

**CARACTERIZACIÓN DE UN MÓDULO DE RECONOCIMIENTO
DE PATRONES PARA UN SERVICIO TIC BASADO EN NARICES
ELECTRÓNICAS**

**PAULA ANDREA LÓPEZ HIGUERA
LUISA FERNANDA RUIZ JIMÉNEZ
YOJANES ANDRÉS RODRÍGUEZ VELÁSQUEZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2015**

**CARACTERIZACIÓN DE UN MÓDULO DE RECONOCIMIENTO
DE PATRONES PARA UN SERVICIO TIC BASADO EN NARICES
ELECTRÓNICAS**

**PAULA ANDREA LÓPEZ HIGUERA
LUISA FERNANDA RUIZ JIMÉNEZ
YOJANES ANDRÉS RODRÍGUEZ VELÁSQUEZ**

**Trabajo de grado para optar al título de
Ingeniero(a) Electrónico**

**Director
HOMERO ORTEGA BOADA
PhD. Ingeniería
Codirector
DANIEL ALFONSO SIERRA BUENO
PhD. en Ingeniería Biomédica**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2015**

AGRADECIMIENTOS

Agradecemos a la Universidad Industrial de Santander y a la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, por formarnos como grandes ingenieros y personas integrales.

Al profesor Homero Ortega Boada como director y al profesor Daniel Sierra Bueno como codirector, por su tiempo, apoyo y dedicación a la realización del presente proyecto de grado. Quienes revisaron la versión inicial de este documento y lo enriquecieron constantemente con sus opiniones.

A la Universidad de Pamplona y su Grupo de Investigación en sistemas multisensoriales y reconocimiento de patrones, dirigido por el profesor Oscar Gualdrón, por el apoyo y recursos prestados para la realización del presente proyecto grado.

Finalmente agradecemos a cada uno de nuestros compañeros que nos llenaron de sonrisas, experiencias y recuerdos que nunca se olvidarán.

*A Dios por haber sido mi guía a lo largo de mi carrera,
A mis padres Rosario del pilar Higuera Acacio y Carlos Julio López Sanabria por
ser junto con mi hermano Juan Carlos mi más grande apoyo y motivación,
A mis amigos y compañeros de trabajo de grado que juntos formamos esa gran
molécula de agua sólida y estable, que estuvo siempre conmigo en esos buenos y
malos momentos de esta experiencia de vida,
A mi amore Anderson Atuesta por su apoyo incondicional y por alegrarme cada
uno de los días,
Y a cada una de esas personas que durante la carrera me brindaron conocimiento,
enseñanza, experiencia y miles de alegrías.
Con cariño Paulis.*

Agradezco infinitamente a aquel hombre que con su mano fuerte y estricta educó bajo principios de nobleza y amor a sus hijas, y junto a la mujer que siempre amó construyeron las bases más sólidas de una familia para superar cualquier obstáculo.

A la más grande educadora de mi vida, que con sus constantes esfuerzos e impresionante paciencia fue dando razón a mi existir. Ella fue mostrándome poco a poco la fortaleza que se debe tener para levantarse de duros amaneceres.

A mis dos mejores amigas, mis hermanas, pues en Vanessa está el espejo en el cual me quiero reflejar por sus virtudes infinitas y en Alejandra su gran corazón y amistad. Su incondicional apoyo me lleva a admirarlas cada día más.

A ese ser que nunca se cansa de saludarnos con una explosión de alegría infinita cuando nos ve, siempre nos entrega su absoluta fidelidad.

Mi familia me ha convertido en la mujer que soy, ahora soy consciente de eso...

Gracias

Luisa Fernanda Ruiz Jiménez

Este libro representa la culminación de un ciclo de formación en mi vida, durante los últimos cinco años llegaron y pasaron personas que de alguna manera contribuyeron a mi formación tanto personal como académica. Intenté aprender de cada una de ellas y también aportar algo a sus vidas; profesores, estudiantes, compañeros, amigos, conocidos de amigos... a todos agradezco por permitirme sacar partida de nuestra interacción, ya que es de la forma en la que estoy acostumbrado a aprender. También agradezco las oportunidades presentadas, los momentos de zozobra por alguna situación académica o personal, y las alegrías de cada semestre; a mi mamá y mis hermanas por apoyar mi progreso y celebrar conmigo mis triunfos, mi primera familia; a mis amigas Paulis y Nanda por acompañar este camino durante cuatro años, y a sus familias porque también se convirtieron en las mías, mi segunda familia; a quienes también han apoyado mi crecimiento musical, Banda Bonita y La Jodencia Ska; a los maestros que hicieron que me enamorara de mi carrera. Finalmente solo me queda agradecer a ese algo que me ha impulsado a seguir adelante, a continuar después de cada derrota y fortalecerme en el proceso.

Yojanes Rodriguez

CONTENIDO

	pág
INTRODUCCIÓN	19
1 ENFOQUE CONCEPTUAL	22
1.1 LA NARIZ ELECTRÓNICA	22
1.2 LENGUAJE UNIFICADO DE MODELADO	24
1.2.1 Diagramas de casos de uso.	26
1.2.2 Diagramas de clases.	26
1.2.3 Diagramas de secuencia.	27
1.3 LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN . .	28
1.4 PLATAFORMAS DE SERVICIOS PARA EL INTERNET DE LAS COSAS	30
2 ESPECIFICACIONES TÉCNICAS DE LA NARIZ ELECTRÓNICA AC- TUAL A LA LUZ DE LA NECESIDAD DE UNA REINGENIERÍA	33
2.1 HARDWARE NARIZ ELECTRÓNICA	33
2.1.1 Módulo neumático.	34
2.1.2 Módulo de sensado.	35
2.1.2.1 Características de los sensores.	36
2.1.3 Módulo de control.	39
2.1.4 Módulo de adquisición de datos.	40
2.1.5 Módulo de alimentación.	41
2.2 SOFTWARE NARIZ ELECTRÓNICA	42
2.2.1 Adquisición de datos.	43
2.2.2 Carga de archivos.	44
2.2.3 Pre-procesado.	44

2.2.4	Análisis de datos.	46
2.2.4.1	Análisis estadístico.	47
2.2.4.2	Redes neuronales.	50
2.2.4.3	Máquinas de soporte vectorial.	53
2.2.5	Selección de datos.	56
2.2.5.1	Método secuencial eliminación regresiva.	57
2.2.5.2	Método secuencial selección progresiva.	58
2.2.5.3	Recocido simulado.	59
3	REPRESENTACIÓN EN DIAGRAMAS UML DEL SISTEMA DE RECO- NOCIMIENTO DE PATRONES	61
3.1	DIAGRAMA DE CASOS DE USO.	61
3.2	CLASES DE LA ETAPA DE ADQUISICIÓN.	64
3.3	SECUENCIA DE LA ETAPA DE ADQUISICIÓN.	65
3.4	CLASES DE LA ETAPA DE PROCESAMIENTO.	67
3.5	SECUENCIA DE LA ETAPA DE PROCESAMIENTO.	70
4	PROPUESTA DE UN SERVICIO TIC BASADO EN NARIZ ELECTRÓNICA	73
4.1	CARACTERÍSTICAS BÁSICAS DEL SERVICIO	73
4.1.1	Capa de sociedad: Usuario.	75
4.1.2	Capa de tecnologías de usuario: Dispositivos.	75
4.1.3	Capas de conectividad y acceso al medio: Red de comunicación y ar- quitectura del software.	77
4.1.4	Capa de <i>cloud computing</i> y aplicaciones: Patrón de Modelo Vista Con- trolador.	77

4.2 LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE LA LÓGICA DE NEGOCIO	79
4.2.1 MATLAB como alternativa de implementación.	79
4.2.2 Lenguajes de programación orientados a objetos.	81
5 CONCLUSIONES	86
6 RECOMENDACIONES PARA TRABAJOS FUTUROS	87
BIBLIOGRAFÍA	88
ANEXOS	92

LISTA DE FIGURAS

		pág
1	Proceso de funcionamiento general de la nariz electrónica	24
2	Etapas del funcionamiento básico de un servicio TIC	28
3	Modelo de capas de las plataformas IoT	31
4	Esquema del módulo general del hardware de la nariz electrónica. . .	34
5	Esquema del módulo neumático de la nariz electrónica.	35
6	Configuración matricial de los 16 sensores y vista del encapsulado. . .	36
7	Esquema del módulo de sensado de la nariz electrónica.	36
8	Circuito básico para realizar mediciones con los sensores Figaro serie 8.	37
9	Curvas de caracterización de sensibilidad del TGS 800R.	38
10	Esquema del módulo de control de la nariz electrónica.	39
11	Esquema del módulo de adquisición de datos.	40
12	Esquema del módulo de alimentación.	42
13	Panel de inicio del software Bnose.	42
14	Panel de inicio del software Bnose.	43
15	Menú para cargar archivos en software Bnose.	44
16	Menú elección de pre procesado, software Bnose.	45
17	Interfaz wavelet, software Bnose.	46
18	Panel de selección, menú análisis de datos	47
19	Interfaz PCA, software Bnose.	48
20	Interfaz Análisis de clúster, software Bnose.	49
21	Interfaz análisis de covarianza, software Bnose.	50
22	Panel de PNN de la interfaz de procesamiento	52
23	Panel de MLP de la interfaz de procesamiento	53
24	La máxima distancia de cada una de las clases con respecto a la frontera.	54
25	Interfaz para parámetros de análisis con SVM	55
26	Interfaz para parámetros de análisis con LS-SVM	56
27	Panel de selección de datos	57
28	Panel de SBS de la interfaz de procesamiento	58
29	Panel de SFS de la interfaz de procesamiento	58
30	Panel de SA PNN de la interfaz de procesamiento	59
31	Panel de SA SVM de la interfaz de procesamiento	60
32	Diagrama Casos de uso, software Bnose.	61
33	Diagrama de clases etapa adquisición	65
34	Diagrama de secuencia etapa de adquisición	66
35	Diagrama de clases etapa de procesamiento.	69
36	Diagrama secuencia etapa de procesamiento.	72
37	Modelo de capas para la propuesta de servicio.	74
38	Esquema del dispositivo de sensado.	76
39	Patrón MVC del servicio TIC basado en nariz electrónica.	78

40	Esquema concepto general de máquina virtual	80
41	Gráfico estadístico de la popularidad lenguajes de programación por CodeEval	81
42	Gráfico estadístico del índice en la calidad del software por TIOBE . . .	82
43	Curvas de caracterización de sensibilidad del TGS 800R.	92
44	Curvas de caracterización de sensibilidad del TGS 813.	93
45	Curvas de caracterización de sensibilidad del TGS 821.	93
46	Curvas de caracterización de sensibilidad del TGS 822.	94
47	Curvas de caracterización de sensibilidad del TGS 825.	94
48	Curvas de caracterización de sensibilidad del TGS 826.	95
49	Curvas de caracterización de sensibilidad del TGS 830.	95
50	Curvas de caracterización de sensibilidad del TGS 831.	96
51	Curvas de caracterización de sensibilidad del TGS 832.	96
52	Curvas de caracterización de sensibilidad del TGS 842.	97
53	Curvas de caracterización de sensibilidad del TGS 880.	98
54	Curvas de caracterización de sensibilidad del TGS 882.	98

LISTA DE TABLAS

	pág
1 Convención de relaciones UML	25
2 Elementos utilizados en los diagramas de casos de uso.	26
3 Elementos utilizados en los diagramas de clases.	27
4 Elementos utilizados en los diagramas de secuencia.	27
5 Condiciones eléctricas del circuito estándar.	37
6 Especificaciones generales de la tarjeta de control de potencia.	40
7 Especificaciones generales de la tarjeta de adquisición de datos NI USB-6210.	41
8 Caracterización usuario para la primera versión del servicio.	75
9 Comparación lenguajes de programación orientados a objetos	82

LISTA DE ANEXOS

pág

ANEXO A. Curvas de caracterización de sensibilidad de los sensores Figaro 8. . 92

RESUMEN

TITULO: CARACTERIZACIÓN DE UN MÓDULO DE RECONOCIMIENTO DE PATRONES PARA UN SERVICIO TIC BASADO EN NARICES ELECTRÓNICAS. *

AUTORES: PAULA ANDREA LÓPEZ HIGUERA, LUISA FERNANDA RUIZ JIMÉNEZ, YOJANES ANDRÉS RODRÍGUEZ VELÁSQUEZ. **

PALABRAS CLAVE: DIAGRAMAS UML, INTERNET DE LAS COSAS, LENGUAJES ORIENTADOS A OBJETOS, MATLAB, NARIZ ELECTRÓNICA, RECONOCIMIENTO DE PATRONES, SERVICIOS TIC, SERVICIOS WEB.

DESCRIPCIÓN:

Esta investigación permite comprender el concepto, funcionalidad y componentes de un sistema multisensorial de gases químicos para el reconocimiento de patrones de olores, que es capaz de emular las funcionalidades de una nariz biológica. Para ello, se presenta la caracterización de cada una de las partes que conforman el software y hardware del sistema mediante un estudio de especificaciones técnicas, seguido del desarrollo de diagramas UML con el fin de tener una visión como desarrollador y diseñador del módulo de reconocimiento de patrones.

Estos aportes sobre la nariz electrónica permiten hacer una redefinición de la misma, con miras a la portabilidad y atención de múltiples usuarios, lo que describe comúnmente a un servicio TIC que está asociado al concepto del internet de las cosas. Por consiguiente se realiza una propuesta del servicio describiendo las características de la implementación y entregando herramientas que permiten comprender las bases y cada uno de los roles a desempeñarse durante la ejecución de una primera versión del servicio, a partir de las potencialidades del sistema actual.

Finalmente, como complemento de análisis para las principales partes que conforman la solución de servicio itinerante se lleva a cabo una evaluación de los posibles lenguajes para la implementación del procesamiento de datos en la nube, buscando el que mejor se adapte a las condiciones deseadas.

* Trabajo de grado.

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director. PhD Homero Ortega Boada

ABSTRACT

TITLE: CHARACTERIZATION OF A PATTERN RECOGNITION MODULE FOR A TIC SERVICE BASED ON ELECTRONIC NOSES. *

AUTHORS: PAULA ANDREA LÓPEZ HIGUERA, LUISA FERNANDA RUIZ JIMÉNEZ, YOJANES ANDRÉS RODRÍGUEZ VELÁSQUEZ. **

KEYWORDS: ELECTRONIC NOSE, INTERNET OF THINGS, MATLAB, OBJECT-ORIENTED PROGRAMING LANGUAGES, PATTERN RECOGNITION, TIC SERVICES, UML DIAGRAMS, WEB SERVICES.

DESCRIPTION:

This research helps to understand the concept, functionality and components of a multisensory system of chemical gases for pattern recognition of smells, which can emulate the functions of a biological nose. To do this, the characterization of each of the parts that make software and hardware system through a study of technical specifications, followed by the development of UML diagrams in order to have a view as a developer and designer of the patterns recognition module is presented.

Through this analysis can be performed for a redefinition of the electronic nose, to provide to portability and treatment multiple users, which commonly this describes a TIC service that is associated with the concept of the internet of things. Therefore, a proposal arises about the service to describe the characteristics of the implementation providing tools to understand the basis and each of the roles to perform during the execution of a first version of the service, from the potential of the current system.

Finally, in addition to analysis for the main parts that integrating of solution to the roaming service performed an assessment of the possible languages for implementing data processing in the cloud, in order to find one that best suits the desired conditions.

* Bachelor Thesis.

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director. PhD Homero Ortega Boada

INTRODUCCIÓN

Los seres humanos estamos bendecidos con una exquisita gama de sentidos y sorprendente habilidad de comprensión del mundo que nos rodea. Gracias a ellos, casi de forma automática, podemos relacionarnos con los objetos del entorno, guiándonos constantemente en la vida. La vista y el oído son los sentidos que creemos son dominantes, sin embargo, el olfato, gusto y tacto intervienen de gran forma en nuestro comportamiento y acciones tomadas en cada instante.

Al igual que para un humano, un perro se siente influido por sus sentidos, sin embargo, el sistema olfativo es el sentido más importante que ha podido desarrollar. A través de este puede identificar objetos con mayor precisión que con cualquier otro sentido. La nitidez olfativa que le proporciona la composición de su nariz y el procesamiento del cerebro, lo ha llevado a utilizarse en beneficio de los humanos, por ejemplo, han sido entrenados para identificar olores determinados como explosivos o drogas, incluso existen perros que realizan alertas médicas cuando hay subidas de azúcar, ataques epilépticos entre muchas más.

La fascinación por la perfección y equilibrio de la naturaleza, particularmente en animales como los perros, ha llevado a la ciencia y tecnología a transformarse alrededor de los sentidos y empezar a darle forma a los objetos con los que interactuamos cotidianamente. Se ha desarrollado toda una industria entorno a la idea de replicar el sistema olfativo biológico denominándolo nariz electrónica.

Esto se ha sumado al aumento exponencial que se está presentando con respecto al uso de las nuevas tecnologías de la información y la comunicación (TIC), generando un gran impacto en el sector industrial, doméstico, médico e incluso investigativo. Por tal razón para el Grupo de Investigación Interdisciplinario en Radiocomunicaciones y Sistemas de Información Georreferenciada (RadioGis) y el Grupo de Investigación en Control, Electrónica, Modelado y Simulación (CEMOS) de la Universidad Industrial de Santander (UIS), es importante ser parte de este crecimiento a nivel tecnológico influenciando a todo el clúster de grupos de investigación (CentroTIC) de la UIS en busca de un objetivo común.

Pensar en el aporte significativo a un país de conflictos internos cambiando la historia de un soldado que lucha por sobrevivir en campamentos de guerra sobre terrenos minados, o a la detección precoz de enfermedades crónicas en pacientes aparentemente sanos, inquieta a este grupo de investigadores en realizar soluciones altamente efectivas a la sociedad. Es así como se ha pensado en que la nariz electrónica juega un papel importante en una propuesta de solución, convertirla en un servicio itinerante a la comunidad agrupa características positivas en pro del desarrollo social.

Este ambicioso objetivo se apoya en el avance obtenido por el grupo de investigación en sistemas multisensoriales y reconocimiento de patrones de la Universidad de

Pamplona (UP), con la implementación de un sistema multisensorial basado en sensores de gases químicos, cuyo software está desarrollado en el entorno de MATLAB (MATrix LABoratory) y denominado bajo el nombre de Bnose.

Sin embargo, este sistema presenta características lejanas a lo que es un servicio TIC, desde el enfoque de ser portable, de bajo costo y de acceso a múltiples usuarios, ya que es un dispositivo de gran tamaño, diseñado para hacer pruebas a solo un conjunto de muestras, además de depender directamente de una conexión cableada a un computador para los procesos de adquisición y procesamiento de la información, elevando el costo propio del sistema.

En pocas palabras, para hacer el sueño realidad de convertir la nariz electrónica en un servicio itinerante que entregue soluciones a la comunidad, es necesario realizar una reingeniería. Sin embargo, para aplicar mejoras pertinentes, se debe en primer lugar hacer un estudio detallado del sistema. Bajo este horizonte, en el segundo capítulo de este documento se entrega la información de las especificaciones técnicas de la nariz, haciendo posible identificar sus características significativas observando las falencias, ventajas y desventajas que presenta.

Para aplicar la visión de ingeniería del software, se presenta en el tercer capítulo de este libro, una generalización del sistema desde el enfoque de desarrollo por medio de la herramienta de modelado UML (por sus siglas en inglés *Unified Modeling Language*), exportando la primera aproximación del software actual al paradigma de orientación a objetos, aprovechando las ventajas que presenta este tipo de programación al ser la arquitectura más utilizada en la implementación de servicios TIC.

Así mismo se realiza un proceso de evaluación de los posibles lenguajes de programación orientados a objetos, sobre el cual se llevaría a cabo el desarrollo del servicio, sin dejar de lado las consideraciones de utilizar el entorno MATLAB para el procesamiento en la nube. Esta información se encuentra expuesta en el cuarto capítulo, en donde adicionalmente se busca implementar soluciones convergentes siendo necesario que las personas implicadas también lo sean, uniendo fuerzas de diferentes áreas como la electrónica, química, sistemas, entre otros. En consecuencia, se aborda una vista genérica por medio de la clasificación en capas del funcionamiento del servicio ubicando a cada uno de los actores en las funciones que debe desempeñar. Además, se utilizan diferentes diagramas que familiarizan el desarrollo y los aportes del ingeniero electrónico y de sistemas con el fin de comunicarlos efectivamente.

