

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
INSTITUTO DE INGENIERÍA**

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA



**“IMPLEMENTACIÓN DE HERRAMIENTAS COMPUTACIONALES
PARA EL DESARROLLO DE UN SISTEMA DE INFORMACIÓN
GEOGRÁFICA DE LA ENERGÍA EÓLICA EN BAJA CALIFORNIA”**

TESIS PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA

PRESENTA

LUIS ANGEL MENDOZA VINIEGRA

DIRECTOR

Dr. ALEJANDRO ADOLFO LAMBERT ARISTA

Mexicali, B. C.

Julio de 2013

ÍNDICE

ÍNDICE.....	¡ERROR! MARCADOR NO DEFINIDO.
LISTA DE ABREVIACIONES.....	IV
LISTA DE FIGURAS.....	VII
LISTA DE TABLAS.....	IX
CAPÍTULO I.....	1
1.1 INTRODUCCIÓN.....	1
1.2 PROBLEMÁTICA.....	7
1.3 OBJETIVOS.....	7
1.3.1 <i>Objetivo general</i>	7
1.3.2 <i>Objetivos específicos</i>	8
1.4 FUNDAMENTACIÓN.....	8
1.5 LIMITACIONES DEL ESTUDIO.....	14
1.6 HIPÓTESIS.....	¡ERROR! MARCADOR NO DEFINIDO.
CAPÍTULO II.....	15
2 MARCO TEÓRICO Y REFERENCIAL.....	15
2.1 ENERGÍA EÓLICA EN BAJA CALIFORNIA.....	15
2.2 SISTEMAS DE INFORMACIÓN GEOGRÁFICA Y SU INFLUENCIA EN LA SOCIEDAD BAJA CALIFORNIANA.....	18
2.3 SISTEMAS DE CÓDIGO ABIERTO Y CERRADO.....	20
2.3.1 <i>Licencia</i>	20
2.3.2 <i>Patente</i>	20
2.3.3 <i>Derecho de autor o copyright</i>	21
2.3.4 <i>Software de licencia cerrada</i>	21
2.3.5 <i>Software de licencia abierta</i>	21
2.3.5.1 <i>Freeware</i>	22
2.3.5.2 <i>Software de código abierto</i>	22
2.4 INTELIGENCIA ARTIFICIAL.....	22
2.4.1 SVD.....	23
2.4.1.1 <i>Concepto de SVD</i>	23
2.4.1.2 <i>Proceso de SVD</i>	23
2.4.2 LDA.....	24
2.4.2.1 <i>Concepto de LDA</i>	24
2.4.2.2 <i>Proceso de LDA</i>	25
2.4.3 PCA.....	25
2.4.3.1 <i>Concepto de PCA</i>	26

2.4.3.2	Proceso de PCA	26
2.4.4	K-Means.....	28
2.4.4.1	Concepto de K-Means.....	28
2.4.4.2	Proceso de K-Means.....	29
2.4.5	ICA	30
2.4.5.1	Concepto de ICA	30
2.4.5.2	Proceso de ICA.....	30
2.4.6	MDS	32
2.4.6.1	Concepto de MDS	32
2.4.6.2	Proceso de MDS	33
2.4.7	Mapeo de Sammon	36
2.4.7.1	Concepto de Mapeo de Sammon	36
2.4.7.2	Proceso de Mapeo de Sammon.....	38
2.4.8	Redes Neuronales.....	38
2.4.8.1	Neurona Biológica	38
2.4.8.2	Neurona Artificial	40
2.4.8.3	Retroalimentación multi capa con propagación hacia atrás.....	41
2.4.8.4	Algoritmo de entrenamiento de propagación hacia atrás.....	44
2.4.8.5	Entrenamiento y generalización	45
CAPÍTULO III.....		51
3	METODOLOGÍA	51
3.1	LOCALIZACIÓN DEL ÁREA DE ESTUDIO	51
3.2	DESARROLLO Y ALIMENTACIÓN DE LA BASE DE DATOS.....	52
3.3	IMPLEMENTACIÓN DE HERRAMIENTAS DE SOFTWARE A UNA APLICACIÓN SIG DE CÓDIGO ABIERTO.	54
3.3.1	Estructura general de las herramientas integradas	56
3.3.2	Commons	58
3.3.2.1	ConexionSQL	59
3.3.2.2	SetupJButton	60
3.3.2.3	SetupJMenuItem	61
3.3.2.4	SetupJSeparator.....	62
3.3.2.5	RegisteredIcon.....	62
3.3.2.6	FileEditButton	63
3.3.2.7	FileInMemory.....	64
3.3.3	Administrador de la Interfaz de Usuario Personalizada	64
3.3.4	Editor RTF	65
3.3.5	Table Editor	66
3.3.6	Editor Simple	67
3.3.7	Console.....	68
3.3.8	Matlab	70
3.3.9	Microsoft Office.....	72

3.3.9.1	Microsoft Word.....	72
3.3.9.2	Microsoft Excel	74
3.3.9.3	Microsoft Access	75
3.3.10	Wind Rose Plot.....	77
3.3.11	Trnsys	78
3.3.12	Openoffice.org	80
3.3.12.1	Openoffice.org Writer.....	80
3.3.12.2	Openoffice.org Calc	82
3.3.13	Scilab	83
3.3.14	R-Project.....	84
3.3.15	MySQL	86
3.4	IMPLEMENTACIÓN DE HERRAMIENTAS DE LA IA A UNA APLICACIÓN SIG DE CÓDIGO ABIERTO	87
3.4.1	Implementación en el sistema.....	88
3.4.2	SVD.....	89
3.4.3	LDA.....	90
3.4.4	PCA.....	90
3.4.5	K-Means.....	91
3.4.6	ICA	92
3.4.7	MDS.....	93
3.4.8	Mapeo de Sammon	94
3.4.9	Redes Neuronales Artificiales	95
3.5	IMPLEMENTACIÓN DE LAS HERRAMIENTAS DESARROLLADAS DENTRO DE UNA APLICACIÓN SIG DE CÓDIGO ABIERTO	96
CAPÍTULO IV		97
4.	RESULTADOS Y DISCUSIÓN.....	97
4.1	BASE DE DATOS	97
4.2	HERRAMIENTAS INTEGRADAS.....	98
4.2.1	Funcionalidad	98
4.2.2	Interfaz grafica.....	103
4.2.3	Herramientas IA.....	109
4.3	ANÁLISIS DE DATOS	109
4.4	COEXISTENCIA DE LOS ELEMENTOS.....	113
CAPÍTULO V		116
5.	CONCLUSIONES Y RECOMENDACIONES.....	116
REFERENCIAS BIBLIOGRÁFICAS.....		117

Lista de Abreviaciones

Descripción	Siglas en Español	Siglas en Ingles
Sistema de Información Geográfica	SIG	GIS
Sistema Inteligente de Información Geográfica	SIIG	IGIS
Instituto Nacional de Estadística y Geografía	INEGI	
Comisión estatal de servicios públicos de Mexicali	CESPM	
Comisión estatal de servicios públicos de Ensenada	CESPE	
Baja California	BC	
Generalidad Valenciana Sistema de Información Geográfica	GvSIG	
GNU No es Unix	GNU	GNU
Licencia Publica General		GPL
Consortio Abierto Geoespacial		OGC
Servicio Web de Mapas		WMS
Servicio Web de Funciones		WFS
Servicio Web de Recuperación		WCS
Suite de Topologías de Java		JTS
Lenguaje de Mercado Geografico		GML
Shapefile		SHP
Formato de Intercambio de Dibujo		DXF
DraWinG		DWG
Diseño		DGN
Lenguaje de Marcado de Cerradura		KML
MultiResolucion Sin costura de Imagen de base de Datos		MrSID
Geoinformación de Formato de Archivos de Imágenes Etiquetadas		GeoTIFF
Medio Ambiente		ENVI
Compresión Mejorada de Wavelet		ECW
Comisión Federal de Electricidad	CFE	
Informacion Referenciada geoespacialmente Integrada en un Sistema	IRIS	
Redes Neuronales Artificiales	RNA	ANN
Programa Incentivo de Fuentes Alternativas de Energía	PROINFA	
Inteligencia Artificial	IA	AI

Kilowatt / hora	KW/h	
Mecanismo de Desarrollo Limpio	MDL	
Energías Renovables	ER	RE
Base de Datos	BD	DB
Sistema de Informacion Geografica Regional		REGIS
		EPURE
		EnTRACK
		REPLAN
		SOLBIO
		SOLARGIS
Sistema de Información Geográfica para las Energías Renovables	SIGER	
Gerencia de Energías No Convencionales	GENC	
Instituto de Investigaciones Eléctricas	IIE	
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas	CIEMAT	
Diseño asistido por computadora	DAC	CAD
	GAMESA	
Mega watts	MW	
Índice de Potencial de Incendio	IPI	
Sistema de Información Geográfica del Sector Agropecuario de Baja California	SigaBC	
Descomposición singular de valores	DSV	SVD
Análisis Discriminante Lineal	ADL	LDA
Análisis de Componentes Principales	ACP	PCA
Error Cuadrático Medio	ECM	MSE
Análisis de Componentes Independientes	ACI	ICA
Escalamiento multi-dimensional	EMD	MDS
Red Neuronal	RN	NN
Retroalimentación multi-capa	RMC	MLF
Centro de Investigación de Científica y de Educación Superior de Ensenada	CICESE	
Comisión Nacional del Agua	CONAGUA	
Universidad Autónoma de Baja California	UABC	
Entidad-Relación	E-R	E-R
Interfaz de Programación de Aplicaciones	IPA	API
		TRNSYS

Wind Rose Plot		WRPLOT
Formato de Texto Enriquecido	ETE	RTF
Sistema Manejador de Base de Datos	SMBD	DBMS
Conexión abierta a Base de Datos		ODBC
		CSV
Máquina Virtual de Java	MVJ	JVM

Lista de Figuras

- Figura 1 Algunos sitios con potencial eólico en México.
- Figura 2 Áreas Potenciales de energía eólica en la península de Baja California.
- Figura 3 Lugares potenciales en BC. Para generación de energía según promedio de velocidad de viento en m/s.
- Figura 4 Proceso de descomposición de valores singulares.
- Figura 5 Esquema del proceso de mezclas y separación.
- Figura 6 Esquema de una neurona natural.
- Figura 7 Red Neuronal Feed-Forward de tres capas típica.
- Figura 8 Conexión entre dos neuronas, i y j .
- Figura 9 Principio de generalización de una red.
- Figura 10 Generalización de una red sobreajustada o sobreentrenada.
- Figura 11 Ubicación geográfica del estado de Baja California y su capital Mexicali
- Figura 12 Diagrama E-R de la base de datos.
- Figura 13 Descripción de la estructura organizacional de gvSIG.
- Figura 14 Diagrama de clases de la paquetería Commons.
- Figura 15 Diagrama de clases de la paquetería Custom UI Manager.
- Figura 16 Diagrama de clases del Editor RTF.
- Figura 17 Diagrama de clases del Editor de Tablas.
- Figura 18 Diagrama de clases del Editor Simple.
- Figura 19 Diagrama de clases de la consola.
- Figura 20 Diagrama de Clases para el módulo de Matlab.
- Figura 21 Diagrama de Clases para el módulo de Microsoft Word.
- Figura 22 Diagrama de Clases para el apartado de Microsoft Excel.
- Figura 23 Diagrama de Clases para el módulo de Microsoft Access.
- Figura 24 Diagrama de Clases para el módulo de Wind Rose Plot.
- Figura 25 Diagrama de Clases para el módulo de Trnsys.
- Figura 26 Diagrama de Clases para el módulo de OpenOffice.org Writer.
- Figura 27 Diagrama de Clases para el módulo de OpenOffice.org Calc.
- Figura 28 Diagrama de Clases para el módulo de Scilab.
- Figura 29 Diagrama de Clases para el módulo de R-Project.

Figura 30 Diagrama de Clases para el módulo de MySQL.

Figura 31 Diagrama de Clases para el módulo de SVD.

Figura 32 Diagrama de Clases para el módulo de LDA.

Figura 33 Diagrama de Clases para el módulo de PCA.

Figura 34 Diagrama de Clases para el módulo de K-means.

Figura 35 Diagrama de Clases para el módulo de ICA.

Figura 36 Diagrama de Clases para el módulo de MDS.

Figura 37 Diagrama de Clases para el módulo de Mapeo de Sammon.

Figura 38 Diagrama de Clases para el módulo RNA MLF BP.

Figura 39 Vista 1 del Editor RFT para procesamiento de texto.

Figura 40 Vista 2 del Editor RFT para procesamiento de texto.

Figura 41 Vista 3 del Editor RFT para procesamiento de texto.

Figura 42 Vista 4 del Editor RFT para procesamiento de texto.

Figura 43 Vista 5 del Editor RFT para procesamiento de texto.

Figura 44 Vista 1 del Editor Simple para procesamiento de texto.

Figura 45 Vista 2 del Editor Simple para procesamiento de texto.

Figura 46 Vista 3 del Editor Simple para procesamiento de texto.

Figura 47 Vista 1 del Table Editor para el manejo de tablas.

Figura 48 Vista 2 del Table Editor para el manejo de tablas.

Figura 49 Vista 3 del Table Editor para el manejo de tablas.

Figura 50 Resultado del procesamiento de PCA.

Figura 51 Resultado del procesamiento de SVD.

Figura 52 Regresión lineal de los resultados obtenidos de una RNA MLF BP.

Figura 53 Vista 2 del Editor Simple para procesamiento de texto.

Figura 54 Vista 3 del Editor Simple para procesamiento de texto.

Figura 55 Vista 1 del Table Editor para el manejo de tablas.

Figura 56 Vista 2 del Table Editor para el manejo de tablas.

Figura 57 Vista 3 del Table Editor para el manejo de tablas.

Figura 58 Resultado del procesamiento de PCA.

Figura 59 Resultado del procesamiento de SVD.

Figura 60 Regresión lineal de los resultados obtenidos de una RNA MLF BP.

Figura 61 Regresión lineal de los resultados obtenidos de una RNA MLF BP.

Lista de Tablas

Tabla 1	Tabla de equivalencias de Kruskal para el nivel de STRESS1.
Tabla 2	Modelo relacional de la BD.
Tabla 3	Funcionalidad mostrada desde la barra de herramientas.
Tabla 4	Funcionalidad que ofrece cada uno de los editores desarrollados.
Tabla 5	Descripción de la funcionalidad que contiene cada evento.

Capítulo I

1.1 Introducción

Un Sistema de Información Geográfica (SIG) es la comunión ordenada entre el *hardware*, el *software* y los *datos* geográficos, un SIG suele ser una aplicación generada para necesidades específicas, pero esto, no lo convierte en un sistema inflexible, ya que este tipo de sistemas son capaces de dar respuestas de manera rápida y oportuna, también permiten la manipulación de datos y mapas para poder brindar resultados de las operaciones realizadas (Bolstad, 2005).

Los SIG han ido incursionando en la vida diaria en las diferentes áreas dando acceso a estudios socioeconómicos, marketing, estudios ambientales entre otros. Estos sistemas son implementados para resolver cuestiones de localización geo-espacial, validación de las condiciones impuestas al sistema, la realización de tendencias temporales o espaciales, el cálculo de la ruta óptima entre dos o más puntos, la generación de modelos a partir de fenómenos o de otras simulaciones.

GvSIG es un proyecto de desarrollo de Sistemas de Información Geográfica en software libre, que incluye principalmente las aplicaciones gvSIG Desktop y gvSIG Mobile. La aplicación gvSIG Desktop fue la primera que se desarrolló dentro del proyecto gvSIG, por lo que también se conoce abreviadamente como gvSIG. Este proyecto fue desarrollado por el gobierno local de la Comunidad Valenciana (Generalidad Valenciana) de España, con el objetivo inicial de realizar la gestión de datos geográficos de esa colectividad; precisamente la sigla gvSIG abrevia la denominación *Generalidad Valenciana Sistema de Información Geográfica* (GvSIG, s.f.).

GvSIG Desktop es un programa informático para el manejo de información geográfica con precisión cartográfica que se distribuye bajo licencia GNU GPL v2. Permite acceder a información vectorial y rasterizada así como a servidores de mapas que cumplan las especificaciones del Open Geospatial Consortium (OGC). Esta es una de las principales características de gvSIG respecto a otros SIG, la importante implementación de servicios OGC: WMS (Web Map Service), WFS (Web Feature Service), WCS (Web Coverage Service), Servicio de Catálogo y Servicio de Nomenclátor (GvSIG, s.f.).

Está desarrollado en lenguaje de programación Java, funcionando con los sistemas operativos Microsoft Windows, Linux y Mac OS X, y utiliza bibliotecas estándar de SIG reconocidas, como Geotools o Java Topology Suite (JTS). Asimismo, gvSIG posee un lenguaje de scripting basado en Jython y también se pueden crear extensiones en Java utilizando las clases de gvSIG.

Entre los formatos gráficos de fichero más habituales cuenta entre otros con acceso a formatos vectoriales GML, SHP, DXF, DWG, DGN, KML y formatos de imagen rasterizada como MrSID, GeoTIFF, ENVI o ECW (GvSIG, s.f.).

Hoy en el estado de Baja California se utilizan aplicaciones SIG en varios sectores públicos como son Instituto Nacional de Estadística y Geografía (INEGI), Comisión Estatal de Servicios Públicos de Mexicali (CESPM), Comisión Estatal de Servicios Públicos de Ensenada (CESPE), Comisión Federal de Electricidad (CFE), entre otros. INEGI por su parte, utiliza su propia aplicación IRIS la cual también se utiliza para distribución, esta

aplicación está dirigida a cubrir las necesidades de los productores de información del país. Además de esta aplicación, INEGI ha desarrollado otras herramientas de fácil acceso para los usuarios, por ejemplo la página “cuéntame” (INEGI, s.f.)

En los últimos años se ha incrementado el uso de los SIG para diversos propósitos, en áreas de investigación, áreas de planeación y zonificación urbana, control de plagas, etc. A continuación se mencionaran algunos casos de éxito utilizando estos sistemas.

Azpurua & Ramos (2010) realizaron varias mediciones geo-referenciadas de campos eléctricos en un área piloto en Caracas, Venezuela, para corroborar que las magnitudes de radio frecuencia de los campos electromagnéticos están por debajo de los límites de exposición del hombre, recomendada por la Comisión Internacional de Protección de Radiación No Ionizada. Los datos recolectados fueron analizados usando un sistema de información geográfica, con el objetivo de utilizar técnicas de interpolación para estimar la magnitud promedio de campos electromagnéticos, el conjunto de datos continuos que puede ser representado en un mapa de toda el área piloto.

En el Valle de Calamuchita (Córdoba-Argentina), la cual posee un área forestada de 36.202 ha. de *Pinus spp.*, se trabaja desde 1996 en un SIG (Sistema de Información Geográfica) como herramienta complementaria para el estudio de la plaga *Sirex noctilio* Fabricius (Hymenoptera-Siricidae). El objetivo es almacenar, organizar, contrastar y analizar los datos obtenidos del estudio de *Sirex noctilio* F., con el fin de generar nueva información, que servirá como base en la definición de una estrategia de manejo de la plaga (López, Demastri, Barotto, Viale, Degioanni, & Zupan, 2002).

Rodríguez, Boullón, Cordero, Miranda, & Crecente (2007) abordaron el problema de la gestión de espacios con la puesta en marcha de un proyecto de investigación sobre la implementación de sistemas de información geográfica dentro de la universidad que consideren al mismo tiempo la oferta de medios existentes (despachos, laboratorios, aulas, seminarios...) y la demanda (ocupación del espacio por el personal docente, investigador, administrativo y de servicios, alumnado...) y en relación a las posibles actividades, edificios, departamentos.

Thurston J, (2002) en su artículo *GIS & Artificial Neural Networks: Does Your GIS Think?* menciona que el propósito de un SIG consta de proveer a la persona y a la organización el conocimiento y el entendimiento de los datos espaciales. Seguido lo usuarios de un SIG subestiman la capacidad de 'toma de decisiones' que estos sistemas pueden dar, en cambio, se centran en la presentación .

Los Sistemas de Información Geográfica (SIG) son una tecnología utilizada para capturar, almacenar, manipular, analizar, y visualizar datos referenciados espacialmente. Los SIG han incrementado su utilidad por la realización de modelos y análisis espaciales, capaces de moverse en áreas para las que no son necesariamente destinados (Paliulionis, 2000).

Actualmente se le han aplicado a los SIG algoritmos para el soporte a la toma de decisiones, estos algoritmos son ramificaciones de la inteligencia artificial, y ayudan a encontrar patrones de búsqueda, para dar un resultado más exacto, este tipo de sistemas se les denomina Sistema Inteligente de Información Geográfica (SIIG), este tipo de

sistemas son utilizados para poder analizar el comportamiento actual y realizar estimaciones a futuro, de tal manera que a raíz de simulaciones pueden proyectar soluciones a largo o corto plazo (Shouman, 2008).

En el caso de las Redes Neuronales Artificiales (RNA), las metodologías de la computación la utilizan para simular el procesamiento los problemas de datos espaciales del cerebro humano. Previendo que muchas aplicaciones futuras incorporaran elementos de inteligencia artificial. Estas redes tienen mucho potencial si las incluimos en una aplicación SIG; paisajes, oceanografía, forestación, *marketing*, transporte, estudios de biosfera, análisis de imágenes, medio ambiente, entretenimiento, anti-terrorismo, análisis de patrones y medicina”. De hecho, en varios de estos casos los SIG ya son utilizados (Thurston, 2002).

El elegir un lugar Industrial envuelve una serie de aspectos críticos que hay que tomar en cuenta como el factor económico, social, técnico, ambiental. La elección del lugar es el proceso para elegir el sitio con las características que necesitamos a través de una selección de criterios. En algunos procesos, el satisfacer multi-criterios y la manipulación de datos espaciales son esenciales para una exitosa toma de decisiones. Porque hay un gran número de herramientas que podemos elegir para llegar a una apropiada solución de problemas complejos. Un enfoque propuesto se encuentra en los Sistemas Expertos, Sistemas de Información Geográfica y Técnicas de decisión de multi-criterios utilizados en conjunto para resolver los problemas para elegir lugares industriales (Shouman, 2008).

Se ha alcanzado un nuevo nivel sobre en evaluación del recurso eólico, debido a los grandes equipos y la reciente disponibilidad de datos detallados sobre la información topográfica y meteorológica. Además de la superficie en las estaciones meteorológicas, estos datos ahora incluyen el datos meteorológicos de altitud, *ship data*, y datos satelitales. El uso en conjunto de estos datos con un SIG con mapas detallados del comportamiento del viento se puede obtener una evaluación detallada sobre el recurso eólico analizado, todo esto en base a una dependencia del grado de los datos. Se puede evaluar en aéreas grandes o pequeñas, y saber cuáles áreas podrían tener mayor viabilidad, así como asistir en el diseño de turbinas *micrositing* (Lew, 2000).

Brasil por tener un amplio potencial eólica, desde la última década ha ido implementando políticas para promover el uso de algunos proyectos, principalmente por el Programa Incentivo de Fuentes Alternativas de Energía (PROINFA), que empezó en el 2002. Antes de empezar la primera fase PROINFA, busca por un programa a largo plazo para la energía eólica en Brasil. Para utilizar la información recabada en la base de datos sobre el comportamiento del viento mediante una aplicación SIG (Dutra & Szklo, 2008).

El estudio emplea un SIG para mapear los recursos de energía eólica y analizar su variabilidad considerando los aspectos estacionales y espaciales del estado de Karntaka. Tomando en cuenta esto, se ha preparado una evaluación del estado actual de la potencia y los mapas de los lugares adecuados para aprovechar la energía eólica. Se desarrolló y utilizó una base de datos espacial con datos de las velocidades de viento para la evaluación de su potencial teórico junto a un continuo monitoreo de los recursos eólicos (Ramachandra & Shruthi, 2005).

Baja de California se encuentra en una fase de investigación para la explotación de energías renovables en esta entidad, donde el recurso eólico ha sido uno de los focos de atención de investigación.

Actualmente se cuenta con información estadística sobre el viento en el estado de Baja California y su potencial energético. Las cuales demuestran que en algunos sectores del mismo hay lugares donde se tiene un alto potencial que podría aprovecharse para cubrir las necesidades energéticas de comunidades sin energía eléctrica.

1.2 Problemática

En la actualidad se necesita utilizar varias tecnologías de software ajenas a la plataforma SIG usada para realizar un estudio completo. Se han realizado esfuerzos de compatibilidad entre las tecnologías, las cuales quedan solo en reconocimiento y generación de archivos entre ellas.

1.3 Objetivos

1.3.1 Objetivo general

Incorporar la información de la velocidad y dirección de viento recolectada, implementar tecnologías de software e integrar herramientas de la Inteligencia Artificial (IA) dentro de una aplicación SIG, para crear una herramienta computacional robusta que de soporte a

la toma de decisiones sobre el manejo de la energía eólica, en la ciudad de Mexicali, Baja California, el cual sea capaz de realizar predicciones, simulaciones e interpolación del comportamiento del viento.

1.3.2 Objetivos específicos

- Incorporar los procesos básicos de las herramientas de software de licencia cerrada dentro de la plataforma SIG.
- Incorporar los procesos básicos de las herramientas de software licencia abierta o gratuito dentro de la plataforma SIG.
- Incorporar la información de la velocidad y dirección del viento analizada dentro del SIG.
- Implementar herramientas de la IA para el manejo de los datos geo-referenciados dentro de la plataforma SIG.
- Crear una base de datos para el manejo de datos de las estaciones meteorológicas.

1.4 Fundamentación

La sustitución en muchos usos del petróleo y el carbón por gas natural, es un proceso en marcha en los países desarrollados, así como el aumento en la eficiencia de su uso. Pero este camino tiene un límite impuesto por las leyes de la física, de forma que no es posible superar ciertos valores para la eficiencia en el uso de la energía, lo que finalmente acaba haciendo necesaria la búsqueda de energías alternativas, eficientes y competitivas económicamente. Esto hace que las energías renovables estén siendo consideradas con gran interés, tratando de hacerlas competitivas con los sistemas convencionales.

Los países desarrollados han dedicado esfuerzos logrando conseguir grandes avances en turbinas eólicas de alto rendimiento, utilizando nuevos materiales y diseños con grandes avances tecnológicos, mejorando ampliamente la eficiencia y disminuyendo sus costos unitarios, lo que las vuelve cada vez más competitivas, así, el costo actual de producción del Kw/h es la quinta parte del conseguido en 1980. Volviéndolas competitivas contra centrales térmicas de carbón, mucho más si se consideraran los beneficios medioambientales (Sardón, 2002).

La energía eólica tiene como ventajas la de ser inagotable, gratuita y no lesiva al medio ambiente, pero cuenta también con los grandes inconvenientes de ser dispersa y aleatoria (Fernández, 2002).

México cuenta con un enorme potencial eólico. Entre los sitios identificados con viento de alta calidad se encuentran sus zonas costeras, especialmente en los estados de Baja California, Baja California Sur, Sonora, Oaxaca y Yucatán. Desafortunadamente, dicho potencial se ha explotado poco (González-Ávila, Beltrán-Morales, Troyo-Diéguéz, & A., 2006).

Se han realizado prospecciones del recurso energético eólico a un nivel exploratorio y de reconocimiento puntual, y pequeñas redes anemométricas realizadas por parte de instituciones federales mexicanas que han servido para confirmar la factibilidad de áreas para establecer un parque o central eólica. En la figura 1 se muestran las zonas con potencial eólico de México, así como las zonas con vientos técnicamente aprovechables y económicamente viables (González-Ávila & et.al., Potencial de aprovechamiento de la energía eólica para la generación de energía eléctrica, 2006).

