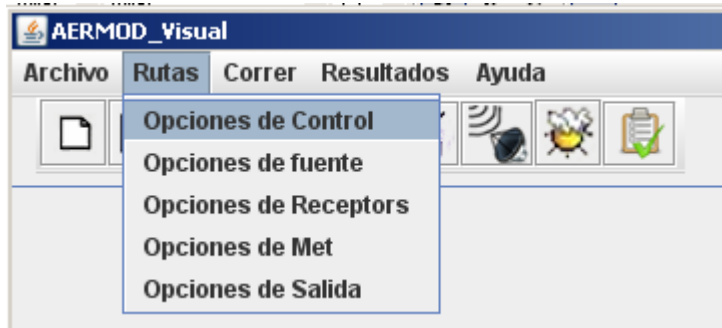
4 Módulos

- ▣ **Control**
- ▣ **Fuentes**
- ▣ **Receptores**
- ▣ **Meteorología**
- ▣ **Salida**
- ▣ **Archivo**
- ▣ **Correr AERMOD**
- ▣ **Gráficos**


4.1 Módulo 01: Control

4.1.1 Descripción

Es donde se editan las opciones de control. Puede ser pulsando el botón  o seleccionando en el menú “Opciones de Control”.



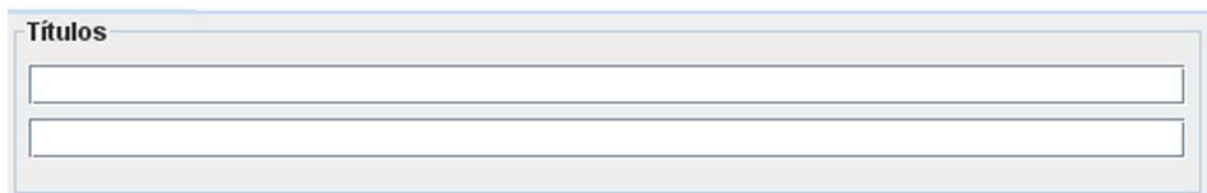
4.1.2 Funcionalidades

Se visualiza un formulario con dos pestañas “Control 1” y “Control 2”  donde se podrán seleccionar las opciones de control e insertar otros valores.

▫ Pestaña “Control 1”

▪ Títulos

Opcionalmente puede insertar dos títulos, pero es obligatorio al menos uno.

A screenshot of a form titled 'Títulos'. It contains two empty text input fields stacked vertically, intended for entering titles.

▫ Opciones predeterminadas

- Opciones predeterminadas
- Opciones no predeterminadas

Al seleccionar las opciones predeterminadas se desactivan las opciones siguientes:

Opciones no predeterminadas

<input type="checkbox"/> No downwash de punta de Pila	
<input type="checkbox"/> Omitir comprob. de fecha para ficheros de datos met non-sec.	
<input type="checkbox"/> SCREEN	
<input type="checkbox"/> TÓXICO	
<input type="checkbox"/> Método de relacion volumen-molar de la pluma (PVMRM) for NO2	
<input type="checkbox"/> Método de límite del Ozono (OLM)	<input type="checkbox"/> Concentración
<input type="checkbox"/> Opción de prueba BETA	<input type="checkbox"/> Dispersión Total
<input type="checkbox"/> Ninguna advertencia de salida	<input type="checkbox"/> Deposición seca
<input type="checkbox"/> Considerar terreno llano (Flat)	<input type="checkbox"/> Deposición húmeda

▣ Opciones no predeterminadas

Opciones predeterminadas
 Opciones no predeterminadas

Al seleccionar las opciones no predeterminadas se activan las opciones:

Opciones no predeterminadas

<input type="checkbox"/> No downwash de punta de Pila	
<input type="checkbox"/> Omitir comprob. de fecha para ficheros de datos met non-sec.	
<input checked="" type="checkbox"/> SCREEN	
<input type="checkbox"/> TÓXICO	
<input type="checkbox"/> Método de relacion volumen-molar de la pluma (PVMRM) for NO2	
<input type="checkbox"/> Método de límite del Ozono (OLM)	<input type="checkbox"/> Concentración
<input type="checkbox"/> Opción de prueba BETA	<input type="checkbox"/> Dispersión Total
<input type="checkbox"/> Ninguna advertencia de salida	<input type="checkbox"/> Deposición seca
<input type="checkbox"/> Considerar terreno llano (Flat)	<input type="checkbox"/> Deposición húmeda

Nota: Las opciones “método de relación volumen de la pluma (PVMRM)” y “Método de límite del Ozono (OLM)” es solo para el contaminante NO2, es decir solo se activa cuando está seleccionado ese contaminante y se puede seleccionar una de ellas a la vez.