Finalmente por medio de este documento se logran extraer las bases fundamentales que permiten entender el concepto y funcionamiento de la nariz electrónica como un sistema artificial de identificación y clasificación, así como la posibilidad de encontrar en esta múltiples características que hacen viable la idea de pensar en una primera versión de servicio, ya sea empleando máquinas virtuales (MATLAB) o mediante la traducción a un lenguaje de programación más familiar al desarrollo en la

nube, como Python.

Por consiguiente, se convierte en una herramienta de información y planeación de tareas base para la reingeniería de la nariz electrónica como servicio TIC, en donde la formulación de las ideas propuestas en este trabajo han sido desarrolladas en conjunto con el CentroTIC y la Universidad de Pamplona que participaron activamente como tutores y co-equipo, por lo tanto, los resultados de este trabajo son compartidos con ellos. Por otro lado, se le permite a los grupos RadioGis y CEMOS usar libremente el contenido de este trabajo para sus actividades naturales de investigación y desarrollo.

1. ENFOQUE CONCEPTUAL

Es importante observar que los principales investigadores en el área del reconocimiento de patrones no se encuentran comúnmente familiarizados con las características, funcionamiento, desarrollo e implementación de servicios TIC. De manera análoga, los expertos en servicios de las comunicaciones buscan constantemente la solución por medio del software y hardware que esté a su alcance, frecuentemente de problemáticas orientadas a la experiencia de clúster de empresas, grandes o pequeñas industrias e incluso, personas del común, sin la necesidad de desarrollar grandes investigaciones y creación de conocimientos, sino por el contrario, en la observación y experiencias de dichos clientes. Por lo tanto, resulta interesante la unión de estas dos fuerzas expertas en cada una de sus áreas y orientarla hacia el desarrollo científico y el aporte tecnológico que implica prestar un servicio orientado al uso de nariz electrónica, que sin lugar a duda permitirá generar nuevas condiciones de vida en el futuro.

Así pues, con el fin de establecer una misma terminología, conocer el estado actual y el marco en la que se desarrollan conceptos generales como principalmente la nariz electrónica, servicios de la tecnología de la información y comunicación, y el modelado de lenguaje unificado, se realiza una búsqueda bibliográfica, consulta a personal pertinente, entre otros, y se compilan en este capítulo los conceptos e ideas que se utilizan para analizar el problema de investigación que se aborda. En consecuencia, no es necesario que el lector sea un experto en el desarrollo de las dos principales áreas relacionadas en este trabajo, pues solo si está dispuesto a familiarizarse con el lenguaje teórico, este capítulo le permitirá rápidamente convertirse en un especialista para la comprensión del desarrollo del estudio realizado.

Finalmente estos aportes se aplicarán al cumplimiento del objetivo de este trabajo de grado orientado a la caracterización del sistema multisensorial de gases que se encuentra en la Universidad de Pamplona, además de desarrollar un panorama general que permite al lector resolver sus dudas y realizar aportes críticos sobre la implementación de un servicio TIC orientado a las narices electrónicas.

1.1. LA NARIZ ELECTRÓNICA

La capacidad para distinguir olores le permite al ser humano buscar alimento y discriminarlo según calidad, estado y tipo. Detectar información sobre el ambiente, le hace posible identificar situaciones relacionadas con presencias odoríficas creando así una relación con el recuerdo mental. En general, esta facultad de los seres que juega un gran papel en la vida, es un sistema sensorial principiante en las investigaciones de la rama tecnológica y por lo tanto, poco popular en las aplicaciones diarias. El afán del hombre por controlar la naturaleza le ha permitido generar grandes avances en todos los campos, siendo así la ciencia y la tecnología la puerta que ofrece soluciones a casi todos los problemas diarios para realizar labores más sencillas y óptimas.

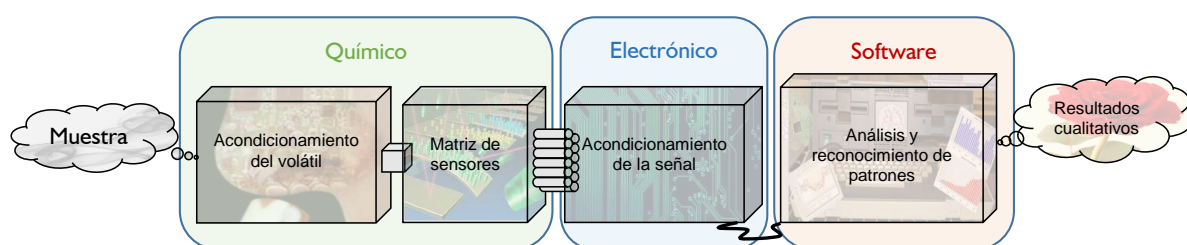
Desde el siglo XX la revolución científica ha traído aportes importantes a la industria, ha podido proveer de sistemas artificiales que emulan algunas funciones de los sentidos humanos a una generación que automatiza tareas garantizando seguridad, confiabilidad y economía. Alrededor de los 60's se crea un dispositivo que se puede considerar como el indicio de la nariz electrónica. Para entonces, la compañía Bacharach Inc., construyó un sensor de gas al cual llamaron "Sniffer", pero no se le consideró como una nariz debido a la incapacidad de reconocer patrones variados de olores (Iveth Moreno, 2009). En esta misma década apareció la propuesta de WF. Wilkens con la publicación "An electronic analog for the olfactory" que pone a prueba el uso de sensores a base de micro-electrodos análogos (Florez, 2007). Según Iveth Moreno y otros (2009), los trabajos históricos de la nariz se remontan a los avances de la Universidad de Warwick en Gran Bretaña y en Argonne National Laboratory (ANL) en Estados Unidos, que en los ochentas trabajaron paralelamente en la posibilidad de imitar las funciones olfativas. Por otra parte, Krishna Persaud y George Dodd orientaron sus estudios al olfato biológico, pensando en sensores de óxidos metálicos, y más tarde polímeros conductores, siendo los primeros en demostrar que una matriz de sensores entrega resultados discriminatorios dependiendo del compuesto. Al mismo tiempo en Estados Unidos se propuso la detección, identificación y medición de una gama de químicos, mezclas, entre otros. Adicionalmente, los estudios realizados en Japón y sus aplicaciones sobre la frescura de peces con sensores de la familia metal-óxido (MOX por sus siglas en inglés *metal-oxide*), generaron gran interés sobre los continentes adyacentes, iniciando el desarrollo de la tecnología de matrices de sensores olfativos. Desde entonces algunos avances importantes y la mayoría de narices electrónicas diseñadas son utilizadas en ámbitos como la seguridad para el gobierno, equipos hospitalarios, medio ambiente, entre otras aplicaciones.

Una nariz electrónica está definida como un "instrumento que comprende una agrupación de sensores químicos con sensibilidades parcialmente solapadas junto a un sistema de reconocimiento de patrones, capaz de analizar y reconocer aromas simples o complejos" (Julian W. Gardner, 1994); en esencia su configuración y funcionamiento emula el sistema olfativo humano. Se diferencia de los sistemas de instrumentación químicos tradicionales, tales como los cromatógrafos de gases debido a que estos analizan los componentes específicos que conforman una muestra. Por el contrario, el olfato electrónico permite un análisis en conjunto, sin identificar particularmente las moléculas presentes, además del uso de algoritmos de inteligencia artificial. Esto es posible gracias a la matriz de sensores químicos que entregan impulsos eléctricos que son interpretados mediante algoritmos computacionales que explotan la multidimensionalidad de los datos permitiendo entrenarse y aprender para clasificar los aromas reconociendo patrones odoríferos, imitando al cerebro humano.

Un sistema olfativo humano comienza desde la nariz, el aire aspirado actúa sobre las neuronas receptoras olfativas, aproximadamente 6 millones, que se prolongan sobre el bulbo olfativo para enviar la información al cerebro con el fin de identificar el aroma del ambiente. El cerebro crea recuerdos de patrones para posteriormente utilizarlos en la tarea de discriminación de aromas (Durán, 2005). En similitud, dentro de un sistema de nariz electrónica, existen tres áreas importantes que se relacionan

entre sí, las cuales son el sistema químico, electrónico y software como se muestra en la figura 1. El proceso químico inicia cuando la muestra es acondicionada para permitir su paso hacia una matriz de sensores. En el caso de ser resistivos, dichas partículas actúan sobre el conjunto de electrodos que tienen una impedancia específica base referida a la composición del aire, generando en cada polímero del sensor un cambio de tamaño y por lo tanto de su resistencia. Esto origina así, señales que pasan por la etapa de acondicionamiento y procesamiento de selección de datos relevantes, con el fin de entregar la huella digital del compuesto a la fase de reconocimiento de patrones de la nariz para detectar, clasificar e identificar la muestra. Actualmente, es usual encontrar que la etapa de procesamiento se implemente por medio de redes neuronales artificiales, pues permite reconocer más compuestos que sensores disponibles gracias a su filosofía de aprendizaje constante.

Figura 1. Proceso de funcionamiento general de la nariz electrónica



Basado en (Durán, 2005, Figura 1.5: Secuencia para el análisis de aromas)

En conclusión, son variados los trabajos que se han realizado para optimizar y convertir la nariz electrónica en una herramienta más accesible, portable, confiable y económica. Por esta razón, la idea de convertirla en un servicio masivo e itinerante, es una propuesta ambiciosa pero no imposible.

1.2. LENGUAJE UNIFICADO DE MODELADO

Un modelo es una representación abstracta y conceptual de un fenómeno real estudiado con el fin de comunicar, especificar y comprender la estructura de un sistema complejo. Gracias al modelado de un proceso, es posible descubrir características de simplificación y optimización, siendo así una herramienta poderosa para el análisis tanto minucioso como general de una actividad científica. Por consiguiente, en el diseño y construcción de software, se hace necesario el uso de técnicas y metodologías para especificar el funcionamiento de este. El lenguaje unificado de modelado (UML por sus siglas en inglés *Unified Modeling Language*), nace como método para facilitar la comprensión de describir, detallar, construir procesos, funciones y aspectos de relación a lenguajes de programación. Enrique Hernández define que “cualquier rama de ingeniería o arquitectura ha encontrado útil desde hace mucho tiempo la representación de los diseños de forma gráfica (...). Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para

servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: *Unified Modeling Language*). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño” (E. Hernández, 2012). Este lenguaje gráfico está respaldado por el Grupo de Gestión de Objetos (OMG por sus siglas en inglés *Object Management Group*), que es una organización sin ánimo de lucro, dedicada a la gestión de diversos estándares de tecnologías orientadas a objetos. UML recibió la aprobación en el 2005 como un estándar por la ISO.

A pesar de que el UML se utiliza generalmente durante la etapa de diseño, a lo largo de este libro se utiliza como medio de descripción de un sistema existente debido a que se hace necesario, en pro de una futura reingeniería del sistema, presentar los elementos de software en un lenguaje de rápido acceso y dándole un enfoque de orientación a objetos para una implementación por módulos. Es por ello que este proyecto se apoya en dos áreas conceptuales del UML: la estructura estática (diagrama de casos de uso y diagramas de clases) y el comportamiento dinámico (diagramas de secuencia) del sistema. En la estructura estática se presentan los conceptos del sistema como clases, mientras que los objetos pertenecientes a cada clase son elementos que almacenan información e interactúan con otros objetos para definir el comportamiento del sistema, así mismo el comportamiento dinámico modela la interacción entre cada objeto mostrando la historia del mismo a lo largo de todo el proceso (James Rumbaugh, 2000, Perspectiva general de UML, Áreas conceptuales).

Tabla 1. Convención de relaciones UML

<i>NOMBRE</i>	<i>SÍMBOLO</i>	<i>DESCRIPCIÓN</i>
<i>Asociación</i>	————	Relación entre dos elementos del diagrama
<i>Asociación unidireccional</i>	————→	Relación entre dos elementos del diagrama en un solo sentido
<i>Generalización</i>	————→	El elemento que se encuentra en la punta de la flecha puede heredar sus propiedades a los elementos del inicio del conector
<i>Extendido</i>	--<<extend>>→	La dirección de la flecha indica que el elemento de inicio puede o no ser necesario para el funcionamiento del elemento al que llega, resulta una extensión del elemento de llegada




Basado en (“Desarrollar modelos para el diseño de software”, 2014)

En la tabla 1 se especifican en general las formas de relación que utilizan los diagramas descritos en el capítulo 3 siendo utilizadas en su mayoría por los diagramas de caso de uso y de clases. En las siguientes secciones se describen las convenciones

propias de cada diagrama y su utilidad en la descripción del sistema.

1.2.1. Diagramas de casos de uso. En los diagramas de casos de uso se presenta el sistema desde la perspectiva del usuario y los actores que intervienen en el proceso, por ende cada caso de uso representa una acción con la que puede estar relacionado un actor y esa interacción se da en el sistema o a través de este (James Rumbaugh, 2000, Vista de los casos de uso). En la tabla 2 se pueden observar las convenciones de las figuras utilizadas en este documento para la representación de los casos de uso.

Tabla 2. Elementos utilizados en los diagramas de casos de uso.

<i>NOMBRE</i>	<i>SÍMBOLO</i>	<i>DESCRIPCIÓN</i>
<i>Sistema</i>		Sistema o aplicación en la que se está trabajando, enmarca los casos de uso que describen su función
<i>Actor</i>		Representa un usuario o organización que interactúa con el sistema
<i>Caso de uso</i>		Representa las acciones que realizan los actores y/o el sistema para conseguir un objetivo

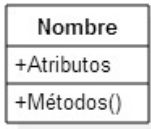
Basado en (“Desarrollar modelos para el diseño de software”, 2014)

1.2.2. Diagramas de clases. Evidencian la estructura básica del sistema ya que cada clase representa un concepto y el conjunto de elementos que lo constituyen, convirtiéndose así en el pilar del modelo ya que se usa para mostrar lo que el sistema puede hacer y cómo puede ser construido.

Cada clase se representa con un cuadrilátero dividido en tres secciones, como se puede ver en la tabla 3, donde en cada sección se especifica:

- Nombre: Hace referencia al nombre de la clase o etiqueta que la representa.
- Atributos: Hace referencia a las propiedades que la representan y variables de las que esta hace uso.
- Métodos: Hace referencia a las operaciones que puede realizar la clase, representando la forma de interacción de cada objeto con su entorno.

Tabla 3. Elementos utilizados en los diagramas de clases.

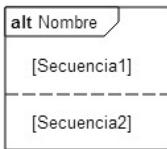

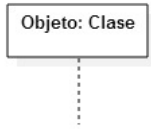
<i>NOMBRE</i>	<i>SÍMBOLO</i>	<i>DESCRIPCIÓN</i>
<i>Clase</i>		Conceptos del sistema, abarcan un grupo de objetos con propiedades similares
<i>Multiplidad</i>	*, N, M..N	Indica cuantos objetos de un extremo se pueden vincular a los objetos del otro extremo. Muchos, N y solo N enteros, de M a N enteros

Basado en (“Desarrollar modelos para el diseño de software”, 2014)

1.2.3. Diagramas de secuencia. Al contrario de los diagramas de clases, en un diagrama de secuencia se muestra el flujo a través del tiempo de un proceso, por ende se ven involucrados los actores, el caso de uso descrito y las clases, pretendiendo así representar las acciones necesarias para cumplir con las funcionalidades del sistema. El tiempo fluye en sentido vertical de arriba hacia abajo, mientras que los objetos, actores y casos de uso se comunican horizontalmente por medio de mensajes o acciones representados por una flecha en la dirección del objeto que recibe el mensaje o realiza la acción (James Rumbaugh, 2000, Vista de interacción)

En la tabla 4 se muestra la descripción de cada uno de los elementos utilizados en este diagrama para la descripción del software Bnose.

Tabla 4. Elementos utilizados en los diagramas de secuencia.

<i>NOMBRE</i>	<i>SÍMBOLO</i>	<i>DESCRIPCIÓN</i>
<i>Alt</i>		Contiene secuencias de mensajes alternativos, cada una representa un camino que puede tomar el flujo del proceso
<i>Loop</i>		Indica una secuencia que se repite tantas veces como se especifique en la variable N
<i>Línea de vida</i>		Representa un objeto perteneciente a una clase determinada y su interacción a través del tiempo con otros objetos del sistema. Los actores y casos de uso también poseen una línea de vida

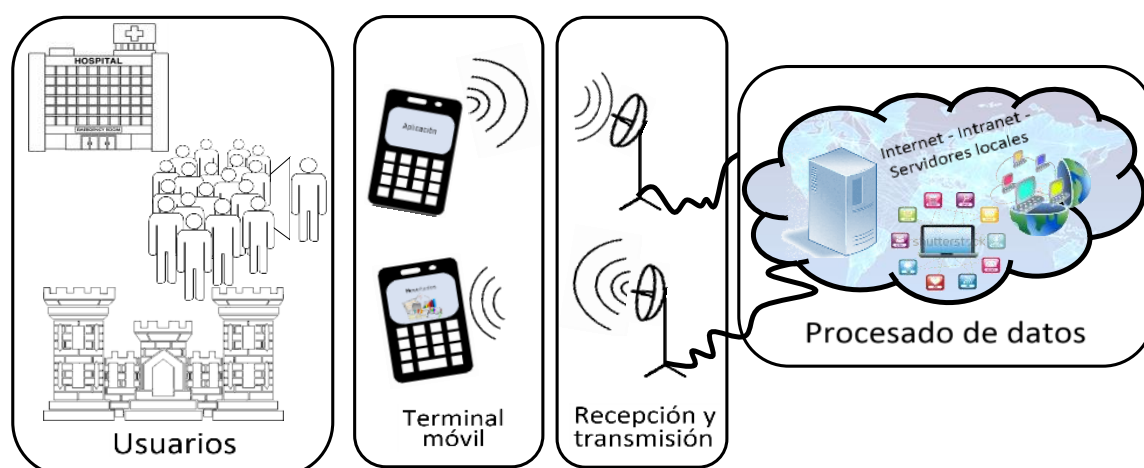
Basado en (“Desarrollar modelos para el diseño de software”, 2014)

1.3. LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

Las tecnologías de la información y comunicación (TIC), son los recursos, procedimientos y técnicas que permiten realizar el almacenamiento, procesamiento y transmisión de la información alojada en la nube. Aunque estas “no son ninguna panacea ni fórmula mágica, pueden mejorar la vida de todos los habitantes del planeta. Se dispone de herramientas para llegar a los Objetivos de Desarrollo del Milenio, de instrumentos que harán avanzar la causa de la libertad y la democracia y de los medios necesarios para propagar los conocimientos y facilitar la comprensión mutua” (Kofi Annan, discurso inaugural de la primera fase de la WSIS, Ginebra, (2003)). Actualmente, la popularización e implementación de las TIC han permitido iniciar la migración de diferentes ámbitos como el Gobierno, Sociedad, Cultura, Seguridad, entre otros, a entornos digitales de fácil acceso.

El funcionamiento de las TIC se basa en la conexión con la nube, concepto que hace referencia a un servicio de computación, y según la *IEEE Computer Society*^{*}, “es un paradigma en el que la información se almacena de manera permanente en servidores de internet y se envía a cachés temporales de cliente” (“Discurso sobre el almacenamiento en la nube”, 2007). Es decir, cualquier sistema móvil de acceso a la red tales como portátiles, celulares, tabletas, SmartTV, entre otros, recibe información constante gracias a los servidores que atienden en cualquier momento las demandas de contenidos en páginas establecidas, finanzas, base de datos, comunicaciones (chat) y muchas más. De esta forma, es posible crear un modelo de servicios en tecnologías que permiten el desarrollo de negocios beneficiando a proveedores y clientes, pues su eficiencia en calidad y tiempo, además de la transparencia del sistema y bajo costo, optimiza los procesos de consumo respondiendo a la ley de oferta-demanda.

Figura 2. Etapas del funcionamiento básico de un servicio TIC



^{*} IEEE Computer Society, organismo líder en proveer información técnica y servicios a estudiantes y profesionales de la computación, la informática y los sistemas a nivel mundial

Un servicio de TIC consta principalmente de las etapas mostradas en la figura 2. Primero se realiza una previa adecuación por medio del sistema electrónico de telecomunicaciones como la depuración y digitalización. La señal que se envía desde la terminal móvil llega al sistema respectivo que tiene como función realizar la transferencia y recuperación de la información. Ésta la redirige al servidor que ejecuta el procesamiento de datos siendo capaz de interpretar la información recibida. Los resultados de salida del ordenador pasan por el mismo proceso anterior siendo enviados nuevamente a la terminal móvil, finalizando el servicio.