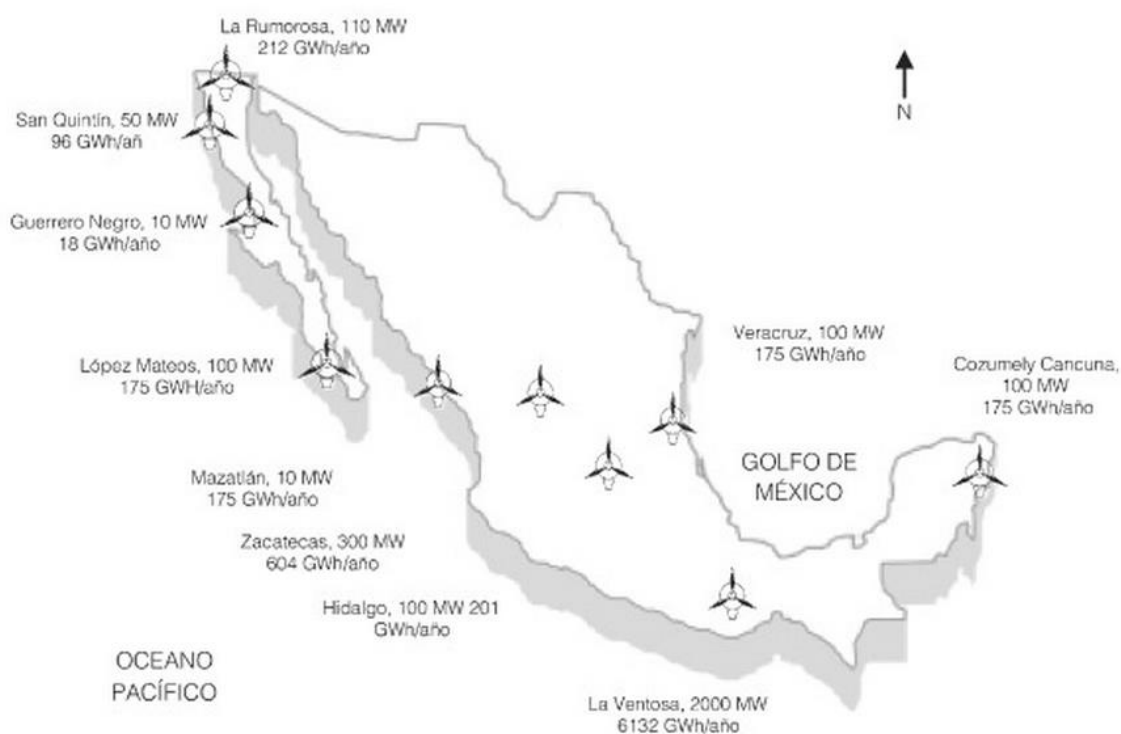


Figura 1. Algunos sitios con potencial eólico en México (González-Ávila & et.al., Potencial de aprovechamiento de la energía eólica para la generación de energía eléctrica, 2006).

La península de Baja California es una zona con una barrera eólica natural perpendicular a los vientos occidentales. Tal es el caso de las áreas cercanas a las pobladas de La Rumorosa y zonas aledañas, así como el paso entre la Sierra de Juárez y la Sierra de San Pedro Mártir (Caldera, 2000).



Figura 2. Áreas Potenciales de energía eólica en la península de Baja California.

En producción de energía eólica, Baja California es el único registrado como proyectos de Mecanismo de Desarrollo Limpio (MDL), en donde además se encuentran registrados proyectos de cogeneración y eficiencia energética que se producen en Nuevo León, Coahuila y Baja California (González M. , 2009).

El área de las energías renovables (ER) no ha sido ajena a la tendencia del uso de SIG que son una tecnología que ha revolucionado al mundo en cuanto a manejo de BD; ya que no se puede entender el manejo de una BD que contenga el componente espacial sin el uso de un SIG, y desde hace varios años, en algunos países del primer mundo, se han desarrollado herramientas basadas en SIG como apoyo para la toma de decisiones en proyectos de ER, tal es el caso de los sistemas REGIS, EPURE, EnTRACK, REPLAN, SOLBIO y SOLARGIS (Ubaldo, Saldaña, & Morales, 2003).

SIGER (El Sistema de Información Geográfica para las Energías Renovables) es un sistema desarrollado por la Gerencia de Energías No Convencionales (GENC) del Instituto de Investigaciones Eléctricas (IIE) dio inicio al proyecto con el fin de participar en el ámbito internacional de este nuevo esquema de manejo de información (Ubaldo, Saldaña, & Morales, 2003).

El desarrollo del Siger, al igual que el de cualquier SIG, depende fundamentalmente de la información disponible. En el caso de las ER existe un gran vacío, por ejemplo, si se trata de encontrar algún mapa de energía eólica en México para adaptarlo e incorporarlo al sistema, el resultado será que no existen mapas de este recurso. Lo mismo ocurre con el de biomasa y con el de micro hidráulica (Ubaldo, Saldaña, & Morales, 2003).

En cuanto a la información general, sucede algo semejante. Salvo en casos muy particulares como son la topografía (que es fundamental), las carreteras y algunas otras; pues no hay información detallada accesible y lo mismo pasa con las líneas eléctricas de distribución o la distribución espacial de cultivos (Ubaldo, Saldaña, & Morales, 2003).

En Ensenada, Baja California también se han realizado proyectos en apoyo de las ER y los SIG, con el sistema IntiGIS. En una aplicación desarrollada por el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) de Madrid para analizar las alternativas tecnológicas encaminadas a la electrificación rural. IntiGIS permite un análisis multi-criterio donde se incluyen variables sociodemográficas, económicas, cartográficas y técnicas (Osuna, 2012).

Durante el proceso de creación y manejo de mapas, y la posterior salida de los mismos se suele utilizar una gran cantidad de programas y no se llegan a cubrir todas las posibles expectativas. Dependiendo de nuestro interés existen programas más competitivos que otros y con mejores resultados. Por tanto los programas acaban especializándose en función del tipo de datos que se supone se van a utilizar, el tipo de aplicaciones y la lógica de trabajo (Gonzalez, Verdugo, & Coronel, 2006)

Existen otros programas de gran importancia por su interacción con los datos SIG, estos son los sistemas CAD, hojas de cálculo, bases de datos, editores de texto, software de programación, programas de retoque fotográfico, diseño gráfico 3D entre otros (Gonzalez, et.al., 2006).

En otros países se han realizado estudios para la predicción de la energía eólica, por medio de la IA, algunos de estos son: *Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models* (Jursa & Rohrig, 2007), *A review on the forecasting of wind speed and generated power*,

Renewable and Sustainable Energy Reviews (Lei, Shiyan, Chuanwen, Hongling, & Yan, 2008), (Sfetsos, 2000), *Analysis of wind power generation and prediction using ANN: A case study* (Carolin & Fernandez, 2008), *Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model*(Cadenas & Rivera, 2010). En México hay pocos estudios aplicando éste tipo de metodologías orientadas a la energía eólica. Áreas como Isla de los Cedros en Baja California, Cerro de la Virgen en Zacatecas y Holbox en Quintana roo se han aplicado modelos híbridos de predicción como es Autoregressive Integrated Moving Average (Morrison, s.f.) Y las RNA.

Debido a lo anterior redactado, se puede observar la importancia una herramienta que sea capaz contribuir al análisis de la energía eólica en baja california integrando las necesidades básicas para realizar un estudio completo.

La realización de este proyecto se vuelve importante ya que la región de Baja California cuenta con un gran potencial eólico y existen pocos trabajos apliquen técnicas alternativas como es la IA. También debido a la poca información histórica se vuelve imperante el abastecer una base de datos que contenga la información necesaria a la mano de las diferentes estaciones meteorológicas ubicadas en BC, así como el fácil acceso y manejo de las herramientas que requiere un SIG para un desarrollo rápido y oportuno.

Para aplicar estudios que permitan explotar el recurso eólico que con el que cuenta la región de Baja California.

1.5 Limitaciones del estudio

Una de las fronteras del estudio es el enfrentarse a la constante evolución de las herramientas de software, esto es, día a día nacen nuevas herramientas que van dejando atrás a programas clásicos.

Otra de las limitaciones es la imposibilidad de entrar al código fuente de los programas de código cerrado, por lo que se tuvo que crear mecanismos que se puedan comunicar con estos sistemas.

Otro de los problemas es la falta de documentación fiable para el desarrollo sobre el armazón que se encuentra en constante evolución, esto ocasiona que durante el transcurso de la codificación se pudieran generar errores (bugs) por cambios en el código fuente.

Capítulo II

2 Marco Teórico y referencial

En este capítulo se presenta el marco teórico-referencial que sustenta el trabajo de investigación realizado, para ello el capítulo se ha organizado en 4 apartados:

Energía eólica en Baja California, Sistemas de Información Geográfica y su influencia en la sociedad, sistemas de código abierto, Inteligencia artificial.

2.1 Energía eólica en Baja California.

El gobierno mexicano ha hecho varios esfuerzos en incluir la energía eólica en las políticas energéticas. De tal manera que se reduzca la dependencia a los combustibles fósiles para la generación eléctrica y por lo tanto disminuir las emisiones dañinas al medio ambiente. A parte de la energía hidráulica y la geotérmica la generación de energía de ER sigue siendo limitada a pesar del potencial energético disponible en el territorio. Sin embargo, la energía eólica se ha incluido en la canasta energética como una de las ER viables para corto y mediano plazo. La meta del gobierno mexicano para el 2012 es lograr que el 5% de la energía generada venga de la energía eólica (Cancino, et.al., 2011).

México cuenta con un enorme potencial eólico. Entre los sitios identificados con viento de alta calidad se encuentran sus zonas costeras, especialmente en los estados de Baja California, Baja California Sur, Sonora, Oaxaca y Yucatán. Desafortunadamente, dicho potencial se ha explotado poco. El presente trabajo analizó zonas rurales con potencial para la generación de energía a partir de los vientos en México, considerando los aspectos sociales y económicos involucrados en el desarrollo de proyectos de energía eólica. Los resultados indican que además del desarrollo tecnológico requerido, se deben fortalecer los aspectos legales que fomenten e incentiven la producción de energía eólica, y establecer regulaciones y normas ambientales que protejan al ambiente y a las poblaciones silvestres (González, et.al., 2006).

Respecto a los vientos en B.C. las estadísticas señalan que las mayores cantidades de viento se pueden encontrar en las zonas costeras, por lo que en teoría los generadores eólicos más eficientes se instalarían en islas o sectores litorales (Moreno, et al., 2006). La figura 3 muestra los lugares con los mejores vientos para la generación de energía eólica.

Donde Primo Tapia e Isla de Cedros se ubican en zonas costeras. Destacando que en la Rumorosa ya se instaló un parque eólico. En todos los lugares el promedio sobrepasa los 5 m/s, sobresaliendo Isla de Cedros con un promedio de 5.46 m/s (López & Marín, 2010).

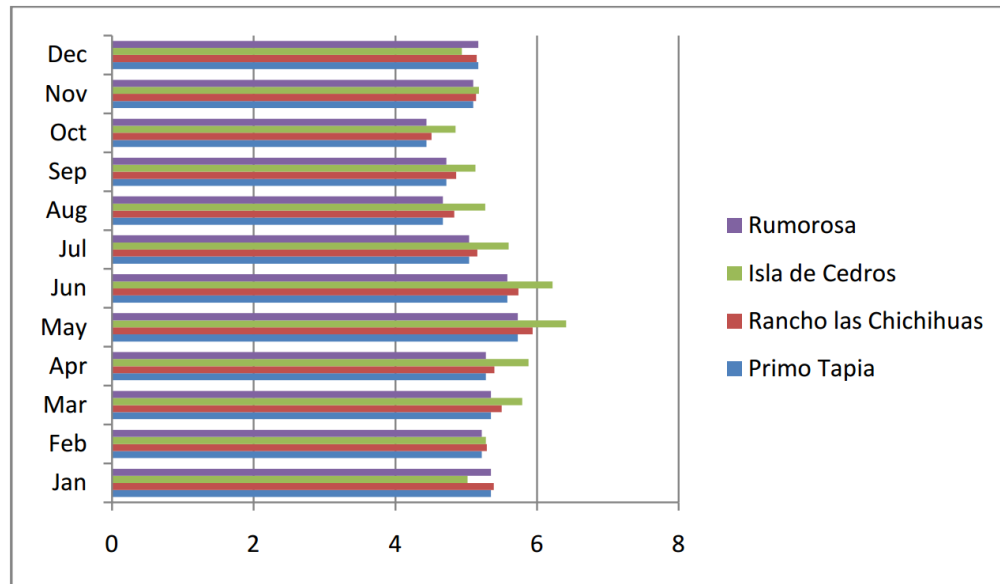


Figura 3. Lugares potenciales en BC para generación de energía según promedio de velocidad de viento en m/s.

En BC el interés por la instalación de proyectos de energía eólica es bastante reciente a pesar de su geografía privilegiada para la instalación de este tipo de generación. Por ejemplo, el parque de la Rumorosa generará diez mega watts de energía eléctrica renovable (López & Marín, 2010).

Baja California, Parque Eólico: La Rumorosa I, presenta grandes ventajas, gracias a sus características geográficas y topográficas, además que las líneas de transmisión, a diferencia de Oaxaca, se encuentran cerca. El proyecto se finalizó en Septiembre de

2009, y entró en operación en Marzo de 2010 con una capacidad de producción de 10 MW (cinco turbinas de 2MW cada una, modelo G-87, marca GAMESA) y una inversión de 26 millones de dólares, la empresa responsable de ejecutar el proyecto fue Turbo Power Services de capital estadounidense (Brambila & Jiménez, 2012).

2.2 Sistemas de Información Geográfica y su influencia en la sociedad baja californiana

Los SIG constituyen una potente herramienta capaz de manipular elevadas cantidades de información, lo que nos permite efectuar una serie de análisis espaciales de sumo interés. Merino, J., & Marauri, (1995) en su documento *Aplicaciones de los sistemas de información geográfica en los estudios geomorfológicos y medioambientales el mapa sintético de riesgos potenciales y el mapa de erosión* sintetizan la problemática geomorfológica y medioambiental de las laderas y Suelos de la zona estudiada (movimientos en masa, inundaciones y erosión), lo que constituye un valioso estudio de cara a la planificación y gestión del territorio.

Tal es el trabajo de pronóstico para el potencial de incendios forestales en baja california, donde se creó un SIG basado en el modelo de Índice de Potencial de Incendio (IPI), con capacidad de generar mapas de pronósticos de 24 horas, utilizando la información disponible y sin detrimento significativo de la confiabilidad de los resultados, pues es importante conocer con anticipación y cierto grado de certeza, donde y cuando se pueden originar; por ello se requiere definir en un contexto espacial y temporal la presencia de los eventos (Sepulveda & et.al., 2011).

También se han realizado esfuerzos para abordar otras áreas como es la agricultura, el Sistema de Información Geográfica del Sector Agropecuario de Baja California (SigaBC), es un instrumento de consulta y análisis diseñado con el objeto de presentar datos geográficos digitales organizados en capas sobre distintos aspectos relacionados con la agricultura, ganadería, pesca, clima y recursos hidráulicos de nuestro Estado. El SigaBC es una aplicación para Internet, diseñado en plataforma MapServer CGI 5.6.3 & p.mapper 3.2.0, Que permite consultar información en un contexto espacial, desde diferentes niveles de acercamiento (Oficina Estatal de Información para el Desarrollo Rural Sustentable OEIDRUS, s.f.).

También otras áreas que se ha visto beneficiadas por los SIG es el área de la mercadotecnia un ejemplo es el área vitícola, Sistema de Información Geográfica para la Identificación y Ubicación de las Zonas Vitícolas de Baja California, ataca Debido a factores tanto de mercado como ambientales (suelo, agua) así como a los planes de desarrollo futuro de la viticultura en la región, se hace necesario conocer donde están ubicados geográficamente los actuales viñedos (distribución espacial) y cuales son su componentes varietales por cada viñedo, quienes son su propietarios, qué características tiene el entorno ambiental y de infraestructura (camino, servicios, etc.) esta información es fundamental para la toma de decisiones.

El ámbito del proyecto comprende las áreas con actividad vitícola del estado de Baja California. La información sobre la localización, y componentes varietales de las áreas vitícolas se llevó a cabo en dos etapas, una de gabinete y otra de campo. Se generó un

plano digital de los viñedos y se construyó una base de datos; tanto el plano digital como la base de datos que se integran en un solo sistema de información. Para establecer comunicación entre el SIG y la página Web del Sistema Producto Vid, se implementó la liga con un acceso restringido mediante clave de usuario y contraseña. Se elaboró un manual que describe cada uno de los pasos a seguir para la operación del mismo, así como para la interpretación de los productos derivados (Cofupro, s.f.).

2.3 Sistemas de código abierto y cerrado.

El software se encuentra dividido en dos partes una es el software libre o de licencia abierta y el software no libre o de licencia cerrada, los cuales se encuentran regidos por diferentes licencias, patentes y/o derechos de autor.

2.3.1 Licencia

Contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software (Free Software Foundation, Inc, 2006).

2.3.2 Patente

Conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto (material o inmaterial) susceptible de ser explotado industrialmente para el bien del solicitante por un periodo de tiempo limitado (OMPI, s.f.).

2.3.3 Derecho de autor o copyright

Forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar (OMPI, s.f.).

2.3.4 Software de licencia cerrada

Este tipo de software es aquel que priva de alguna manera el uso, distribución, modificación, mejoras al código del programa.

Este tipo de software es usualmente respaldado por derechos de autor utilizando un copyright (OMPI, s.f.). Debido a que varios de los programas más populares son de licencia cerrada y algunos son usados para desarrollo matemático, redacción de documentos, administración de bases de datos, uso de sistema operativo, etc. Se pensó en implementar un vínculo entre nuestro sistema de desarrollo de SIG (GvSIG) y este software.

2.3.5 Software de licencia abierta

Este tipo de licencias permiten el acceso al código fuente o la ejecución del software para cualquier propósito, ya sea: modificación o redistribución, el programa original o cualquiera de las modificaciones ejercidas sobre el mismo. Dándose la distribución del producto derivado bajo las mismas condiciones de licenciamiento que el programa original (Free Software Foundation, Inc, 2006).

2.3.5.1 Freeware

Este tipo de licencias permite al usuario adquirir el software de manera gratuita y con una duración ilimitada sobre el uso del programa. Cabe mencionar que este tipo de sistemas no se permite la venta, modificación o acceso al código fuente. Generalmente este tipo de software no es de licencia libre.

2.3.5.2 Software de código abierto

Es el tipo de licencia que te permite tener acceso a todas las funcionalidades del programa y al código fuente, permitiendo libertad de uso para cualquier propósito, libertad de modificación a conveniencia, libertad de distribución del programa y libertad de realizar mejoras y hacer publicación sobre las mejoras del mismo.

A diferencia de las licencias freeware o shareware, el software realizado bajo este tipo de licencias suele ser gratuito y libre.

2.4 Inteligencia artificial

2.4.1 SVD

2.4.1.1 Concepto de SVD

El termino SVD se refiere al proceso de descomponer una matriz disminuyendo la dimensionalidad al problema. Este método tiene como objetivo perder la menor cantidad de información maximizando la varianza total, y mantener la independencia lineal conservando la estructura original de los datos (Rios, 2009).

2.4.1.2 Proceso de SVD

El proceso SVD como se mencionó en el punto 2.4.1.1 busca descomponer un conjunto de datos de valores singulares en la figura X se muestra el proceso de este algoritmo.

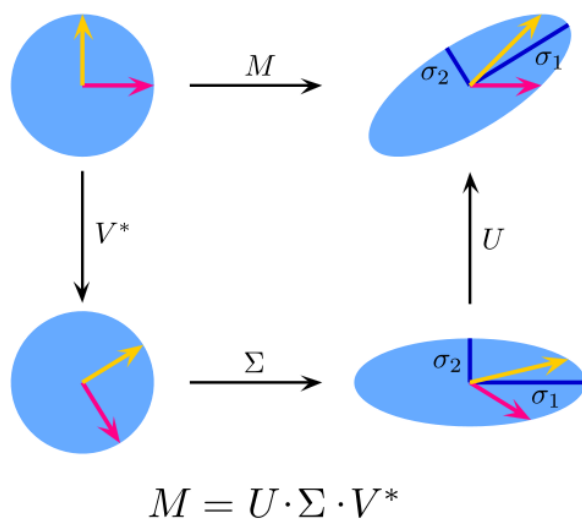


Figura 4. Proceso de descomposición de valores singulares (Wikipedia, s.f.).

Sea $A \in \mathbb{C}^{m \times n}$. Su SVD es: $A = USV^*$

Dónde:

$U \in \mathbb{C}^{m \times n}$ Es unitaria ($UU^* = I_m$),

$S \in \mathbb{R}^{m \times n}$ Es “diagonal”, con “diagonal” no negativa y no creciente,

$V \in \mathbb{C}^{m \times n}$ Es unitaria.

En lo que sigue, sea $p = \min\{m, n\}$. Usualmente la definición de matriz diagonal se aplica únicamente a matrices cuadradas. Una matriz S rectangular es diagonal si $s_{ij} = 0$ para $i \neq j$. Su diagonal esta compuesta por los elementos diagonales, es decir, los elementos s_{ii} , $i = 1, \dots, p$. Su diagonal no es negativa si $s_{ii} \geq 0$ para $i = 1, \dots, p$. Su diagonal es no creciente si $s_{11} \geq s_{22} \geq \dots \geq s_{pp}$. Indicando como resultado fundamental que toda matriz tiene descomposición en valores singulares. Los valores s_{ii} se llaman los valores singulares de A (a veces son restringidos a valores positivos). Frecuentemente los valores singulares se denotan $\sigma_1 = s_{11}$, $\sigma_2 = s_{22}$, $\dots$, $\sigma_p = s_{pp}$ (Mora, 2011).

2.4.2 LDA

2.4.2.1 Concepto de LDA

El análisis discriminante equivale a un análisis de regresión donde la variable dependiente es categórica y tiene como categorías la etiqueta de cada uno de los grupos, y las variables independientes son continuas determinan a que grupos pertenecen los objetos.

Se trata de encontrar relaciones lineales entre las variables continuas que mejor discriminen en los grupos dados a los objetos. Además, se trata de definir una regla de decisión que asigne un nuevo objeto, que no sabemos clasificar previamente, a uno de los grupos prefijados.

2.4.2.2 Proceso de LDA

Este tipo de análisis presenta una serie de restricciones o supuestos:

- i. Se cuenta con una variable categórica y el resto de las variables son de intervalo, siendo las últimas independientes de la primera.
- ii. Es necesario que existan al menos dos grupos y para cada grupo se necesitan dos o más casos.
- iii. El número de variables discriminantes p debe ser menor que el número de objetos n menos dos.

$$p < (n - 2) \quad (1)$$

- iv. Ninguna variable discriminante puede ser combinación lineal de otras variables discriminantes.

Este tipo de análisis de estadística multivariante tiene la finalidad de comparar y describir las medias entre g grupos de objetos sobre los que se observan p variables (Hervé, 2007).

2.4.3 PCA

2.4.3.1 Concepto de PCA

Este método recurre a examinar la relación existente entre las componentes principales y las variables originales (u otras auxiliares), para por medio de esta relación tratar de darles un contenido a su significado, para lo que la información básica para esta tarea es la matriz de correlaciones entre las componentes principales y las variables originales, que toman en este ámbito el nombre de *matriz de estructura* (Quinn, 1992).

Un componente principal es una combinación lineal de las variables originales. En este método los nuevos factores son independientes entre sí, es decir, no están correlacionados. Por medio de dicha técnica se logra disminuir el número de variables que se tenía originalmente, y con las recién creadas se puede explicar gran parte de la variabilidad total de los datos (Hair, Rolph, & Tatham, 1987).

Es un procedimiento que utiliza una transformación ortogonal para convertir un grupo de variables correlacionadas en un grupo de valores, de variables separadas. El número de componentes principales es menor o igual al número original de variables (Hair, Rolph, & Tatham, 1987).

2.4.3.2 Proceso de PCA

En PCA se trata de buscar las componentes principales de las observaciones que consisten en los autovalores de la matriz de covarianzas de éstas. Para poder aplicar el

método, debemos centrar el vector x . Para ello, simplemente, tenemos que sustraerle la media:

$$\tilde{x} = x - E\{x\} \quad (2)$$

Denotaremos como y la nueva representación de los datos. Se trata de un vector de dimensión m . En general, se cumplirá que $m \leq n$ siendo n la dimensión de los datos originales.

La nueva representación de los datos tras aplicar las técnicas PCA vendrá dada por:

$$Y = W^T X \quad (3)$$

Donde W es una matriz cuyas filas son vectores de dimensión n , tales que maximizan la varianza entre las componentes del vector y . Para dicha matriz sus columnas serán ortogonales entre sí, es decir:

$$w_i^T w_j = \delta_{ij} \quad (4)$$

Puede demostrarse, según (Hyvärinen, Karhunen, & Oja, 2001) en “Independent Component Analysis”, que las columnas de la matriz W coinciden con una base ortogonal expandida a partir de los autovectores $e_1, e_2, \dots, e_m$ de la matriz de covarianzas, C_x del vector x . Dichos autovectores serán distintos entre sí y se ordenarán en función del valor de sus autovalores $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$. La mayor parte de la información estará

contenida en el primer autovector e_1 . La cantidad de información contenida en cada autovalor será mayor cuanto mayor sea su autovalor asociado, de forma que en las primeras componentes se concentrará la mayor parte de la información. Al valor $y_1 = e_1^T x$ se le denomina componente principal de x , y es la que contiene más información de todas.

En el caso de que $m < n$, estaremos cometiendo un error en la representación de los datos. Como ventaja estaremos comprimiendo la información contenida en el vector inicial. El error cuadrático mínimo que podemos cometer se dará cuando tomemos una base ortogonal expandida a partir de los m autovectores $e_1, e_2, \dots, e_m$ de la matriz de covarianzas. Dicho error vendrá dado por:

$$J_{MSE}^{PCA} = \sum_{i=m+1}^n d_i \quad (5)$$

Si $m = n$ no se cometerá error alguno en la representación de los datos en la nueva base formada por los autovectores de la matriz de covarianzas de los datos. Este caso será de especial interés, ya que al aplicar un análisis PCA a los datos, conseguiremos decorrelacionarlos (Leñero & Cruces, 2005).

2.4.4 K-Means

2.4.4.1 Concepto de K-Means

K-means (MacQueen, 1967) es un método que persigue de manera sencilla clasificar un conjunto de datos dado y una cantidad de clúster a buscar (k clúster) (Luke, s.f.).

2.4.4.2 Proceso de K-Means

La idea principal es ir encontrando los centros del conjunto de datos e irlos desplazando hacia los centroides (MacQueen, 1967). Los centroides deben de colocarse de manera estratégica debido a que diferentes lugares generan diferentes resultados. Por lo tanto, la mejor manera de colocarlos es lo más lejos posible uno de otro (Luke, s.f.).

El siguiente paso es tomar cada punto dado en el conjunto de datos y asociarlo al centroide más cercano. Cuando ya no existan puntos por agrupar, se recalculan nuevos k-centroides de los clúster resultantes del paso anterior. Después se realiza una nueva asociación entre el mismo conjunto de datos y los nuevos centroides. Esto genera un ciclo hasta que ya no existan cambios, es decir hasta los centroides no se muevan.

Después se obtiene un error usualmente el Error Cuadrático Medio (ECM).

Este algoritmo se basa en 4 pasos los cuales son:

1. Colocar los K puntos en un espacio representado por los objetos que los clasificarán. Esos puntos representan el punto de partida de los centroides.
2. Asignar cada objeto al centroide más cercano.

3. Mientras los objetos se estén reasignando, recalcular la posición de los k -centroides.
4. Repetir los pasos 2 y 3 hasta que los centroides ya no se reubiquen.

Este proceso produce una separación de los objetos en grupos donde las métricas son minimizadas y pueden ser calculadas. (Lloyd, 1982)

2.4.5 ICA

2.4.5.1 Concepto de ICA

El ICA es una técnica pensada para obtener o extraer una representación de cada una de las distintas componentes o señales independientes entre sí que forman parte de una señal de mezcla. Veremos que esto será posible siempre que las distintas señales que componen la mezcla sean independientes entre sí; es decir, el valor de una de ellas no está relacionado con el de las otras, y que no más de una de ellas tenga una distribución de tipo gaussiana (Leñero & Cruces, 2005).

Entre las numerosas aplicaciones de ICA, nos centraremos en las aplicaciones para BSS (Separación ciega de fuentes) que consisten en la recuperación de un conjunto de señales a partir de mezclas lineales de ellas sin saber nada sobre las fuentes que queremos recuperar (Leñero & Cruces, 2005).

2.4.5.2 Proceso de ICA

Consideraremos que observamos m variables aleatorias en presencia de ruido a las que denotaremos como $x_1, x_2, \dots, x_m$. Estas variables aleatorias son el resultado de combinaciones de otras variables aleatorias, a las que denotaremos como $s_1, s_2, \dots, s_n$, que queremos obtener a partir de las variables que hemos observado. En general, podremos expresar cada una de las variables aleatorias que hemos observado de la forma:

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \dots + a_{in}s_n + n_i \text{ con } i = 1, \dots, m \quad (6)$$

Donde los coeficientes a_{ij} son coeficientes reales a priori desconocidos, con $i = 1, \dots, m$ y $j = 1, \dots, n$ y n_i es el ruido recibido en el i -ésimo receptor. En general, supondremos que las variables aleatorias s_i son estadísticamente independientes y que, como máximo, una de ellas tiene una distribución de tipo gaussiana. También supondremos que el número de variables aleatorias observadas cumple que $m > n$ en el caso de que las observaciones se hagan en presencia de ruido y que $m \geq n$ en ausencia de ruido.