▣ Selección del contaminante

Contaminante

NO2 ▼

Agregar

AERMOD-VISTA

Cuando se selecciona el contaminante NO₂, el sistema permite seleccionar las opciones “método de relación volumen de la pluma (PVMRM)” y “Método de límite del Ozono (OLM)” que a su vez necesitan la entrada de otros valores como son:

“Pila NO₂”, “Valor de Ozono” y “archivo de Ozono” que solo se activan con la selección de una de las opciones y si se selecciona una de las opciones es obligatorio llenar los valores.

Nota: las opciones “Pila NO₂”, “Valor de Ozono” y “archivo de Ozono” se encuentran en la pestaña Control 2.

También se pueden agregar o quitar contaminantes del listado de selección (hasta 12), pero esto debe manejarse con precaución, deben de ser los contaminantes aprobados por la EPA¹ para el modelo de dispersión AERMOD, dentro de ellos esta el CO₂, SO₂, No₂ y otros.

Agregar Contaminante

Seleccionar nombre de contaminantes

Nombres de Contaminantes

Contaminant.1	<input type="text" value="OTHER"/>
Contaminant.2	<input type="text" value="NO2"/>
Contaminant.3	<input type="text"/>
Contaminant.4	<input type="text"/>
Contaminant.5	<input type="text"/>
Contaminant.6	<input type="text"/>
Contaminant.7	<input type="text"/>
Contaminant.8	<input type="text"/>
Contaminant.9	<input type="text"/>
Contaminant.10	<input type="text"/>
Contaminant.11	<input type="text"/>
Contaminant.12	<input type="text"/>

OK

Cancel

¹ Environmental Protection Agency.

- **Pestaña “Control 2”**

- **Periodos de promediado**

Se selecciona los periodos de promediado, como máximo 4.

Periodos de promediado

1-hora ^ 4-horas^ 12-horas^ Period
 2-horas^ 6-horas^ 24-horas^ Anual
 3-horas^ 8-horas^ Mensualmente^

*Máx. de 4 periodos, Actualmente: 1

Nota: Si se selecciona en opciones no predeterminadas la opción “SCREEN”, se selecciona por defecto solo la opción para “1-Hora”.

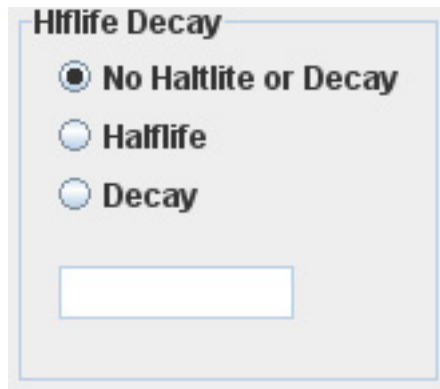
- **Banderilla de receptores**

Receptores Banderilla

Sin altura de banderilla
 Alturas individuales
 Banderilla Ht.=

Si se selecciona la opción “Banderilla Ht.=”, se debe introducir la altura para dicha banderilla que es un valor flota.

▫ Período/Decaimiento



Half-life Decay

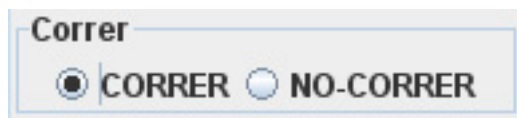
No Half-life or Decay

Half-life

Decay

Si se selecciona “Half-life” se activa un editor para introducir el half-life en (sec.), si se selecciona “Decay” se activa un editor para introducir el decaimiento en (1/sec).

▫ Correr o no correr



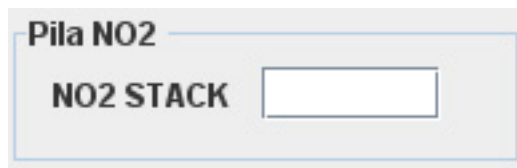
Correr

CORRER NO-CORRER

La opción **CORRER O NO CORRER** ha sido incluida en la las opciones de control para permitirle al usuario especificar, **correr** el modelo y realizar todos los cálculos, o **no correr** y sólo procesar los datos de entrada.

▫ Pila NO2

Esta opción permite entrar el valor (float) de NO2 de la pila.



Pila NO2

NO2 STACK

▫ Valor de Ozono

Se introduce el valor de ozono y se selecciona los parámetros de unidades.