El constante intercambio de información desde los orígenes mismos de la vida ha sido, tal vez, la mayor de las promotoras del progreso del desarrollo actual seguido por una revolución social. De esta forma, creció un interés de reglamentar y organizar las telecomunicaciones por parte de los gobernantes del mundo durante la segunda guerra mundial, creando así la primera organización internacional llamada Comité Consultativo Internacional de Telegrafía y Telefonía (CCITT), que más adelante inspiró a la creación de la Organización de las Naciones Unidas (ONU). La CCITT se convirtió en la Unión Internacional de Telecomunicaciones (UIT) a mediados del año 1992 y se mantiene adscrito a la ONU. Sus principales funciones son estudiar, formular y mantener las políticas internacionales en torno de las TIC (Martínez, 2007).

En Colombia, el plan del gobierno conocido como Vive Digital es el esfuerzo más grande que se centra en las TIC intentado “impulsar el gran salto tecnológico a través de la masificación del uso de Internet con el fin de reducir la pobreza y generar empleo. Para lograrlo, el Plan impulsa el ecosistema digital del país conformado por 4 grandes componentes: Infraestructura, Servicios, Aplicaciones y Usuarios. (...) La infraestructura corresponde a los elementos físicos que proveen conectividad digital. (...) Los servicios ofrecidos por los operadores hacen uso de la infraestructura y permiten desarrollar la conectividad digital. (...) Las aplicaciones son herramientas informáticas que permiten a los usuarios (...) realizar una serie de tareas de manera práctica y desde distintos tipos de terminales. (...) Los usuarios hacen uso de las aplicaciones e indirectamente de los servicios e infraestructura para consumir y producir información digital.” (“Plan vive Colombia digital”, 2010). Así surge el término de las Tecnologías de la información y las comunicaciones (TIC) como aquel sistema tecnológico que permite procesar la información y transmitirla de forma remota.

Según la página web informativa (“Grupo de investigación RadioGis”, 2010), actualmente el grupo de investigación RadioGis, tiene un amplio conocimiento en la implementación de servicios, estado de las redes y relaciones con entidades interesadas, gracias a los constantes trabajos orientados a las tecnologías de la comunicación. La integración de esfuerzos de la electrónica, informática y telecomunicaciones en el desarrollo de soluciones útiles e innovadoras que brinda el grupo de investigación, ofrecen un respaldo para el planteamiento de nuevos desafíos del sector.

El Centro de Investigación Científica y de Desarrollo en Tecnologías de la Información y las Comunicaciones (CentroTIC) es una unidad adscrita a la Vicerrectoría de Investigación y Extensión de la UIS que orienta “al desarrollo y uso de las TIC para los diferentes sectores industriales, gubernamentales y sociales de la región y el país” (“Centro de Investigación Científica y de Desarrollo en Tecnologías de la Información y las Comunicaciones (CentroTIC)”, 2012). Al pertenecer a este clúster de investigadores, RadioGis puede complementarse con otras áreas por medio de grupos de investigación de ingenierías, ciencias básicas, salud, entre otros, permitiendo concluir que posee las capacidades para realizar la implementación de la propuesta de la nariz electrónica como servicio itinerante.

1.4. PLATAFORMAS DE SERVICIOS PARA EL INTERNET DE LAS COSAS

La idea de interconectar todos los elementos que se encuentran en el entorno resulta ser real gracias a los avances tecnológicos. Ya hablar de Internet como una red de comunicación entre personas a través de dispositivos limita el verdadero uso que se le está asignando, pues se ha convertido en la forma como una alta gama de objetos infinitos interactúa entre sí para dar información, recibir notificaciones e incluso tomar decisiones (Roca, 2014).

El Internet de las cosas (IoT por sus siglas en inglés *Internet of Things*) se basa principalmente en sensores, redes de comunicación e inteligencia que controla los procesos y datos generados (Roca, 2014). Por lo tanto es posible redefinir la nariz electrónica considerándosele como un objeto que ahora estará conectado a la red y desde allí cumplirá sus funciones sistemáticas. En general, la nariz electrónica se convierte en una de las múltiples opciones que existen en el desarrollo de soluciones IoT, pues se pueden implementar soluciones de todo tipo según las capacidades de los desarrolladores que ya van liderando en este sector.

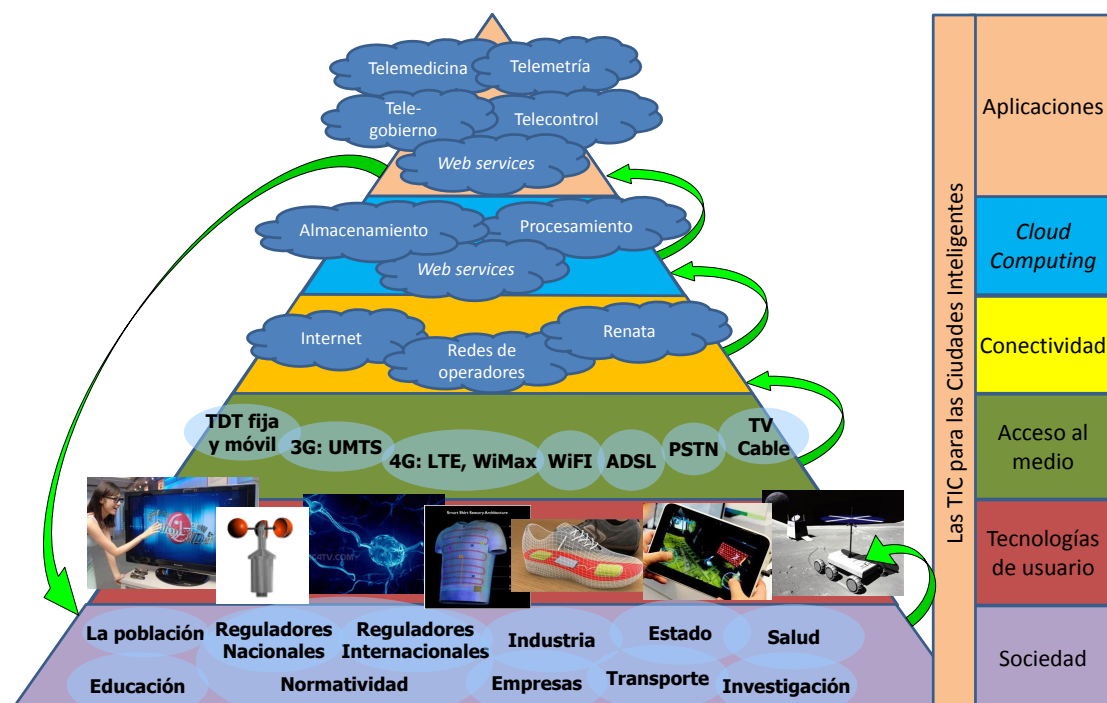
El CentroTIC utiliza una representación en capas para exponer la conformación de las IoT, basándose en modelos conocidos como el de las NGN (*next generation networks*) y las IMT (*International Mobile Telecommunication*), reuniendo en detalle los componentes que interceden en la realidad del servicio. Este modelo integrador, mostrado en la figura 3, permite que todos los actores que se relacionan con el sistema encuentren rápidamente su rol dentro de la industria, reconociendo sus habilidades y aportes (Ortega, 2013).

En la base de la pirámide se encuentra la sociedad, que es un actor real dentro de cualquier servicio, entregando una historia particular, la cual es la principal base del desarrollo de la solución. Como sucede en general, es la imaginación de los usuarios quien saca partido a esta tecnología, por lo tanto el centro TIC debe tener acciones que involucren a personajes que realicen la vigilancia de las necesidades de la industria.

La sociedad se relaciona con la aplicación por medio de la capa que reúne a todas las tecnologías a la que tiene acceso. Aquí se encuentran objetos inteligentes tales como smartphones, Smart-TV, ropa inteligente y cualquier otro dispositivo electrónico que permita capturar señales y realizar cualquier tipo de control, es decir, las cosas que se encuentren conectadas a Internet.

Sin embargo, antes de pensar en la simple interacción de la sociedad con tecnologías del usuario, es necesario que exista algún método de comunicación que permita tener acceso al medio. En esta capa, se encuentran tecnologías tales como TDT fija o móvil, 3G: UMTS, 4G:LTE, WiMax, WiFi, ADSL, PSTN, TV Cable entre muchas más, las cuales permiten ingresar a la capa de conectividad, donde existen redes como Internet, Renata u operadores móviles, que intervienen con servicios que entregan valor a la aplicación IoT, con el uso de amplificadores, priorización de actividades, ancho de banda específico, incluso mensajes cortos (SMS), entre otros. Especificando el proceso de las arquitecturas de comunicación, posteriormente se establece el tipo de procesamiento en la nube (*cloud computing*) la cual permite realizar almacenamiento de datos y cómputo de lógicas (o algoritmos) que entregan funcionalidades a la aplicación, además de contar con el uso de servicios web (Web Services) que brindan a las aplicaciones independencia de las plataformas sobre las que son instaladas, ya que son la integración de varias soluciones implementadas por diferentes organizaciones.

Figura 3. Modelo de capas de las plataformas IoT



Tomado de (Ortega, 2013)

Finalmente, se encuentra en la capa más alta, y como consecuencia de los aportes

de las capas anteriores, la aplicación que entrega a la sociedad la solución esperada. Algunos ejemplos de estos pueden orientarse a la telemedicina, telemetría, telegobierno, telecontrol y muchos más. A pesar de la interrelación y dependencia entre las capas, realmente los sujetos desarrolladores que se encuentran en una específica, generalmente necesita conocer poca información de las otras, pues solo con el cumplimiento de funcionalidades dentro del marco del rol, están contribuyendo con el desarrollo de las tecnologías y en este caso particular, en la realidad de soluciones IoT.

2. ESPECIFICACIONES TÉCNICAS DE LA NARIZ ELECTRÓNICA ACTUAL A LA LUZ DE LA NECESIDAD DE UNA REINGENIERÍA

Para convertir la nariz electrónica en un servicio itinerante que se pueda ofrecer a la comunidad, es necesario analizar aspectos importantes del “sistema multisensorial basado en sensores de gases químicos” desarrollado por la UP. Para ello, se realiza el estudio de las especificaciones técnicas, es decir, describirla a partir de las capacidades reales que la componen y la interacción entre ellas, logrando conocer la nariz electrónica desde el punto de vista de un desarrollador.

Por medio de esta caracterización, cualquier persona (sea investigador, desarrollador, diseñador, entre otros) se encuentra en la capacidad de plantear soluciones a partir de observaciones personales en base a falencias, ventajas y desventajas que pueda extraer según su necesidad o continuar con el mejoramiento de algoritmos de reconocimiento de patrones, ampliación de la cobertura de reacciones químicas de los sensores, entre otros aspectos que conlleven a la reingeniería de la nariz electrónica.

A continuación se presenta detalladamente el hardware y el software del sistema. La descripción de los componentes físicos se basa principalmente en resultados de experimentación, documentación de la Universidad de Pamplona y la Universidad Rovira i Virgili (España), y de las hojas de datos de los sensores Figaro serie 8, estableciendo el funcionamiento por etapas a través de diagramas de bloques por su facilidad de comprensión e identificación de las contribuciones de cada componente, además de ser uno de los diagramas más utilizados para modelar sistemas. Este define las entradas, salidas y organización interna, permitiendo conocer el flujo de señales del proceso. En lo que respecta al software del sistema multisensorial, se muestra cada panel programado por medio de la interfaz gráfica de usuario GUIDE de la herramienta MATLAB. Se muestra también la interacción y el funcionamiento esperado del programa para cumplir con el objetivo de controlar y procesar los datos adquiridos.

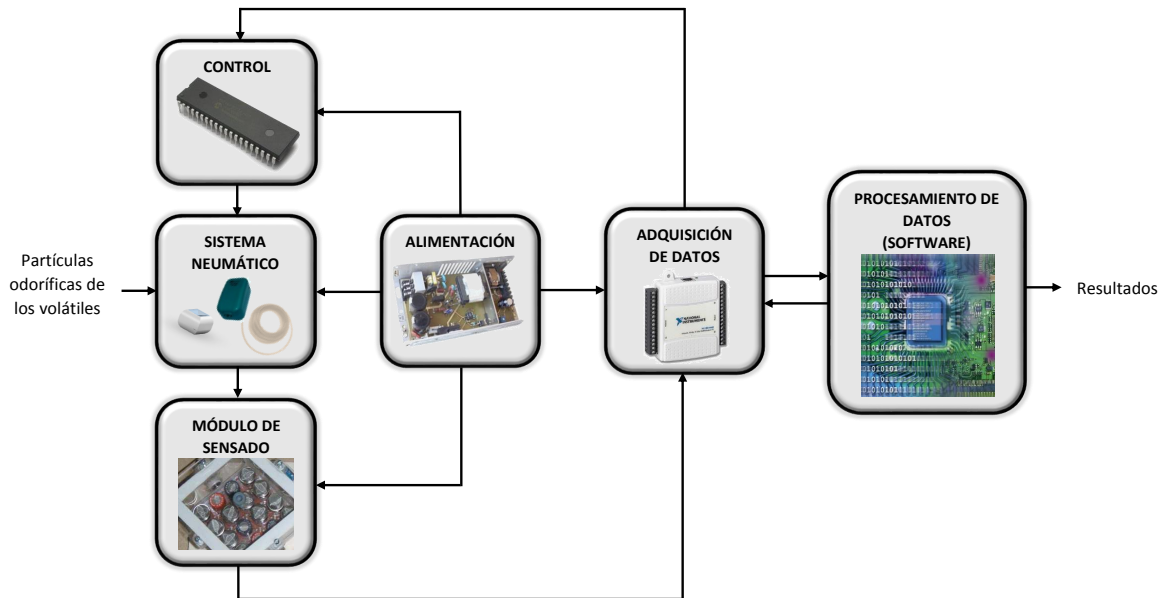
2.1. HARDWARE NARIZ ELECTRÓNICA

El sistema de hardware de la nariz electrónica consta de cinco módulos de funcionamiento relacionados entre sí. En primera instancia, el módulo del sistema neumático, conformado principalmente por electroválvulas, electrobombas, tuberías y cámara de muestras, permite el tratamiento de los volátiles acondicionando las muestras para su recepción por parte del módulo de sensado. En este último módulo, conformado por una matriz de sensores, se generan reacciones electroquímicas que se transforman en señales de tensión, las cuales son muestreadas posteriormente por el módulo de adquisición de señales, el cual permite la comunicación entre el software y el hardware de la nariz electrónica.

Adicionalmente, el sistema general cuenta con un módulo de control que actúa sobre

el sistema neumático y un módulo de potencia que suministra la energía demandada para el funcionamiento del Hardware. A continuación, se analiza en detalle las especificaciones de los componentes propios de cada módulo del hardware, los cuales se muestran en la figura 4.

Figura 4. Esquema del módulo general del hardware de la nariz electrónica.

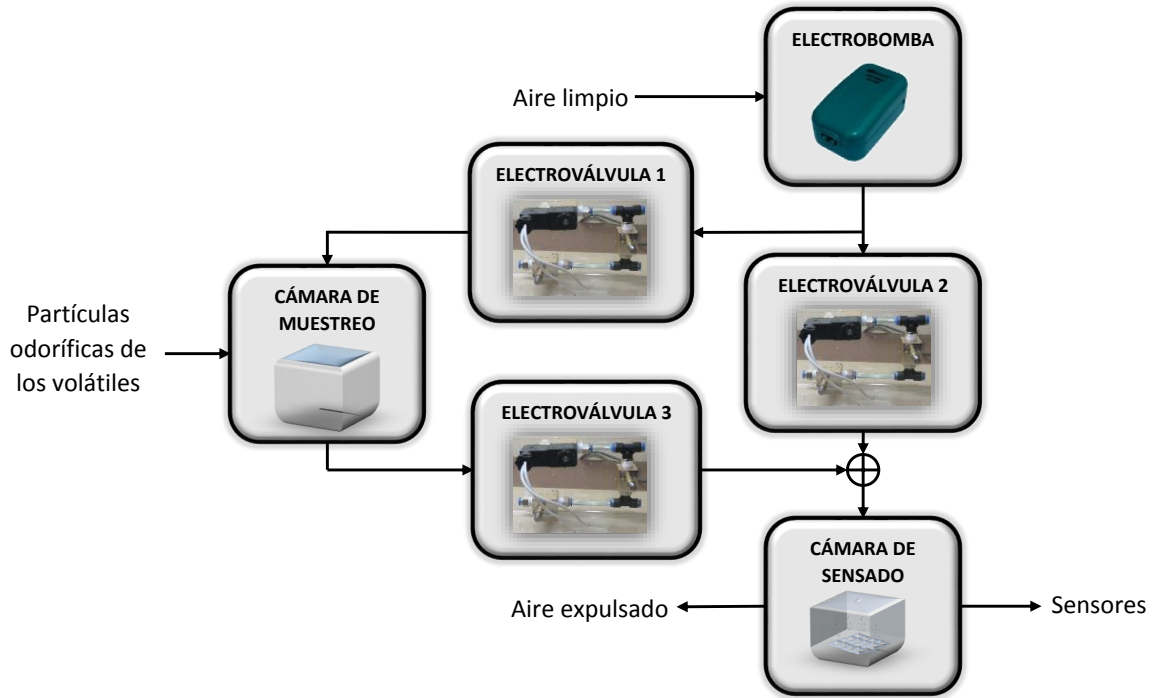


2.1.1. Módulo neumático. El sistema neumático, esquematizado en la figura 5, se encuentra conformado por una cámara de muestreo, la cual corresponde a un recipiente hermético de dimensiones condicionadas al objeto de interés, según la función otorgada a la nariz electrónica. Esta cámara presenta una entrada y una salida del flujo de aire, con el fin de acumular en su interior los volátiles, que serán transportados a través de las tuberías según la fase del procedimiento de medida. A su vez, la estructura de tuberías está complementada por una electrobomba y tres electroválvulas que establecen el flujo de aire.

Existen tres fases de desarrollo para realizar la medición por medio de los sensores. La primera fase corresponde a la concentración de partículas, donde las electroválvulas 1 y 3 correspondientes a la trayectoria hacia la cámara de muestreo se cierran acumulando las muestras adecuadas de volátiles, y a su vez la electrobomba impulsa el aire en la trayectoria de la cámara de sensado donde la electroválvula 2 se encuentra abierta. A continuación se establece la fase de medida, donde el flujo de aire pasa a través de la cámara de muestreo y desplaza las moléculas odoríficas hacia la cámara de sensado. Para esto, las electroválvulas 1 y 3 se abren, y a su vez la electroválvula 2 se cierra. Después de un tiempo prudente para la estabilización de la respuesta de los sensores (se recomienda un aproximado de 10 minutos), se procede a la fase final, denominada limpieza y reposo. Con el fin de retirar los restos

de volátiles presentes en los sensores y llevarlos a su estado de reposo, se redirige el flujo de aire con la misma configuración de la fase de concentración.

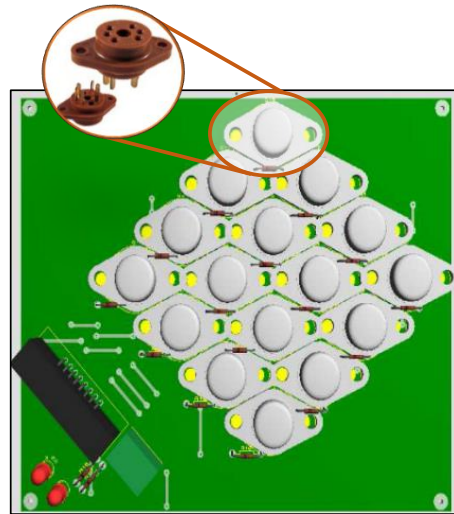
Figura 5. Esquema del módulo neumático de la nariz electrónica.



La nariz electrónica actualmente tiene dos versiones de cámara de muestreo una de vidrio de 7215.4 cm^3 (17.7cm ancho, 26.3cm largo, 15.5cm alto) y una cámara de acrílico de 1728 cm^3 (12cm ancho, 12cm largo, 12cm alto). Una electrobomba de marca RESUN modelo AC1500 con una capacidad de 2.2 l/min, una presión $> 0,016 \text{ MPa}$ y potencia de 3W. Las electroválvulas son solenoides 2/2 way de la marca E.B.C, modelo 91110 2V025-08 de tipo *on/off* de 110 V y consumo de 5.5 VA a 50 mA. Se cuenta además, con una cámara de acrílico transparente cúbica con una entrada y salida de aire de volumen de 401.157 cm^3 para alojar los sensores.