En el presente documento se empleará notación vectorial para referirse a las fuentes y a las observaciones. Denotaremos como $\mathbf{x}$ al vector que contiene cada una de las mezclas observadas $x_1, x_2, \dots, x_m$, como $\mathbf{s}$ al vector que contiene cada una de las componentes independientes $s_1, s_2, \dots, s_n$ que queremos separar y como $\mathbf{n}$ al vector que contiene las fuentes de ruido recibidas en cada sensor. Asimismo denotaremos como $\mathbf{A}$ la matriz de mezclas cuyos elementos son los coeficientes a_{ij} . De forma compacta, podemos expresar el proceso de mezcla de la siguiente forma:

$$x = As + n \quad (7)$$

En la siguiente figura se ilustra gráficamente el proceso de mezcla y de estimación de las señales originales a partir de sus mezclas:

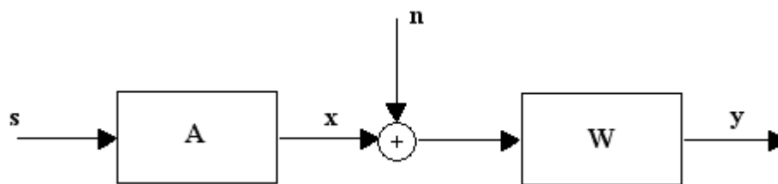


Figura 5. Esquema del proceso de mezclas y separación.

Nuestro objetivo será determinar las componentes independientes a partir de las observaciones. Para ello, trataremos de buscar una estimación de inversa de la matriz de mezclas W , que nos permita recuperar las señales originales a partir de las observaciones. Normalmente la matriz A será invertible, aunque si el número de observaciones es mayor que el de fuentes (tendremos redundancia) la matriz de mezclas no será cuadrada. En estos casos, la identificación de las fuentes a partir de las mezclas también es posible y nos limitaremos a buscar una matriz que al multiplicarla por la matriz de mezclas nos dé la matriz identidad (Leñero & Cruces, 2005).

2.4.6 MDS

2.4.6.1 Concepto de MDS

MDS no es exactamente un método para reorganizar objetos de una manera más eficiente con el fin de llegar a una configuración que mejor se aproxime a las distancias observadas. Este algoritmo de hecho mueve los objetos en un espacio definido a través de un número de dimensiones dadas, y verifica que tan bien se puede reproducir la muestra con la nueva configuración. Utilizando una función de minimización, la cual evalúa diferentes configuraciones para llegar a una meta ya sea por maximización o minimización de factor de ajuste (StatSoft, s.f.).

El MDS es un algoritmo que se enfoca en detectar las características más significativas que permitan al investigador explicar las distancias (d_i) similares de las muestras observadas. Con este método se puede analizar cualquier tipo de matriz de (d_i) similitud, además de matrices de correlación (StatSoft, s.f.).

La técnica de este método difiere de varias maneras con casi todos los otros métodos de ordenación. En los cuales se calculan varios ejes y solo se muestran algunos pocos. En cambio, con MDS se fija una cantidad exacta de ejes y se ajustan los datos a esas dimensiones para su análisis (Holland, 2008).

2.4.6.2 Proceso de MDS

En las técnicas de escalamiento multidimensional, el primer requisito trata con los conceptos de objeto-estímulo y de similitud-disimilitud-distancia y el segundo requisito se refiere al procedimiento para alcanzar una configuración de puntos que refleje

las similitudes observadas o percibidas (Linares, Escalamiento Multidimensional: Conceptos y Enfoques, 2001).

Las medidas de semejanza, como una aplicación de valores numéricos que permiten expresar numéricamente el vínculo existente entre estímulos, son aquí de importancia capital. Los conceptos de similitud, disimilitud y distancia, como medidas de semejanza, poseen propiedades específicas que deben tenerse en cuenta (Linares, 1990).

Se inicia el procedimiento con una matriz de (di) similitudes entre n objetos, con el elemento δ_{ij} , donde se considera i para las filas, j para las columnas y se fija el número de dimensiones p .

1) Se arreglan los n objetos en una configuración inicial en p dimensiones, donde para cada objeto se contienen las coordenadas $(x_1, x_2, \dots, x_p)$ en el espacio de p dimensiones.

2) Después se calculan las distancias normalmente la euclídea es utilizada los objetos de esta configuración, esto es, calcular las d_{ij} , que son las distancias entre el objeto i y el objeto j (Linares, 2001).

Donde es una distancia euclídea si $x = (x_1, \dots, x_p)'$ e $y = (y_1, \dots, y_p)'$ son dos observaciones de dimensión p (Sanchez, 2009).

La distancia euclidiana se calcula de la siguiente manera:

$$d(x, y) = \sqrt{\sum_{k=1}^p (x_k - y_k)^2} \quad (8)$$

3) Después se realiza una regresión de d_{ij} sobre δ_{ij} . Donde la regresión puede ser lineal, polinomial o monótona.

4) Es conveniente medir el facto de ajuste o índice de esfuerzo (stress) de las distancias de la configuración y las disparidades a través de algún método. Uno de los criterios más utilizados es el siguiente:

$$STRESS1 = \sqrt{\frac{\sum \sum (d_{ij} - \hat{d}_{ij})^2}{\sum \sum d_{ij}^2}} \quad (9)$$

Todas las sumatorias sobre i y j van de 1 a p y las disparidades dependen del tipo de regresión utilizado en el tercer paso del procedimiento.

El STRESS1 es la fórmula introducida por Kruskal quien ofreció la siguiente guía para su interpretación la cual se muestra en la tabla 1.

Tamaño del STRESS1	Interpretación
0.2	Pobre
0.1	Regular
0.05	Bueno
0.025	Excelente
0.00	Perfecto

Tabla 1. Tabla de equivalencias de Kruskal para el nivel de STRESS1

- 5) Las coordenadas $(x_1, x_2, \dots, x_t)$ de cada objeto se cambian ligeramente de tal manera que la medida de ajuste se reduzca.
- 6) Los pasos del 2 al 5 se repiten hasta que al parecer la medida de ajuste entre las disparidades y las distancias de configuración no puedan seguir reduciéndose. El resultado final del análisis es entonces las coordenadas de los n objetos en las p dimensiones (Linares, Escalamiento Multidimensional: Conceptos y Enfoques, 2001).

2.4.7 Mapeo de Sammon

2.4.7.1 Concepto de Mapeo de Sammon

El mapeo o proyección de Sammon trata de encontrar una proyección no lineal óptima para los datos en alta dimensión, de manera que los vectores que se proyectan en la

superficie bidimensional, conservan la misma distancia euclídea relativa entre ellos que la tenían en alta dimensión (Marín, s.f.).

La técnica consiste en la proyección no lineal (Non-linear mapping) de los datos procedentes de un espacio vectorial de dimensión elevada en otro espacio de dimensión más reducida. Habitualmente, por claridad de la representación, la reducción se limita a un simple par de variables. De esta forma, se obtiene una representación en un plano como resultado de la proyección. Para obtener una representación dimensional menor de los datos, se minimiza una función de las diferencias en las distancias entre puntos en el espacio original y aquellos en el espacio reducido.

Esta consideración particular tiende a preservar una estructura “local”, en el sentido de que cada punto en el espacio reducido, sostiene aproximadamente la misma relación con sus vecinos más cercanos como lo hizo en el espacio original, sus relaciones con puntos relativamente distantes en el espacio original, pueden ser un poco diferentes en el espacio reducido. Empezando con una configuración arbitraria de dos dimensiones, los puntos son ajustados, usando el procedimiento descendiente con más pendiente, para que el error sea minimizado. Esto da una configuración final de dos dimensiones, que conserva mucho de la estructura presente en el original de n dimensiones (Barreto, Ordieres, & Castejón, 2009).

Cabe mencionar que esta técnica es un proceso aleatorio, es decir consigue un punto al azar y de ahí se calculan las distancias entre todos los puntos. La buena praxis en los procesos aleatorios es obtener un promedio corriendo el proceso varias veces.

2.4.7.2 Proceso de Mapeo de Sammon

Se denota la distancia entre los objetos i-ecimos y j-ecimos en el espacio original, y la distancia entre sus proyecciones por d_{ij} . El mapeo de Sammon tiene como objetivo minimizar la siguiente función de error, que se conoce como el estrés de Sammon o de error de Sammon menudo:

$$E = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} \quad (10)$$

La minimización se puede realizar ya sea por descenso de gradientes tal como se propone inicialmente, o por otros medios, por lo general implican métodos iterativos. El número de iteraciones se determina a manera de prueba-error y las soluciones convergentes no siempre están garantizadas. Muchas implementaciones prefieren utilizar los primeros componentes principales como una configuración inicial (ver punto 2.4.3.2) (Sammon, 1969).

2.4.8 Redes Neuronales

2.4.8.1 Neurona Biológica

La neurona biológica está formada por tres partes (ver figura 6):

1. Soma o núcleo celular: es el núcleo de la célula que dirige la actividad de la misma.
2. Axón: es una prolongación muy larga que permite transmitir el impulso eléctrico producido por la soma a otras neuronas.
3. Dendritas: Son pequeñas prolongaciones mediante las cuales se reciben los impulsos eléctricos de otras neuronas (Cota, 2010).

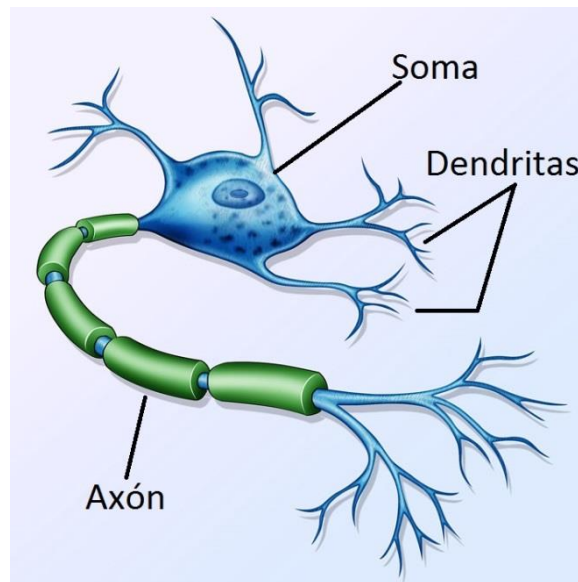


Figura 6. Esquema de una neurona natural.

Sinapsis es la conexión entre axones y dendritas que permite la comunicación con otras neuronas. Una neurona es sometida a través de sus dendritas a muchos estímulos eléctricos, cuando la suma de todos los estímulos eléctricos recibidos de sus dendritas es lo suficientemente grande se produce la activación de la neurona y su consiguiente emisión eléctrica a través de su axón.

De esta forma una neurona natural es un sistema que procesa varias entradas que le llegan a través de las dendritas, las suma y si el resultado supera el umbral se activa y produce una salida que la lleva al axón, y éste a las siguientes neuronas (González D. , 2004).

2.4.8.2 Neurona Artificial

Una neurona artificial se puede definir como un dispositivo que a partir de un conjunto de entradas $X_i, i = 1, \dots, n$ (o vector x), genera una única salida Y (binaria). Cada entrada X_i está asociada con un peso w_i que, representa la importancia que le damos a esa entrada, por lo tanto también tenemos como entrada el vector w . Además es necesario conocer el umbral θ por el que se activará o no la neurona (Cota, 2010).

Donde la Y resultante cumple con la siguiente relación funcional:

$$Y = f(xw - \theta) = f\left(\sum x_i w_i - \theta\right) \quad (11)$$

Al comparar la suma de los productos de las con el umbral de activación, si la resta da un resultado positivo la neurona se activa y si es negativo la neurona no se activa. El umbral puede ser considerado como una entrada más X_0 , en la que su peso w_0 tiene un valor -1, así se simplificar la fórmula de la siguiente manera (Cota, 2010) :

$$Y(x w) = f\left(\sum x_i w_i\right) \quad (12)$$

Las RN representan una técnica de modelación matemática, que intenta imitar el proceso de aprendizaje que ocurre en el sistema nervioso (Freeman & Skapura, 1991).

Las RN se basan en una estructura de neuronas unidas por enlaces que transmiten información a otras neuronas, las cuales entregan un resultado mediante funciones matemáticas. Las RN aprenden de la información histórica a través de un entrenamiento, proceso mediante el cual se ajustan los parámetros de la red, a fin de entregar la respuesta deseada, adquiriendo entonces la capacidad de predecir respuestas del mismo fenómeno. El comportamiento de las redes depende entonces de los pesos para los enlaces, de las funciones de activación que se especifican para las neuronas, que pueden ser de tres categorías: lineal, de umbral (o escalón) y sigmoidea, y de la forma en que propagan el error (Freeman & Skapura, 1991).

La red neuronal debe ser sometida a un proceso de aprendizaje antes de su utilización, en dicho proceso los pesos de cada una de sus neuronas se modifican para dar más relevancia a unas entradas frente a otras y conseguir la especialización de cada neurona que forma la red neuronal. Este ajuste de los pesos debe hacerse basándose en alguna regla de aprendizaje, que se plasma en un algoritmo de aprendizaje, el cual determina el procedimiento numérico de ajuste de los pesos (Cota, 2010).

2.4.8.3 Retroalimentación multi capa con propagación hacia atrás

Las RMC o MLF (por sus siglas en ingles Multi-Layer Feed-forward) entrenadas con el algoritmo de aprendizaje de propagación hacia atrás son las más populares. Estas son aplicadas ampliamente a problemas relacionados a la química (Zupan & Gasteiger, 1993).

Una red neuronal MLF consta de neuronas, que se ordenan en capas (figura 7). La primera capa se denomina capa de entrada, la última capa se denomina capa de salida, y las capas están entre las capas ocultas.

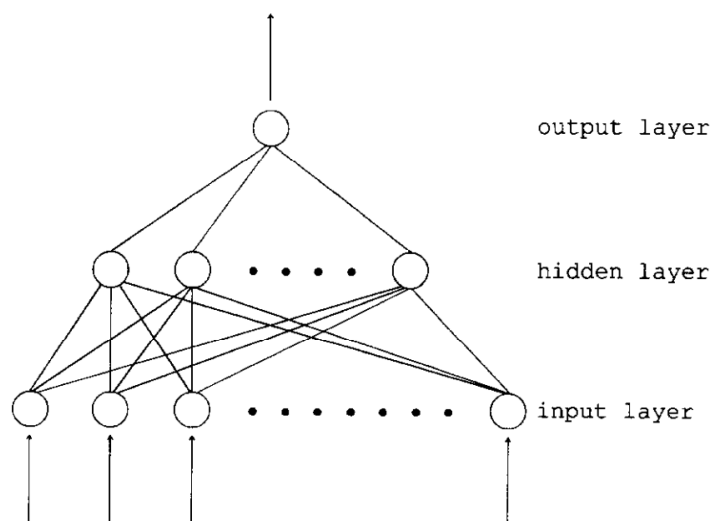


Figura 7. Red Neuronal Feed-Forward de tres capas típica.

Para una descripción formal de las neuronas podemos utilizar la función de mapeo Γ , que asigna a cada neurona i a un subconjunto $\Gamma(i) \subseteq V$, que consiste en todos los antepasados de la neurona determinada. Un subconjunto $\Gamma^{-1}(i) \subseteq V$ que consta de todos los predecesores de la neurona i .

Cada neurona en una capa particular está conectada con todas las neuronas de la capa siguiente. La conexión entre la neurona i -ésima y j -ésima se caracteriza por el coeficiente de ponderación ω_{ij} , y la neurona i -ésima por el coeficiente de umbral $\vartheta(i)$ (Svozil, Kvasnička, & Pospíchal, 1997), ver (figura 8).

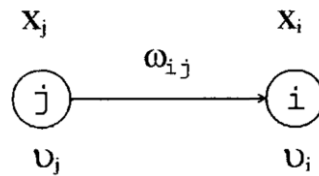


Figura 8. Conexión entre dos neuronas, i y j .

El coeficiente de ponderación refleja el grado de importancia de la conexión dado en la red neuronal. El valor de la producción (actividad) de la i -ésima neurona x_i está determinado por las ecuaciones (13) y (14). Sostiene que:

$$x_i = f(\xi_i) \quad (13)$$

$$\xi_i = \vartheta_i + \sum_{j \in \Gamma_i^{-1}} \omega_{ij} x_j \quad (14)$$

Donde ξ_i es el potencial de la neurona i -ésima y la función $f(\xi_i)$ es la denominada función de transferencia (la suma en la ecuación (14). se lleva a cabo sobre todas las neuronas j transferir la señal a la neurona i -ésima). El coeficiente de umbral puede ser entendido como un coeficiente de ponderación de la conexión con el añadido de formalmente neurona j , donde $x_j = 1$ (el llamado sesgo).

Para obtener la función de transferencia se cumple que:

$$f(\xi) = \frac{1}{1 + \exp(-\xi)} \quad (13)$$

El proceso de adaptación supervisada varía el umbral de coeficientes de ϑ_i y el peso coeficientes ω_{ij} para reducir al mínimo la suma de las diferencias al cuadrado entre los valores de salida calculados y requeridos. Esto se logra mediante la reducción al mínimo de la función objetivo E :

$$E = \sum_o \frac{1}{2} (x_o - \hat{x}_o)^2 \quad (14)$$

Donde x_o y $\hat{x}_o$ son los vectores formados por las actividades calculados y se requiere de las neuronas de salida y la suma se extiende sobre todos o salida de las neuronas.

2.4.8.4 Algoritmo de entrenamiento de propagación hacia atrás

En el algoritmo de propagación hacia atrás se utiliza el método de minimización más empinada-descenso. Para el ajuste de los coeficientes de ponderación y el umbral se cumple que:

$$\omega_{ij}^{k+1} = \omega_{ij}^k - \lambda \left(\frac{\partial E}{\partial \omega_{ij}} \right)^k \quad (15)$$

$$\vartheta_i^{k+1} = \vartheta_i^k - \lambda \left(\frac{\partial E}{\partial \vartheta_i} \right)^k \quad (16)$$

Donde λ es la tasa de aprendizaje $\lambda > 0$. Para un análisis más elemental del cálculo de las derivadas ver el *Introduction to multi-layer feed-forward neural networks* (Svozil, KvasniEka, & Pospichal, 1997).

Basado en el enfoque dado por encima de los derivados de la función objetivo para la capa de salida y, a continuación de las capas ocultas se pueden calcular recurrentemente. Este algoritmo se llama la propagación hacia atrás, debido a que el error de salida se propaga desde la capa de salida a través de las capas ocultas a la capa de entrada.

2.4.8.5 Entrenamiento y generalización

La red neuronal MLF funciona en dos partes:

- a) Modo de entrenamiento
- b) Modo de predicción

Para el entrenamiento y para la predicción de la red neuronal MLF se necesitan dos conjuntos de datos, el conjunto de entrenamiento y el conjunto que queremos predecir (de prueba).

El entrenamiento comienza con valores arbitrarios de los pesos (los cuales pueden ser números aleatorios) y procede de manera iterativa. Cada iteración completada del conjunto de entrenamiento es llamado *época* (Svozil, KvasniEka, & Pospichal, 1997). La red ajusta los pesos en cada época en dirección de reducir el error (véase punto 2.4.8.4). A medida que continúa el proceso iterativo de ajuste por incrementos, los pesos

convergen gradualmente a un conjunto local óptimo de valores. Suelen ser necesarias muchas épocas antes de completar el entrenamiento.

Para un conjunto de entrenamiento dado, de aprendizaje de propagación hacia atrás puede proceder en una de dos formas básicas: modo de patrón y el modo por lotes.

En el modo de patrón de aprendizaje propagación hacia atrás, es evaluada la actualización de pesos antes de mostrar cada patrón de entrenamiento. En el modo continuo de aprendizaje de propagación hacia atrás, es evaluada la actualización de pesos antes de mostrar los ejemplos de entrenamiento (i.e., después de toda la época).

Desde un punto de vista 'en-línea', se prefiere el modo de patrón sobre el modo por lotes, ya que requiere menos almacenamiento local para cada conexión sináptica. Sin embargo, dado que los patrones se presentan a la red de una manera aleatoria, la actualización de los pesos patrón por patrón, hace la búsqueda se vuelva estocástica en relación a los pesos, lo que hace que sea menos probable a ser atrapado en un mínimo local en el algoritmo de propagación hacia atrás (Svozil, KvasniEka, & Pospichal, 1997).

Por otro lado, el uso del entrenamiento por lotes proporciona una estimación más precisa del vector gradiente. El entrenamiento por patrones se utiliza, por ejemplo, en el control de procesos en línea, debido a que no son todos los patrones de formación disponibles en el momento.

En el análisis final de los dos modos de entrenamiento depende de la eficiencia relativa de del problema resuelto tal como lo explican en *Introduction to the Theory of Neural* (Computation, Hertz, et. al., 1991) y *Neuro-Dynamic Programming* (Bertsekas & Tsitsiklis, 1996).

En el modo de predicción, la información fluye hacia delante a través de la red, a partir de las entradas a las salidas. La red procesa un ejemplo a la vez, produciendo una estimación del(os) valor(es) de salida sobre la base de los valores de entrada. El error resultante se utiliza como una estimación de la calidad de la predicción de la red entrenada.

En el aprendizaje de propagación hacia atrás, por lo general se comienza con un conjunto de entrenamiento y es utilizado el algoritmo backpropagation para calcular los pesos sinápticos de la red. La intención es que la red neuronal diseñada pueda ser generalizará. Una red se dice que se encuentra bien generalizada cuando la relación entrada-salida calculada por la red es correcta (o casi correcta) por patrones de entrada / salida no utilizados en el entrenamiento de la red (Svozil, KvasniEka, & Pospichal, 1997).

La generalización de las redes neuronales no es una propiedad mística, pero puede ser comparada con el efecto de una buena interpolación no lineal de los datos de entrada (Wieland & Leighton, 1987). El principio de generalización es mostrado en la figura 9.

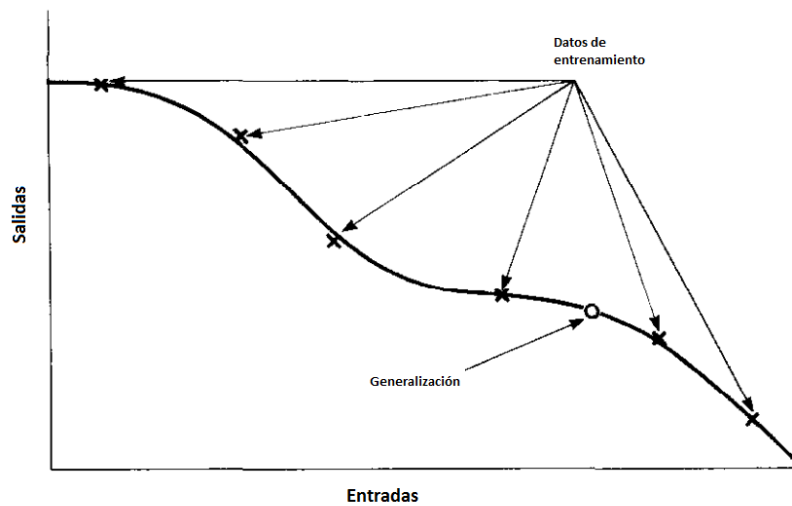


Figura 9. Principio de generalización de una red.

Cuando el proceso de aprendizaje se repite demasiadas veces (i.e. la red neuronal esta sobreentrenada o sobre ajustada, entre sobre entrenamiento y sobreajuste hay diferencia), la red puede memorizar los datos de entrenamiento y por lo tanto será menos capaz de generalizar entre patrones similares de entrada y salida. La red da resultados casi perfectos para ejemplos del conjunto de entrenamiento, pero no para los ejemplos del conjunto de prueba. El sobreajuste puede ser comparado a una elección inapropiada de grados de polinomios en una regresión polinomial (figura 10). Puede suceder un grave sobreajuste con los datos ruidosos, aunque hay muchos más casos de entrenamiento que de pesos (Svozil, KvasniEka, & Pospichal, 1997).

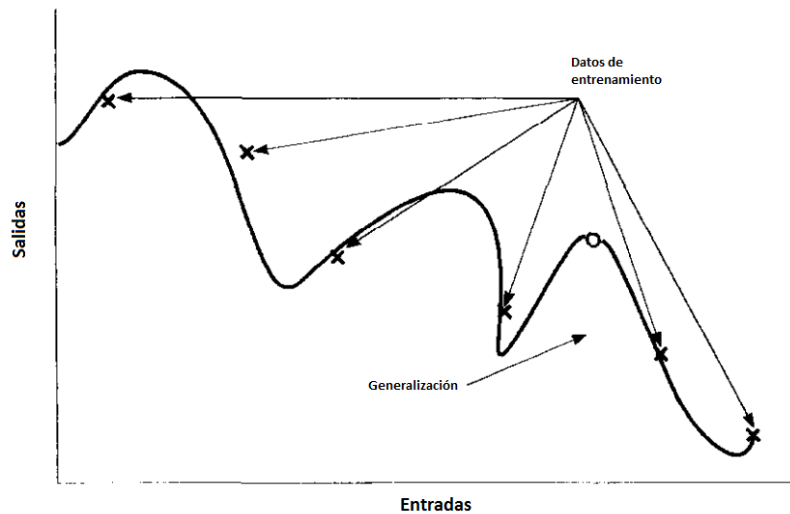


Figura 10. Generalización de una red sobre ajustada o sobreentrenada.

La condición básica para una buena generalización es suficientemente amplio conjunto de los casos de entrenamiento. Este conjunto de entrenamiento debe estar en el mismo tiempo el subconjunto representativo del conjunto de todos los casos que se desea generalizar. La importancia de esta condición está relacionada con el hecho de que hay dos tipos diferentes de generalización: de interpolación y extrapolación (Svozil, KvasniEka, & Pospichal, 1997).

La interpolación se aplica a los casos en que están más o menos rodeados por casos de entrenamiento cercanos; todo lo demás es extrapolación. En particular, los casos que están fuera del rango de los datos de entrenamiento requieren extrapolación. La interpolación puede hacerse a menudo fiable, pero la extrapolación es muy poco fiable. Por lo tanto, es importante contar con suficientes datos de entrenamiento para evitar la

necesidad de extrapolación. Los métodos para la selección de buenos conjuntos de entrenamiento surgen de diseño experimental (Wu, et al., 1996).

Para un análisis más elemental de sobreajuste, ver *Networks for Statistical Modelling* (Smith, 1993) y para un enfoque más riguroso, ver el artículo de *Neural networks and the bias / variance dilemma* (Geman, Bienenstock, & Doursat, 1992). Dada una cantidad fija de datos de entrenamiento, hay algunos métodos eficaces para evitar el sobreajuste, y por lo tanto conseguir una buena generalización, ver artículo *Introduction to multi-layer feed-forward neural networks* (Svozil, Kvasnička, & Pospíchal, 1997).

Capítulo III

3 Metodología

En este capítulo se explicaran las cuatro fases en las que se compuso el proyecto, las cuales son las siguientes:

1. Desarrollo y alimentación de la base de datos
2. Implementación de tecnologías de software a la aplicación GvSIG
3. Implementación de herramientas de la IA para el uso de energía eólica en baja california.
4. Implementación de las herramientas anteriores dentro de un SIG.

3.1 Localización del área de estudio

Baja California está ubicado en el extremo noroeste de México. Comparte la península de Baja California con el estado mexicano de Baja California Sur. Tiene un área de 71.576 km², es decir, el 3,57% del país. Limita al oeste con el océano Pacífico, al este con el Mar de Cortés o Golfo de California, el estado de Sonora y el estado estadounidense de Arizona, al sur con el estado de Baja California Sur y al norte con el estado estadounidense de California (INEGI, 2010).



Figura 11. Ubicación geográfica del estado de Baja California y su capital Mexicali. Fuente: Anuario estadístico de Mexicali, 2008.

3.2 Desarrollo y alimentación de la base de datos

Se recabo información sobre las condiciones climatológicas del estado de Baja California de distintas instituciones. Se obtuvo información del 2000 al 2010 de la estación climatológica Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), del año 2007 de la estación climatológica Comisión Nacional del Agua (CONAGUA), y del año 2006 y 2007 de la estación climatológica de la Universidad Autónoma de Baja California (UABC).

La base de datos fue creada en MySQL y nombrada como SIGER_EO (Sistema de Información Geográfica de Energías Renovables – Eólica), se encuentra normalizada en la 3ra forma normal según (Ullman, 2002) aplicando las reglas descritas en (Valera,

2003) en un modelo relacional como se muestra en la tabla 2 y su Diagrama entidad-relación mostrado en la figura 12.