AERMOD-VISTA

Valor de Ozono

ValorOzono:

Parámetros de unidades O3

PPM PPB UG/M3

▣ Archivo Ozono

Aquí se introduce el archivo que contiene los datos de ozono y el formato bajo el cual se creó así como los parámetros de unidades.

Archivo OZONO

ArchOZONE


Parámetros de unidades O3

PPM PPB UG/M3

Formato:

4.2 Módulo 02: Fuentes

4.2.1 Descripción

Es donde se editan las opciones de fuente. Puede ser pulsando el botón  o seleccionando en el menú la “Opciones de Fuentes”.



4.2.2 Funcionalidades

Se visualiza un formulario con tres pestañas “Fuentes”, “Grupo de Fuentes” y “Factores de Emisión” donde se podrán insertar los datos de fuentes, grupo de fuentes y factores de emisión.

▫ Pestaña “Fuentes”

▪ Insertar y Modificar Fuentes

Aquí se muestran los identificadores de las fuentes insertadas, y se pueden definir y editar fuentes mediante el botón “Definir Fuentes” y borrar mediante el botón “Borra Fuente”.

	Fuentes
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

Nota: Al borrar una fuente, también se borran los datos relacionados con ella.

Este formulario se muestra al pulsar el botón “Definir Fuente”, y es aquí donde se editan, insertan y pueden borrar los datos de las fuentes. Si presiona el botón “Aceptar” se guardarán permanentemente los datos modificados o insertados, lo contrario pasa si se pulsa el botón “Cancelar” con este se descartan los cambios realizados.

Edición de Fuentes

Tipo de Fuente

Tipo: POINT ID Fuente:

Parámetros de Localización de la Fuente

Coordenada X o Este (XS): (m)

Coordenada Y o Norte (YS): (m)

Altura de la Fuente desde la bas... (m)

Altura de la Fuente (HS): (m)

Parámetros de liberación de la fuente

Tasa de Emisión (Q1): (g/s)

Temperatura de la Fuente (TS): °K



Diámetro exterior (DS): (m)

Velocidad de Salida (VS): (m/s)

Tasa de Flujo de Salida (FS): (acm/s)

Borrar < 1 of 0 > Nuevo Primero Ultimo

Aceptar Cancelar

Para ver o modificar los datos de alguna fuente insertada se pueden usar los botones  , el primero para ver las fuentes anteriores y el segundo para ver las siguientes, también se puede modificar el tipo de fuente que por defecto es “POINT” y desplazarse hasta la ultima o primera fuente mediante los botones “Ultimo” y “Primero”. Al cambiar el tipo de fuente se emite un mensaje de advertencia para evitar la pérdida de datos por alguna equivocación.

Para insertar una nueva fuente se pulsa en el botón “Nuevo”, es prepara las entradas para los datos que se agregarán.

▣ Pestaña “Grupo de Fuentes”

▪ Insertar y modificar Grupos de Fuentes

Aquí se muestran los identificadores de los grupos de fuentes insertadas, y se pueden definir y editar grupos de fuentes mediante el botón “Definir grupo Fuentes” y borrar mediante el botón “Borra grupo de Fuente”.

Factores” se puede borrar un factor a la vez, seleccionándolo y pulsando el botón.

Agregar Factores de Emisión

	NOMBRES	
1	sa	▲
2		
3		
4		
5		
6		
7		
8		
9		
0		▼

En el siguiente panel se selecciona la tasa de emisión para un factor seleccionado en el panel anterior.

Tasa de Emisión

Ninguno Estación Mes

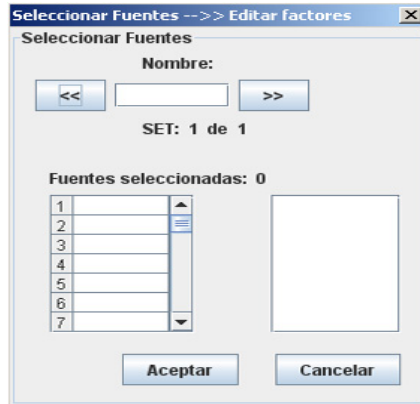
HoraDelDía STAR EstaciónHora

En este panel se insertan los valores de emisión para el factor seleccionado.

Tasa de Factores de Emisión Estacional


Invier.	1
Primav.	2
Verano	43
Otoño	4

La siguiente ventana es la que se utiliza para insertar o modificar las fuentes por factor y se pueden agregar más factores.