2.1.2. Módulo de sensado. El módulo de sensado, esquematizado en la figura 7 se encuentra conformado por la matriz de sensores de gases y el circuito que acondiciona la señal generada, que se ubica en la cámara de medida. Una configuración matricial permite optimizar el espacio utilizado para la ubicación de los sensores, de esta forma, se minimiza la diferencia de lectura medida entre ellos, como se observa en la figura 6, donde también se representa la vista frontal y posterior de uno de los encapsulados.

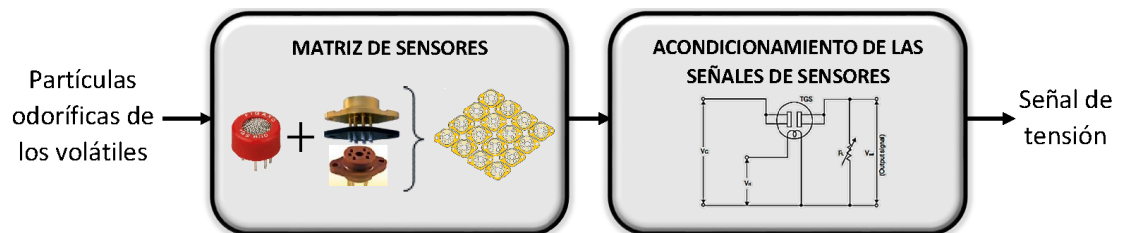
Figura 6. Configuración matricial de los 16 sensores y vista del encapsulado.



Basado en (Rojas, 2010)

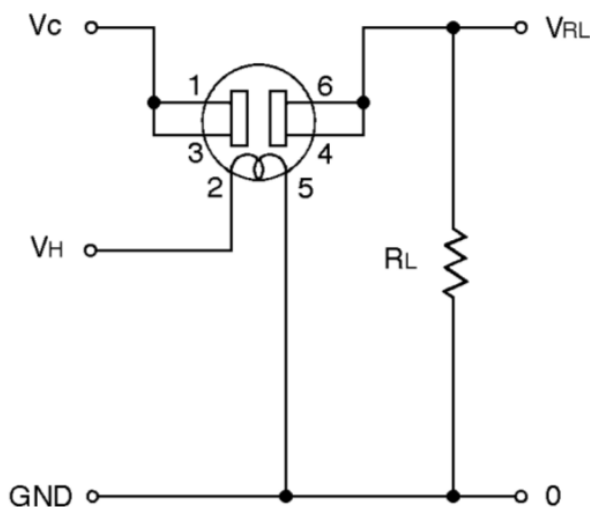
La cantidad y tipo de sensores se encuentran condicionados a la funcionalidad específica de la nariz electrónica, para este caso, el propósito de su diseño ha sido enfocado a pruebas de laboratorio. Por esta razón, su arreglo matricial se encuentra constituido por 16 sensores de 12 tipos distintos de la marca Figaro de la serie 8, de los cuales se presentan características relevantes más adelante para la selección de los mismos en caso de requerir una variación.

Figura 7. Esquema del módulo de sentido de la nariz electrónica.



2.1.2.1. Características de los sensores. Los 16 sensores utilizados en la nariz electrónica son de tipo semiconductor de óxido de estaño, que se caracterizan por su alta sensibilidad ante la presencia de diversos volátiles orgánicos. Su funcionamiento se basa en la variación resistiva de la capa activa, que se produce por la reacción de los volátiles desprendidos por el objeto a analizar (gases reductores) sobre el semiconductor del dióxido de estaño, bajo condiciones específicas de temperatura, alrededor de 400°C.

Figura 8. Circuito básico para realizar mediciones con los sensores Figaro serie 8.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

En general, todos los sensores presentan la configuración circuital básica para realizar mediciones que se muestra en la figura 8. Donde V_C corresponde a la tensión de alimentación, V_H la tensión del calefactor del sensor, R_L es una resistencia asociada, y V_{RL} , la tensión respuesta de salida del sensor. A partir de V_{RL} se puede extraer la resistencia R_S que describe la reacción del sensor como se muestra en la ecuación 1.

$$R_S = \frac{V_C - V_{RL}}{V_{RL}} * R_L \quad (1)$$

En la tabla 5 se muestran las especificaciones máximas de trabajo del circuito de acondicionamiento de cada sensor.

Tabla 5. Condiciones eléctricas del circuito estándar.

<i>VARIABLE</i>	<i>SÍMBOLO</i>	<i>RANGO DE VALORES</i>	<i>OBSERVACIONES</i>
<i>Voltaje calentador</i>	V_H	$5,0 \pm 0,2V$	AC o DC
<i>Voltaje circuito</i>	V_C	Máximo 24V	DC para $P_s \leq 15mW$
<i>Resistencia carga</i>	R_L	Variable	Mínimo 0.45 K Ω

Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

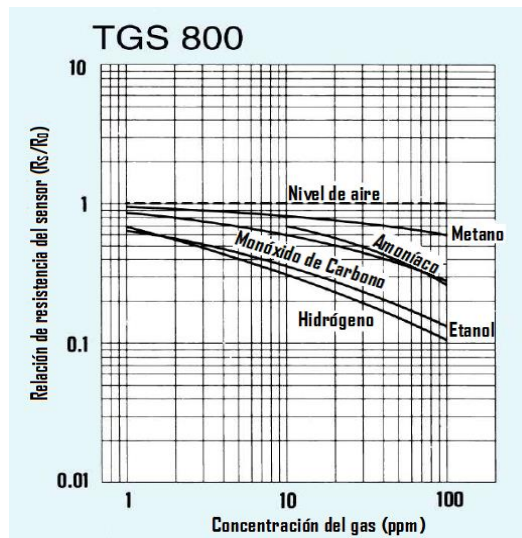
Para caracterizar la sensibilidad correspondiente a cada sensor, se analiza la razón de cambio entre la resistencia media R_s/R_o y la concentración medida en partes por

millón, donde R_s corresponde a la resistencia media del sensor ante la variación de concentración de cualquier tipo de volátil, y R_o la resistencia base ante el volátil al que reacciona cada uno de los sensores específicamente bajo una concentración determinada. Por lo tanto, cuando la relación de resistencia del sensor se aproxima a 1 la respuesta ante el volátil presente tiende a ser igual a la obtenida bajo las condiciones de R_o .

Sensor TGS 800R: Este sensor tiene una aplicación dirigida al control del aire, es decir, reacciona ante presencia de aire contaminado con elementos odoríficos tales como tabaco, gasolina, etc. El TGS 800R, tiene una curva característica de sensibilidad mostrada en la figura 43, con un R_o medido bajo condiciones de aire limpio.

En la curva de sensibilidad se observa que el sensor no presenta reacción alguna ante distintas concentraciones del aire limpio ($R_s = R_o$), cuando existe presencia de componentes distintos responde con una disminución en la resistencia R_s a medida que aumenta dicha concentración. Específicamente el sensor responde más rápido ante la presencia del amoníaco que ante al metano.

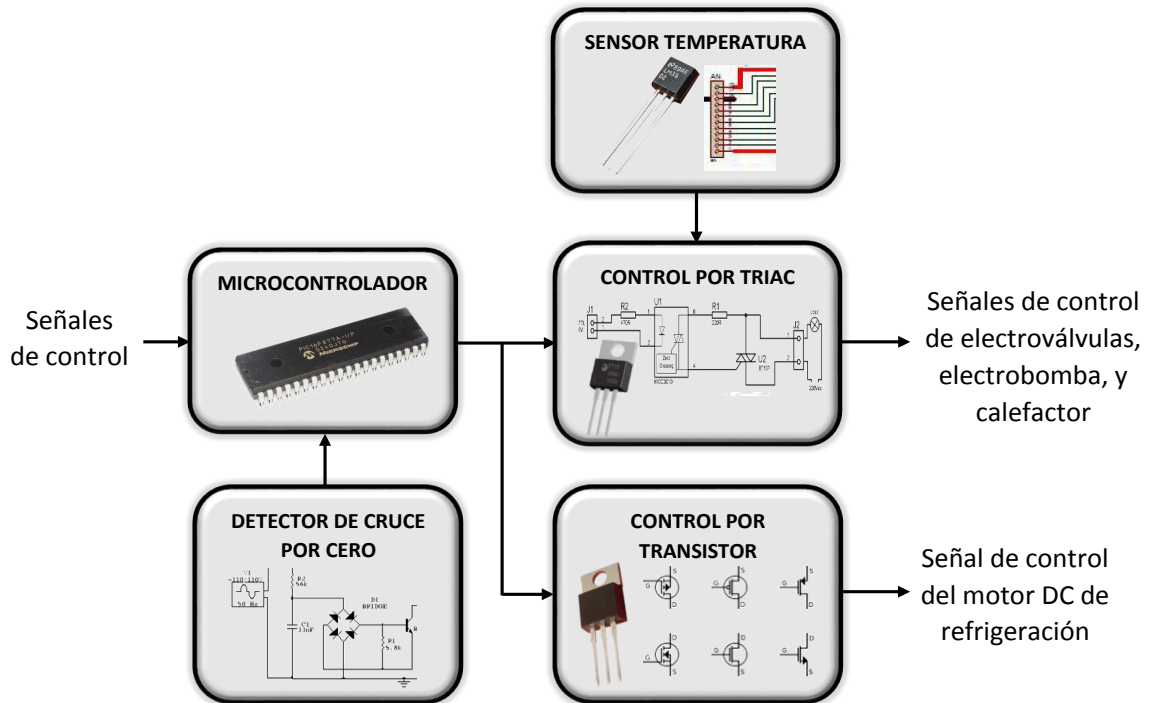
Figura 9. Curvas de caracterización de sensibilidad del TGS 800R.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Para hacer revision en detalle de las curvas asociadas a los otros sensores dirigirse a la sección de anexos.

Figura 10. Esquema del módulo de control de la nariz electrónica.



2.1.3. Módulo de control. El sistema de control, esquematizado en la figura 10, se encarga de controlar la potencia suministrada al módulo del sistema neumático, al motor DC de refrigeración y la calefacción de los sensores. Un microcontrolador recibe órdenes del módulo de adquisición además de señales de un detector de cruce por cero y un sensor de temperatura, con el fin de accionar los elementos necesarios para llevar un flujo de proceso adecuado (proceso descrito en la sección módulo neumático). Actualmente esta etapa se implementa en una PCB con un microcontrolador PIC 16F877A-I/P de cuarenta pines, 8 *triac* BTA12 para control de potencia, 8 opto-acopladores MOC3021 y MOC3041 para la interfaz microcontrolador-triac y un opto-acoplador 4N25 para el circuito de detección de cruce por cero. En la tabla 6 se muestran las características generales de la tarjeta de control de potencia implementada.

Tabla 6. Especificaciones generales de la tarjeta de control de potencia.

<i>GENERAL</i>	<i>CARACTERÍSTICAS</i>
<i>Entrada de control</i>	4 Bit de control correspondientes al puerto b del microcontrolador 16f877a.
<i>Control de AC por Triac</i>	8 Salidas de control.
<i>Entradas análogas disponibles</i>	8 Entradas disponibles y una configuración de hardware para sensor de temperatura lm35.
<i>Control de motor DC o carga en DC</i>	2 Salidas configurables a PWM.
<i>Entrada para detección de cruce por cero para control de potencia en AC</i>	1 Entrada internamente por hardware habilitada por software.
<i>Conector de fase y neutro</i>	1 Conector tipo bornera.
<i>Conector de polarización DC</i>	1 Conector tipo bornera.

Tomado de (Rojas, 2010)

2.1.4. Módulo de adquisición de datos. El módulo de adquisición, esquematizado en la figura 11 permite la comunicación directa entre el hardware y software que conforman la nariz electrónica, ya que consiste en la adquisición de las señales de respuesta del módulo de sensado, y posteriormente son muestreadas, adquiriendo datos que pueden ser procesados por el software almacenado en un computador. Por otra parte, este módulo también recibe señales desde el software del sistema que son redirigidos hacia el módulo de control, realizando acciones específicas sobre el módulo neumático.

Figura 11. Esquema del módulo de adquisición de datos.



Tabla 7. Especificaciones generales de la tarjeta de adquisición de datos NI USB-6210.

<i>ESPECIFICACIONES</i>	<i>ENTRADA ANALÓGICA</i>	<i>ENTRADA/SALIDA DIGITAL</i>
<i>Canales</i>	16 de una sola terminal 8 diferenciales.	4 de salida y 4 de entrada.
<i>Resolución</i>	16bit.	—
<i>Velocidad de muestreo</i>	250 kS/s.	—
<i>Máximo voltaje de entrada</i>	10 V.	Niveles lógicos TTL de entrada y de salida.
<i>Rango voltaje máximo</i>	-10 V – 10 V.	Entrada: 0 V – 5.25 V Salida: 0 V – 3.8 V.
<i>Precisión máxima del rango de voltaje</i>	2,69 mV.	—
<i>Sensibilidad máxima del rango de voltaje</i>	91,6 μ V.	—
<i>Rango de voltaje mínimo</i>	-200 mV – 200 mV.	—
<i>Precisión mínima del rango de voltaje</i>	0,088 mV.	—
<i>Sensibilidad mínima del rango de voltaje</i>	4,8 μ V.	—
<i>Memoria interna</i>	4095 muestras.	—

Tomado de (“NI USB-621x Specifications”, 2009)

Actualmente esta fase se compone de una tarjeta de adquisición de datos (DAQ) de referencia NI USB-6210 multifunción de la serie M, cuyas principales características se muestran en la tabla 7.

2.1.5. Módulo de alimentación. El módulo de alimentación, cuyas entradas y salidas se muestran en la figura 12, se encarga del suministro adecuado y establecer la energía necesaria para el funcionamiento de la tarjeta de control de potencia y del módulo de sensado. Se compone por una fuente de poder conmutada de 600 W de la marca UNITEC y de una placa reguladora que permite garantizar una entrega de voltaje de polarización para señales análogas no superior a los 10 V, límite de funcionamiento de la tarjeta NI USB 6210.

Figura 12. Esquema del módulo de alimentación.



2.2. SOFTWARE NARIZ ELECTRÓNICA

Asociado al hardware de la nariz electrónica se presenta el software Bnose, desarrollado mediante una interfaz gráfica de usuario de MATLAB, el cual cumple principalmente la función de sincronizar el funcionamiento general de las etapas que conforma el proceso de la nariz, desde el punto en donde se realiza la toma de las muestras (adquisición) hasta cuando se aplica una técnica de procesamiento, que permite realizar el análisis y la clasificación de las respectivas medidas (procesamiento), como se observa en la interfaz principal en la figura 13.

Figura 13. Panel de inicio del software Bnose.



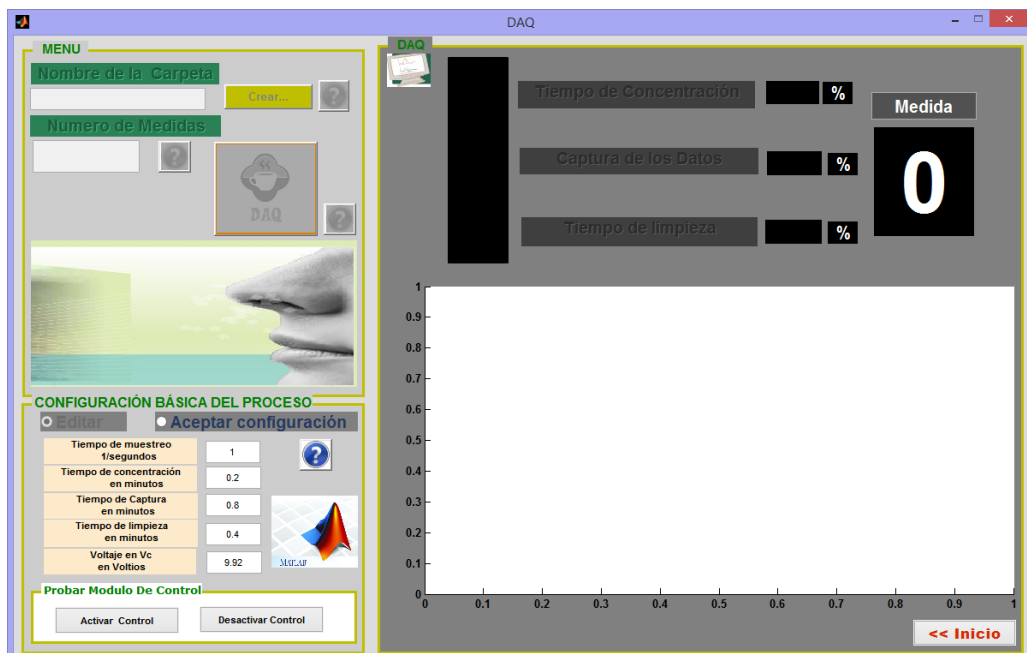
El software se encuentra dividido en cinco secciones fundamentales, la adquisición de datos, la carga de archivos, el pre procesamiento, el análisis de datos directamente

relacionado con los métodos de clasificación y visualización de resultados, y la selección de datos. A continuación se hace una descripción detallada de cada una de estas secciones.

2.2.1. Adquisición de datos. La etapa de adquisición del software Bnose se encuentra al ingresar desde la interfaz principal haciendo clic en el botón “Modulo DAQ”, esta nueva interfaz, mostrada en la figura 14, permite configurar los parámetros adecuados para llevar a cabo el proceso de toma de las muestras del volátil con el cual se desea trabajar. Para iniciar, el usuario puede hacer una verificación de la correcta acción de control que se debe ejercer sobre el sistema neumático de la nariz electrónica desde la sección probar módulo de control y sobre esta encontrar el menú para la modificación e ingreso de los valores de las variables como tiempo de muestreo, concentración, captura, limpieza y voltaje de la señal, procediendo a definir el número de muestras a obtener y el nombre de la carpeta en donde se va a almacenar dicha información.

El proceso de adquisición inicia apenas el usuario oprime el botón “DAQ”, a partir del cual se observa en la parte derecha del panel las gráficas de respuesta asociada a cada uno de los sensores durante las fases de concentración, captura y limpieza. Este procedimiento se lleva a cabo la cantidad de veces indicada por el usuario en la variable número de medidas. En caso del usuario desear regresar a la interfaz principal del software puede dirigirse al botón “inicio” que se encuentra en la esquina inferior derecha de la pantalla.

Figura 14. Panel de inicio del software Bnose.

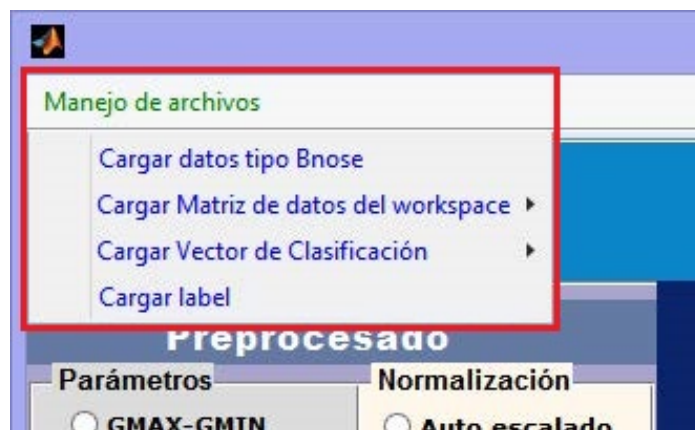


2.2.2. Carga de archivos. La etapa de procesamiento del software Bnose fue desarrollada con el objetivo de utilizar archivos de datos previamente almacenados en la etapa de adquisición, sin tener la facultad de tomar y procesar los datos en tiempo real.

Los archivos se manejan en 4 formas distintas, como se muestra en la figura 15. La primera llamada datos tipo Bnose, es utilizada para cargar en el *workspace* de Matlab Inc las matrices de datos almacenados en formato “.txt” en la etapa de adquisición. En la segunda se admite la posibilidad de utilizar archivos que ya se encuentren en el *workspace*, para lo cual se permiten archivos tipo Bnose, matrices de entrenamiento, vector de validación y matriz de wavelet inversa, los cuales están pensados para la aplicación de métodos de reconocimiento de patrones. La tercera forma se utiliza para cargar vectores de clasificación, los cuales contienen información sobre las clases que existen en una serie de datos previamente analizados, esto permite al software conocer los tipos de olores que hay en un grupo de muestras previamente caracterizadas, ya sea con vectores multiclase (varias clasificaciones) o de una clase (identificación de un solo olor). Finalmente el software permite la introducción de archivos tipo “Label”, para permitir al usuario identificar las clases de un grupo de datos en formato texto a partir de la clasificación hecha en formato entero en el vector de clases.