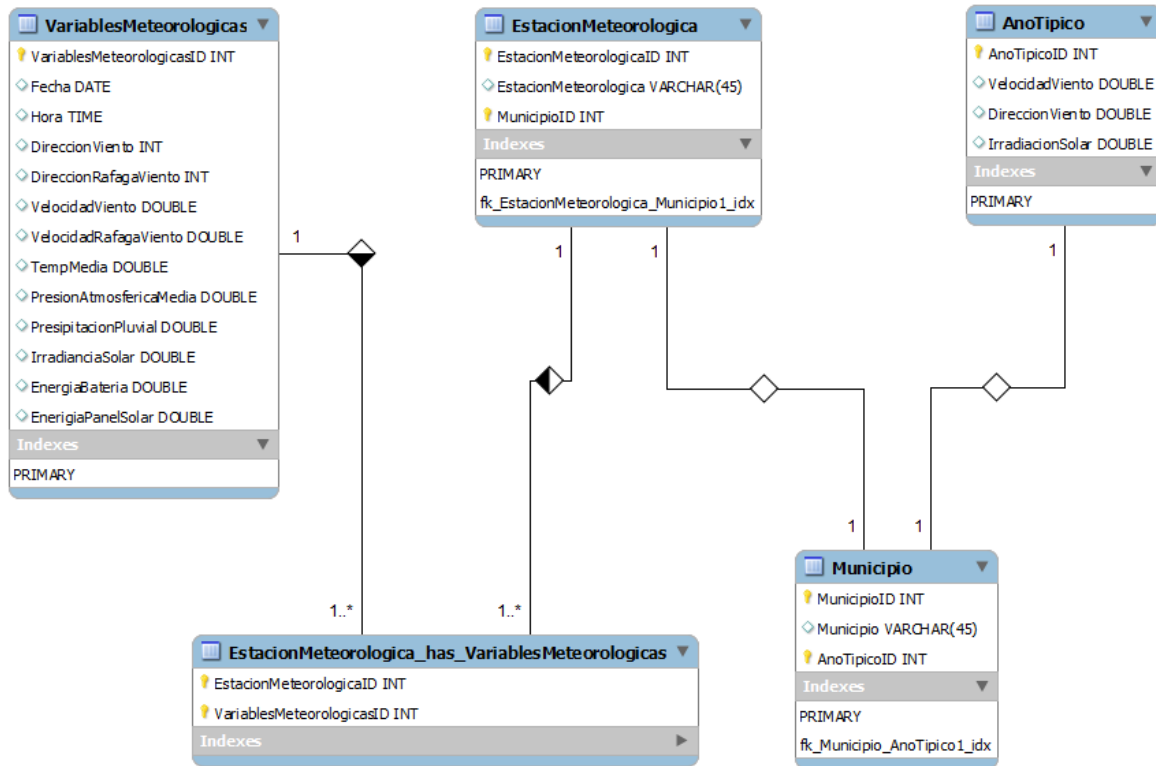


Figura 12. Diagrama E-R de la base de datos SIGER_EO.

Modelo relacional de la SIGER_EO

VariablesMeteorologicas (VariablesMeteorologicasID, Fecha, Hora, DireccionViento, DireccionRafagaViento, VelocidadViento, VelocidadRafagaViento, TempMedia, PresionAtmosfericaMedia, PresipitacionPluvial, IrradianciaSolar, EnergiaBateria, EnergiaPanelSolar)

EstacionMeteorologica (EstacionMeteorologicaID, EstacionMeteorologica, MunicipioID)

AnoTipico (AnoTipicoID, VelocidadViendo, DireccionViento, IrradianciaSolar)

Municipio (MunicipioID, Municipio, AnoTipicoID)

EstacionMeteorologica_has_VariablesMeteorologicas (EstacionMeteorologicaID, VariablesMeteorologicasID)

Tabla 2. Modelo relacional de la BD.

3.3 Implementación de herramientas de software a una aplicación SIG de código abierto.

Se utilizaron las API por sus siglas en inglés *Application Programming Interface* (*Interfaz de Programación de Aplicaciones*) de librería de terceros para enlazar las distintas aplicaciones del software de licencia y libre al software gvSIG versión 2.0.0, la cual será la aplicación SIG utilizada. GvSIG tendrá acceso a los software desde su interfaz, de tal manera que no será necesario ir a abrir el software directamente sino que se podrán realizar funciones básicas cada software integrado, trabajando de manera directa sobre los archivos, y las funciones más específicas de cada software se realizarán lanzando el software especificado desde gvSIG, esto evita que se manipulen varios software a la vez. Realizando los procesos de manera interna.

La tecnología de software que se integró a GvSIG fueron las siguientes:

1. MATLAB
2. MICROSOFT OFFICE
 - a. MICROSOFT WORD
 - b. MICROSOFT EXCEL
 - c. MICROSOFT ACCESS
3. TRNSYS
4. WRPLOT
5. R
6. SCILAB

7. OPENOFFICE.ORG

- a. WRITER
- b. CALC

8. MYSQL

Se integraron TRNSYS y WRPLOT para tener una función más especializada sobre las ER y un uso especializado en la Energía Eólica. Además se crearon módulos extra para el desarrollo de estas funcionalidades, estos módulos fueron creados en java, los cuales se señalan a continuación.

- 1. Editor Simple
- 2. Editor RTF
- 3. Console
- 4. TableEditor

El lenguaje de programación que se utilizará es Java, lenguaje que deriva en gran parte de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede correr en cualquier máquina virtual Java (JVM), sin importar la arquitectura de la computadora. Java es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo

que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados (Wikipedia, s.f.).

Para una información mas detallada sobre el lenguaje referirse a *Introducción a la programación orientada a objetos con Java*(Thomas, 2001).

A continuación se presenta la metodología para la adaptación de cada una de las tecnologías antes mencionadas dentro de GvSIG.

3.3.1 Estructura general de las herramientas integradas

La versión del sistema fue bautizada como GvSIG-HIIAEREO, nombre recibido por la funcionalidad que desempeña.

La estructura organizacional de la extensión del sistema se encuentra separada por paquetes los cuales contienen subpaquetes del sistema o clases las cuales constituyen la funcionalidad del software tal como se muestra en la figura 13.

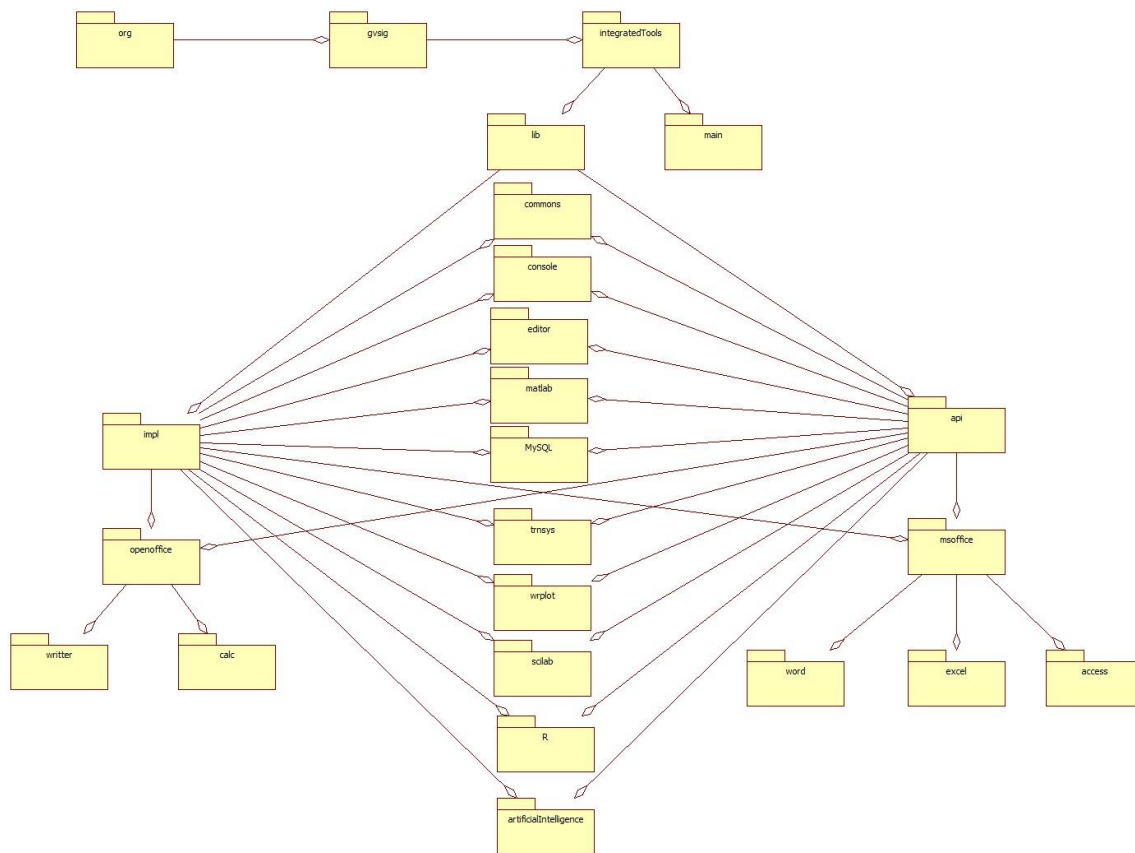


Figura 13. Descripción de la estructura organizacional de GvSIG-HIIAEREO.

La jerarquía de la estructura inicia con el paquete org, este paquete contiene clases nativas del armazón andami, que a su vez contiene el paquete gvsig. El paquete GvSIG contiene las clases propias del sistema principal, que además contiene el paquete IntegratedTools. Este paquete es el nodo “raíz” de la versión de GvSIG que se está desarrollando.

IntegratedTools contiene clases demo para poder ejecutar el sistema en modo de prueba antes de liberar versión al principal, además de este paquete se desprenden el paquete lib y main. En la paquetería main se encuentran alojadas clases y un pequeño GUI (Graphic

User Interface o Interfaz Grafica de Usuario) que permiten lanzar la versión de prueba del la versión. La paquetería lib se divide en dos apartados, impl de implementación, y api definición dada en el punto 3.3. El paquete impl se encarga de contener las clases que se mostraran en el GUI y el paquete api se encarga de contener las clases.

Cada ramificación (impl y api) contiene la misma cantidad de paquetes internos, los cuales son: commns, console, editor, matlab, mysql, trnsys, wrplot, scilab, r, artificialintelligence, msoffice y openoffice. Los paquetes msoffice y openoffice contienen los paquetes Word, Excel, Access para el paquete msoffice y, writter y calc para oppenoffice.

3.3.2 Commons

Commons (ver diagrama de clases, figura 14) es una paquetería que implementa las clases ConexionSQL, SetupJButton, SetupJMenuItem, RegisteredIcon, SetupJSeparator, FileInMemory, FileEditButton. Estas clases controlan el acceso a los datos, los archivos abiertos en *runtime*, así como iconos y configuración de los controles que son llamados por cada módulo integrado al sistema.

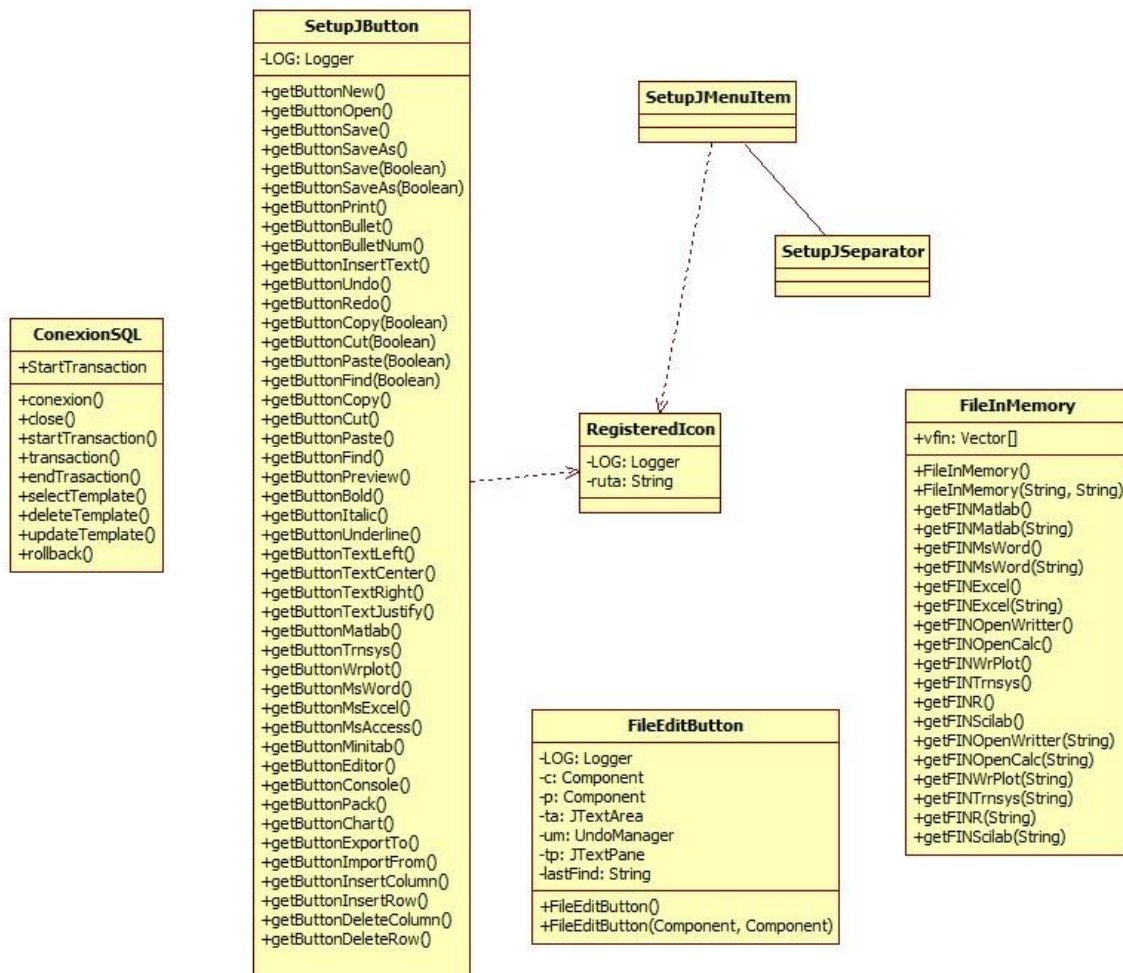


Figura 14. Diagrama de clases de la paquetería Commons.

3.3.2.1 ConexionSQL

Esta clase contiene los métodos que se encargan de controlar los accesos a las bases de datos a través de rutinas simplificadas para un manejo más dinámico sobre las órdenes de ejecución.

Por ejemplo:

```
public string selectTemplate(bool condicion, bool orderby, bool distinct) {  
    String selectTemplate = "SELECT [FIELDS] FROM [TABLE]";  
    if (condicion)  
        selectTemplate += " WHERE [CONDITION]";  
    if(orderby)  
        selectTemplate += " ORDER BY [ORDER_FIELDS]";  
    if (distinct)  
        selectTemplate = selectTemplate.replace("[FIELDS]", "DISTINCT [FIELDS]");  
    return selectTemplate;  
}
```

Este metodo recibe una serie de criterios los cuales forman dinámicamente la cadena de para consulta a la base de datos, que va variando su complejidad dependiendo de los criterios dados. Esta forma permite crear cadenas tan complejas que incluyan comandos como Union, Union All, subselect o joins.

3.3.2.2 SetupJButton

Esta clase contiene las definiciones de las propiedades del control JButton para cada funcionalidad utilizada en el sistema.

Por ejemplo:

```
public JButton getButtonNew(){  
    JButton bt = new JButton();  
    bt.setIcon(new ImageIcon(new RegisteredIcon().getIconNewFile(1)));  
    bt.setToolTipText("Crear nuevo archivo...");  
    return bt;  
}
```

JButton es un control del lenguaje de programación Java que dibuja y controla las acciones y propiedades de dicho control.

3.3.2.3 SetupJMenuItem

Esta clase contiene las definiciones de las propiedades del control JMenuItem para cada funcionalidad utilizada en el sistema.

Por ejemplo:

```
public JMenuItem getMenuitemMatlab(){  
    JMenuItem mi = new JMenuItem("Matlab (R2011a)");  
    mi.setIcon(new ImageIcon(new RegisteredIcon().getIconMatlab(2)));  
    return mi;  
}
```

JMenuItem es un control del lenguaje de programación Java que dibuja y controla las acciones y propiedades de dicho control.

3.3.2.4 SetupJSeparator

Esta clase contiene el método con las características personalizadas del sistema que define las propiedades del control JSeparator.

Por ejemplo:

```
public JSeparator getSeparator(){  
    JSeparator s = new JSeparator();  
    s.setBorder(null);  
    return s;  
}
```

JSeparator es un control del lenguaje de programación Java que dibuja y controla las acciones y propiedades de dicho control.

3.3.2.5 RegisteredIcon

Esta clase se encarga de agregar los iconos registrados a GvSIG. Los iconos que se encuentren dentro serán los mostrados en la interfaz de usuario. Cualquier otro icono colocado en otro punto será discriminado por el sistema.

Por ejemplo:

```
public String getIconMatlab(int size){  
    if(size==1)  
        return ruta+"matlab16.png";  
    if(size==2)  
        return ruta+"matlab24.png";  
    return "";  
}
```

3.3.2.6 FileEditButton

Esta clase se encarga de controlar los eventos (cortar, copiar, pegar, deshacer, rehacer, etc.) de los editores desarrollados, es decir, para el table editor, simple editor y editor rtf.

Por ejemplo:

```
public void actionCopy(){  
    if(ta!=null)  
        ta.copy();  
    if(tp!=null)  
        tp.copy();  
}
```

Donde el metodo actionCopy identifica el control sobre el cual debe trabajar a través de otro metodo constructor que se define previamente, llamado FileEditButton.

3.3.2.7 FileInMemory

Esta clase se encarga de gestionar la persistencia de la ruta del archivo durante la ejecución del proceso, es decir, mientras se trabaja con la utilería actual (matlab, Access, Word, etc.).

Por ejemplo:

```
public String getFINMsExcel(){  
    return (String)vfin[1].elementAt(vfin[2].indexOf("ms-excel"));  
}
```

3.3.3 Administrador de la Interfaz de Usuario Personalizada

La clase CustomUIManager (ver figura 15) se encarga de detectar y establecer la interfaz del sistema operativo configurada por el usuario a través de la clase que identifica el LookAndFeel.



Figura 15. Diagrama de clases de la paquetería Custom UI Manager.

Codigo:

```
public void useUIManager(){
    try {

        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

    } catch (ClassNotFoundException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    } catch (InstantiationException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    } catch (IllegalAccessException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    } catch (UnsupportedLookAndFeelException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }
}
```

3.3.4 Editor RTF

Es una interface desarrollada (ver figura 16) para implementar las funcionalidades básicas de este tipo de procesadores de texto, este editor toca el formato de texto dándole mas vista al documento.

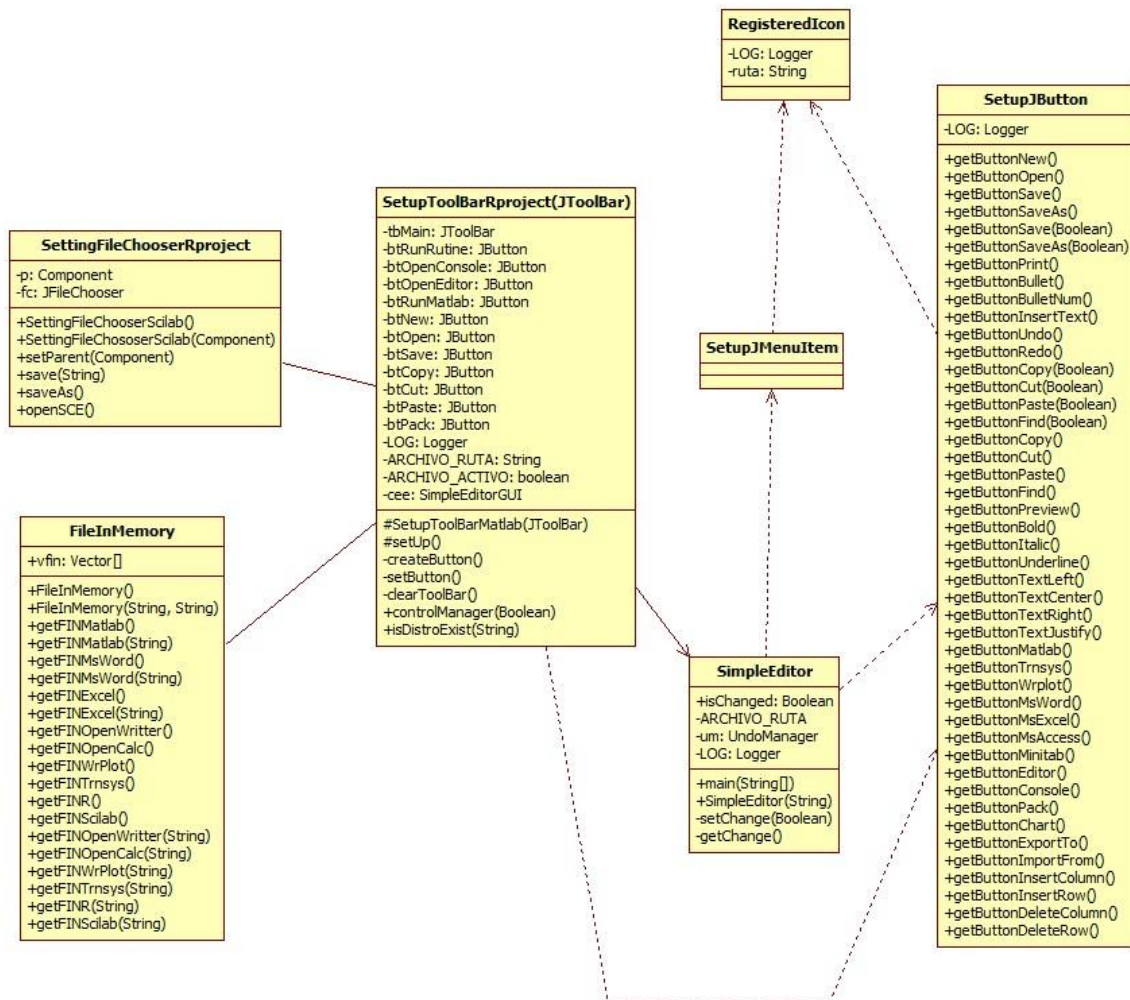


Figura 16. Diagrama de clases del Editor RTF.

3.3.5 Table Editor

Es una interface desarrollada (ver figura 17) para implementar las funcionalidades básicas de este tipo de procesadores de hojas de calculo, además este editor toca funcionalides de texto enriquecido.

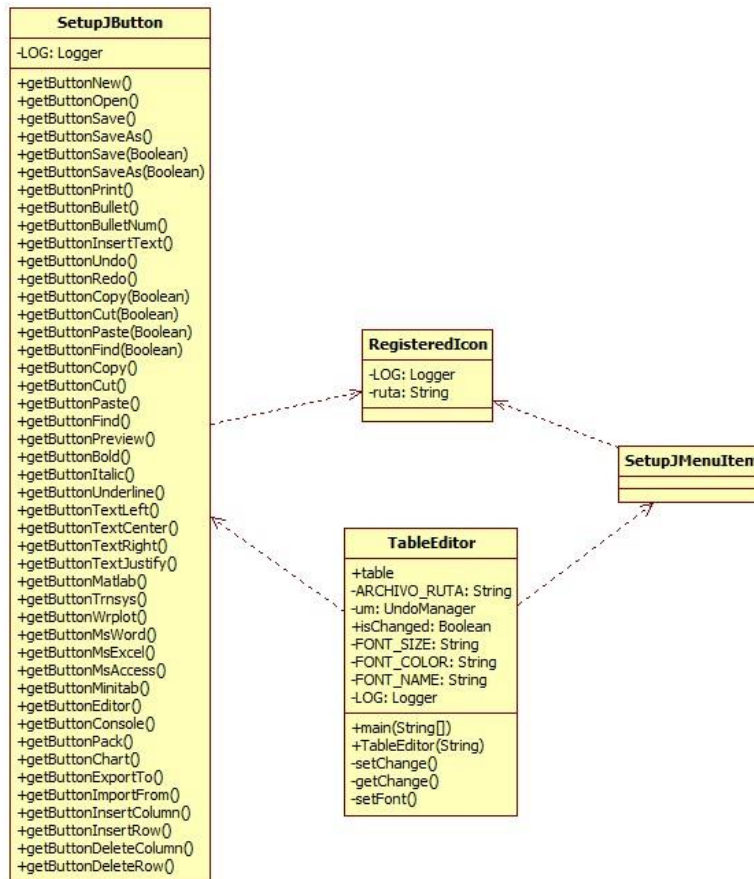


Figura 17. Diagrama de clases del Editor de Tablas.

3.3.6 Editor Simple

Es una interface desarrollada (ver figura 18) para implementar las funcionalidades básicas de procesamiento de texto, este editor fue desarrollado con la finalidad de tener un componente ligero para los modulos que solo requieran generar scripts (i.e. matlab, mysql).

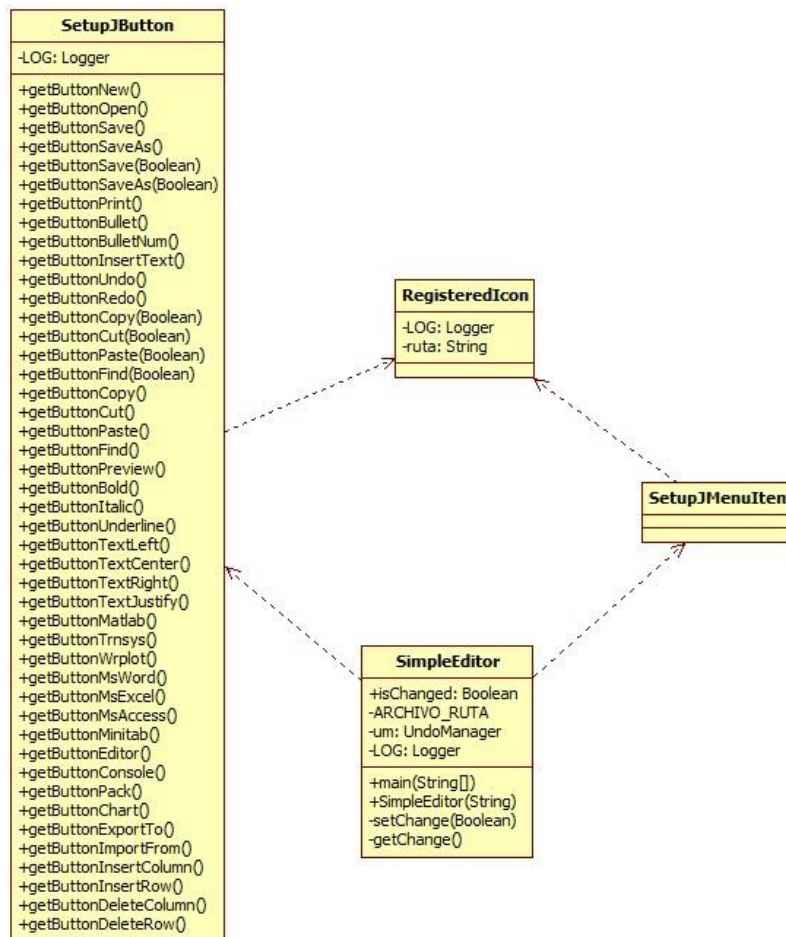


Figura 18. Diagrama de clases del Editor Simple

3.3.7 Console

Es una interface desarrollada (ver figura 19) para comunicar en modo consola GvSIG con las aplicaciones Matlab y Scilab.

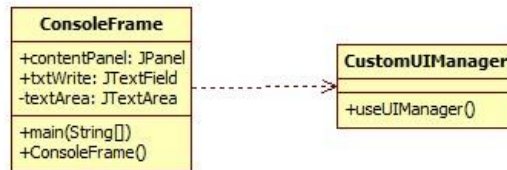


Figura 19. Diagrama de clases de la consola.

Para Matlab se utilizo la librería JMatLink que permite hacer un puente entre estas dos interfaces y para Scilab se utilizo la librería Javasci.