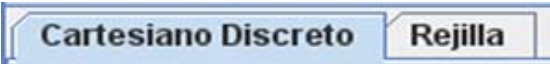
4.3 Módulo 03: Receptores

4.3.1 Descripción

Es donde se editan las opciones de Receptor. Puede ser pulsando el botón  o seleccionando en el menú la “Opciones de Receptor”.



4.3.2 Funcionalidades

Se visualiza un formulario con dos pestañas “Cartesiano Discreto” y “Rejilla”  donde se podrán insertar los datos de los receptores.

- **Pestaña “Cartesiano Discreto”**

- **Insertar, Modificar y Eliminar datos de Receptores Cartesianos Discretos**

- En el panel siguiente se insertan los datos de los receptores cartesianos discretos, dando las coordenadas norte y este en metros, la inserción de los datos de elevación y banderilla dependen de la selección en las opciones de control de terreno elevado y banderilla respectivamente.

Para eliminar, se selecciona una fila de datos equivalente a un receptor se se pulsa el botón “Borrar Fila”.

	Este (X) (m)	Norte (Y) (m)	Elevación	Banderilla
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

▣ Pestaña “Rejilla”

El siguiente panel permite insertar modificar o eliminar rejillas de receptores (hasta 5) del tipo (Polar, Cartesiano, Cartesiano Generado y Polar Generado). Los datos de origen pueden ser las coordenadas x, y o el id. de la fuente y como nombre “id de la red”.

A cada rejilla le corresponden datos de elevación y/o altura de banderilla, dependiendo de la selección de las opciones “banderilla” y terreno elevado.

Receptores	Tipo de Rejilla	Id de Red	X(Este)'	Y(Norte)'	O Id de Fuente
<input checked="" type="radio"/> Rejilla1 0	None				
<input type="radio"/> Rejilla2 0	None				
<input type="radio"/> Rejilla3 0	None				
<input type="radio"/> Rejilla4 0	None				
<input type="radio"/> Rejilla5 0	None				

Al pulsar el botón “Editar Rejilla” se muestra la siguiente ventana:

AERMOD-VISTA

ss Elevation in meters

ELEVATIONS

A	1	2	3	4
5				
6				
7				
87				

HILL

A	1	2	3	4
5				
6				
7				
87				

ok Cancel

En esta ventana se introducen los datos de elevación y colina para la rejilla seleccionada. La cantidad de valores a introducir para elevación y colina dependen del tipo de rejilla, de la cantidad de direcciones radiales y la distancia entre anillos.

En la ventana siguiente se introducen los datos de banderilla para la rejilla seleccionada.


v Elevación en metros

A	1	2	3
5			
6			

Aceptar Cancelar

4.4 Módulo 04: Meteorología

4.4.1 Descripción

Es donde se editan las opciones de datos meteorológicos. Puede ser pulsando el botón  o seleccionando en el menú “Opciones de Met”.



4.4.2 Funcionalidades

Se visualiza un formulario donde se podrán insertar las opciones de datos meteorológicos.




	Sta. No.	Año	Ubicación
Superficie:	99999	1990	UNK
Aire Superior:	99999	1990	UNK

Nota: Los datos meteorológicos se toman de los ficheros .SFC que son datos meteorológicos de superficie y .PFL datos de perfil. Hay que especificar de “Superficie” el año número de la estación y ubicación y lo mismo para el “Aire Superior”. Para el funcionamiento del sistema es imprescindible que se copien estos archivos en la misma ubicación del ejecutable del sistema.

4.5 Módulo 05: Salida

4.5.1 Descripción

Es donde se editan las opciones de salida. Puede ser pulsando el botón  o seleccionando en el menú "Opciones de Salida".



4.5.2 Funcionalidades

Se visualiza un formulario donde se podrán insertar las opciones de salida.

Nota: Los periodos de promedio dependen de las opciones seleccionadas en "Las Opciones de Control".

Opciones de Salida

Tablas

Valores más altos para Cada Receptor (Tabla de salida Principal)

Escriba las Siguietes Tablas para Indicar los Periodos de Promedio

1-Hora	<input type="checkbox"/> Superior	<input type="checkbox"/> 2do	<input type="checkbox"/> 3ro	<input type="checkbox"/> 4to	<input type="checkbox"/> 5to	<input type="checkbox"/> 6to
None	<input type="checkbox"/> Superior	<input type="checkbox"/> 2do	<input type="checkbox"/> 3ro	<input type="checkbox"/> 4to	<input type="checkbox"/> 5to	<input type="checkbox"/> 6to
None	<input type="checkbox"/> Superior	<input type="checkbox"/> 2do	<input type="checkbox"/> 3ro	<input type="checkbox"/> 4to	<input type="checkbox"/> 5to	<input type="checkbox"/> 6to
None	<input type="checkbox"/> Superior	<input type="checkbox"/> 2do	<input type="checkbox"/> 3ro	<input type="checkbox"/> 4to	<input type="checkbox"/> 5to	<input type="checkbox"/> 6to