Los datos provenientes de la adquisición están conformados por grupos de matrices, cada una correspondiente a una muestra que contiene información de un olor, y estas matrices contienen los datos de las curvas resistivas en la respuesta de cada uno de los 16 sensores al olor específico.

Figura 15. Menú para cargar archivos en software Bnose.



2.2.3. Pre-procesado. En el ámbito del reconocimiento de patrones es importante adecuar las señales en función de parámetros que se adapten al problema que se requiera tratar, en el caso del software Bnose los datos se reciben como curvas que representan la reacción en resistencia eléctrica de los sensores ante un olor, sin embargo en el caso de la identificación de olores el análisis de datos se realiza a

partir de las curvas de conductancia en los sensores con el fin de elegir parámetros más adecuados y mejores criterios de selección de datos.

Dentro de la etapa de pre procesado el usuario tiene la opción de elegir uno de cuatro parámetros predefinidos para ser calculado sobre la matriz de datos y posteriormente ser utilizado en las demás etapas de procesamiento. Dentro de los parámetros disponibles se encuentra el cálculo de mayor incremento en la conductancia, es decir la diferencia entre conductancia máxima y conductancia mínima (“Gmax-Gmin”), el incremento de conductancia definida como la diferencia entre conductancia final y conductancia inicial (“Gf-Gi”), y sus versiones normalizadas o fraccionales “Gmax-Gmin/Gmax” y “Gf-Gi/Gf”.

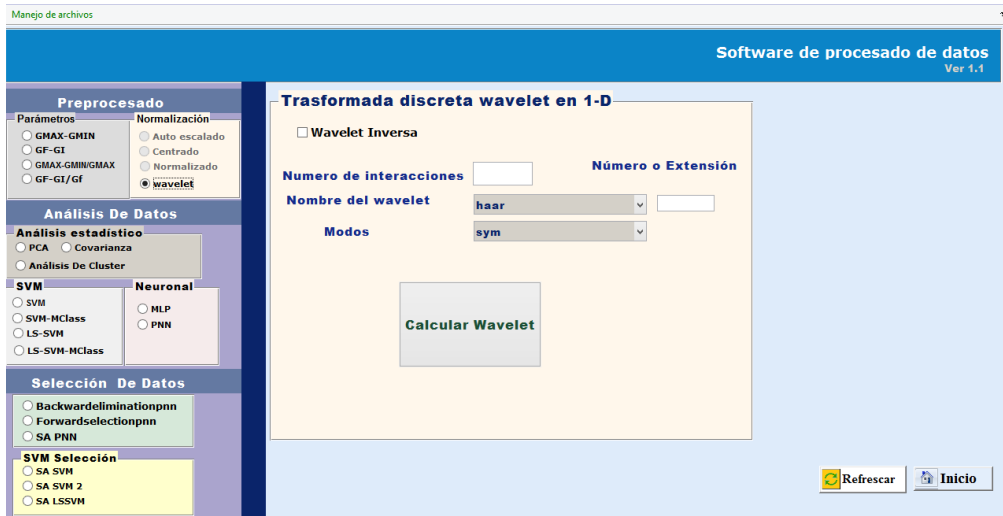
En el software Bnose se dispone de tres tipos de normalización pudiendo ser seleccionado cada uno de ellos como se muestra en la figura 16. Este proceso ayuda a realizar una primera limpieza de los datos ya que permite organizarlos o referirlos a un mismo punto en común, lo que lleva a una mejor selección de datos en los procesos posteriores, cabe aclarar que se puede realizar después de haber parametrizado los datos. Los tipos a utilizar son: normalizado, en el cual cada dato es dividido en el dato de mayor valor, centrado, donde a cada dato se le resta la media del total obteniendo datos con media nula, y el auto escalado, un proceso en el que se centra y posteriormente se normaliza por lo que la media es nula y la varianza unitaria.

Figura 16. Menú elección de pre procesado, software Bnose.



Adicionalmente en este menú se incluye la opción de realizar la transformada wavelet directamente a las curvas de resistencia o las curvas de conductancia parametrizadas, para su posterior procesamiento logrando identificar otras propiedades en las muestras. En la interfaz de Wavelet mostrada en la figura 17 se puede observar que el usuario tiene en un principio la opción de realizar la transformada wavelet o wavelet inversa seleccionando los parámetros de criterio de cálculo y el número de iteraciones en cada caso.

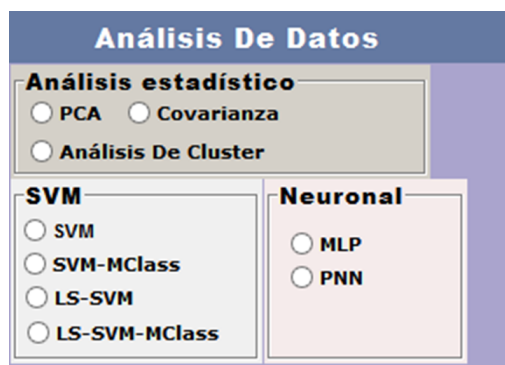
Figura 17. Interfaz wavelet, software Bnose.



2.2.4. Análisis de datos. Posteriormente de obtener los datos a partir de las muestras volátiles de la fase de adquisición, es necesario examinar, seleccionar y transformarlos con el objetivo de extraer información relevante para el usuario, la cual es la principal función del software implementado. Se utilizan técnicas estadísticas tales como análisis de componentes principales (*PCA*), “análisis de cluster” y “covarianza”, permitiendo considerar comportamientos característicos, que concluye en la toma de decisiones trascendentales. Esto permite por ejemplo lograr una compresión de la densidad de los datos y por lo tanto una reducción de latencias en el procesamiento. También es posible suprimir ciertos sensores implementados en el hardware entre otras peculiaridades del sistema. A partir de ese análisis, el usuario podrá utilizar sistemas de aprendizaje artificial, que le permiten entrenar y obtener resultados clasificatorios para el funcionamiento de la nariz electrónica. Se dispone de dos tipos de redes neuronales principales para el aprendizaje, que son *back-propagation multi-layer perceptrons* (*MLP*) y *probabilistic neural network* (*PNN*), además de dos métodos de clasificación y selección con resultados comparables con las redes neuronales, que son *support vector machines* (*SVM*) y *least squares support vector machines* (*LS-SVM*).

El análisis de datos implementado en la nariz electrónica de la Universidad de Pamplona tiene un amplio abanico de posibilidades de uso, que se limita a ser utilizado por especialistas en el procesamiento de datos, con el fin de explotar correctamente cada una de las posibilidades que la misma ofrece. El panel de selección de análisis de datos se muestra en la figura 18

Figura 18. Panel de selección, menú análisis de datos

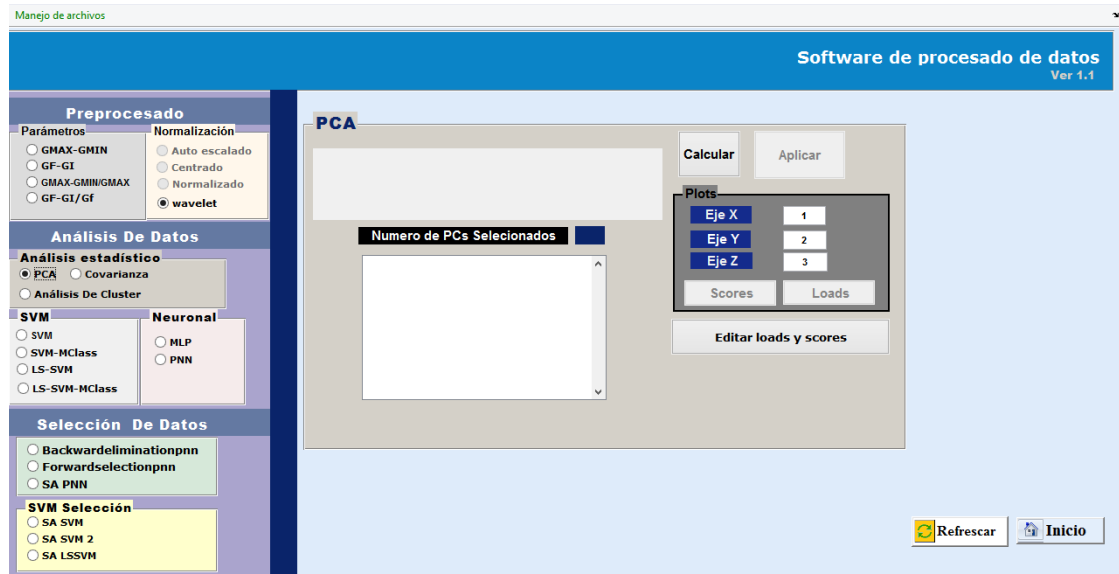


2.2.4.1. Análisis estadístico. La estadística se define como la aplicación del método científico en el análisis de datos numéricos con el fin de tomar decisiones racionales (Mark Berenson, 1987), siendo importante en todo proceso de análisis de datos ya sea para reducir la cantidad de muestras, identificación de variables relevantes y/o relación entre muestras, avistamiento de comportamientos reiterativos en los datos, o directamente concluir resultados en torno al análisis estadístico realizado.

En esta sección del software el usuario tiene la opción de realizar tres tipos de análisis, dos de ellos con el fin de mostrar gráficamente la distribución de las muestras y que el usuario pueda identificar la relación de estas como diagrama de árbol (análisis de clúster) o si se diferencian sus componentes como grupos pertenecientes a distintos tipos de elementos (olor) a través de PCA, y otro con el fin de realizar una fase más de pre procesado en los datos que es el análisis de covarianza. A continuación se definen estos tres tipos de análisis.

Análisis de componentes principales: (*Principal Components Analysis (PCA)* en inglés) es un método de análisis estadístico que permite extraer información y relación entre variables a partir de los mismos parámetros para un conjunto de datos muy numeroso, este método fue ideado en 1981 debido a la creciente cantidad de instrumentos que podían otorgar gran cantidad de datos, convirtiéndose en un problema a la hora de interpretar la información obtenida. En PCA se reduce la dimensionalidad de los datos a partir de encontrar los ejes que definen el espacio de componentes principales (PC por sus siglas en inglés *Principal Components*), estos definen la relación entre variables o colinealidad entre datos por lo que no es recomendable en sistemas altamente no lineales. En el caso de la nariz electrónica se utiliza como elemento de agrupación o identificación de relación entre muestras, es decir como método de reconocimiento de patrones no supervisado, sin embargo existen algoritmos que permiten aplicarlo en modo supervisado.

Figura 19. Interfaz PCA, software Bnose.

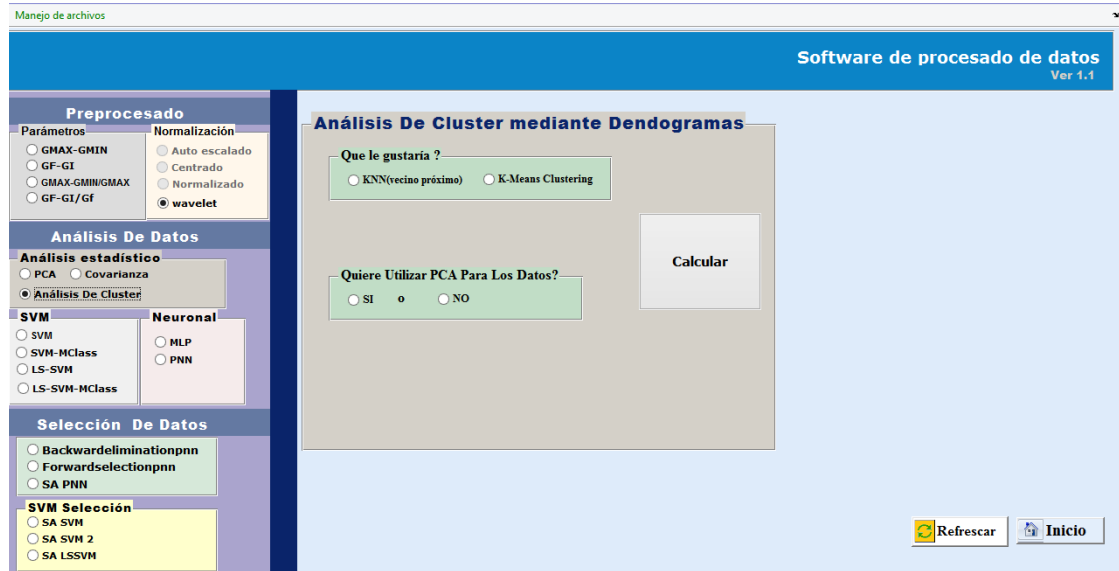


En el software Bnose la pestaña PCA, como se observa en la figura 19, permite al usuario calcular los PCs del sistema con los datos de entrada, botón calcular, con lo que aparece una interfaz (a la izquierda) con información de los PCs calculados en este orden, número del PC, autovalor, porcentaje de varianza en ese PC y porcentaje de varianza en la sumatoria con anteriores PCs; luego se debe seleccionar la cantidad de PCs a utilizar para la caracterización y asociar el PC con un eje del espacio cartesiano (X, Y, Z) (a la derecha de la interfaz). Con el botón aplicar PCA, se calcula el PCA en función de los componentes seleccionados y se guarda la información para ser utilizado por análisis de clúster. La ventana PCA permite observar dos análisis, “Scores” grafica los valores de cada muestra en función de los componentes seleccionados como ejes, lo que permite visualizar la cercanía entre muestras para determinar si pertenecen a un mismo elemento o ver que tan relacionado está un grupo de elementos, y “Loads” grafica el valor obtenido por cada sensor en función de los PCs seleccionados como ejes. Esto permite visualizar la percepción de los sensores ante cada muestra dejando agrupar los dispositivos que reaccionan de forma similar ante el mismo o distintos elementos. En la práctica puede servir para discriminar la cantidad y uso de determinados sensores. También existe la opción editar “Loads” y “Scores”, logrando personalizar la ventana de visualización de los gráficos anteriores, facilitando la interpretación de datos al ubicar etiquetas pertinentes.

Análisis de clúster: Con el fin de encontrar homogeneidad entre muestras, en este tipo de análisis se toma un conjunto de variables que evaluadas sobre las muestras obtenidas permiten organizar grupos o clústeres con ciertas similitudes entre muestras y diferencias entre grupos. A su vez se evalúan relaciones o parentescos entre grupos para crear clústeres de mayor nivel, de esta forma se permite facilitar la aproximación segmentada de un determinado análisis. El análisis de clúster puede ser utilizado para reducir la dimensionalidad de los datos para posterior procesamiento

o como en el caso de Bnose, con el fin de explorar la relación entre muestras a lo que se le llama análisis taxonómico.

Figura 20. Interfaz Análisis de clúster, software Bnose.

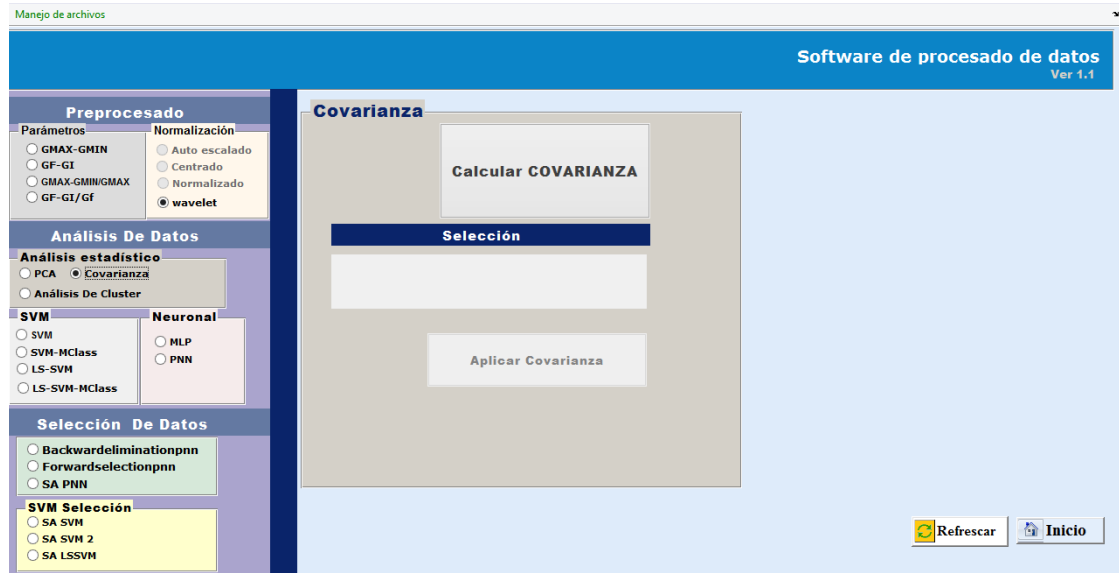


En la herramienta Bnose antes de realizar análisis de clúster los datos deben tener un tipo de normalización auto escalado o centrado, lo anterior debido a que resaltan las características de clasificación clásicas como el caso de media nula. En la figura 20 se puede observar la interfaz Bnose en este tipo de análisis. Para realizar el análisis se puede elegir uno entre dos métodos, el *K nearest neighbor* o *KNN* (primer mención por Fix y Hodges en 1951), un algoritmo de clasificación de objetos basado en el entrenamiento mediante ejemplos cercanos en el espacio de los elementos a clasificar donde la elección de K se hace mediante una optimización dependiendo del uso, y el *means clustering*, método en el que se agrupan las muestras de acuerdo a su cercanía en el valor de la media entre las mismas. Adicionalmente el software permite, si el usuario lo desea, realizar una clasificación previa con ayuda del algoritmo PCA, tomando los resultados del anterior para aplicar el método de clasificación, seleccionando esta opción también se permite aplicar medida de distancia de Mahalanobis, medida en la que se tiene en cuenta la correlación entre variables, si no se elige este tipo de medida o PCA la distancia entre muestras es calculada con la medida euclidiana.

Análisis de covarianza: Es un método que permite verificar la repetitividad y selectividad de los datos. Para la realización del método se toma cada muestra para calcular el centroide o tendencia central de la muestra, posteriormente es calculada la distancia media de cada dato con dirección al centroide de su respectiva muestra, intervianza media, evaluando así la repetitividad, y la distancia media de cada centroide con respecto a los de otras muestras, intravarianza media, es calculada para definir un criterio de selectividad obteniendo de esta forma información para reconocer cuales datos y muestras contienen la mayor información y cuales pueden ser

redundantes. El criterio de selección de muestras para este método se toma a partir de un índice definido como la razón entre la intervarianza media y la intravarianza media el cual es llamado poder de resolución.

Figura 21. Interfaz análisis de covarianza, software Bnose.



En la interfaz del software Bnose, figura 21, inicialmente tenemos la opción de generar el cálculo de covarianza, con lo cual obtenemos en el panel frontal “Selección”, un vector que indica cuales son las muestras más apropiadas para iniciar el procesamiento de datos a la vez que un gráfico donde se muestra el valor de índice de poder de resolución para cada muestra. Al presionar la opción aplicar covarianza se guarda la matriz de datos inicial agregando una columna que indica las muestras más apropiadas para el procesamiento.

2.2.4.2. Redes neuronales. Una de las técnicas con que cuenta el desarrollo de los actuales productos electrónicos, se basa principalmente en las ventajas que entrega la inteligencia artificial, pues a través de ella, es posible realizar prototipos con procesos de aprendizaje autónomo y adaptable. Particularmente, en la nariz electrónica implementada en la UP, se explora la funcionalidad de las redes neuronales aplicadas a un objetivo específico, más que clasificar, identificar muestras volátiles.

Las redes neuronales son algoritmos matemáticos que generan una salida particular a partir de una matriz, o vector de entradas, gracias a un conjunto de “pesos” que representan la relación entrada-salida, y que son auto-modificados en la fase de entrenamiento, es decir, en función de las entradas y la memoria local, produce una salida que actualiza la memoria local.

Existen dos posibles algoritmos de redes neuronales que se pueden implementar en el software que son las redes neuronales probabilísticas y el perceptrón multicapa de retro-propagación.