Ejemplo de enlace a Matlab:

```

public void ejecutarScript(String script){
    JMatLink engine = new JMatLink();
    engine.engOpen();
    engine.engEvalString(script);
}
  
```

Ejemplo de enlace a Scilab:

```

public void ejecutarScript(String script){
    Scilab sci = new Scilab();
    sci.open();
    sci.exec(script);
}
  
```

3.3.8 Matlab

Para este sistema se implementó la funcionalidad de la comunicación con el software Matlab versión R2010a, este software es un sistema de aplicación matemática, basado en el lenguaje M, el entorno de esta aplicación se basa en el desarrollo de scripts, los cuales permiten el manejo de algoritmos complejos, que en otros motores resulta complicada su ejecución. Este modulo utilizó el EditorSimple descrito en el punto 3.3.6.

La organización del apartado para matlab se puede apreciar en el diagrama de clases de la figura 20.

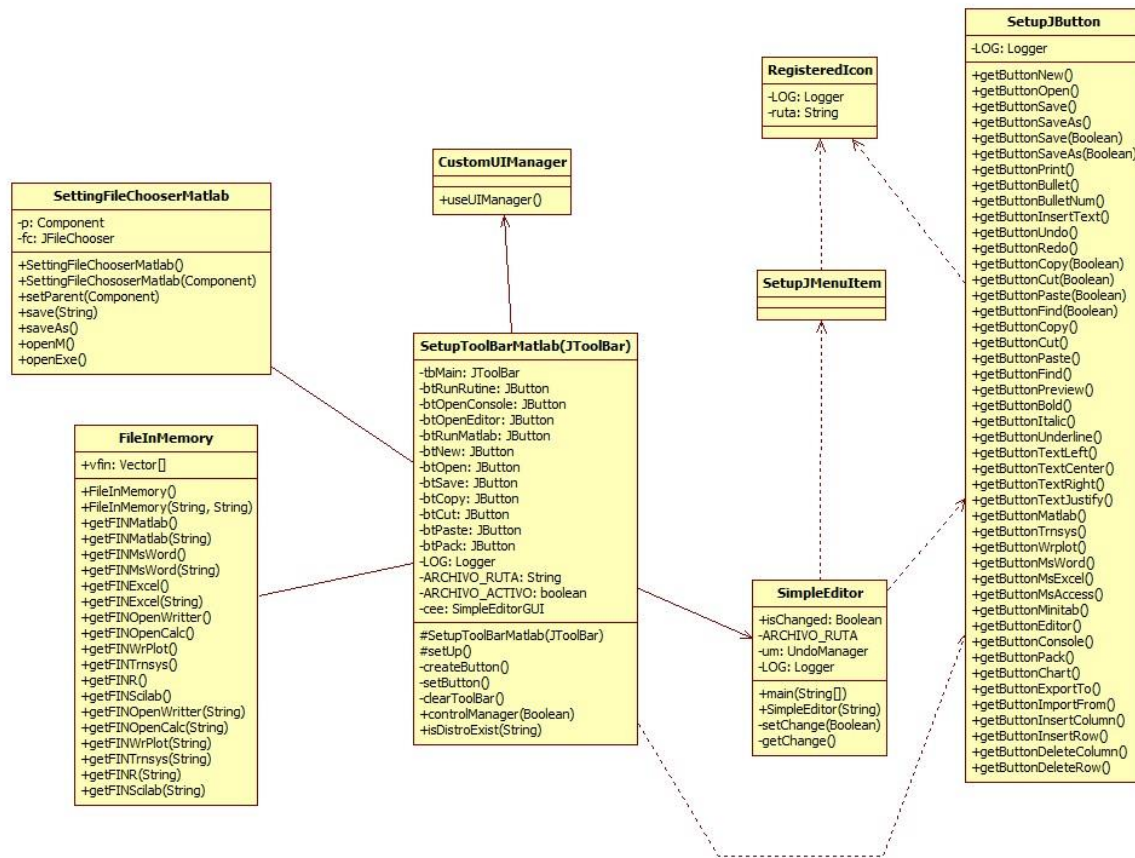


Figura 20. Diagrama de Clases para el módulo de Matlab.

En la figura 20 se puede apreciar que el apartado de matlab se compone principalmente de dos clases SettingFileChooserMatlab y SetupToolBarMatlab. Esta última clase se puede apreciar que se encuentra asociada con la clase SimpleEditorGUI y depende directamente de SetupJButton.

3.3.9 Microsoft Office

3.3.9.1 Microsoft Word

Para este sistema se implementó la funcionalidad de comunicación con el software Microsoft Office 2007, este software es un procesador de texto, esto es, un programa que utiliza un editor de texto para dar formato. Se puede dar ya sea en formato de texto plano, o formato de texto enriquecido.

Se utilizó el Editor de RTF explicado en el punto 3.3.4. Además se implementaron las librerías de Apache POI, de las cuales específicamente se utilizaron las clases XWPFWordExtractor, IBodyElement, XWPFDocument, para la extracción de la información del documento, interpretación y colocación del texto dentro del editor RTF desarrollado.

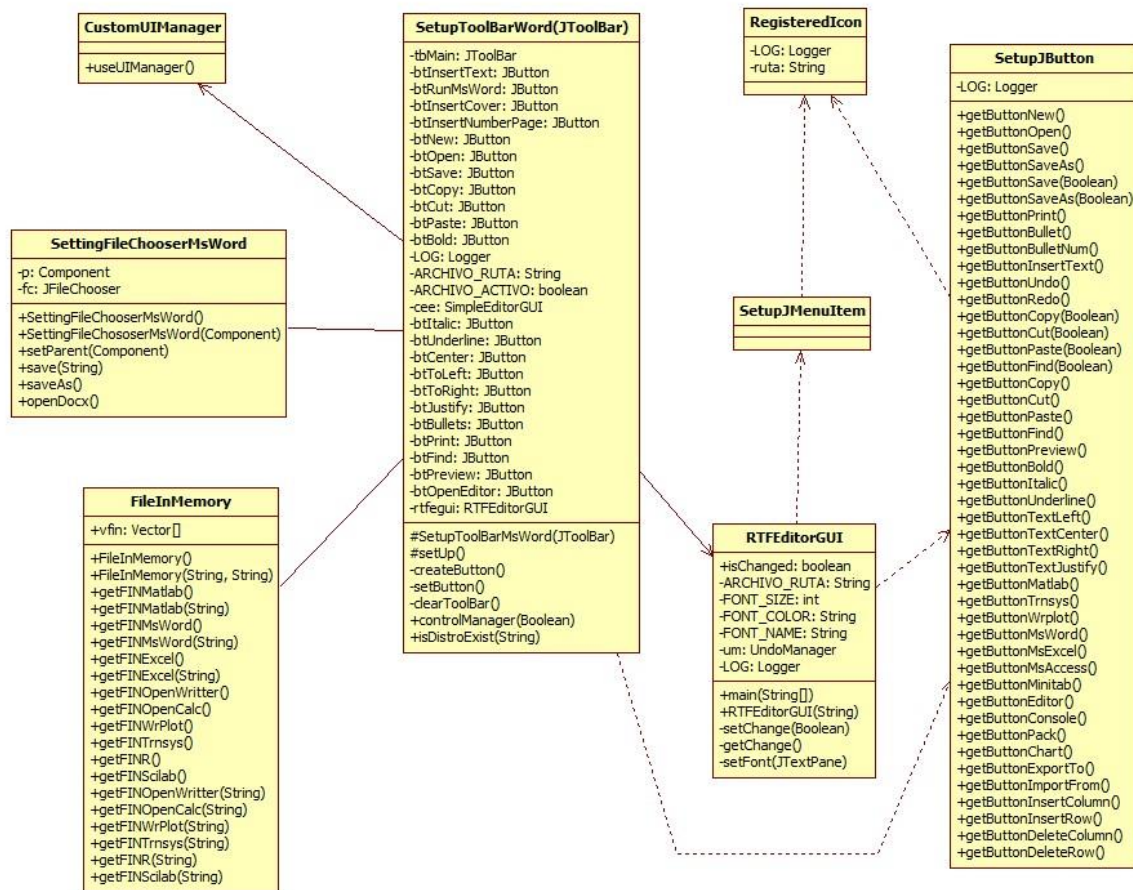


Figura 21. Diagrama de Clases para el módulo de Microsoft Word.

En la figura 21 se puede apreciar que el apartado de matlab se compone principalmente de dos clases SettingFileChooserMsWord y SetupToolBarMsWord. Esta última clase se puede apreciar que se encuentra asociada con la clase SimpleEditorGUI y depende directamente de SetupJButton.

3.3.9.2 Microsoft Excel

Para este sistema se implementó la funcionalidad de comunicación con el software Microsoft Office Excel 2007, este software es un libro que contiene diversas hojas de cálculo, en el cual se pueden efectuar diferentes tipo de cálculos ya sean contables, estadísticos, etc.

Se creó un módulo en java Editor de Tablas explicado en el punto 3.3.5. Se utilizó la librería JExcelAPI, de las cuales específicamente se utilizaron las clases Sheet, Workbook, BiffException, para la extracción de la información del documento, interpretación y el colocado dentro del editor de tablas de Excel.

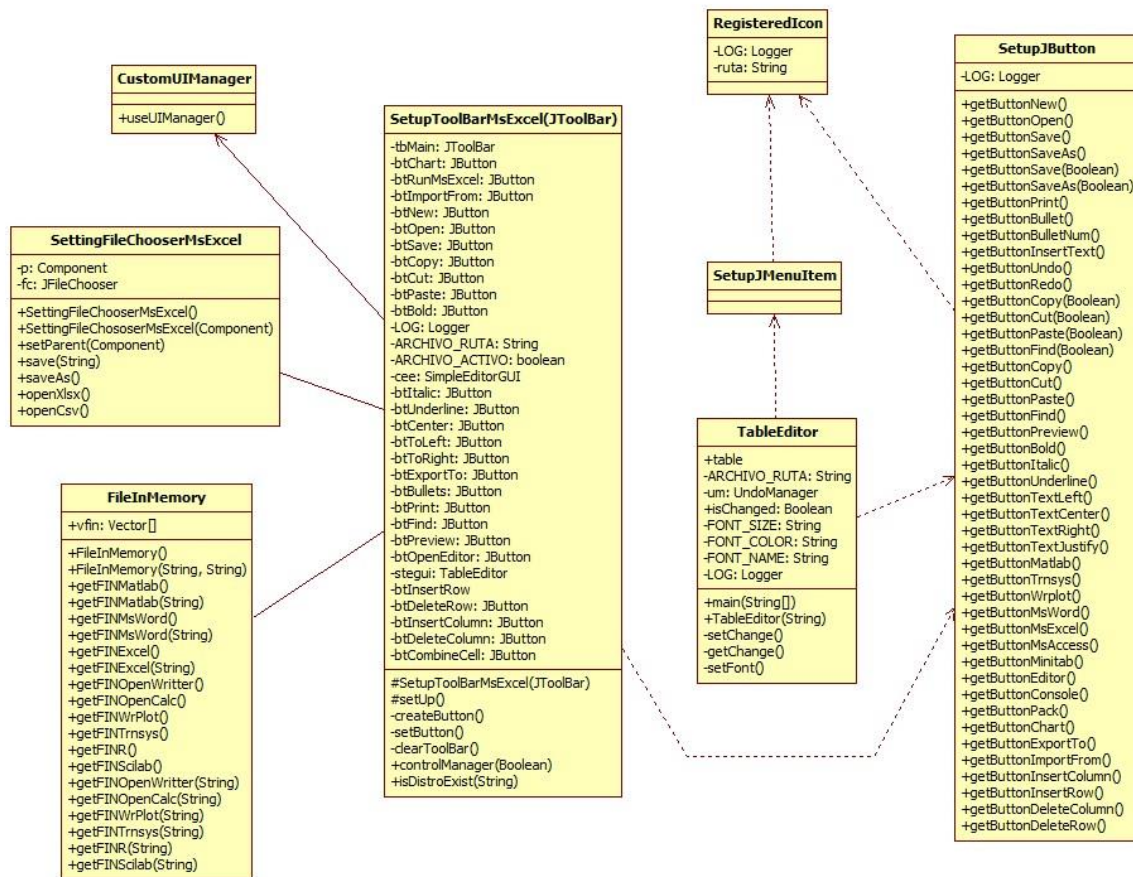


Figura 22. Diagrama de Clases para el apartado de Microsoft Excel.

En la figura 22 se puede observar las relaciones que conserva este modulo.

3.3.9.3 Microsoft Access

Para este sistema se implementó la funcionalidad de comunicación con el software Microsoft Office Access 2007, este software es un sistema manejador de bases de datos (DBMS), en el cual se pueden administrar y almacenar extensas cantidades de

información a las cuales se les puede aplicar diferentes usos, catálogos, minería de datos, etc. Este modulo utilizó el EditorSimple descrito en el punto 3.3.6.

La conexión a los procesos de este software se realizó a través de ODBC (Open Database Connectivity), realizando operaciones básicas de este sistema. En la figura 23 se puede observar la estructura de este modulo.

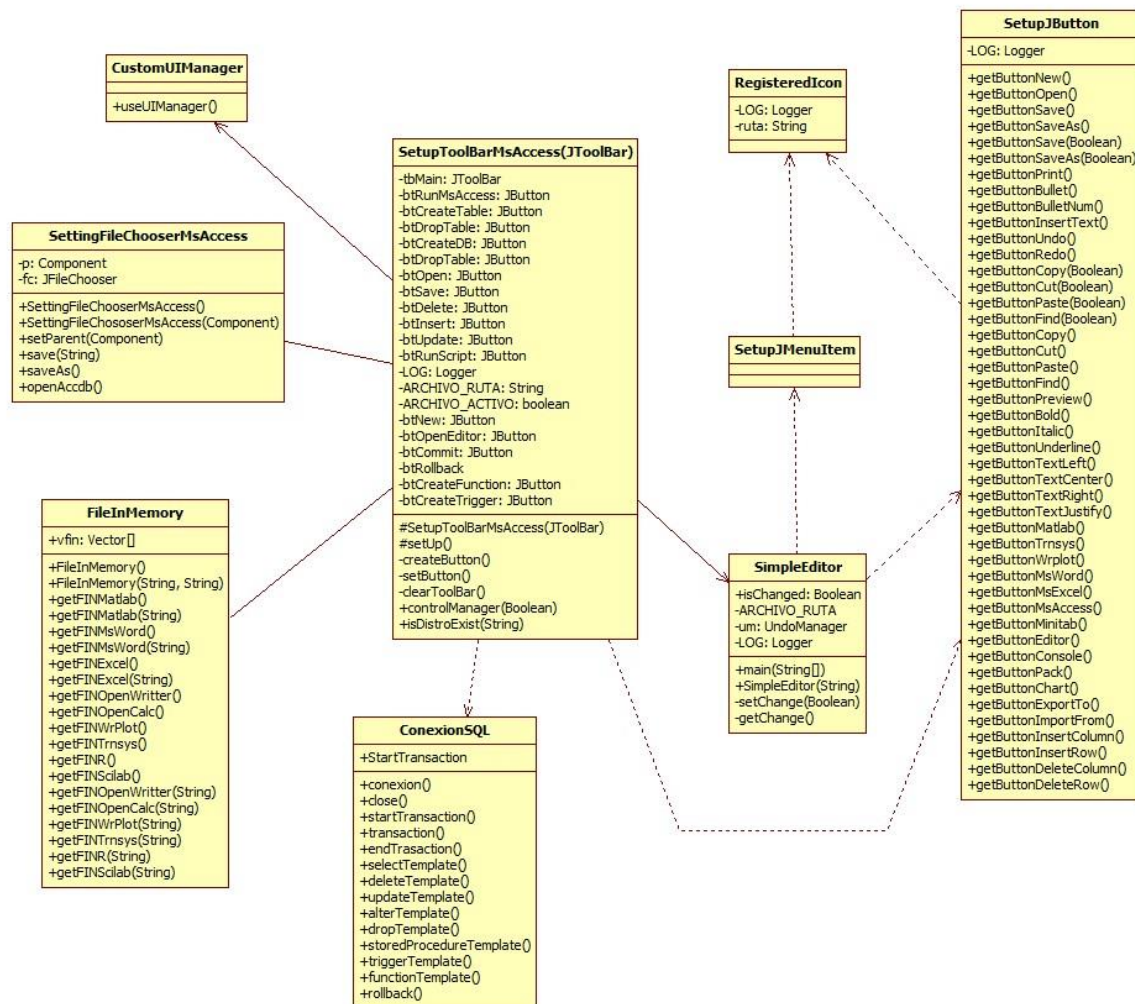


Figura 23. Diagrama de Clases para el módulo de Microsoft Access.

3.3.10 Wind Rose Plot

Para este sistema se implementó la funcionalidad de comunicación con el software Wind Rose Plot (WRPlot), este software es un sistema que genera gráficas y estadísticas sobre velocidad y dirección de viento.

La rosa de los vientos suele usarse para demostrar gráficamente la dirección y velocidad de viento predominante sobre un área, se encuentra basada en la escala Beaufort, ver *Escala de Beaufort* (Waypointgijon, s.f.).

La conexión a los procesos de este software se realizó a través de java, realizando operaciones básicas de este sistema. En la figura 24 se puede observar la estructura de este modulo.

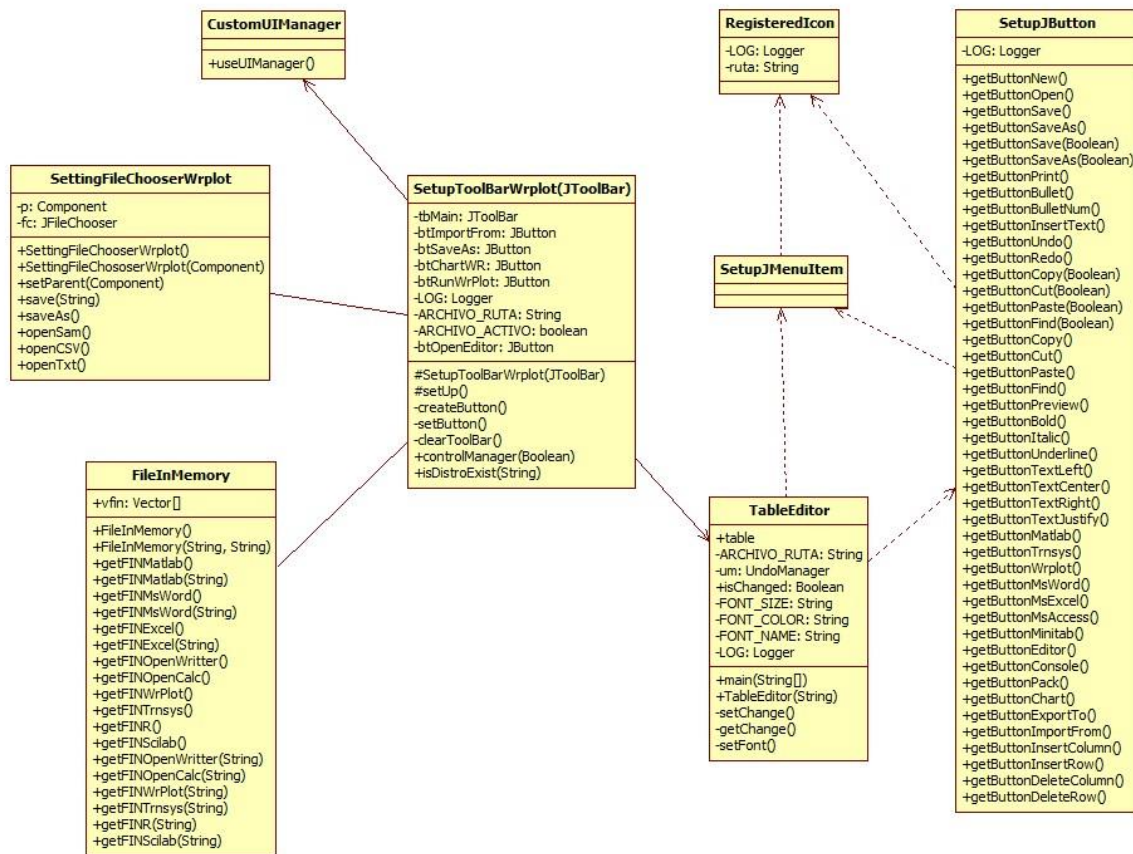


Figura 24. Diagrama de Clases para el módulo de Wind Rose Plot.

3.3.11 Trnsys

Para este sistema se implementó la funcionalidad de comunicación con el software Trnsys, este software es un sistema el cual permite crear un ambiente para la simulación de sistemas, para validacion de teorías o conceptos de usos sobre la energía.

Este software es open source (código abierto), y cuenta con enlaces para pre y post procesamiento de datos tales como Microsoft Excel, Matlab, COMIS, entre otros. La conexión a los procesos de este software se realizó a través de java (ver figura 25), realizando operaciones básicas de este sistema.

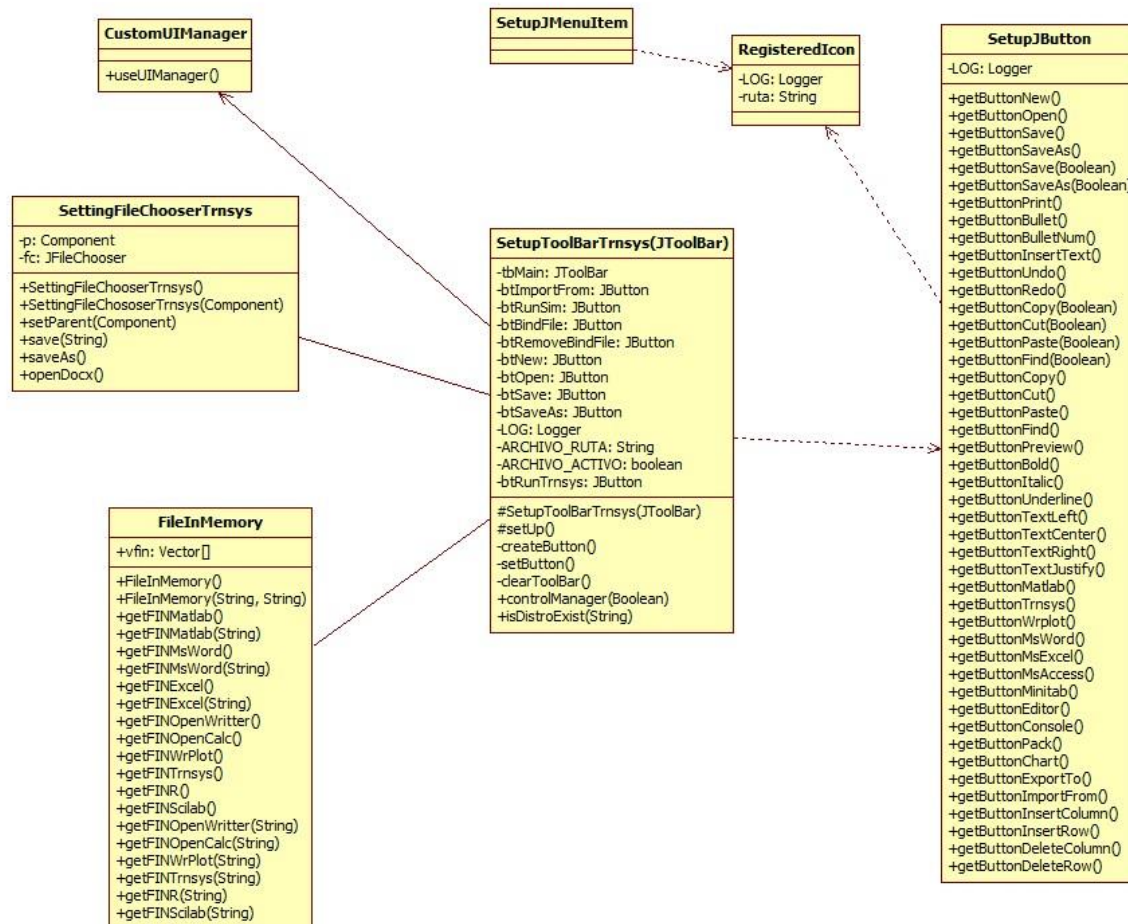


Figura 25. Diagrama de Clases para el módulo de Trnsys.

3.3.12 Openoffice.org

Apache OpenOffice es una suite de oficina de código abierto para el procesamiento de text, hojas de cálculo y más. Almacena todos sus datos en un formato que es un estándar internacional y puede también leer y escribir archivos producidos por otros paquetes de oficina. Puede ser descargado y utilizado completamente sin cargo para cualquier propósito. Este software es liberado bajo la licencia Apache 2.0. Esto significa que se puede utilizar el programa para cualquier propósito: doméstico, comercial, educacional, administración pública. Se puede instalar en tantas computadoras como sean necesarios.

3.3.12.1 Openoffice.org Writer

Se implementó la funcionalidad de comunicación con este software. un programa que utilizada un editor de texto para dar formato. Se puede dar ya sea en formato de texto plano, o formato de texto enriquecido.

Se utilizó el Editor de RTF explicado en el punto 3.3.4. El cual sirve de intérprete entre los dos sistemas. Se usaron las librerías de Apache POI, de las cuales específicamente se utilizaron las clases XWPFWordExtractor, IBodyElement, XWPFDocument, para la extracción de la información del documento, interpretación y colocación del texto dentro del editor RTF desarrollado.

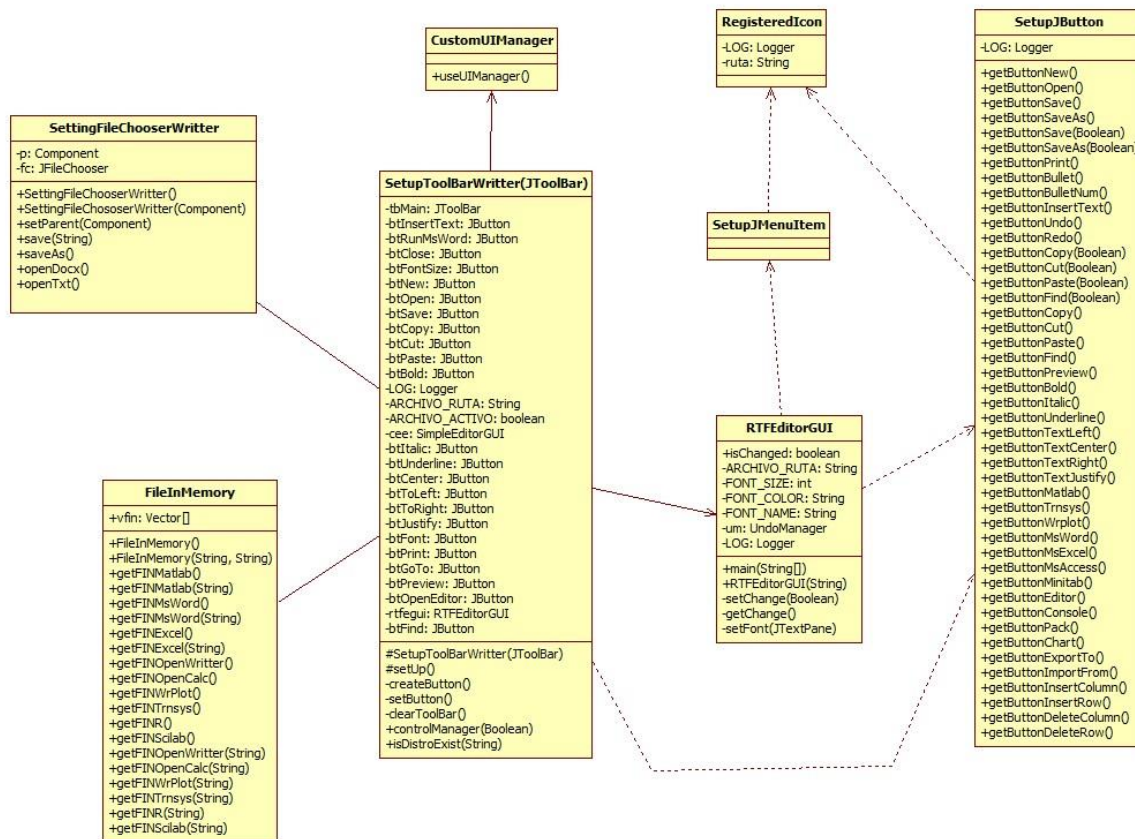


Figura 26. Diagrama de Clases para el módulo de OpenOffice.org Writer.

En la figura 26 se puede apreciar que el apartado de matlab se compone principalmente de dos clases SettingFileChooserWriter y SetupToolBarWriter. Esta última clase se puede apreciar que se encuentra asociada con la clase SimpleEditorGUI y depende directamente de SetupJButton.

3.3.12.2 Openoffice.org Calc

Para este sistema se implementó la funcionalidad de comunicación con el software OpenOffice.Org Calc, este software es un libro que contiene diversas hojas de cálculo, en el cual se pueden efectuar diferentes tipo de cálculos ya sean contables, estadísticos, etc.