Valores máximos En conjunto

Valores Máximos	1-Hora	<input type="text"/>
Para periodo	None	<input type="text"/>
de promedio	None	<input type="text"/>
indicado.	None	<input type="text"/>


Tablas Diarias


Seleccionar Periodos para Tablas Diarias


<input type="checkbox"/> 1-Hora
<input type="checkbox"/> None
<input type="checkbox"/> None
<input type="checkbox"/> None

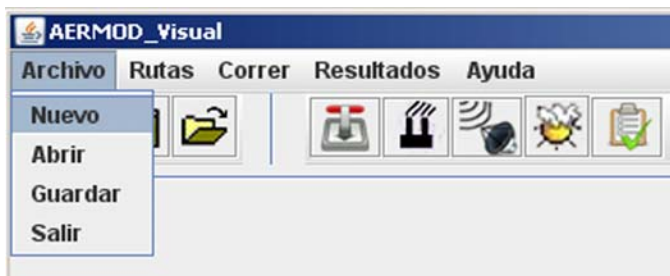
4.6 Módulo 06: Archivo

4.6.1 Descripción

En este módulo se pueden crear, guardar o abrir una sesión de AERMOD-VISTA. Para abrir se pulsa el botón  o se selecciona la opción “Abrir” en el menú “Archivo”

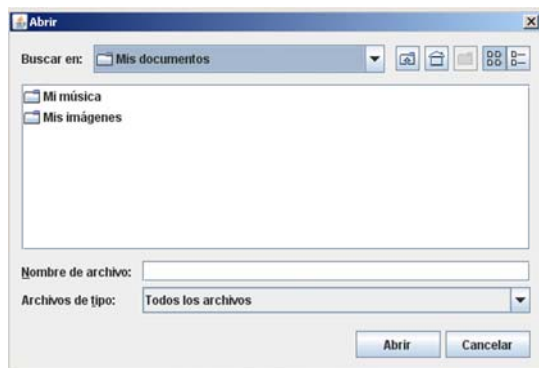
Para guardar se pulsa el botón  o se selecciona la opción “Guardar” en el menú “Archivo”

Para crear una sesión nueva de AERMOD-VISTA se pulsa el botón  o se selecciona la opción “Nuevo” en el menú “Archivo”.



4.6.2 Funcionalidades

Al pulsar el botón “abrir” se muestra una ventana que permite localizar el archivo que contiene la salva de alguna sesión.




Al pulsar el botón “guardar” se muestra una ventana que permite guardar el archivo en la ubicación deseada.

Al pulsar el botón “Nuevo” se muestra un mensaje de advertencia, se crea una nueva sesión descartando cualquier información insertada anteriormente.

4.7 Módulo 07: Correr Aermod

4.7.1 Descripción

Al pulsar el botón  o seleccionar la opción “AERMOD” en el menú “Correr” se ejecuta el modelo de dispersión AERMOD con los parámetros (archivos .INP, .SFC, .PFL) que le pasa el AERMOD-VISTA.




4.7.2 Funcionalidades

Este módulo ejecuta el modelo de dispersión AERMOD y le pasa como parámetros el archivo (.INP) creado por la aplicación AERMOD-VISTA y los archivos (.SFC), (.PFL) y opcionalmente el (Ozone.dat)

Nota: Los archivos (.SFC) y (.PFL) deben copiarse en la misma localización donde se encuentra el ejecutable de la aplicación AERMOD-VISTA.

4.8 Módulo 08: Gráficos

4.8.1 Descripción

Al pulsar el botón  se muestra una ventana que muestra las opciones para graficar las concentraciones en las direcciones disponibles.

4.8.2 Funcionalidades

Periodos	Grupos de F.	ID de la red de recep.	Tipo de red de recept...	Dirección	Rectable
1	ALL	CAR1	Cartesian	500	1ST

En este módulo se muestra una ventana que contiene las opciones, “Períodos”, “Grupos de F.”, “Id de la red de receptores”, “Dirección” y “Valores más altos para cada receptor” que el usuario puede seleccionar. Estas opciones determinan la forma del gráfico según los datos leídos del fichero (.OUT) generado por el modelo de dispersión AERMOD. Al pulsar el botón “Graficar” se crea el gráfico y se visualiza.