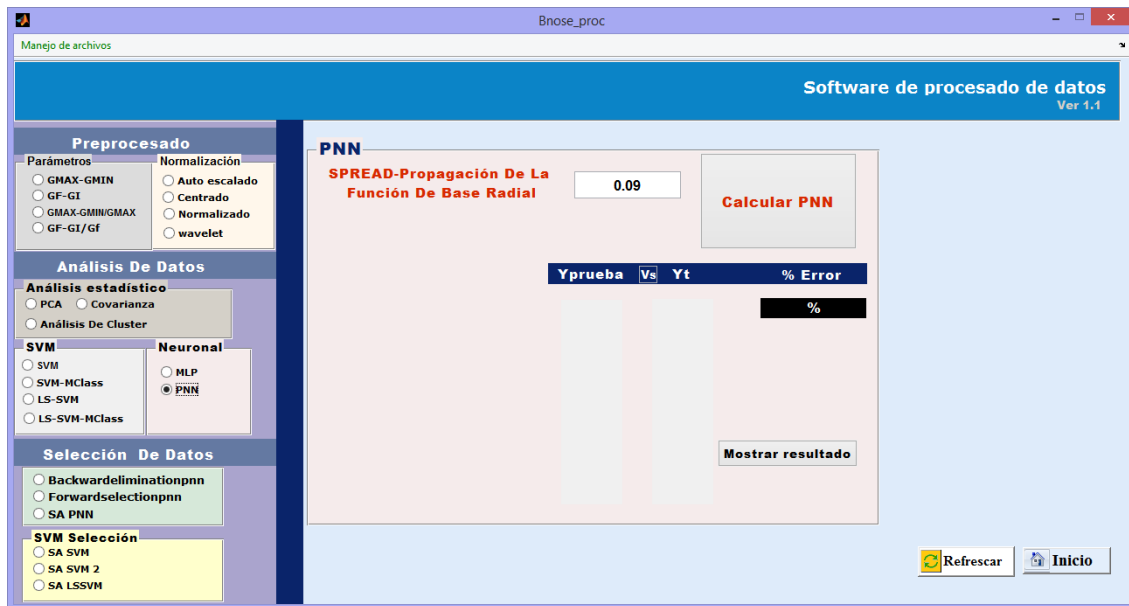
Redes neuronales probabilísticas (PNN): Las redes neuronales probabilísticas se utilizan en la práctica como método de clasificación y reconocimiento en procesos donde existen más de una variable a evaluar. “Los autores referidos han reportado una eficiencia de reconocimiento mayor al 90 % en algunas de las investigaciones, ofreciendo un menor rango de error y tiempo de ejecución en comparación de otras redes neuronales como las que utilizan el algoritmo *backpropagation* (Cadena, 2008) o el perceptrón simple y de doble capa (Pérez Ramírez, 2007)” (Paloma Teresita Gutiérrez Rosas, 2013).

Las PNN, por sus siglas en inglés *probabilistic neural network*, pertenecen a la familia de redes neuronales con función base radial. Constan de dos capas, una con un conjunto de neuronas de base radial igual al número de vectores de entrenamiento, y la otra capa competitiva con la misma cantidad de neuronas que de categorías o clases del problema de clasificación.

Esta red se implementó por medio de la librería de MATLAB, software de *Math-Works Inc*, llamada *Neural Network Toolbox*, utilizando la función “newpnn” como método para calcular una red neuronal probabilística a partir de los parámetros de entrada, las cuales son el vector de entrenamiento, vector de clases y la propagación de la base radial (*spread*). La lógica programada en el software de la nariz electrónica obedece a calcular una red neuronal con base al conjunto de datos ingresados sin una de las filas (datos de muestra temporal) que lo componen, para posteriormente, a partir del entrenamiento, identificar la clase a la que pertenece dicha fila excluida. Lo descrito se realiza para cada una de las filas de la matriz de información suministrada en el momento de utilizar la función PNN del menú neuronal.

Para que el usuario pueda hacer uso de la red neuronal PNN como método de análisis de datos, debe previamente escoger el tipo de pre-procesado, y si cree conveniente, realizar alguna selección de datos y/o análisis estadístico de covarianza, con el fin de editar según corresponda la matriz de datos de información. A continuación, se escoge la opción PNN del menú “Neuronal”, la cual imprime en pantalla el panel de procesamiento mostrado en la figura 22. Se debe suministrar un valor de propagación de la función de base radial (por defecto 0.09) que el usuario debe escoger de acuerdo a sus necesidades y se hace clic en el botón calcular PNN. El resultado del procesamiento se muestra principalmente en dos columnas, una que muestra las verdaderas clases a las que pertenece cada vector de datos, y la segunda entrega los resultados de clasificación obtenido por la red neuronal. Adicionalmente se le entrega al usuario el porcentaje de error de clasificación. Cada vez que se hace clic en el botón “Calcular PNN”, se crea una nueva red suprimiendo las anteriormente calculadas.

Figura 22. Panel de PNN de la interfaz de procesamiento



Redes de retro-propagación - perceptrón multicapa: Las redes MLP, por sus siglas en inglés *multi-layer perceptrons*, son redes neuronales que asignan a un conjunto de entradas, un valor de salida específico mediante el método de aprendizaje supervisado llamado *back propagation*, es decir, el valor de los pesos de conexión van evolucionando a partir del error calculado como la razón entre la salida de la red neuronal y el resultado esperado. Los algoritmos implementados para discriminar de forma lineal en la red neuronal, son llamados perceptrón, el cual, en este caso está constituido por dos o más capas de unidades de computación, por lo tanto existe un conjunto de neuronas en una capa que se interconecta con otro conjunto de neuronas en una siguiente capa, creando una red más robusta. Sin embargo, este tipo de redes neuronales presentan una convergencia de aprendizaje considerablemente lenta. La principal ventaja que presenta la red multicapa sobre una red de una capa, es la capacidad de distinguir entre los datos que no son linealmente separables, lo que resulta la capacidad para procesar varias clases, trabajando como clasificador no lineal o como función de regresión. (Pavlov, 2013)

Esta red se implementó en MATLAB al igual que la red PNN por medio de la librería *Neural Network Toolbox*, utilizando la función “newff” como método para calcular una red neuronal multicapa según los parámetros de ingreso tales como la matriz de vectores de entrada, matriz de clases correspondientes, tamaños de las capas internas, entre otros.

Para crear una red neuronal tipo *back-propagation* MLP, el usuario debe seleccionar la opción MLP del panel de selección, dentro del menú análisis de datos – neuronal. Como resultado, el sistema presenta en pantalla las opciones de la figura 23, las cuales le permiten configurar la red neuronal, desde la cantidad de neuronas utilizadas y la función de transferencia de las capas intermedias, la forma de aprendizaje del

perceptrón, el tipo de aprendizaje de los pesos y polarizaciones, las iteraciones para la convergencia de los resultados del entrenamiento, entre otras características importantes. Al hacer clic en el botón “Calcular MLP” seguido por el botón “Mostrar resultados”, se puede obtener el error que ha tenido la red neuronal al procesar los datos, además de los resultados clasificatorios.

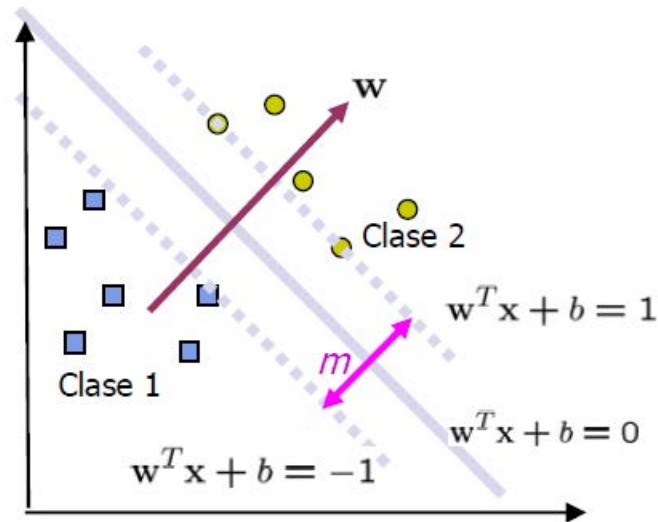
Figura 23. Panel de MLP de la interfaz de procesamiento



2.2.4.3. Máquinas de soporte vectorial. (SVM por sus siglas en inglés *Support Vector Machines*), son un conjunto de algoritmos consecuencia de la investigación sobre la minimización del riesgo estructural realizada por Vapnik (Vapnik, 1999). Este tipo de análisis ha venido siendo implementado en sistemas artificiales del olfato (Cosino Distante, 2003), como una herramienta para el reconocimiento de patrones o clasificación (Burges, 1998; Useche, 2009).

Su objetivo consiste principalmente en determinar la combinación de algunos puntos referidos a los datos de entrada que se definen como vectores de soporte, los cuales permiten describir un hiperplano óptimo que separe al conjunto de muestras en distintas clases, a partir de una formulación matemática que depende directamente de la naturaleza de los datos a emplear, ya que se pueden presentar dos casos: el linealmente separable o el no linealmente separable (Betancourt, 2005). Consecuentes con la idea anterior, el software Bnose emplea la teoría para un conjunto de datos no linealmente separables, en donde se realiza el mapeo de los datos proporcionados por medio de una función “kernel” (puede ser lineal, polinomial o Gaussiana) a un espacio dimensional mayor donde se busca la máxima separación entre las clases, al ser traída de regreso al espacio de entrada la frontera resultante permite obtener un agrupamiento entre las distintas clases. Como se puede observar en la figura 24.

Figura 24. La máxima distancia de cada una de las clases con respecto a la frontera.



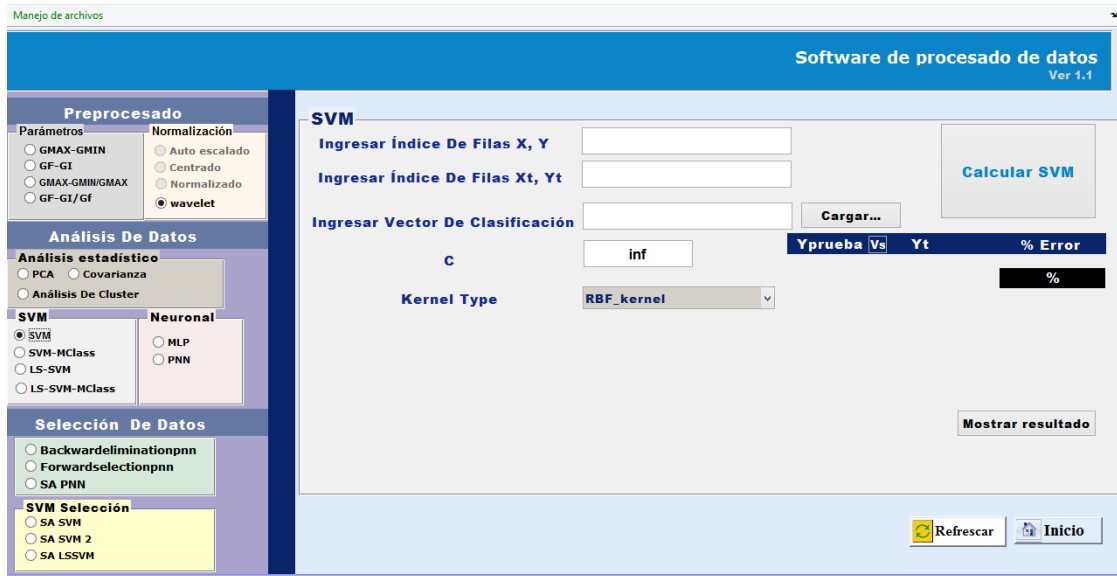
Tomado de (Betancourt, 2005)

Los parámetros requeridos por la interfaz constan de un conjunto de datos de entrada para el entrenamiento, un conjunto de datos para validación, el vector de clasificación, el tipo de función “kernel” que se desea aplicar y finalmente una constante C que está directamente relacionada con regular la maximización del margen y el error de clasificación, como se observa en el figura 25. Después del ingreso de estos parámetros el usuario presiona el botón calcular SVM seguido de mostrar resultados, obteniendo dos columnas de números que hacen referencia a cada una de las clases, donde Y_{prueba} es el resultado de clasificación dado para el vector de validación ingresado y $Y(t)$ el verdadero valor de clasificación según el vector de clasificación ingresado, permitiendo obtener un porcentaje de error al comparar dichos resultados.

La principal consideración para el empleo de este tipo de método es su facilidad y rapidez de análisis a comparación de redes neuronales (Betancourt, 2005). Sin embargo *SVM* da solución a un problema cuadrático donde el número de coeficientes es igual a número de datos de entrenamiento, lo que en casos de gran cantidad de datos se vuelve un limitante en términos computacionales (Trejo, 2006), por esta razón no se recomienda el empleo de máquinas de soporte vectorial en casos donde las entradas son obtenidas al tiempo que se va realizando un aprendizaje, y a su vez es un método altamente dependiente de la correcta selección de la función “kernel”.

Normalmente este tipo de algoritmo se emplea como clasificador de dos clases (Betancourt, 2005), sin embargo se puede adaptar para ser clasificador multiclase, a partir de la modificación en el algoritmo de programación cuadrática del método, por lo cual en el software Bnose surge la posibilidad de un análisis *SVM-Mclass*.

Figura 25. Interfaz para parámetros de análisis con SVM



Máquinas de soporte vectorial de mínimos cuadrados: En el software Bnose se presenta a la vez del método de máquinas de soporte vectorial, una reformulación de este, que consiste en la aplicación de la técnica de los mínimos cuadrados sobre las máquinas la cual fue propuesta por Sunkeys y Vanderwalle (Suykens y Vandewalle, 1999). A partir de esto logra una clasificación por medio de la solución a un conjunto de ecuaciones, las cuales permiten la determinación de la máxima distancia entre la frontera que separa cada una de las clases que se quieren obtener a partir de los datos de entrada a analizar, los cuales pueden ser de un cantidad relativamente mayor en comparación al caso de las SVMs (Luis A. Munoz, 2013).

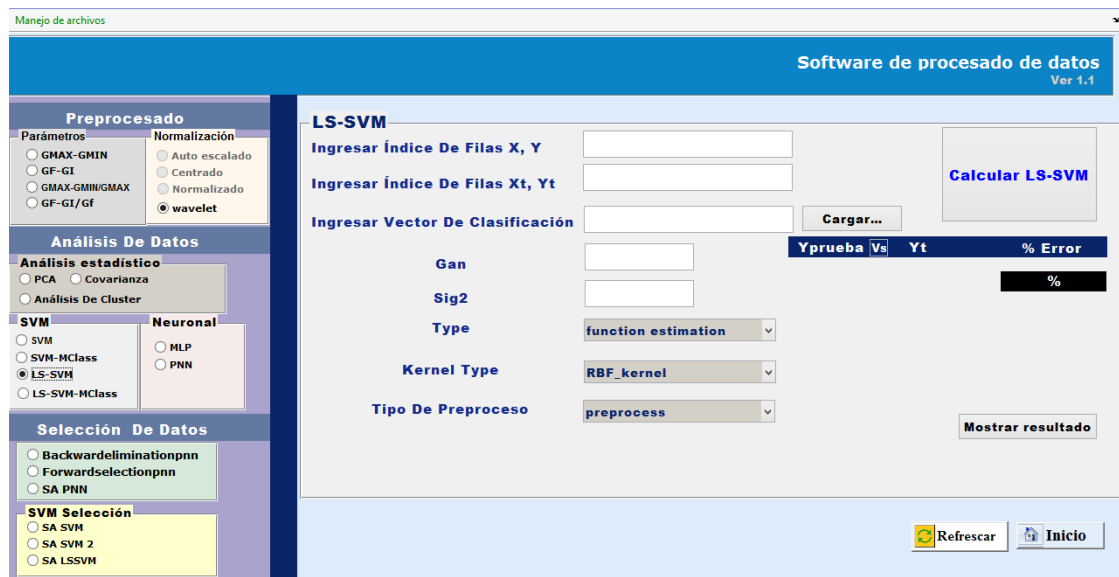
Al igual que en el caso de SVM, se presenta la clasificación por medio de LSSVM, por sus siglas en inglés *Least Squares Support Vector Machines*, que consiste en dos fases: Una primera de entrenamiento a partir de un conjunto de datos ingresados por medio de los cuales se encuentran los parámetros que permiten construir un clasificador. En la siguiente fase de prueba, este clasificador busca con un conjunto de datos ingresados determinar a qué tipo de clase pertenece para finalmente comparar con los datos reales, obteniendo así un error que permite ver la efectividad del modelo hallado.

Además de estos vectores tanto de entrenamiento como de validación el software como se observa en la figura 26, requiere del ingreso de un vector de clasificación, y a su vez parámetros para el entrenamiento como el valor de gamma, y sigma al cuadrado, así como el tipo de “kernel” y adicionalmente habilita la opción de si se desea trabajar con los datos de forma pre procesada u original, y si el método se va a emplear para la clasificación o como una función de estimación. Ingresando correctamente cada uno de los parámetros el usuario debe proceder a presionar el botón calcular LSSVM seguido de mostrar resultados, obteniendo dos columnas de números que hacen referencia a cada una de las clases, donde Yprueba es el resulta-

do de clasificación dado para el vector de validación ingresado y $Y(t)$ el verdadero valor de clasificación según el vector de clasificación ingresado, permitiendo obtener un porcentaje de error al comparar ambos resultados.

La técnica de LSSVM fue inicialmente planteada para el análisis binario (Luis A. Muñoz, 2013), sin embargo permite mediante estrategias desarrollar el ajuste para realizar la clasificación de múltiples clases, brindando la herramienta “LS-SVM-Mclass”.

Figura 26. Interfaz para parámetros de análisis con LS-SVM



2.2.5. Selección de datos. En todo sistema que recurre a la adquisición de señales es normal encontrarse con un gran conjunto de datos, entre los cuales algunos contienen bastante información útil para el análisis a realizar y otros que simplemente son generadores de ruido, es decir, son datos que no poseen información relevante o son redundantes, por lo que generan una mayor desviación entre los datos. Por esta razón el sistema de procesamiento del software Bnose presenta la posibilidad de realizar, mediante un criterio de selección de datos, una disminución de la dimensión de datos ingresada sin perder la información principal (Acevedo, 2013).

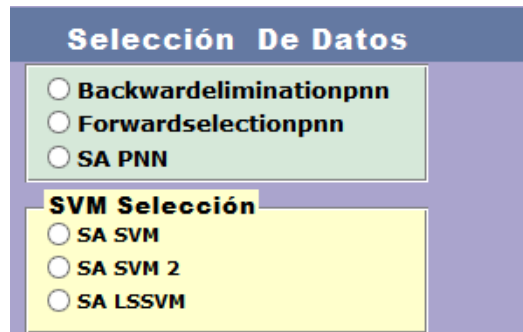
Las redes neuronales y las SVM son técnicas de aprendizaje para la clasificación, las cuales presentan una alta dependencia con respecto a la cantidad de datos suministrados para el correcto entrenamiento y la calidad en términos de ruido asociado a estos datos (Masso, 2014). Por esta razón se aplica un proceso de selección, ya sea independiente o dependiente de las técnicas de reconocimiento de patrones sobre las cuales se van a aplicar dicho conjunto de datos resultantes de la selección de variables empleada.

El resultado final de aplicar un pre procesamiento como la selección de variables sobre los datos ingresados, brinda la posibilidad de aumentar la eficiencia propia de los

algoritmos de aprendizaje empleados en el reconocimiento de patrones, ya que les permite a estos centrarse en un grupo reducido de datos al momento de aplicar el análisis, obteniendo mejores resultados.

Los procedimientos de selección de variables constituyen dos fases: el criterio de selección simplificado y un procedimiento de búsqueda óptimo, en donde para este último se han desarrollado múltiples estrategias (Gualdrón, 2005), de las cuales en el software Bnose se aplicaron de tipo secuencial el *forward selection* y el *backward selection*, además del algoritmo de búsqueda aleatoria *simulated annealing*. El panel de selección se puede observar en la figura 27.