Se creó un módulo en java Editor de Tablas ver punto 3.3.5, el cual sirve de interprete entre los dos software. Se usaron la librería *JExcelAPI*, de las cuales específicamente se utilizaron las clases *Sheet*, *WorkBook*, *BiffException*, para la extracción de la información del documento, interpretación y colocación dentro del editor de tablas desarrollado.

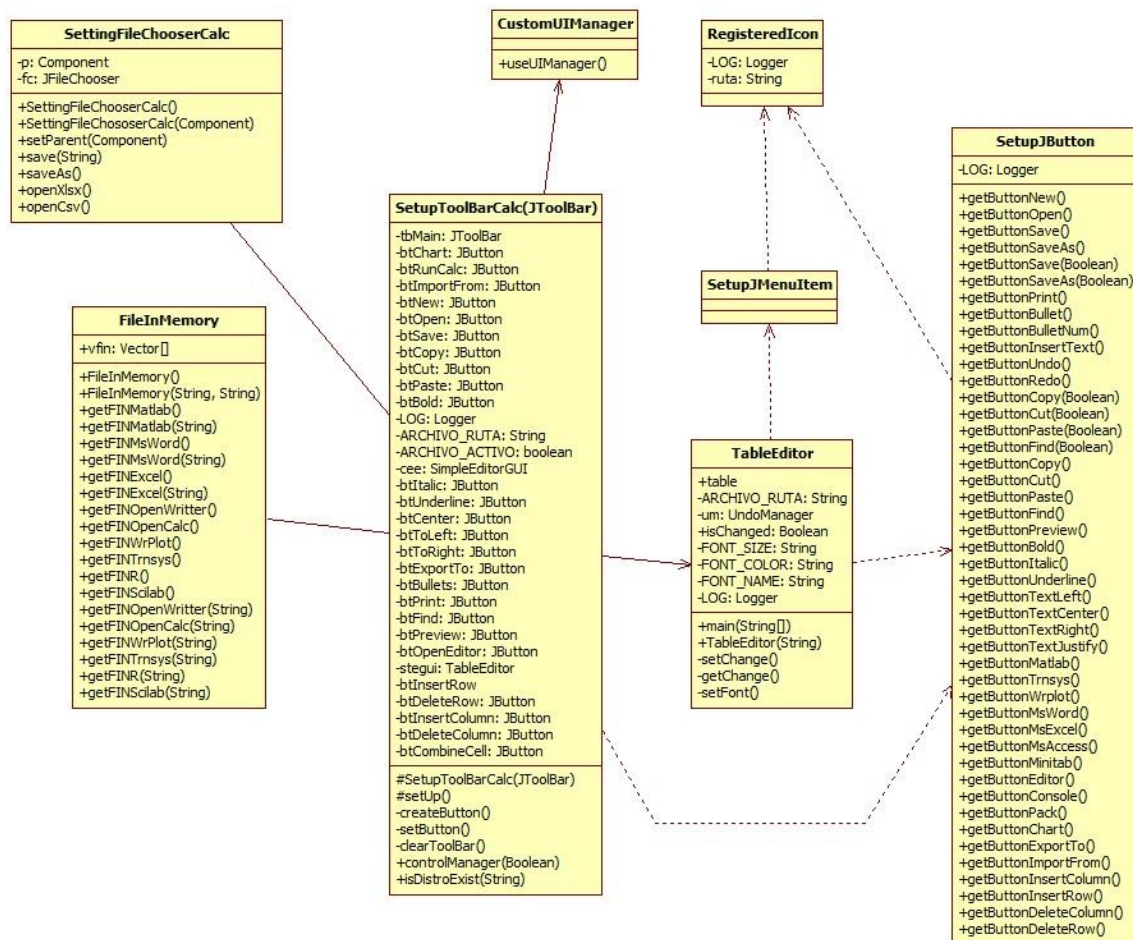


Figura 27. Diagrama de Clases para el módulo de OpenOffice.org Calc.

En la figura 27 se puede ver la estructura desarrollada para este modulo.

3.3.13 Scilab

Para este sistema se implementó la funcionalidad de la comunicación con el software Scilab versión 5.4.1, este software es un sistema de aplicación matemática, el entorno de esta aplicación se basa en el desarrollo de scripts, los cuales permiten el manejo de

algoritmos complejos, que en otros motores resulta complicada su ejecución, ver figura 28 para conocer la estructura del desarrollo de este modulo.

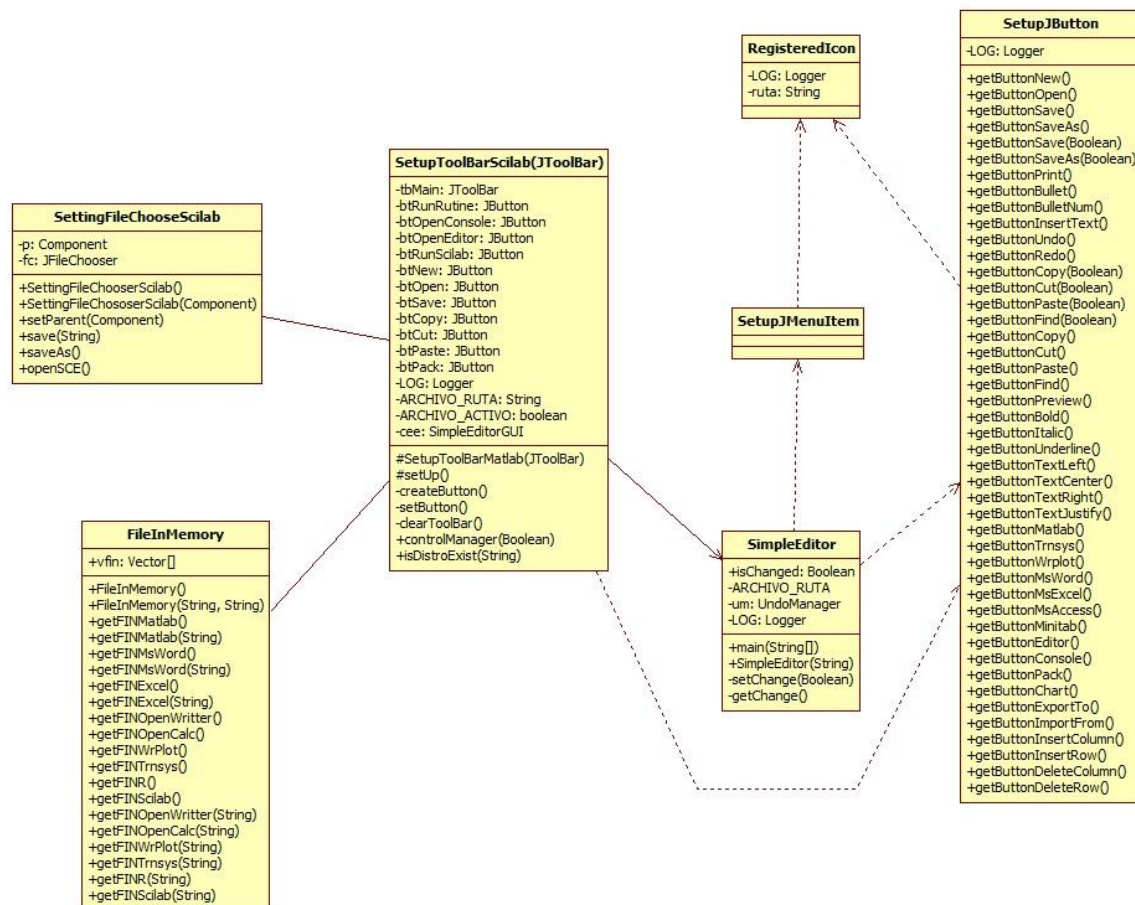


Figura 28. Diagrama de Clases para el módulo de Scilab.

3.3.14 R-Project

Para este sistema se implementó la funcionalidad de la comunicación con el software R-Project versión 2.15.1, este software es un sistema de aplicación estadística, basado en el lenguaje R, el entorno de esta aplicación se basa en el desarrollo de scripts, los cuales

permiten el manejo de algoritmos complejos, que en otros motores resulta complicada su ejecución, ver figura 29 para conocer la estructura del desarrollo de este modulo.

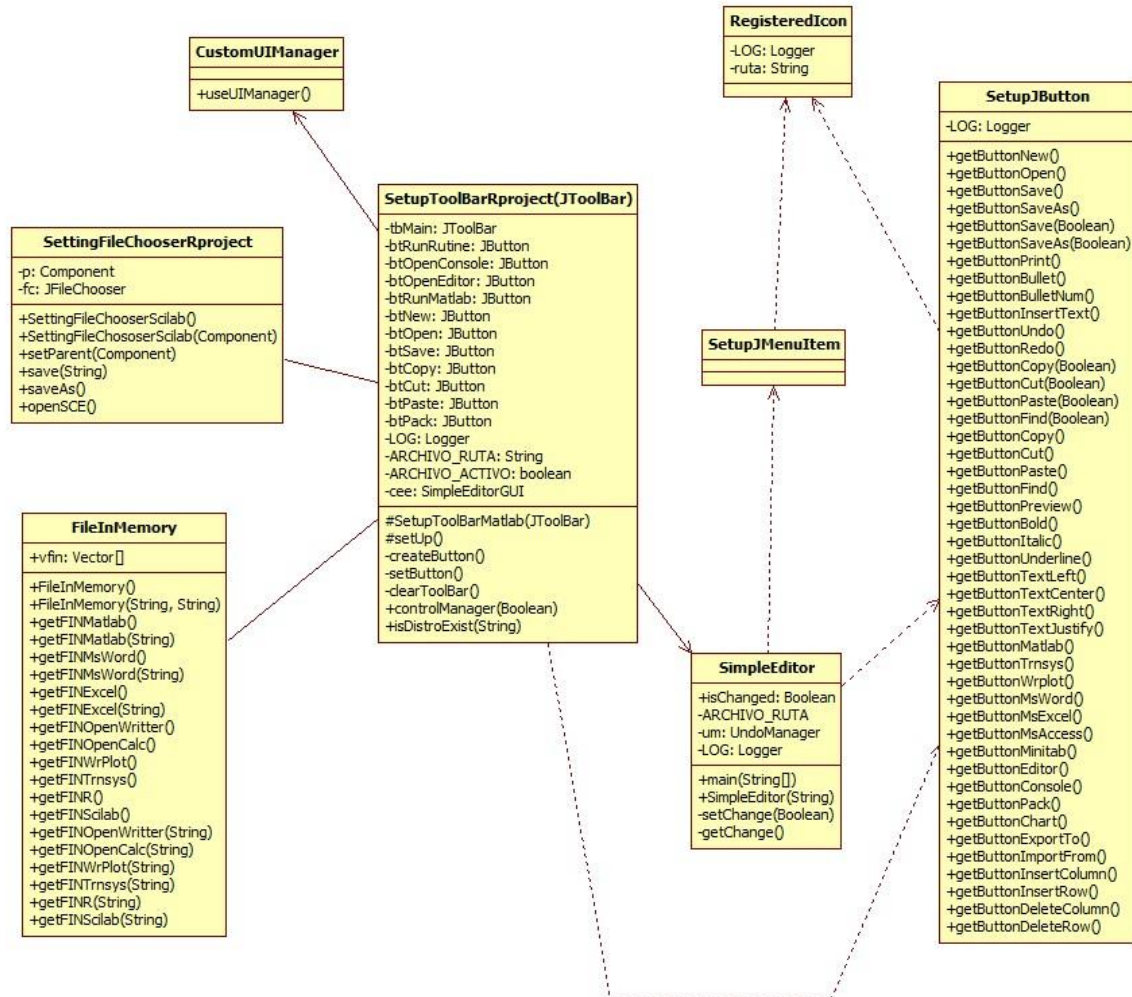


Figura 29. Diagrama de Clases para el módulo de R-Project.

3.3.15 MySQL

Para este sistema se implementó la funcionalidad de comunicación con el software MySQL, este software es un sistema manejador de bases de datos (DBMS), en el cual se pueden administrar y almacenar extensas cantidades de información a las cuales se les puede aplicar diferentes usos, catálogos, minería de datos, etc.

La conexión a los procesos de este software se realizó a través de jdbc (Java Database Connectivity), realizando operaciones básicas de este sistema. En la figura 30 se puede observar la estructura desarrollada por este modulo.

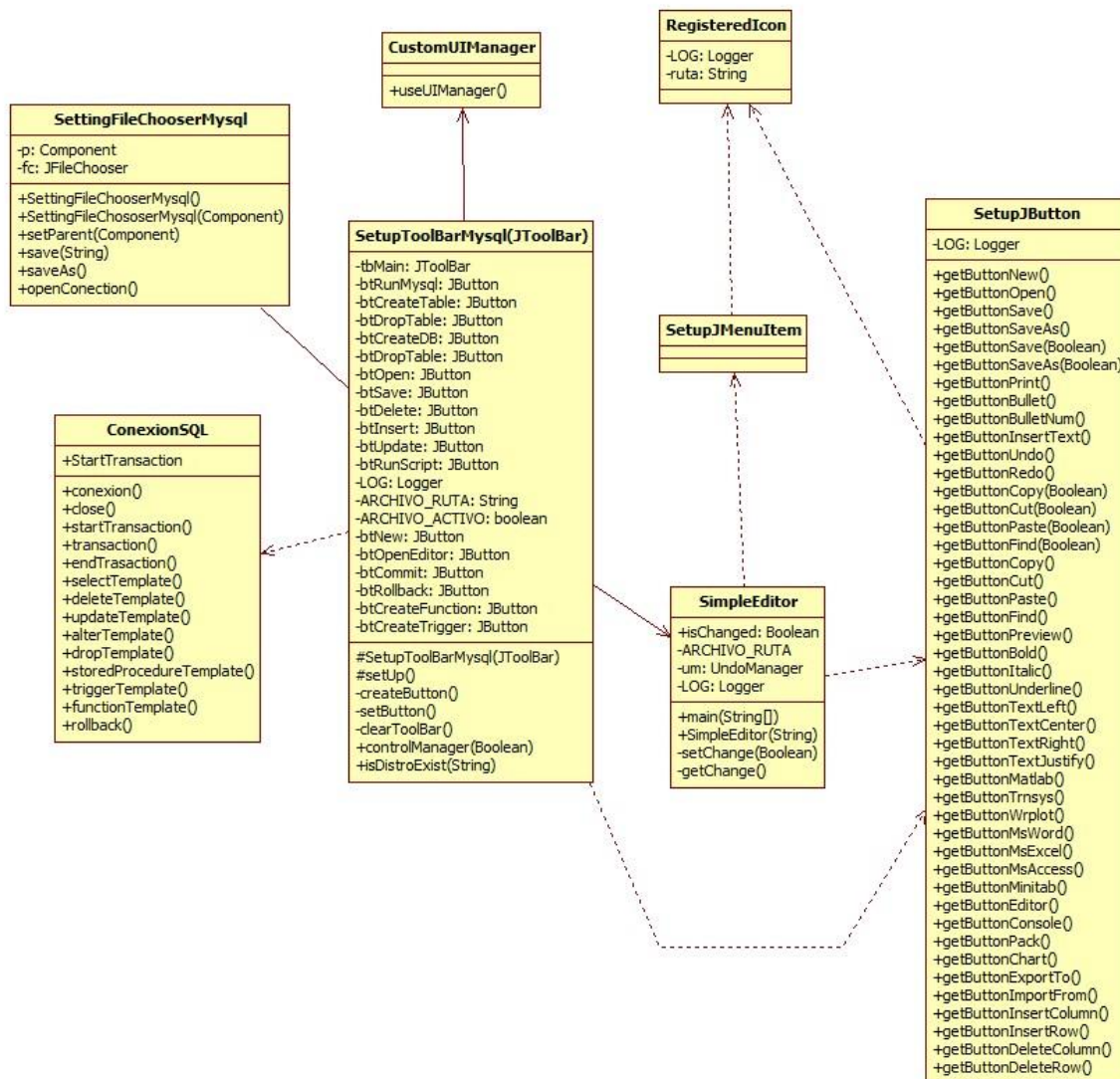


Figura 30. Diagrama de Clases para el módulo de MySQL.

3.4 Implementación de herramientas de la IA a una aplicación SIG de código abierto

Esta etapa consintió en integrar las diferentes herramientas que presentan la inteligencia artificial, tales como herramientas para clasificación de datos, como herramientas de

regresión de comportamiento, esto con la finalidad de poder dar soporte a la toma de decisiones.

Las herramientas que se integraron se presentan de la siguiente manera:

- Singular Value Decomposition (SVD)
- Linear Discriminant Analysis (LDA)
- Principal Components Analysis (PCA)
- K-means
- Independent Components Analysis (ICA)
- Multi-dimensional scaling (MDS)
- Redes neuronales artificiales
 - Perceptron Multicapa Feedforward Backpropagation
- Mapeo de Sammon

3.4.1 Implementación en el sistema

Los algoritmos fueron desarrollados en lenguaje m de Matlab, e integrados en lenguaje java al sistema GvSig, la manera en que se unieron estos dos sistemas fue por medio de sistema de archivos y la librería JMatlink, esta librería permite enlazar los dos motores y puedan interpretar la comunicación del envío de información entre sí.

Cada interface crea un archivo el cual se comunica con Matlab, este último sistema toma, interpreta y comienza a desarrollar el algoritmo actual. Después del procesamiento arrojan los resultados obtenidos en un archivo CSV, y muestra la gráfica si fue pedida.

Para mostrar los resultados obtenidos se utilizó el editor mencionado en el punto 3.3.1. Cada interface está preparada para leer los archivos dbf para los datos geo referenciados y archivos CSC con información externa.

Debido a la naturaleza de cada algoritmo se desarrolló una interfaz diferente la cual se describe a continuación.

3.4.2 SVD

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto 3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n
- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

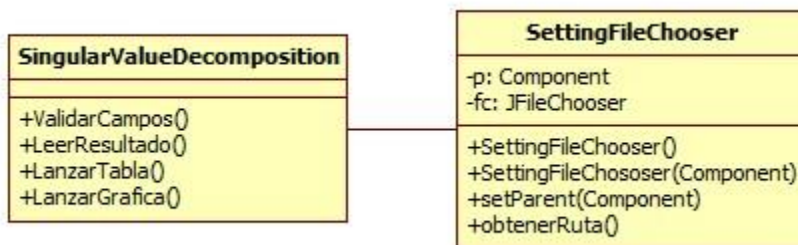


Figura 31. Diagrama de Clases para el módulo de SVD.

3.4.3 LDA

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n
- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

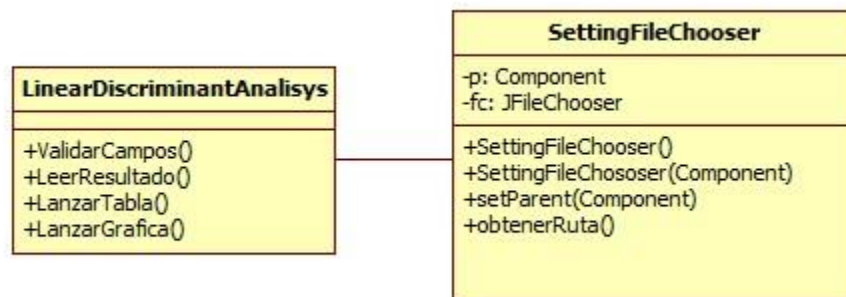


Figura 32. Diagrama de Clases para el módulo de LDA.

3.4.4 PCA

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n
- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

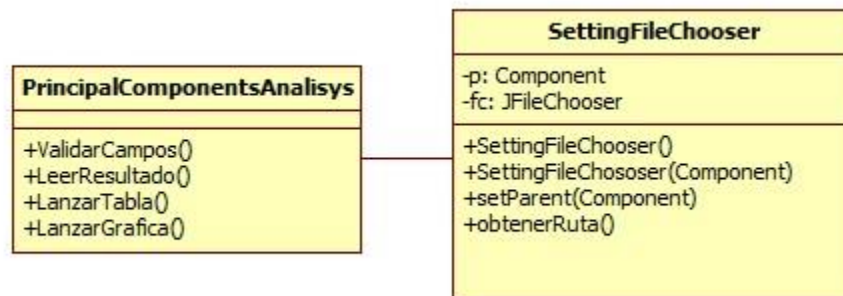


Figura 33. Diagrama de Clases para el módulo de PCA.

3.4.5 K-Means

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n

- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

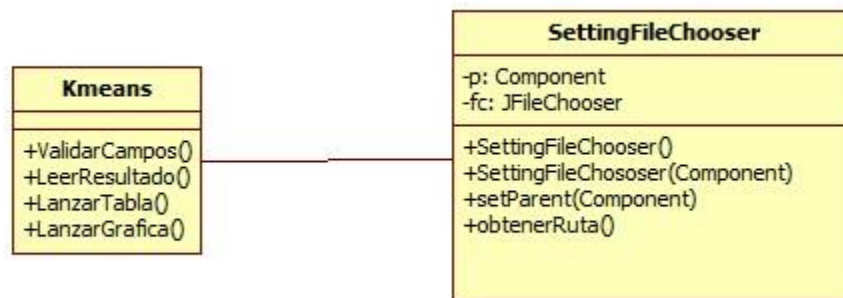


Figura 34. Diagrama de Clases para el módulo de K-means.

3.4.6 ICA

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n
- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

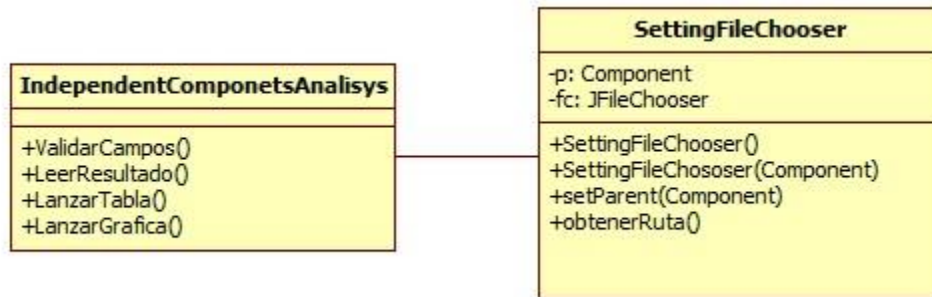


Figura 34. Diagrama de Clases para el módulo de ICA.

3.4.7 MDS

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n
- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

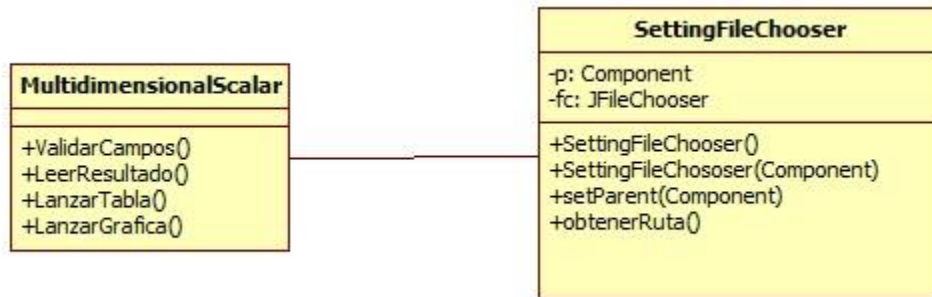


Figura 38. Diagrama de Clases para el módulo de MDS.

3.4.8 Mapeo de Sammon

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- Las columnas que se tomaran en cuenta de 1 a n
- Los renglones que se tomaran en cuenta de 1 a m.
- Cuál es la columna que contiene la clase.
- Los componentes principales a obtener.
- Si se generara gráfica.

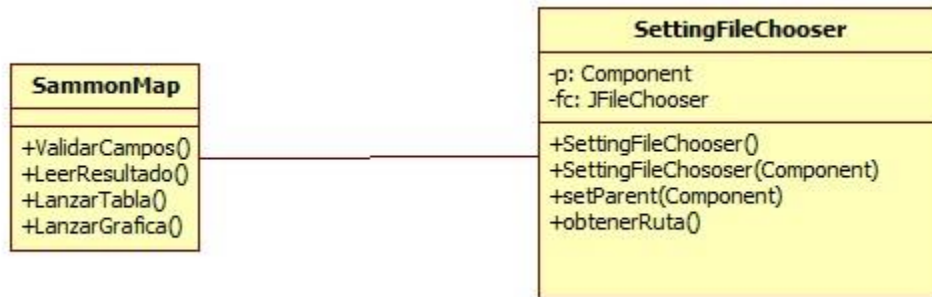


Figura 37. Diagrama de Clases para el módulo de Mapeo de Sammon.

3.4.9 Redes Neuronales Artificiales

Para el desarrollo de este componente se recurrió al procedimiento descrito en el punto

3.4.1. Además fue necesario crear una interface en java la cual pide:

- Ruta de archivo. Este archivo es un conjunto de datos previamente formateado.
- La cantidad de capas
- El peso de la Neurona
- La meta para el mejor desempeño
- Porcentaje en que se separa el conjunto de datos
- Funciones de transferencia.
- Función de entrenamiento.
- Función de Aprendizaje
- Tipo de error a calcular.
- Se mostraran las gráficas.

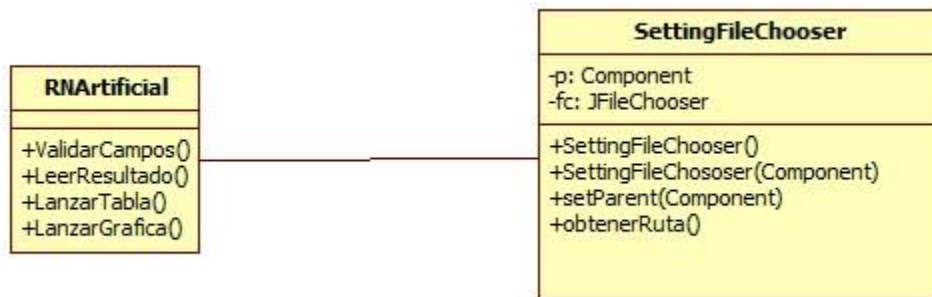


Figura 38. Diagrama de Clases para el módulo RNA MLF BP.

3.5 Implementación de las herramientas desarrolladas dentro de una aplicación SIG de código abierto

Como objetivo de una representación más palpable de utilidad de las herramientas antes descritas, se recurrió a desarrollar la aplicación de los temas anteriores, es decir, se desarrolló un SIG utilizando como zona de estudio la ciudad de Mexicali, a motivo de utilizar una muestra más pequeña para control de las pruebas piloto.

Se obtuvieron los mapas de edafología, geología, topografía, uso de suelo y vegetación de INEGI y se obtuvo imágenes satelitales de mapas WMS de Cartografía Catastral de la Dirección General del Catastro (<http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx>), Con una proyección EPSG: 23030, del año 2010.

Se aplicaron a los datos de velocidad y viento los algoritmos de PCA, RNA-FF-BP, SVD, Mapeo de Sammon.

CAPÍTULO IV

4. Resultados y Discusión

En este capítulo se presentan los resultados de obtenido de integrar diferentes herramientas de software e inteligencia artificial, la información contenida de la base de datos con el fin de detectar patrones de comportamiento del vector de viento en Baja California para los análisis correspondientes de casa estudio.

El capítulo se encuentra seccionado en cuatro apartados; el primero muestra los registros obtenidos listos para análisis; el segundo explica la funcionalidad y muestra las interfaces desarrolladas para integrar los distintos sistemas; el tercer apartado de igual manera explica la funcionalidad y muestras las interfaces desarrolladas para integrar los métodos de la IA; el último apartado muestra los resultados obtenidos de trabajar con los sistemas a la par.

4.1 Base de datos

Se obtuvo una base de datos con 690680 registros, de los cuales 490600 pertenecen a la estación meteorológica de CICESE, 180080 a la estación meteorológica de CONAGUA y 20000 de la estación meteorológica de la UABC.

4.2 Herramientas Integradas

4.2.1 Funcionalidad

En la tabla 3 y 4 se muestra un mapa que indica que funcionalidad tiene integrada cada programa y/o complemento desarrollado. En la tabla 5 se da una descripción de la naturaleza de cada evento desarrollado.