Figura 27. Panel de selección de datos

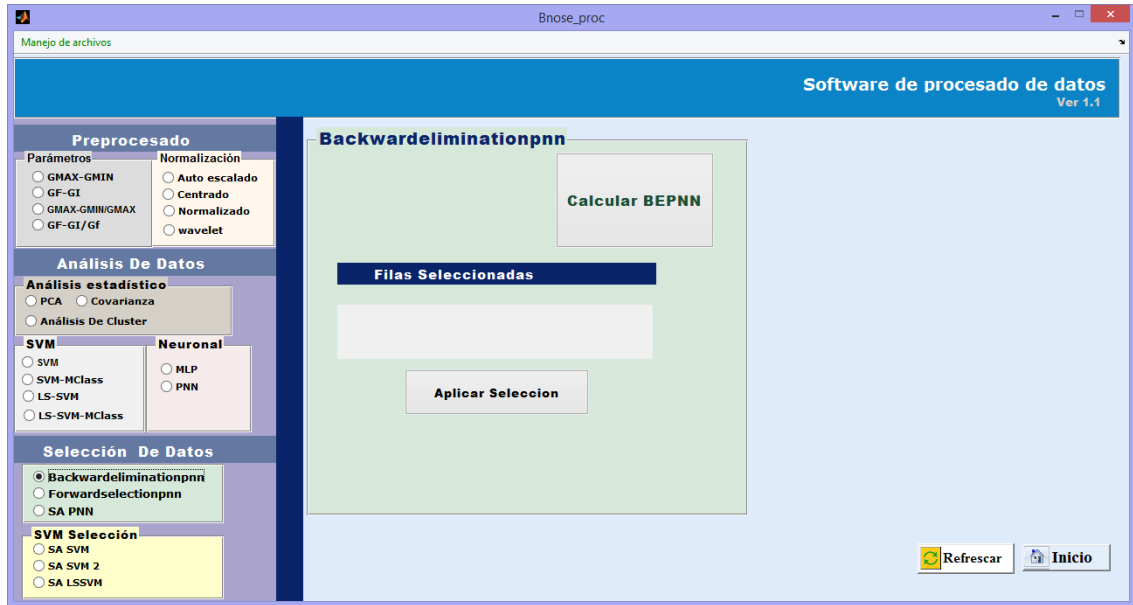


2.2.5.1. Método secuencial eliminación regresiva. El método de selección de variables denominado eliminación regresiva, por sus siglas en inglés *sequential backward selection* (SBS) o también llamado simplemente *backward elimination*, realiza la selección del conjunto de datos que representan en mayor medida la información, basándose en el cálculo del error de predicción del clasificador.

Inicialmente se considera todo el conjunto de variables y se calcula a cada una el error de predicción del clasificador neuronal PNN con respecto a las variables restantes. A partir de ello, se descarta las variables de exclusión.

En el software, escrito en MATLAB e implementado en la nariz electrónica, se programa cada uno de los pasos necesarios, utilizando principalmente la librería de redes neuronales para la función PNN. El usuario puede realizar este procedimiento a la matriz de datos después de aplicado algún tipo de pre-procesamiento. La pantalla de presentación proporcionada para esta etapa se presenta en la figura 28, donde basta con que el usuario haga clic en "Calcular BEPNN", verifique las filas seleccionadas como el nuevo grupo de datos y aplique la selección.

Figura 28. Panel de SBS de la interfaz de procesamiento



2.2.5.2. Método secuencial selección progresiva. A diferencia del *backward elimination*, la selección progresiva o SFS por sus siglas en ingles *sequential forward selection*, comienza con un conjunto de variables vacío y secuencialmente va agregando cada una de los parámetros individualmente según el criterio de selección dado por el error de predicción del clasificador neuronal PNN.

Figura 29. Panel de SFS de la interfaz de procesamiento

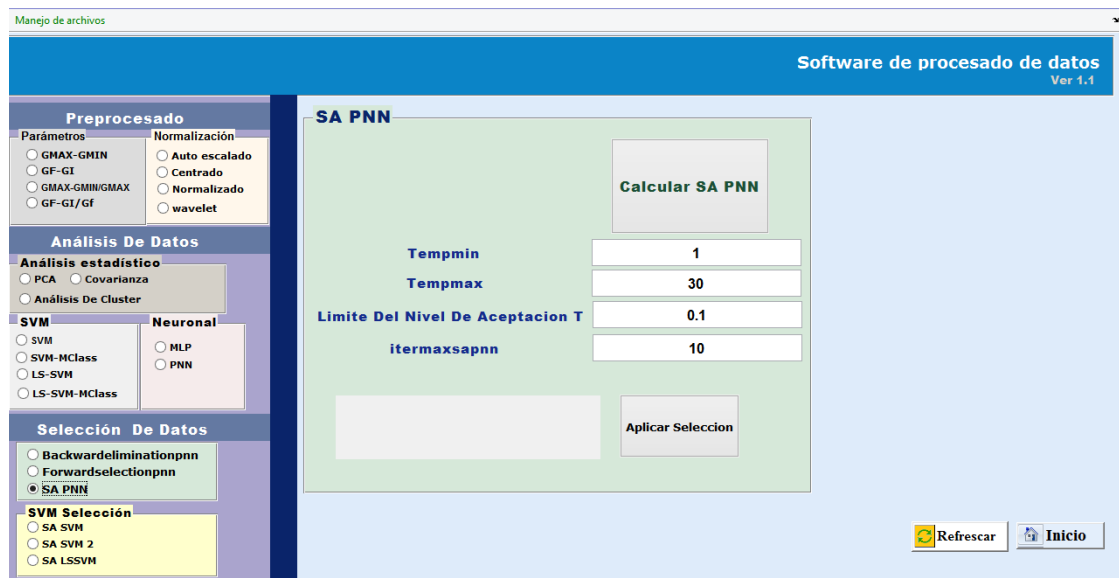


El usuario, al habilitar en el panel de selección la opción de “forwardselectionpnn”

dentro del menú se selección de datos, se imprime en pantalla el cuadro mostrado en a figura 29, que contiene las mismas dos opciones que para el modelo de eliminación regresiva, siendo su uso intuitivo.

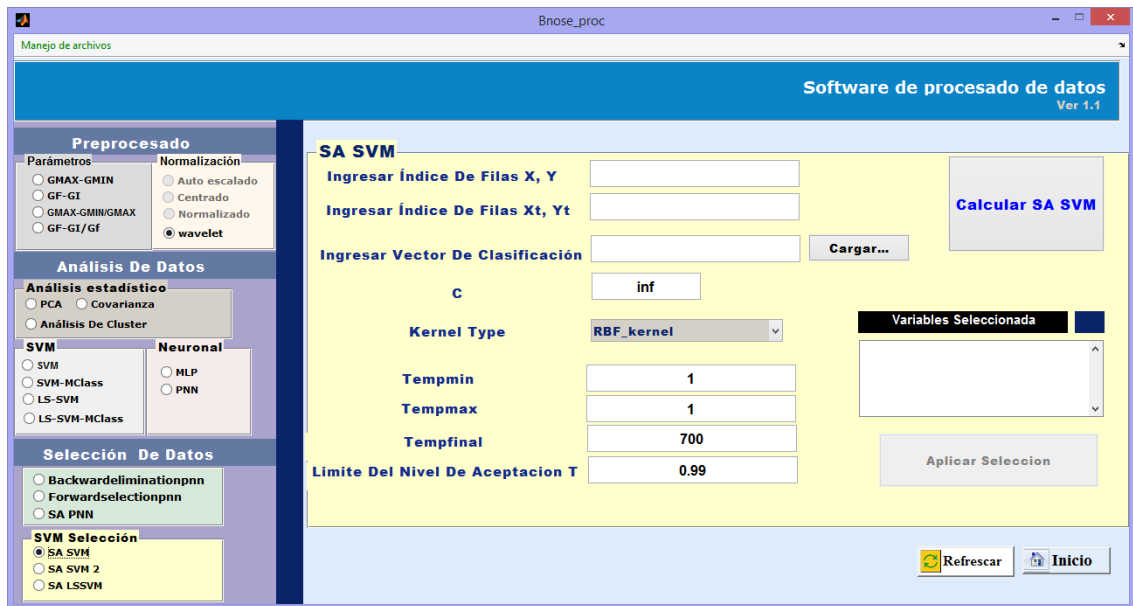
2.2.5.3. Recocido simulado. La técnica SA (por sus siglas en inglés *simulated annealing*) implementada en el software de MATLAB de la nariz electrónica, consiste en un algoritmo estocástico que permite optimizar la selección de las variables a emplear para un análisis, fue introducido por Kirkpatrick y colaboradores en 1983 (S. Kirkpatrick, 1983). Se basa principalmente en el concepto de templado de metales, en donde se calienta un metal a una temperatura inicial t_0 , para posteriormente empezar a disminuirla hasta el punto que el sistema queda en un estado de enfriamiento con energía mínima, punto que se define como el óptimo global(L. Nolle, 2002).

Figura 30. Panel de SA PNN de la interfaz de procesamiento



Equivalente al proceso de análisis que se aplica en la metalúrgica, el método emplea una regla de aceptación que permite llevar a cabo la selección de datos en el software Bnose. La técnica parte de un conjunto de variables, más específicamente los vectores de entrada, para sobre estos aplicar un método de reconocimiento de patrones, que permite obtener un valor de error de predicción, denominado *fitness*, el cual es nuevamente calculado sobre la matriz resultado de haber eliminado de manera aleatoria algunas variables, de tal forma que se hace la comparación del *fitness* nuevo con el viejo, y con el valor de delta obtenido se procede a analizar si se ha mejorado la clasificación con la eliminación de estas variables, o si en caso contrario es necesario proseguir a la segunda fase donde se hace un estudio de probabilidad, hasta que finalmente después de múltiples iteraciones se llega a un conjunto ideal de variables para una temperatura determinada, es decir, el método converge, quedando el conjunto de datos listo para ser aplicado sobre algún método clasificador específico (Gualdrón, 2005).

Figura 31. Panel de SA SVM de la interfaz de procesamiento



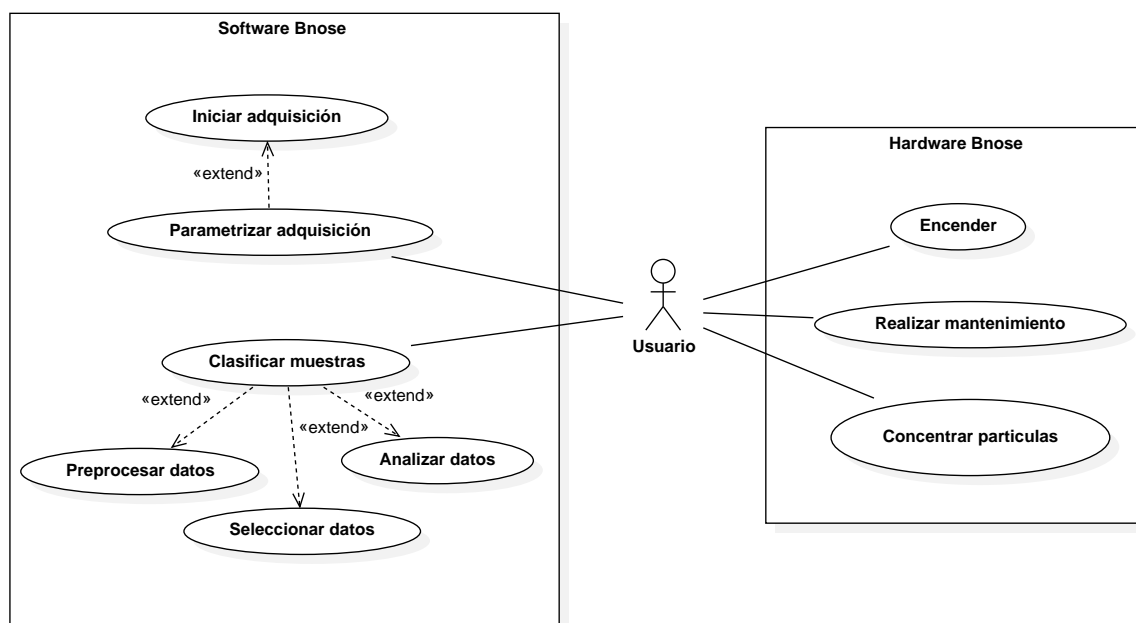
Para el caso de Bnose este método de selección es aplicado en conjunto con las técnicas de PNN, SVM y LS SVM de tal forma que los parámetros a ingresar son los mismos que se presentaron para el caso de análisis de datos, sin embargo adicionalmente se deben ingresar los parámetros que definen un número fijo de iteraciones para cualquier temperatura dada y el coeficiente de enfriamiento como parámetro de decisión, para un correcto desempeño del algoritmo de selección de variables, seguido al ingreso de los parámetros el usuario debe pulsar el botón calcular SA SVM o SA PNN según sea el caso, obteniendo en la esquina inferior derecha del panel del análisis SA el listado con los índices de vectores de entrada que se deben mantener después de aplicar el proceso de selección, el software con el botón aplicar selección le permite al usuario tomar la decisión de si desea o no hacer aplicación del resultado obtenido. Se pueden observar los paneles de selección del PNN en la figura 30, y el SA SVM en la figura 31.

3. REPRESENTACIÓN EN DIAGRAMAS UML DEL SISTEMA DE RECONOCIMIENTO DE PATRONES

Encontrar la manera de clarificar el funcionamiento del módulo de reconocimiento de patrones de la nariz electrónica para un desarrollador o ingeniero de sistemas en pro del posterior diseño de un servicio, se convierte en el primer reto a resolver en aras de transmitir la información mostrada del software en el capítulo anterior. Por ello se hace necesario utilizar un lenguaje más familiar y simplificado, uno en el que se presenten los conceptos del software a través de interacciones con un propósito definido y parta de las acciones del usuario para presentar las funcionalidades del sistema. En consecuencia se utiliza el UML como medio de caracterización para describir los componentes estructural y dinámico del software Bnose. Para esto se emplea el diagrama de casos de uso (interacción del usuario), diagrama de clases (estructura del sistema) y diagrama de secuencia (interacción entre componentes) desarrollados con ayuda del software StarUML 2014 versión 2.0.1.0 de MKLab.

3.1. DIAGRAMA DE CASOS DE USO.

Figura 32. Diagrama Casos de uso, software Bnose.



El diagrama de casos de uso se muestra en la figura 32, en este se expresa la relación que tiene el usuario con el sistema nariz electrónica, evidenciando las características de interfaces de trabajo o acciones del mismo con los dispositivos del sistema. En el caso del hardware, el usuario debe realizar acciones directas con los componentes de la nariz electrónica, siendo estas, encender el equipo (asegurarse de la correcta alimentación de cada dispositivo), realizar mantenimiento pertinente para conservar en buen estado el sistema (si es necesario cambiar piezas en mal estado) y concentrar

partículas, acción que consiste en encargarse del posicionamiento de las muestras de olor en la cámara de muestras cada vez que se realiza una toma de datos. Con respecto al software, el usuario se comunica a través de una interfaz visual (pantalla) con tres ventanas principales, siendo la primera el pre-requisito para acceder a las funcionalidades del sistema, es decir los casos de uso del software Bnose a los que el usuario tiene acceso directo como lo son “Clasificar muestras” y “Parametrizar adquisición”.

Correspondiente al caso de uso “Parametrizar Adquisición”, se presentan todas las acciones que decide el usuario con apoyo de una interfaz gráfica para realizar la adquisición de señales, como puede ser la cantidad de muestras o tiempo de duración del muestreo, y por consiguiente “Iniciar adquisición”. Cuando el usuario interactúa con el sistema a través del caso de uso “Clasificar muestras”, accede a herramientas que le permiten realizar una diferenciación de las muestras que adquirió anteriormente, es decir, puede extraer conclusiones a partir de la separación (realizada por cada algoritmo empleado) de los datos en grupos o clases (clasificación). Como una extensión de este caso de uso, se encuentran los casos de uso de preprocesado, selección de datos y análisis de datos. Cabe destacar que ante la carencia de un aprendizaje continuo de las redes neuronales y las máquinas de soporte vectorial, el caso de uso correspondiente a la identificación de muestras no se encuentra implementado.

A continuación se describe en detalle el proceso desde realizar una captura de datos hasta utilizar una o algunas de las opciones de procesamiento que posee el software Bnose:

1. Usuario verifica correcta conexión de los dispositivos físicos.
2. Usuario energiza el sistema.
3. Usuario posiciona la muestra en la cámara de muestras del sistema.
4. Usuario accede a la interfaz “Bnose” y decide Parametrizar adquisición a través del boton “Módulo DAQ”.
 - a) Usuario accede a interfaz de adquisición.
 - b) Sistema verifica conexión de tarjeta DAQ para correcta adquisición.
 - c) Usuario tiene la opción de verificar el correcto funcionamiento de los dispositivos físicos (hardware) a través del menú “Probar módulo de control”.
 - d) Usuario registra parámetros de adquisición (T muestreo, t concentración, t captura, t limpieza, Voltaje V_c), y se confirman los parámetros con el botón “Aceptar configuración”.
 - e) Usuario ingresa el nombre de archivo para guardar los datos de la adquisición en la carpeta DAQ, luego finaliza con el botón “crear”.
 - f) Usuario ingresa el número de medidas que desea tomar (cantidad de veces que se repite el ciclo de adquisición).
 - g) Usuario indica Iniciar adquisición con el botón “DAQ”.

- 1) Sistema habilita tiempo de concentración en hardware (clase).
 - 2) Sistema habilita tiempo de adquisición en hardware, se guardan los datos en el archivo indicado anteriormente por el usuario.
 - 3) Sistema habilita tiempo de limpieza en hardware.
 - 4) Se eliminan variables y se habilita la interfaz para adquirir nuevos datos.
- h)* Fin del proceso de adquisición. El usuario puede realizar una nueva toma de muestras repitiendo el proceso anterior o regresar a la interfaz principal con el botón “Inicio” con lo que el flujo de proceso continua como sigue.
5. Usuario decide Clasificar muestras a través del botón “Procesar Datos”.
- a)* Usuario accede a la interfaz que le permite preprocesar y procesar los datos.
 - b)* Usuario ingresa datos (adquiridos previamente) a través de la pestaña “Manejo de archivos”, estos se guardan en una matriz en la memoria temporal. Pueden ser tipo “Bnose”, matriz de entrenamiento, de clasificación y/o *label*. Sistema envía mensaje “Carga de archivos exitosa...” en caso positivo. En este nivel el usuario puede realizar cualquiera de los pasos desarrollados desde el item *c* al *g*.
 - c)* Usuario decide Preprocesar datos.
 - 1) Usuario indica “Parámetros” señalando uno o ninguno de los siguientes: GMAX-GMIN, GF-GI, GMAX-GMIN/GMAX, GF-GI/GF. El sistema calcula el parámetro indicado a partir de la matriz en la memoria temporal y la reescribe.
 - 2) Usuario indica “Normalización” señalando uno o ninguno entre: Auto escalado, Centrado, Normalizado, Wavelet. El sistema calcula el valor indicado a partir de la matriz en la memoria temporal y la reescribe.
 - a'* Si el usuario indica transformada Wavelet, accede a la interfaz Wavelet.
 - b'* Usuario indica parámetros de cálculo. El sistema calcula transformada wavelet a partir de la matriz en la memoria temporal y la reescribe.
 - d)* Usuario decide Seleccionar datos, para lo cual tiene las siguientes opciones como alternativa de reducción de los datos:
 - 1) Indicar “Backwardeliminationpnn”.
 - 2) Indicar “Forwarselectionpnn”.
 - 3) Indicar SA PNN.
 - 4) Indicar SA SVM.
 - 5) Indicar SA SVM 2.
 - 6) Indicar SA LSSVM.
 - e)* Usuario decide Analizar datos, por lo cual puede:
 - 1) Indicar PCA.

- a'* Usuario accede a interfaz de PCA.
 - b'* Se indica “calcular”, el sistema procesa los datos en matriz.
 - c'* Usuario indica el número de PCs para el registro de *loads* y *scores*.
 - d'* usuario selecciona “Aplicar”.
 - e'* El sistema muestra resultados de *loads* y *scores* a través de una gráfica.
- 2) Usuario indica Covarianza.
- a'* Usuario accede a interfaz de covarianza.
 - b'* Se indica “Calcular COVARIANZA”, el sistema procesa los datos y muestra un gráfico al usuario.
 - c'* El usuario puede indicar “Aplicar covarianza” para guardar los datos calculados.
- 3) Usuario indica Análisis de clúster.
- a'* Usuario accede a interfaz de Análisis de clúster.
 - b'* Usuario indica parámetros de cálculo. El sistema procesa los datos y muestra resultados a través de un diagrama de clúster.
- 4) Usuario decide usar SVM para la clasificación, con lo cual posee cuatro opciones de análisis, cada una de ellas con parámetros y entrega de resultados independiente.
- a'* SVM.
 - b'* SVM-MClass.
 - c'* LS-SVM.
 - d'* LS-SVM-MClass.
- 5) Usuario decide usar Neuronal para la clasificación, con lo cual posee dos opciones de análisis, cada una de ellas con parámetros y entrega de resultados independiente.
- a'* MLP.
 - b'* PNN.
- f)* Usuario puede realizar nuevamente la Clasificación de muestras limpiando variables usando el botón “Refrescar”.
- g)* Fin del procesado de datos. Botón “Inicio” para volver a interfaz principal.
6. Final del proceso “Bnose”, botón “Salir”.