	Matlab	Scilab	R-Project	Microsoft Word	Microsoft Excel	Microsoft Access	MySQL	OpenOffice. Org Writer	OpenOffice. Org Cal	Tmsys	WrPlot
Nuevo	x	x	x	x	x	x		x	x	x	
Abrir	x	x	x	x	x			x	x		
Abrir Consola	x	x	x								
Abrir [Programa en ejecución]	x	x	x	x	x	x	x	x	x	x	x
Abrir Editor Simple	x	x	x								
Abrir Editor RTF				x				x			
Abrir Editor de Tablas					x				x		x
Guardar	x	x	x	x	x			x	x	x	
Guardar Como				x	x			x	x	x	x
Ejecutar	x	x	x								
Compilar	x	x	x								
Imprimir				x	x			x	x		
Insertar texto				x				x			
Vista Previa				x	x			x	x		
Exportar A					x				x		
Importar De					x				x	x	x
Nueva Base de Datos							x				
Eliminar Base de Datos						x	x				
Nueva Tabla						x	x				
Eliminar Tabla						x	x				

Commit						X	X				
Rollback						X	X				
Crear procedimiento almacenado						X	X				
Crear función						X	X				
Crear trigger						X	X				
Insertar						X	X				
Eliminar						X	X				
Actualizar						X	X				
Crear SQL						X	X				
Generar Rosa de los vientos											X
Importar proyecto										X	
Lanzar Simulación										X	
Asociar archivo										X	
Eliminar asociación de archivo										X	

Tabla 3. Funcionalidad mostrada desde la barra de herramientas.

	Editor Simple	Editor RTF	Table Editor	Console
Nuevo	X			
Abrir	X			
Guardar	X			
Guardar Como	X			
Cerrar	X	X	X	X
Copiar	X	X	X	
Cortar	X	X	X	
Pegar	X	X	X	
Imprimir	X	X		
Deshacer	X	X	X	
Rehacer	X	X	X	
Buscar	X	X	X	
Buscar Siguiente	X	X	X	
Ir a	X	X	X	
Seleccionar todo	X	X	X	

Negrita		X	X	
Itálica		X	X	
Subrayado		X	X	
Fuente		X	X	
Tamaño		X	X	
Color de Fuente		X	X	
Alinear a la Izquierda		X	X	
Alinear a la Derecha		X	X	
Centrar		X	X	
Justificar		X	X	
Insertar Columna			X	
Eliminar Columna			X	
Insertar Fila			X	
Eliminar Fila			X	
Exportar A			X	
Importar De			X	
Crear grafica			X	

Tabla 4. Funcionalidad que ofrece cada uno de los editores desarrollados.

Nuevo	Crea un nuevo archivo con la extensión de los archivos soportados para sistema activado en el momento. El total de archivos que soporta es M, XLS, DOC, TXT, RTF, ACCDB, CSV, (TRN, TRNSYS).
Abrir	Abre un archivo del tipo soportado por el sistema activado.
Abrir Consola	Abre la consola diseñada en java, que permite la comunicación por comandos para los Scilab, Matlab y R
Abrir [Programa en ejecución]	Manda ejecutar el programa que se encuentra activado.
Abrir Editor Simple	Abre un Editor Simple de procesamiento de texto que permite el desarrollo de scripts y documentos simples de texto.
Abrir Editor RTF	Abre un Editor RTF que permite el procesamiento de texto enriquecido para lograr documentos con decorados básicos.
Abrir Editor de Tablas	Abre el un editor que permite la edición básica de hojas de cálculo.
Guardar	Guarda los cambios efectuados sobre el documento abierto.
Guardar Como	Guarda el contenido del archivo actual con un nombre otorgado

	por el usuario nombre y/o formato.
Ejecutar	Ejecuta una rutina compilada por Scilab, Matlab o R.
Compilar	Compila alguna rutina guardada por Scilab, Matlab o R.
Imprimir	Manda a imprimir el archivo en uso.
Insertar texto	Lanza una ventana en el cual el texto capturado se inserta al final del documento, sin necesidad de abrir el editor RTF o abrir Microsoft Word.
Vista Previa	Muestra una vista previa del documento a imprimir
Exportar A	Exporta la tabla utilizada actualmente a un tipo de formato y nombre especificado.
Importar De	Importa un archivo en formato CVS o XLS.
Nueva Base de Datos	Pide el nombre de la base de datos y ejecuta un script que crea una nueva base de datos en MySQL.
Eliminar Base de Datos	Elimina la base de datos indicada.
Nueva Tabla	Crea una nueva tabla dentro de la base de datos especificada.
Eliminar Tabla	Elimina una tabla dentro de la base de datos especificada.
Commit	Acepta todos los procesos en espera de guardarse en la base de datos.
Rollback	Regresa a la última transacción aceptada en la base de datos.
Crear procedimiento almacenado	Crea una plantilla de SQL para crear un procedimiento almacenado.
Crear función	Crea una plantilla de SQL para crear una función.
Crear trigger	Crea una plantilla de SQL para crear un disparador.
Insertar	Inserta un registro dentro de la base de datos utilizada en la tabla especificada.
Eliminar	Elimina el registro indicado dentro de la base de datos utilizada en la tabla especificada.
Actualizar	Actualiza los datos especificados dentro de la base de datos utilizada, en la tabla especificada.
Crear SQL	Abre el editor simple para la creación de sentencias SQL.
Generar Rosa de los vientos	Genera la gráfica de la rosa de los vientos en base a los datos del archivo cargado en memoria.
Importar proyecto	Carga en el sistema otro proyecto de Trnsys indicado.
Lanzar Simulación	Ejecuta una simulación de Trnsys guardada.

Asociar archivo	Asocia documentos de texto de tipo FORTRAN (FOR), Microsoft Word (DOC) y Texto plano (TXT).
Eliminar asociación de archivo	Elimina la asociación creado a los documentos de texto de tipo FORTRAN (FOR), Microsoft Word (DOC) y Texto Plano (TXT).
Cerrar	Cierra el editor actualmente abierto (simple, RTF o Tablas). Y conserva el archivo en memoria.
Copiar	Copia el texto seleccionado en el editor.
Cortar	Corta el texto seleccionado en el editor.
Pegar	Pega el texto seleccionado en el editor.
Imprimir	Manda a imprimir el archivo en actualmente activado.
Deshacer	Regresa a un cambio previo.
Rehacer	Vuelve a un cambio realizado.
Buscar	Busca la primera coincidencia de acuerdo a un criterio otorgado.
Buscar Siguiente	Busca la siguiente coincidencia de acuerdo a un criterio otorgado.
Ir a	Se le indica la línea a la cual moverá el cursor.
Seleccionar todo	Selecciona todo el texto del archivo.
Negrita	Pone en negrita el texto seleccionado.
Itálica	Pone en itálica el texto seleccionado.
Subrayado	Subraya el texto seleccionado.
Fuente	Cambia la fuente del texto seleccionado.
Tamaño	Cambia el tamaño del texto seleccionado.
Color de Fuente	Cambia el color del texto seleccionado.
Alinear a la Izquierda	Alinea el texto seleccionado a la izquierda.
Alinear a la Derecha	Alinea el texto seleccionado a la derecha.
Centrar	Centra el texto seleccionado.
Justificar	Justifica el texto seleccionado.
Insertar Columna	Inserta una columna a la tabla de la hoja de cálculo.
Eliminar Columna	Elimina una columna a la tabla de la hoja de cálculo.
Insertar Fila	Inserta una fila a la tabla de la hoja de cálculo.
Eliminar Fila	Elimina una fila a la tabla de la hoja de cálculo.
Exportar A	Exporta la tabla utilizada actualmente a un tipo de formato y nombre especificado.
Importar De	Trae a la tabla editor, las tablas indicadas de un archivo

	especificado.
Crear grafica	Crea una gráfica de puntos, barras, pastel, de los datos.

Tabla 5. Descripción de la funcionalidad que contiene cada evento.

4.2.2 Interfaz grafica

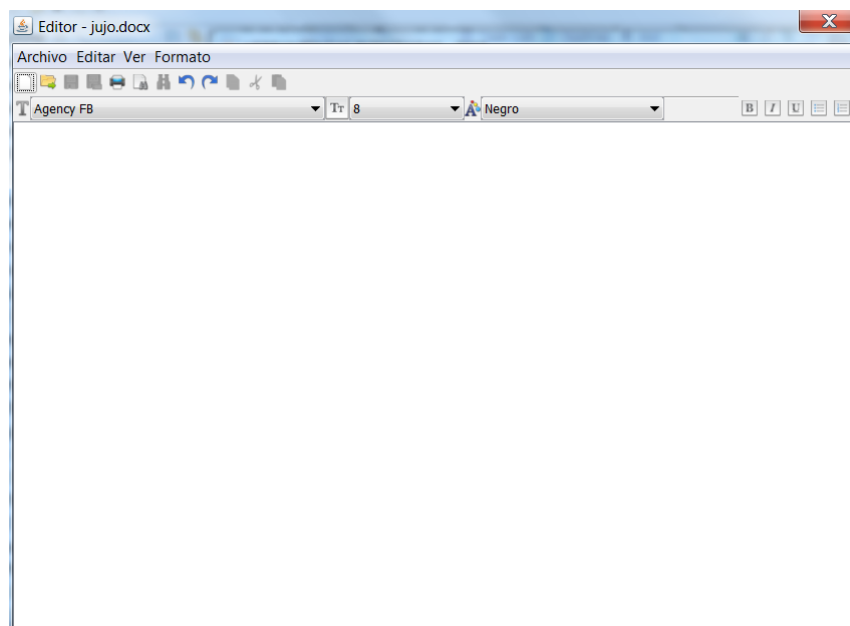


Figura 39. Vista 1 del Editor RFT para procesamiento de texto.

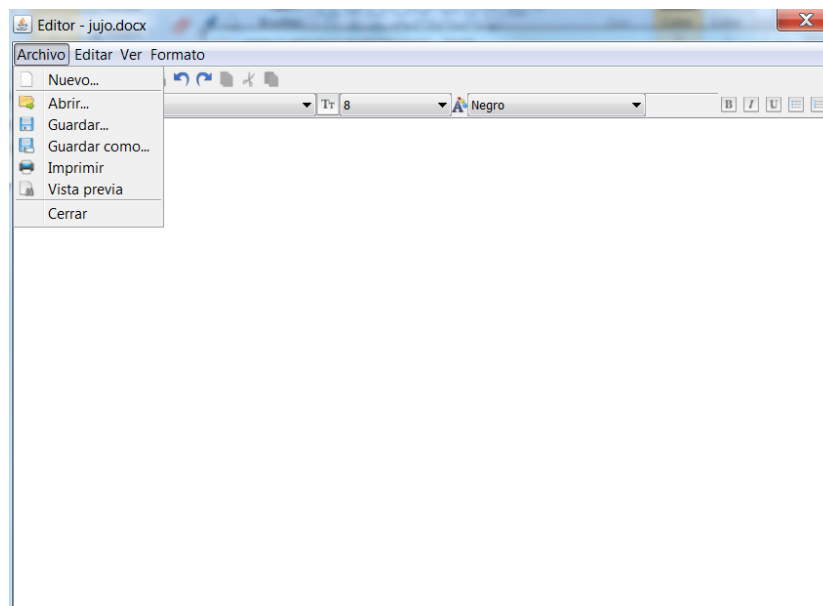


Figura 40. Vista 2 del Editor RFT para procesamiento de texto.

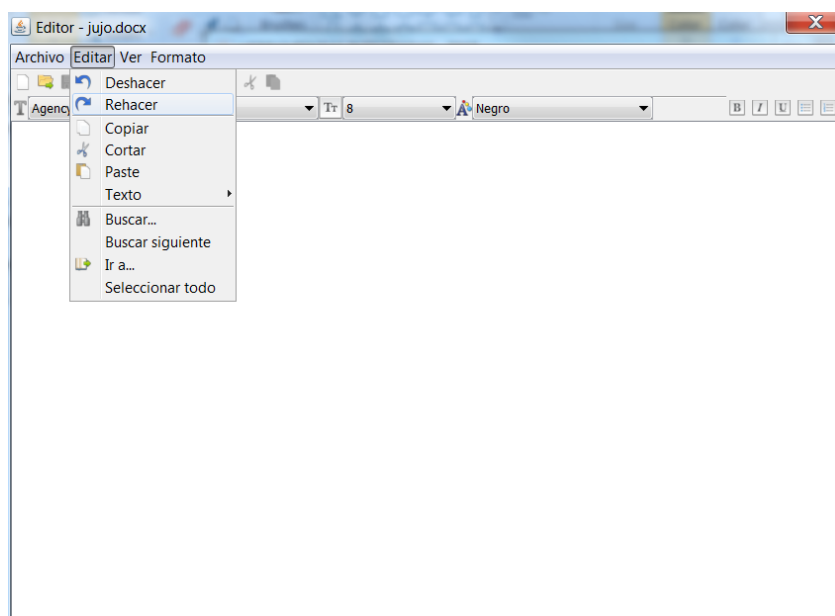


Figura 40. Vista 3 del Editor RFT para procesamiento de texto.

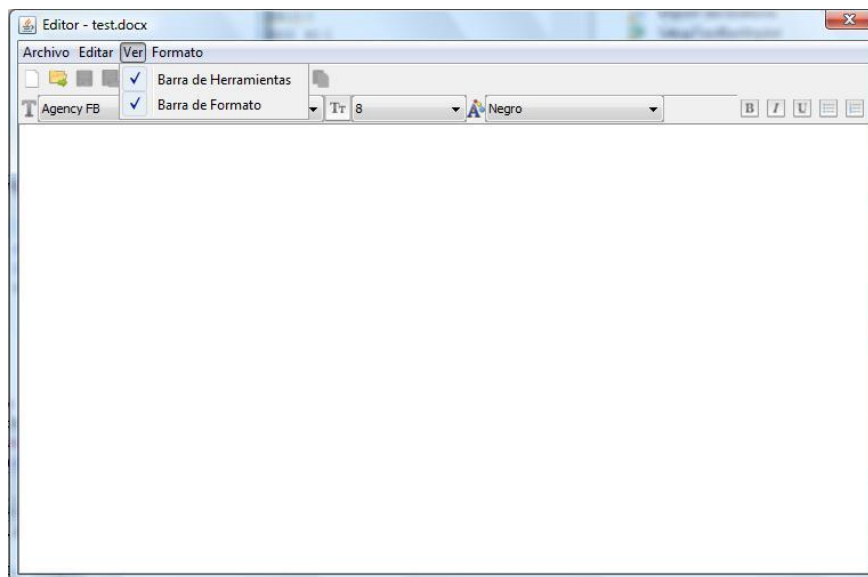


Figura 41. Vista 4 del Editor RFT para procesamiento de texto.

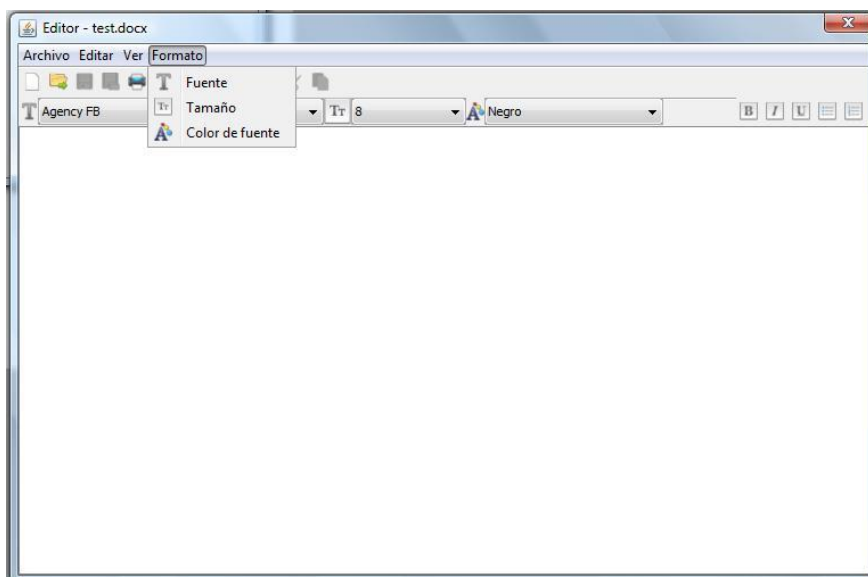


Figura 42. Vista 5 del Editor RFT para procesamiento de texto.

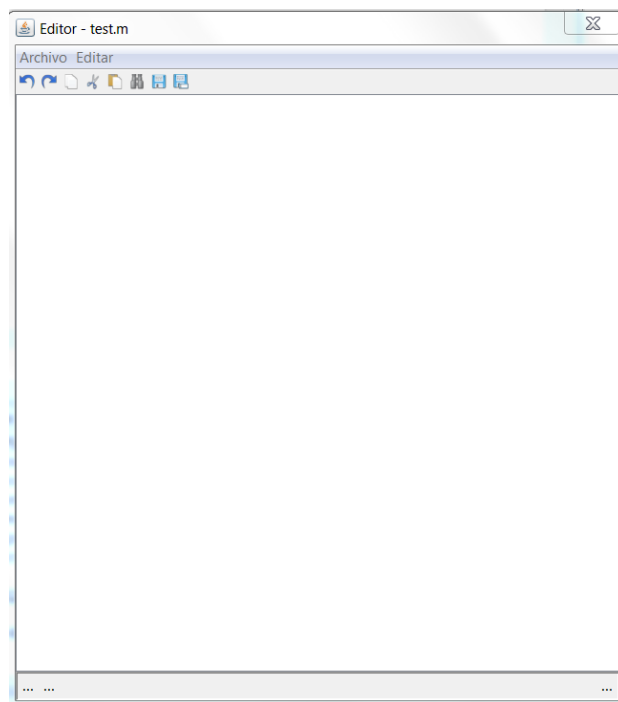


Figura 43. Vista 1 del Editor Simple para procesamiento de texto.

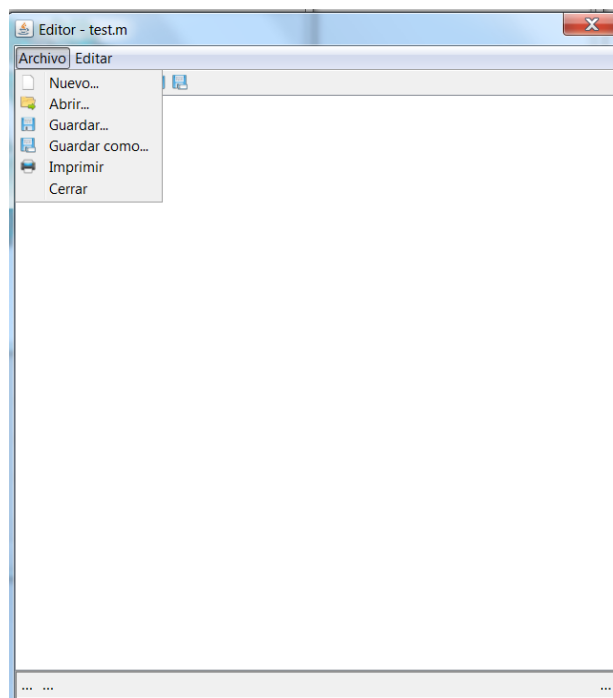


Figura 44. Vista 2 del Editor Simple para procesamiento de texto.

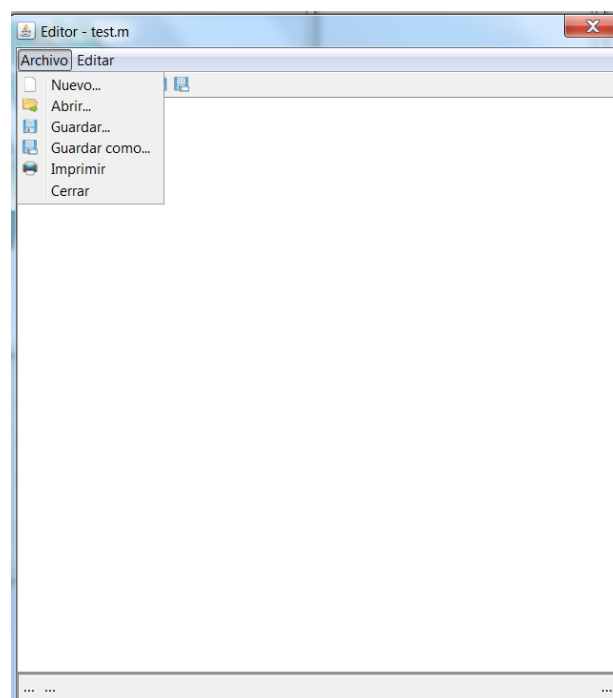


Figura 45. Vista 3 del Editor Simple para procesamiento de texto.

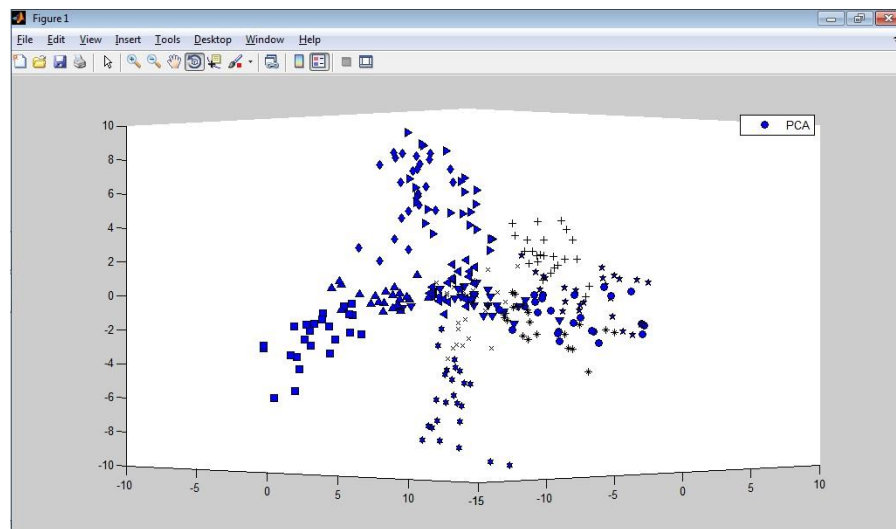


Figura 50. Resultado del procesamiento de PCA.

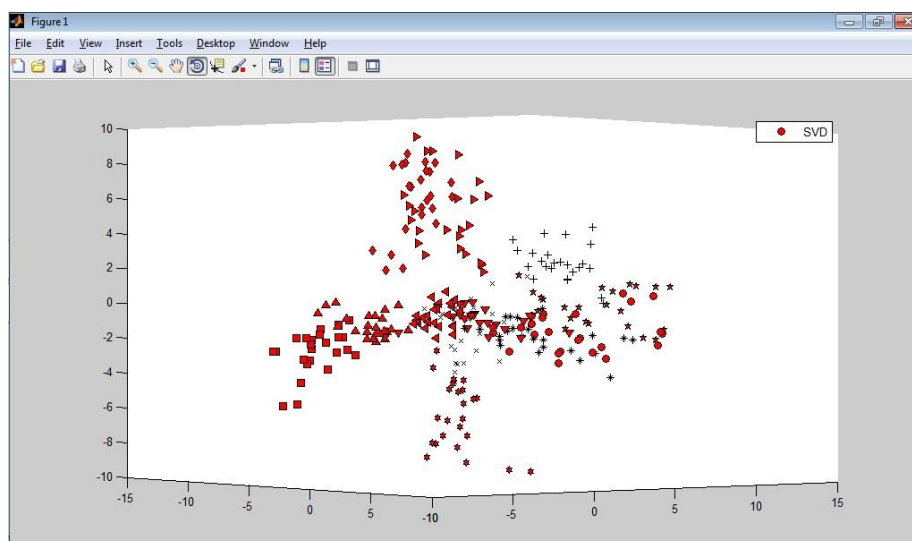


Figura 51. Resultado del procesamiento de SVD.

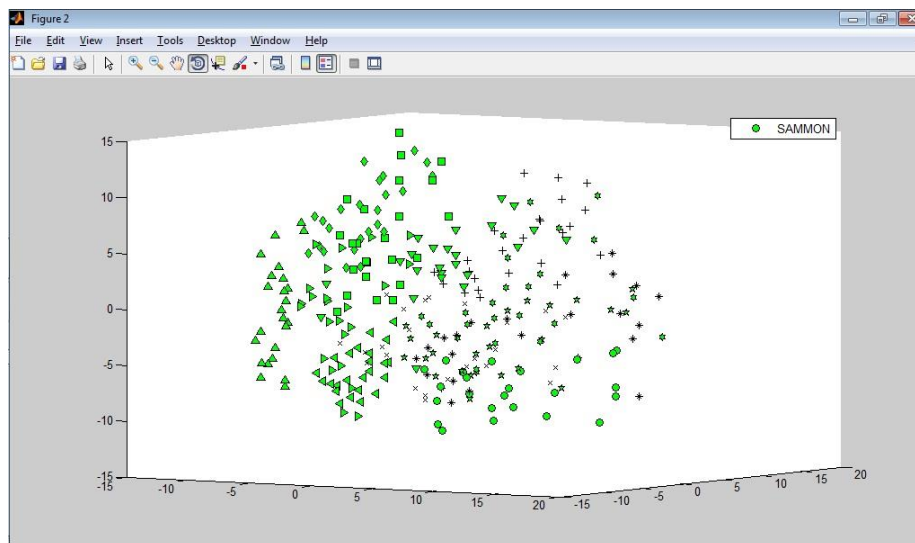


Figura 52. Resultado del procedimiento de Mapeo de Sammon.

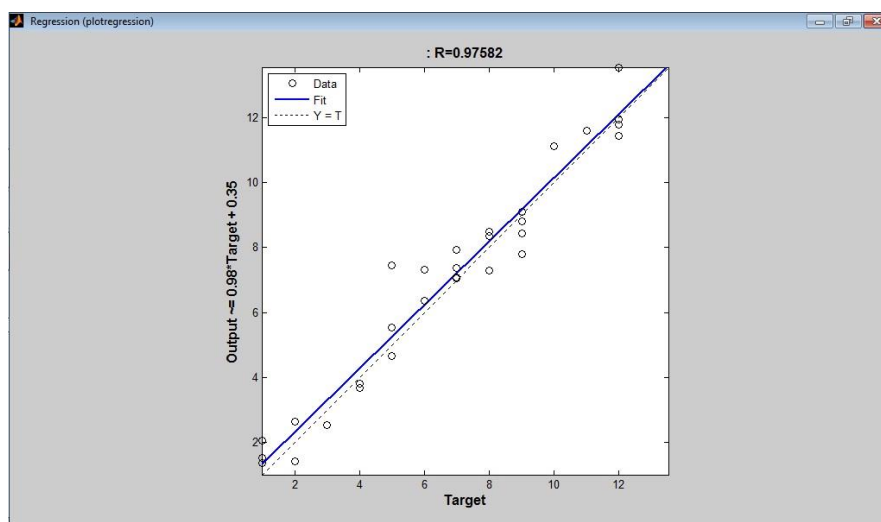


Figura 53. Regresión lineal de los resultados obtenidos de una RNA MLF BP.

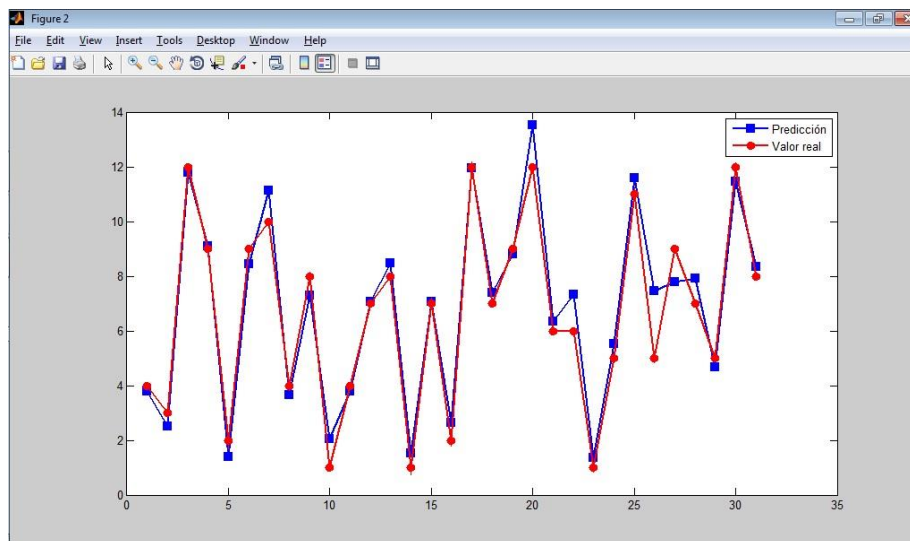
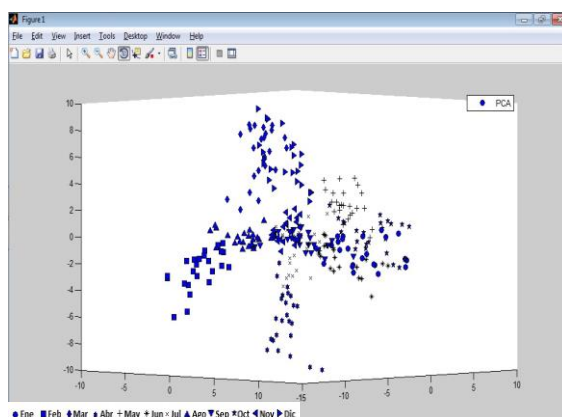


Figura 54. Resultado de la predicción de los datos.

a)



b)

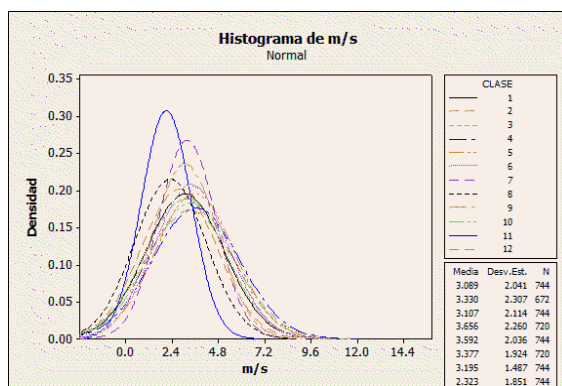


Figura 55. Resultado de la predicción de los datos

4.4 Coexistencia de los elementos

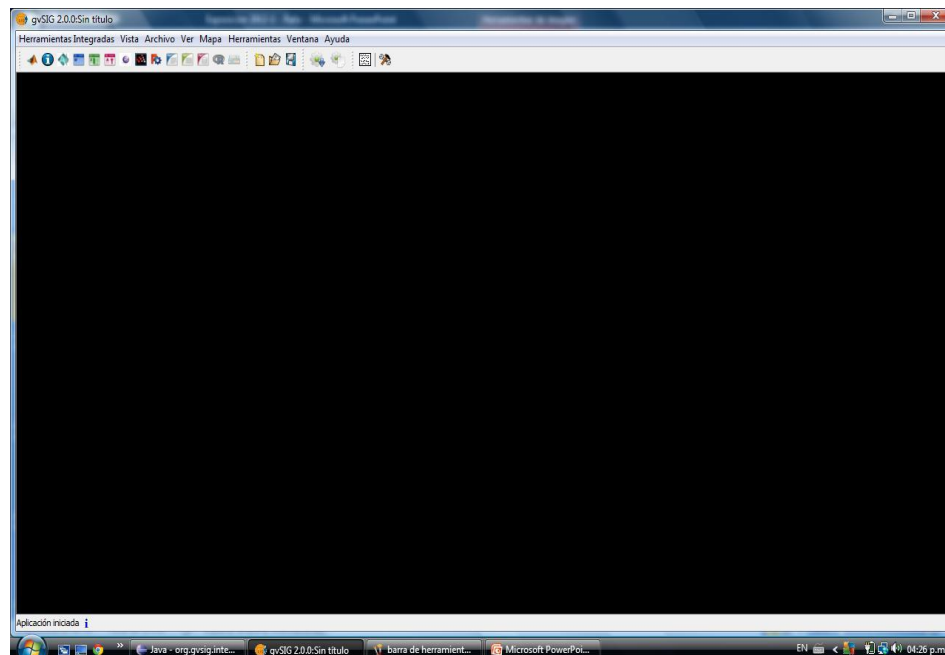


Figura 56. Resultado de la prediccion de los datos.

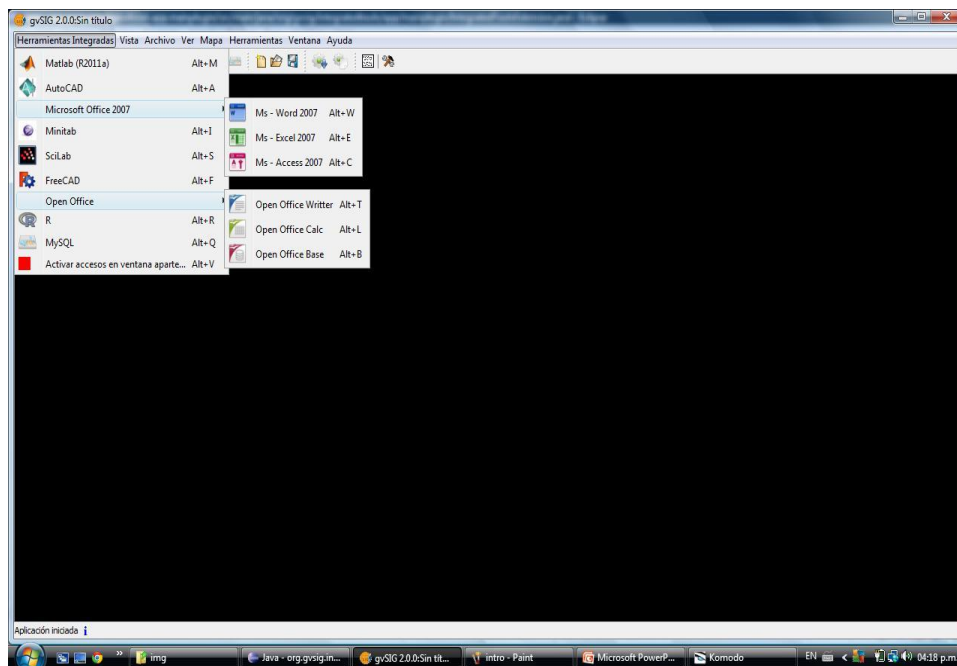


Figura 57. Resultado de la prediccion de los datos.

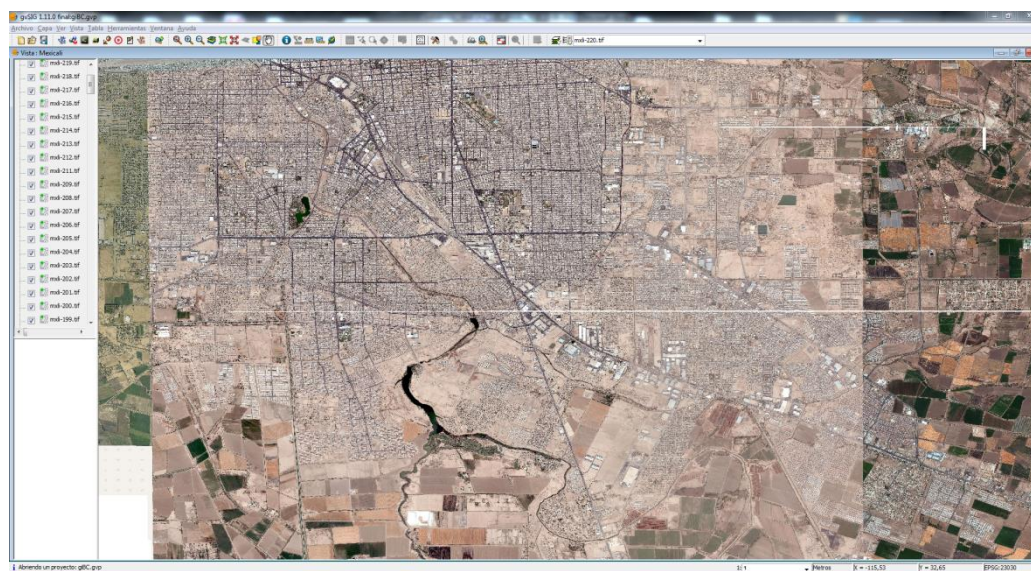


Figura 58. Resultado de la prediccion de los datos.

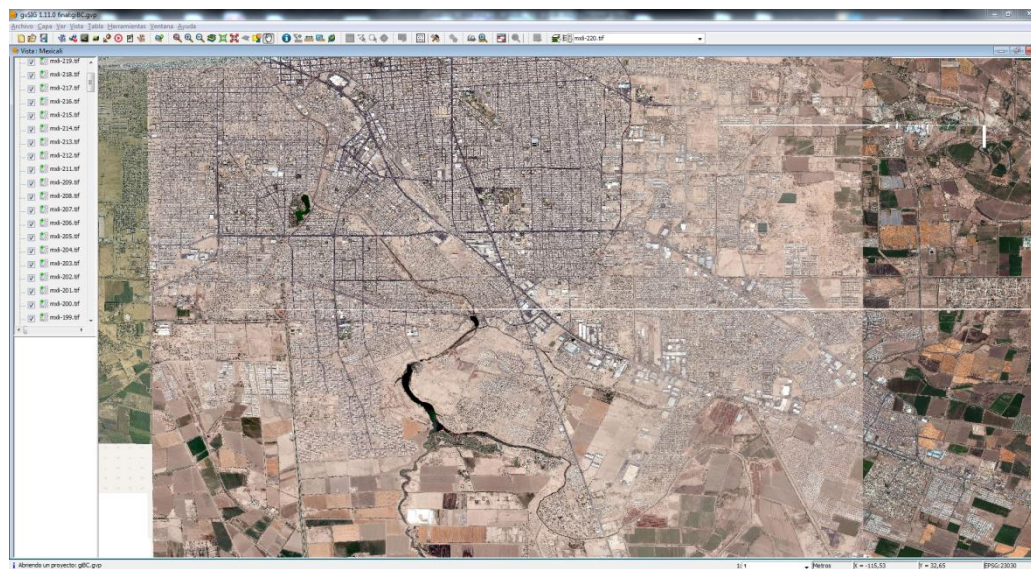


Figura 59. Resultado de la prediccion de los datos.

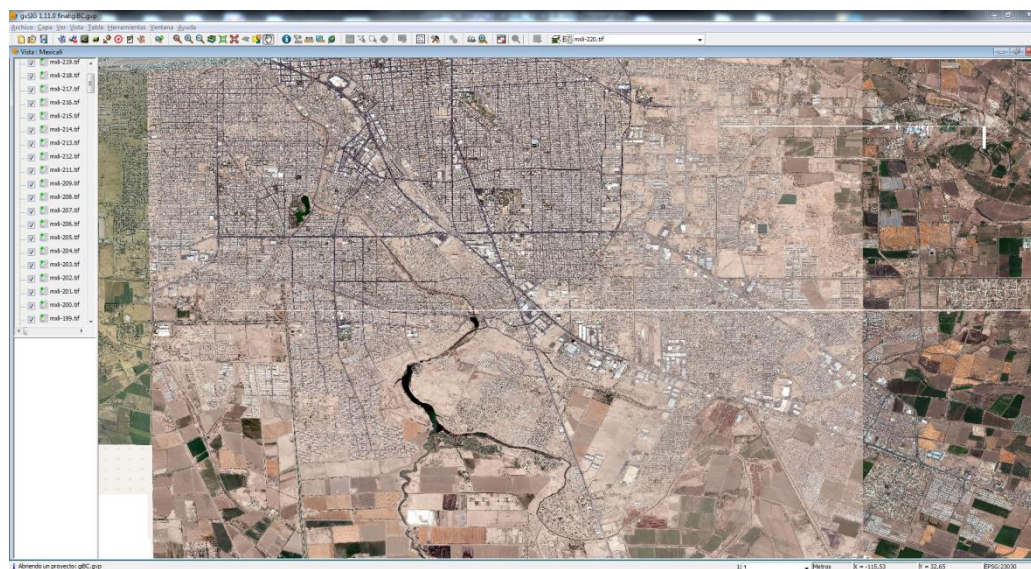


Figura 60. Resultado de la prediccion de los datos.

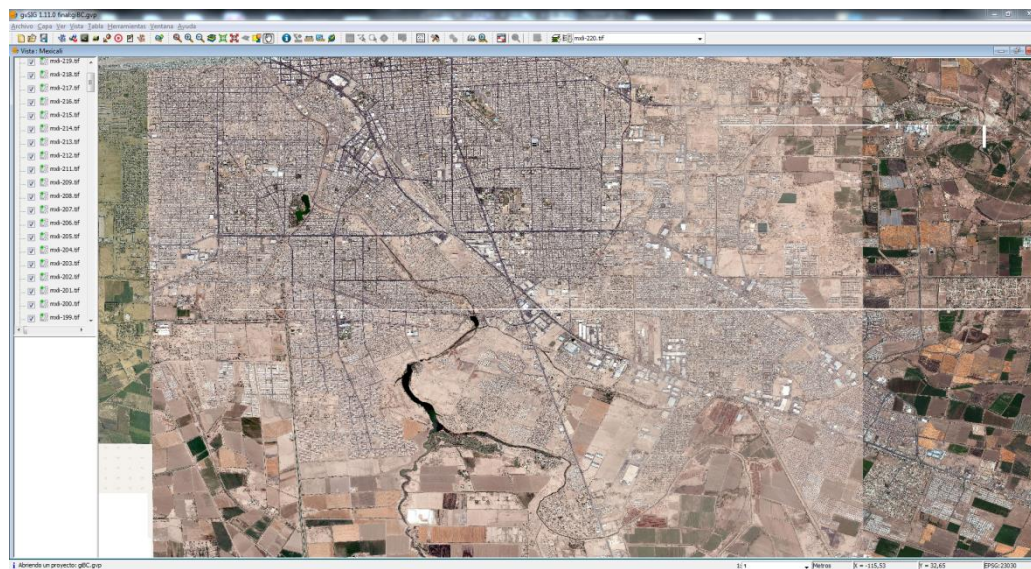


Figura 61. Resultado de la prediccion de los datos.

CAPÍTULO V

5. Conclusiones y Recomendaciones

El integrar diferentes herramientas dentro de una aplicación SIG puede ser un arma de dos filos, porque al incrementar las fronteras del alcance del proyecto, se puede crear una herramienta muy pesada y de difícil manipulación para que el usuario pueda lograr sus objetivos. O por el contrario tener una herramienta robusta con un amplio potencial de manejo y de carácter altamente intuitivo para el usuario final.

El implementar estas distintas herramientas que utilizamos para el desarrollo de nuestros SIG en un concepto minimalista, es decir, solo lo esencial, es un camino para atacar esta necesidad en la que el usuario se encuentra al momento de desarrollar un SIG. Con una herramienta capaz de coexistir con otras herramientas en tiempo de ejecución se puede tener una aplicación capaz de ayudar en el soporte a la toma de decisiones, apoyando con algoritmos de IA que nos brinden otra perspectiva sobre la información que manejamos.

Por lo tanto un trabajo futuro importante sería importante continuar este proyecto y fortalecer potencial que se puede lograr incorporando más técnicas de la IA que ayude a la toma de decisiones y aprovechar el alcance que tiene un proyecto de código abierto al permitir poder implementar nuevas herramientas con forme requiera el tipo de análisis que necesitamos.

REFERENCIAS BIBLIOGRÁFICAS

- Azpurua, M., & Ramos, K. (2010). *A comparison of spatial interpolation methods for estimation of average electromagnetic field magnitude*. Progress In Electromagnetics Research M, Vol. 14, 135-145, 2010. doi:10.2528/PIERM10083103.
- Barreto, C., Ordieres, J., & Castejón, L. (2009). *Minería de datos aplicada a la mejora de procesos de extrusión de elastómeros*. Industrial Engineering. Universidad Politécnica de Valencia.
- Bertsekas, D., & Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific.
- Bolstad, P. (2005). *GIS Fundamentals: A first text on Geographic Information Systems, Second Edition*. White Bear Lake.
- Brambila, A., & Jiménez, R. (2012). *UN ESTUDIO SOBRE LA ENERGÍA EÓLICA EN MÉXICO: APLICACIÓN DE INNOVACIONES TECNOLOGICAS POSIBLES PARA LA GENERACIÓN DE*

ENERGÍA ELÉCTRICA. Recuperado de <http://www.repositoriodigital.ipn.mx/handle/123456789/6421> el 16 de Agosto de 2011.

Cadenas, E., & Rivera, W. (2010). *Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model*. Renewable Energy, Volume 35, Issue 12, December 2010, Pages 2732-2738, ISSN 0960-1481, 10.1016/j.renene.2010.04.022.

Caldera, M. (2000). *Potencial de la energía eoloelectrica en México*. Greenpeace México. pp. 16-17.

Cancino, Y., Gutiérrez, A., & Xiberta, J. (2011). *Current state of wind energy in Mexico, achievements and perspectives*. Renewable and Sustainable Energy Reviews, Volume 15, Issue 8, October 2011, Pages 3552-3557, ISSN 1364-0321, 10.1016/j.rser.2011.05.009.

Carolín, M., & Fernandez, E. (2008). *Analysis of wind power generation and prediction using ANN: A case study*. Renewable Energy, Volume 33, Issue 5, May 2008, Pages 986-992, ISSN 0960-1481, 10.1016/j.renene.2007.06.013.

Cofupro. (s.f.). *Sistema de Información Geográfica para la Identificación y Ubicación de las Zonas Vitícolas de Baja California*. Recuperado de http://www.cofupro.org.mx/cofupro/archivo/fondo_sectorial/Baja%20California/19baja.pdf el 12 de Noviembre del 2011.

Cota, E. (2010). *CARACTERIZACIÓN SOCIO-ECONÓMICA DE LAS FUGAS DE GASES REFRIGERANTES. CASO: SECTOR RESIDENCIAL MEXICALI, BAJA CALIFORNIA*. Instituto de Ingeniería, Maestría y Doctorado en Ciencias e Ingeniería, Universidad Autónoma de Baja California.

Dutra, R., & Szklo, A. (2008). *Assessing long-term incentive programs for implementing wind power in Brazil using GIS rule-based methods*. Renewable Energy 33(12): 2507-2515.

Fernández, P. (2002). *Energía Eólica*. DEPARTAMENTO DE INGENIERÍA UNIVERSIDAD DE CANTABRIA .

Free Software Foundation, Inc. (2006). *Categorías de Software Libre y No Libre*. Recuperado de <http://www.analfatecnicos.net/archivos/53.CategoriasDeSoftwareLibreyNoLibre.pdf> el 15 de Noviembre de 2011.

Freeman, J., & Skapura, D. (1991). *Neural Networks: Algorithms, Applications, and Programming Techniques*. Addison-Wesley.

Geman, S., Bienenstock, E., & Doursat, R. (1992). *Neural networks and the bias/variance dilemma*. Neural Computation 4, pp. 1-58.

- González, D. (2004). *REDES NEURONALES ARTIFICIALES Y MAPAS AUTOORGANIZADOS*. Universidad de Burgos.
- Gonzalez, G., Verdugo, D., & Coronel, J. (2006). *Nociones básicas de los sistemas de información geográfica (S.I.G.)*. Recuperado de <http://dspace.ups.edu.ec/handle/123456789/204> el 15 de Octubre de 2011.
- González, M. (2009). *Producción de bioenergía en el norte de México: Tan lejos y tan cerca...* Frontera norte v.21 n.41 México ene./jun. 2009, ISSN 0187-7372.
- González, M., Ortega, A., Beltrán, F., & Troyo, .. (2006). *Potencial de aprovechamiento de la energía eólica para la generación de energía eléctrica en zonas rurales de México*. Revista de ciencia y tecnología de América, ISSN 0378-1844, Vol. 31, Nº. 4, 2006 , págs. 240-245).
- González-Ávila, M., & et.al. (2006). *Potencial de aprovechamiento de la energía eólica para la generación de energía eléctrica*. Interciencia, ISSN 0378-1844, INCI v.31 pp 240-245 Recuperado de http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0378-18442006000400002&nrm=iso 30 de Mayo de 2010.
- González-Ávila, M., Beltrán-Morales, L., Troyo-Diéguez, E., & A., O.-R. (2006). *Potencial de aprovechamiento de la energía eólica para la generación de energía eléctrica*. Interciencia, ISSN 0378-1844, INCI v.31 pp 240-245 Recuperado de http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0378-18442006000400002&nrm=iso 30 de Mayo de 2010.
- GvSIG. (s.f.). *Quiénes somos?* Recuperado el 10 de Octubre de 2012 de <http://www.gvsig.org/web/organization/quienes-somos/quienes-somos>.
- Hair, J., Rolph, A., & Tatham, R. (1987). *Multivariate Data Analysis*. 2nd ed., Macmillan New York, NY.
- Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA.
- Hervé, A. (2007). *Discriminant Correspondence Analysis*. Program in Cognition and Neurosciences, MS: Gr.4.1, The University of Texas at Dallas.
- Holland, S. (2008). *NON-METRIC MULTIDIMENSIONAL SCALING (MDS)*. Department of Geology, University of Georgia, Athens, GA 30602-2501.
- Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. Wiley_Interscience.

- INEGI. (2010). *Principales resultados por localidad 2010 (ITER) - Baja California*. Consultado el 5 de marzo de 2011.
- INEGI. (s.f.). *Cuentame...* Recuperado el 03 de Mayo de 2010 de <http://cuentame.inegi.org.mx/>.
- Jursa, R., & Rohrig, K. (2007). *Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models*. Institut für Solare Energieversorgungstechnik (ISET) e.V., Königstor 59, D-34119 Kassel, Germany: An Advanced Statistical Method for Wind Power Forecasting," Power Systems, IEEE Transactions on , vol.22, no.1, pp.258,265, Feb. 2007.
- Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., & Yan, Z. (2008). *A review on the forecasting of wind speed and generated power, Renewable and Sustainable Energy Reviews*. Volume 13, Issue 4, May 2009, Pages 915-920, ISSN 1364-0321, 10.1016/j.rser.2008.02.002.
- Leñero, J., & Cruces, S. (2005). *IMPLEMENTACIÓN DE UN ALGORITMO PARA LA SEPARACIÓN CIEGA DE FUENTES CON PEQUEÑOS RETARDOS*. Universidad de Sevilla, DPTO. TEORÍA DE LA SEÑAL Y COMUNICACIONES.
- Lew, D. (2000). *Alternatives to coal and candles: wind power in China*. Energy Policy, Volume 28, Issue 4, April 2000, Pages 271-286, ISSN 0301-4215, 10.1016/S0301-4215(99)00077-4.
- Linares, G. (1990). *Análisis de Datos*. Universidad de La Habana, MES, Cuba, 349-354.
- Linares, G. (2001). *Escalamiento Multidimensional: Conceptos y Enfoques*. Departamento de Matemática Aplicada, Facultad de Matemática y Computación, Universidad de La Habana.
- Lloyd, S. (1982). *Least Squares Quantization in PCM*. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-28, NO. 2, MARCH 1982, pp 129.
- López, A., Demaestri, M., Barotto, O., Viale, S., Degioanni, A., & Zupan, E. (2002). *Aplicación de un Sistema de Información Geográfica (SIG), en el manejo de "Sirex noctilio" F. "La avispa barrenadora de los pinos", en el valle de Calamuchita - Córdoba - Argentina*. Dialnet. Retrieved at October 29, 2012, from the website temoa : Open Educational Resources (OER) Portal at <http://www.temoa.info/node/510094>.
- López, G., & Marín, M. (2010). *Proyecto Valle Verde, una aplicación de triple hélice para el impulso de la generación y uso de las energías renovables: solar y eólica en Baja California*. Redes de colaboración academia y sector industrial: casos exitosos.
- Luke, B. (s.f.). *K-Means Clustering*. Recuperado de <http://fconyx.ncifcrf.gov/~lukeb/kmeans.html> el 11 de Enero de 2010.

- MacQueen, J. (1967). *Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, University of California Press, 1:281-297.
- Marín, J. (s.f.). <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf>. Recuperado de <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf> el 12 de Mayo de 2010.
- Merino, A., J., E., & Marauri, P. (1995). *Aplicaciones de los sistemas de información geográfica en los estudios geomorfológicos y medioambientales el mapa sintético de riesgos potenciales y el mapa de erosión*. Lurralde: Investigación y espacio, ISSN 0211-5891, Nº 18, 1995, págs. 257-291).
- Mora, H. (2011). *SVD: descomposicion en valores singulares: resultados, aplicaciones, cálculo*. Universidad Central, Bogotá.
- Morrison, J. (s.f.). *AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODELS (ARIMA)*. Recuperado de <http://www.forecastingsolutions.com/arima.html> el 12 de Abril de 2012.
- Oficina Estatal de Información para el Desarrollo Rural Sustentable OEIDRUS. (s.f.). *Sistema de Información Geografica del Sector Agropecuario de Baja California (SigaBC)*. Recuperado de http://www.oeidrus-bc.gob.mx/oeidrus_bca/siga.html el 10 de Marzo de 2010.
- OMPI. (s.f.). *Portal de la OMPI sobre las patentes y el sistema internacional de patentes*. Recuperado de <http://www.wipo.int/patentscope/es/> el 12 de Mayo de 2011.
- Osuna, B. (2012). *Analisis de competitividad tecnológica para la electrificación rural del Municipio de Ensenada, México*. E-Prints Complutense.
- Paliulionis, V. (2000). *Intelligent GIS: Architectural Issues and Implementation Methods*. Informatica, Lith. Acad. Sci., 11(3):269-280, 2000.
- Quinn, J. (1992). *Intelligent Enterprise*. Free Press, New York, NY.
- Ramachandra, T., & Shruthi, B. (2005). *Wind energy potential mapping in Karnataka, India, using GIS*. Energy Conversion and Management 46(9-10): 1561-1578.
- Rios, R. (2009). *Métodos de descomposición en valores singulares para tensores*. Biblioteca Sede Medellín Universidad Nacional de Colombia Recuperado de <http://www.bdigital.unal.edu.co/1797/> el 11 de Julio de 2010.

- Rodríguez, C., Boullón, M., Cordero, M., Miranda, D., & Crecente, R. (2007). *Desarrollo de una herramienta de código abierto para la gestión de espacios en la universidad*. A I Jornadas de SIG Libre. Girona: Universitat. [Consulta: 18 diciembre 2008]. Disponible a: <http://hdl.handle.net/10256/1204>.
- Sammon, J. (1969). *A nonlinear mapping for data structure analysis*. IEEE Transactions on Computers 18: 401–409.
- Sanchez, C. (2009). *Análisis Multivariante*. Departamento de Estadística e Investigación Operativa, Universida de Santiago de Compostela.
- Sardón, J. (2002). *ENERGÍAS RENOVABLES PARA EL DESARROLLO*. THOMSON PARANINFO.
- Sepulveda, J., & et.al. (2011). *Sistema de pronósticos para el potencial de incendios forestales en baja california, mexico, Mapping centroamerica y el caribe*. Revista Mapping Mayo-Junio 2011 pp26.
- Sfetsos, A. (2000). *A comparison of various forecasting techniques applied to mean hourly wind speed time series*. Renewable Energy, Volume 21, Issue 1, 1 September 2000, Pages 23-35, ISSN 0960-1481, 10.1016/S0960-1481(99)00125-1.
- Shouman, M. (2008). *Industrial site selection: An intelligent GIS-based decision analysis approach*. ASCE Journal, U.S.A.
- Smith, M. (1993). *Networks for Statistical Modelling*. New York: Van Nostrand Reinhold.
- StatSoft. (s.f.). *Multidimensional Scaling*. Recuperado de <http://www.statsoft.com/textbook/multidimensional-scaling/#general> el 20 de Octubre del 2011.
- Svozil, D., KvasniEka, V., & Pospichal, J. (1997). *Introduction to multi-layer feed-forward neural networks*. Chemometrics and Intelligent Laboratory Systems 39 (1997) 43-62.
- Thomas, C. (2001). *Introducción a la programación orientada a objetos con Java*. España: McGraw-Hill Interamericana de España, ISBN: 84-481-3194-0.
- Thurston, J. (2002). *GIS & Artificial Neural Networks: Does Your GIS Think?* GISVision Magazine. February, 2002.
- Ubaldo, M., Saldaña, F., & Morales, M. (2003). *El Sistema de Información Geográfica para las Energías Renovables (SIGER) en México*. Boletín IIE, octubre-diciembre del 2003.
- Ullman, J. (2002). *Principles of Database Systems*. Computer Science Press. 2da ed. .

Valera, F. (2003). *Introducción a los Sistemas de Bases de Datos*. La Habana.

Waypointgijon. (s.f.). *Escala de Beaufort*. Recuperado de http://www.waypointgijon.com/wabpoint_cp_041.htm el 1 de Enero de 2012.

Wieland, A., & Leighton, R. (1987). *Geometric analysis of neural network capabilities*. San Diego, CA: 1st IEEE Int. Conf. on Neural Networks, Vol. 3, p. 385.

Wikipedia. (s.f.). *Java (lenguaje de programación)*. Recuperado de [http://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)) el 12 de Mayo de 2012.

Wikipedia. (s.f.). *Proceso de Descomposicion de Valores Singulares*. Recuperado de <http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/Singular-Value-Decomposition.svg/512px-Singular-Value-Decomposition.svg.png> el 15 de Febrero de 2012.

Wu, W., Walczak, B., Massart, D., Heurding, S., Emi, F., Last, I., et al. (1996). *Artificial neural networks in classification of NIR spectral data: Design of the training set*. Chemom. Intell. Lab. Syst. 33 , pp. 35-46.

Zupan, J., & Gasteiger, J. (1993). *Neural Networks for Chemists*. New York: VCH.