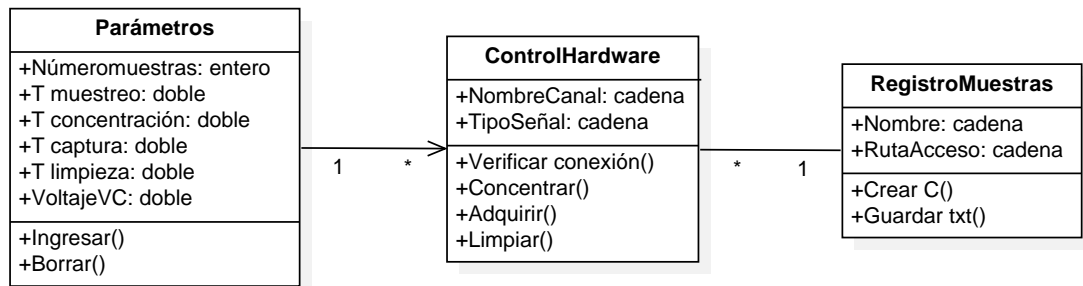
3.2. CLASES DE LA ETAPA DE ADQUISICIÓN.

En el contexto de la nariz electrónica existente, se hace importante realizar un esquema general de la parte de adquisición del sistema, ya que esta es la etapa fundamental que permite la extracción de muestras para un posterior procesamiento, en base a esto se obtiene el diagrama de clases de la figura 33, en donde se observa que está conformado por tres clases. La clase “Parámetros” hace referencia a cada

una de las variables asociadas al proceso de adquisición la cuales son ingresadas por el usuario, esta posee atributos como numero de muestras, tiempo de muestreo, de concentración, de captura y de limpieza, así como un voltaje Vc, los cuales son de tipo entero y doble. Las acciones a las cuales responde esta clase son ingresar y borrar.

Otra entidad es la de “Control HW” quien es la asociación directa entre software y hardware, ya que como su nombre lo indica realiza el control del hardware mediante el envío de señales, las características de esta clase son nombre de canal y tipo, haciendo referencia a si es analógico o digital según como se trabaja en la tarjeta “DAQ”, bajo estos atributos se tienen las operaciones de “verificar conexión”, “concentrar”, “adquirir” y “limpiar” siendo los principales pasos que debe cumplir para llevarse a cabo una correcta toma de medidas.

Figura 33. Diagrama de clases etapa adquisición



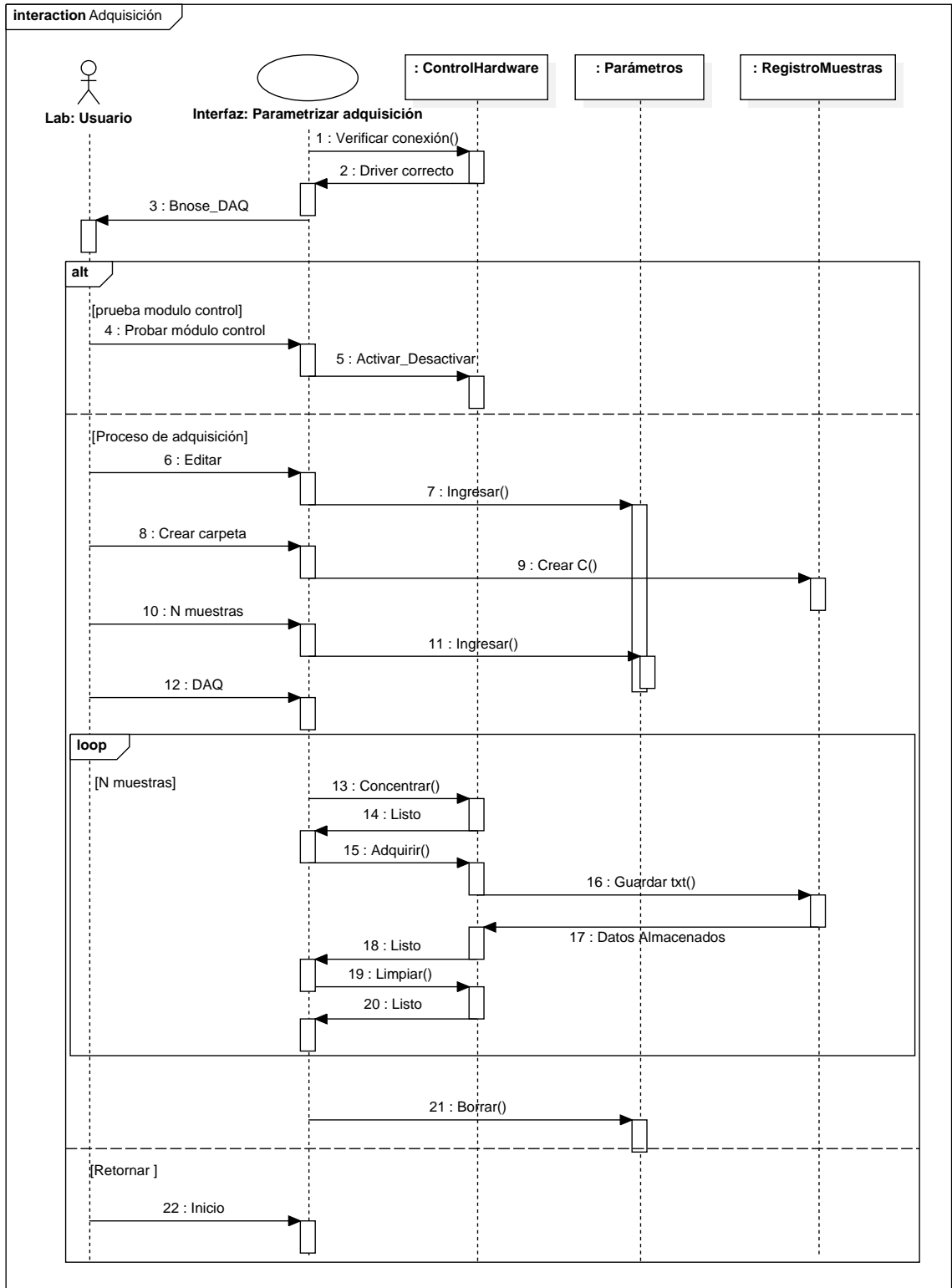
Finalmente se tiene una clase adicional definida como “RegistroMuestras” la cual posee atributos como nombre y ruta de acceso, y a su vez permite llevar a cabo las operaciones de creación de una carpeta “Crear C” y guardar los datos “Guardar txt”, es decir, es la parte encargada de generar el archivo de texto donde se guardan las muestras adquiridas por el sistema, siendo el punto final de recepción de la información necesaria para proceder a un estudio sobre los datos obtenidos.

3.3. SECUENCIA DE LA ETAPA DE ADQUISICIÓN.

El software Bnose está conformado por una etapa de adquisición que además de ser vista desde el diagrama de clases, se presenta en el diagrama de secuencia que se observa en la figura 34, en donde se tiene un usuario de laboratorio como el actor principal, un caso de uso denominado interfaz: parametrizar adquisición y tres clases: “Control hardware (HW)”, “Parámetros” y “RegistroMuestras”.

Para iniciar la clase Control HW verifica la conexión del sistema y la presencia del driver, devolviendo un mensaje con referencia al correcto estado de conexión de la tarjeta de adquisición, y permitiendo que se ejecute la interfaz “Bnose_DAQ”, donde el usuario tiene la opción de dirigirse a la sección “probar módulo de control” para realizar la verificación del estado de la clase “control HW”. A continuación el usuario tiene la posibilidad de configurar los parámetros para llevar a cabo el

Figura 34. Diagrama de secuencia etapa de adquisición



proceso de adquisición, para esto hace clic sobre la opción “editar” que genera el llamado de la interfaz a la clase parámetros para ingresar el valor de las variables, después el usuario solicita crear la carpeta a la clase RegistroMuestras para el posterior almacenamiento de la información. Como parámetro final el usuario envía el mensaje “N de muestras” a la interfaz, con lo cual esta le solicita a parámetros el ingreso número de muestras a realizar, quedando configurada la etapa de adquisición.

Para iniciar la toma de medidas el usuario oprime sobre el botón con el mensaje “DAQ” por medio del cual acciona a la clase “Control HW” a realizar la operación de concentrar, luego este avisa la finalización del proceso devolviendo un mensaje de listo, en donde empieza la operación de adquirir, quien trabaja en conjunto con la acción guardar en la clase RegistroMuestras la cual avisa el correcto almacenamiento de los datos, cuando acaba la clase “Control HW” envía un mensaje de listo a interfaz para dar inicio a la operación de limpieza en la clase control HW la cual de aviso de su finalización, este procedimiento se repite la cantidad N de muestras definidas previamente por el usuario.

Para finalizar la interfaz solicita a la clase parámetros la operación de borrar de tal forma que se pueda dar inicio a una nueva toma o proceder a la opción de oprimir el botón “inicio”, el cual hace un retorno a la interfaz principal del software Bnose.

3.4. CLASES DE LA ETAPA DE PROCESAMIENTO.

Para representar de una forma más técnica el funcionamiento de la etapa de procesamiento del software de la nariz electrónica, se realiza el diagrama de clases mostrado en la figura 35. El diagrama se conforma por 17 clases que interactúan entre sí según el seguimiento secuencial que se desarrolla al usar el programa, de las cuales, contiene 5 clases principales de las que se derivan clases hijas que especifican procesos lógicos y que por concepto heredan los atributos de sus clases madre o superclase. Estas cinco principales clases se encuentran conectadas entre sí debido a que todas son capaces de generar cambios en la matriz de datos (invocando las funciones de la clase archivo) y realizar acciones secuenciales que involucran las funciones de las demás. Todas las clases de la etapa de procesamiento se explican a continuación:

Principales clases madres del procesamiento:

1. **Archivo:** Esta clase representa todos los objetos que interactúan con los documentos de datos guardados, teniendo como atributos, el nombre, extensión y ruta de acceso. Además, incluye a la matriz de datos que se alojan en dichos documentos, por lo tanto, también contiene características que corresponden a datos y dimensión de la matriz. Tiene como funciones, buscar, cargar y guardar, las cuales son utilizadas por todas las clases principales (pre-procesado, procesado, selección de datos y análisis estadístico).
2. **Preprocesado:** Toda la etapa de preprocesamiento que se puede realizar en el software pertenece a esta clase. Los objetos que pertenecen aquí tienen como

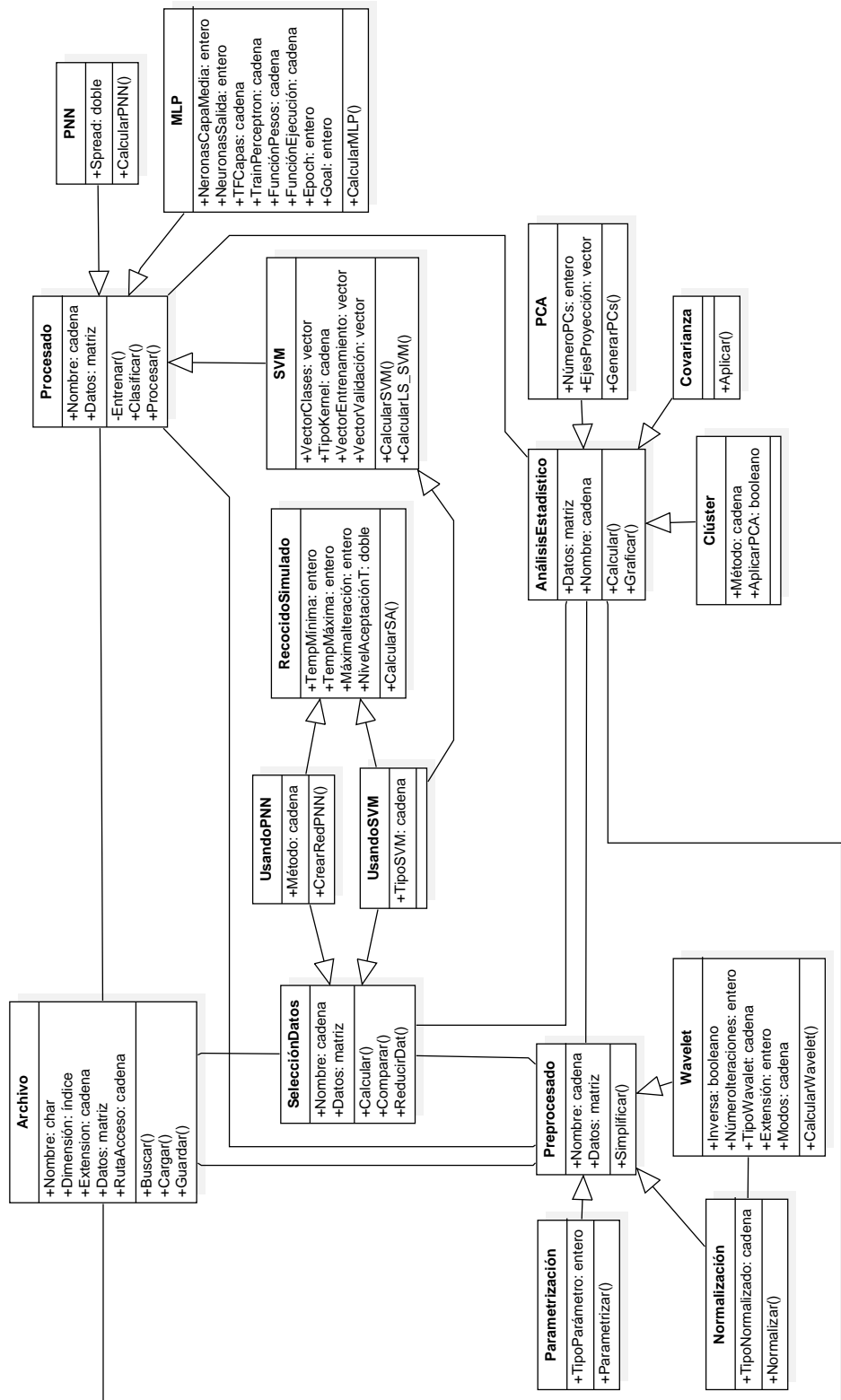
atributos un nombre y los datos tipo matriz. Las acciones que realiza es calcular el resultado del algoritmo seleccionado y transformar los datos ya guardados, de esta forma, simplifica la cantidad de datos adquiridos. A esta pertenecen tres clases hijas, “Parametrización”, “Normalización” y “Wavelet”, donde los objetos de cada una son los tipos de algoritmos que se pueden implementar.

3. **Análisis estadístico:** Principalmente las clases “PCA”, “Covarianza” y “Cluster” pertenecen a esta clase, y representa a los tres algoritmos disponibles para realizar el análisis estadístico de los datos. Los atributos esta clase son nombre y matriz de datos, por lo tanto, sus funciones son calcular (según el algoritmo que se quiera implementar) y graficar los resultados obtenidos.
4. **Selección de datos:** Los objetos de esta clase tienen un nombre que los identifica tipo cadena de caracteres y los datos que se utilizan. Las principales funciones son calcular, comparar y reducir la matriz de datos por medio de alguna de las dos clases hijas que se incluyen en esta: “UsandoPNN” y “UsandoSVM”, las cuales heredan dichos atributos además de las características de las clases de “Recocido Simulado” y “SVM”.
5. **Procesado:** Una de las principales etapas del procesamiento de datos del software se incluyen en esta clase, asignándoles atributos como nombre y datos, funciones como entrenar, clasificar y procesar la información obtenida. Las clases “hijas” son el tipo de procesamiento “MLP”, “PNN” y “SVM”, esta última es superclase de “UsandoSVM” heredando sus atributos.

Subclases del procesamiento:

1. **Parametrización:** Esta clase hija hereda los atributos de preprocesado al igual que la función de simplificar los datos. Además incluye la el tipo de parametrización que se desea realizar, las cuales son los objetos pertenecientes a esta clase. La parametrización de la matriz de datos genera cambios significativos.
2. **Normalización:** Como parte del preprocesado, esta clase agrupa los tipos de normalización que ofrece el software de procesamiento, por lo tanto adiciona este atributo tipo cadena de caracteres (nombre de la normalización), además de simplificar los datos por medio de algoritmos guardados en los objetos que le pertenecen, para ello, es necesario interrelacionarse con la clase Wavelet invocando sus funcionalidades.
3. **Wavelet:** Los atributos relacionados en esta clase hacen referencia al conjunto de parámetros que se deben ingresar para establecer e implementar el algoritmo que calcula la transformada wavelet de los datos.
4. **Cluster:** Esta clase hija tiene la clase madre “AnálisisEstadístico” y agrega dos atributos nuevos referentes al método a aplicar y el uso del algoritmo de PCA.

Figura 35. Diagrama de clases etapa de procesamiento.



5. **Covarianza:** Al igual que Cluster, la clase madre de análisis estadístico le hereda sus atributos y funciones. Este solo adiciona la función aplicar la covarianza a la matriz de datos adquiridos.
6. **PCA:** La clase madre de esta subclase es también análisis estadístico. Adiciona los atributos correspondientes a la cantidad de PCs y los ejes de proyección que se desean utilizar. La función desarrollada es generar los PC correspondientes a la última matriz de datos guarda.
7. **PNN:** Esta subclase pertenece a la clase procesado, heredando sus atributos y funciones como clase hija. Solo incluye el atributo spread, variable de tipo doble, y la función de calcular la red PNN de la matriz de datos.
8. **MLP:** La clase hija MLP adiciona atributos relacionados con cada uno de los parámetros que se ingresan para definir el procesamiento de la red neuronal. Su superclase es el Procesado y tiene como función calcular la red.
9. **SVM:** Esta clase hija de la superclase procesado, tiene como atributos los parámetros que permiten generar la máquina de soporte vectorial, siendo sus funciones calcular tanto el SVM como el LSSVM. A partir de estas variables, se permite implementar este tipo de análisis a la selección de datos, por lo tanto, tiene la clase hija “UsandoSVM” heredando sus atributos y funciones.
10. **UsandoSVM:** Esta clase integra los atributos y funciones de sus superclases. Esta es hija de las clases SVM, recocido simulado y selección de datos, ya que necesitan calcular en su totalidad la máquina virtual usando sus atributos, por lo tanto, solo contiene el tipo de algoritmo que se desea implementar.
11. **UsandoPNN:** Al igual que la clase UsandoSVM, esta clase es hija de las clases selección de datos y recocido simulado, donde integra el atributo que tiene de alusión al método que se desea implementar y la función de crear la red probabilística.
12. **RecocidoSimulado:** Esta clase tiene como atributos las temperaturas, la cantidad de interacciones y el nivel de aceptación que se deben establecer para realizar el recocido simulado (función de la clase). Esto es posible implementarse por medio de PNN o SVM, razón por la cual existen estas dos clases hijas asociadas.

3.5. SECUENCIA DE LA ETAPA DE PROCESAMIENTO.

El diagrama presentado en la figura 36 modela la interacción entre las clases “Archivo”, “Preprocesado”, “SelecciónDatos”, “Análisis Estadístico”, “Procesado” y los casos de uso “Usuario” e “Interfaz: Clasificar muestras”. Se puede acceder a la etapa de procesamiento al finalizar la adquisición de datos, o simplemente accediendo a ella. La interfaz presenta al usuario el panel de inicio correspondiente a la etapa de procesamiento, en donde se ingresan los datos correspondientes y descritos en la sección anterior (hardware y software), la cual se puede consultar para conocer de

forma detallada los procesos de cada una de las clases y objetos para desarrollar las funciones especificadas. Posteriormente, se genera la acción buscar por parte de la interfaz sobre la clase “Archivo”, con el fin de entregar al usuario el mensaje de “CargaExitosa”.

Al cargarse los datos a la plataforma, el usuario tiene varias alternativas para ejecutar, si el usuario desea preprocesar los datos, genera la petición sobre la interfaz, y esta a su vez invoca la función “Transformar” de la clase “Preprocesado”, la cual necesita “Cargar” la información, función que desarrolla la clase “Archivo”. La clase “Preprocesado” utiliza la función “Calcular” y se guardan los cambios generados.

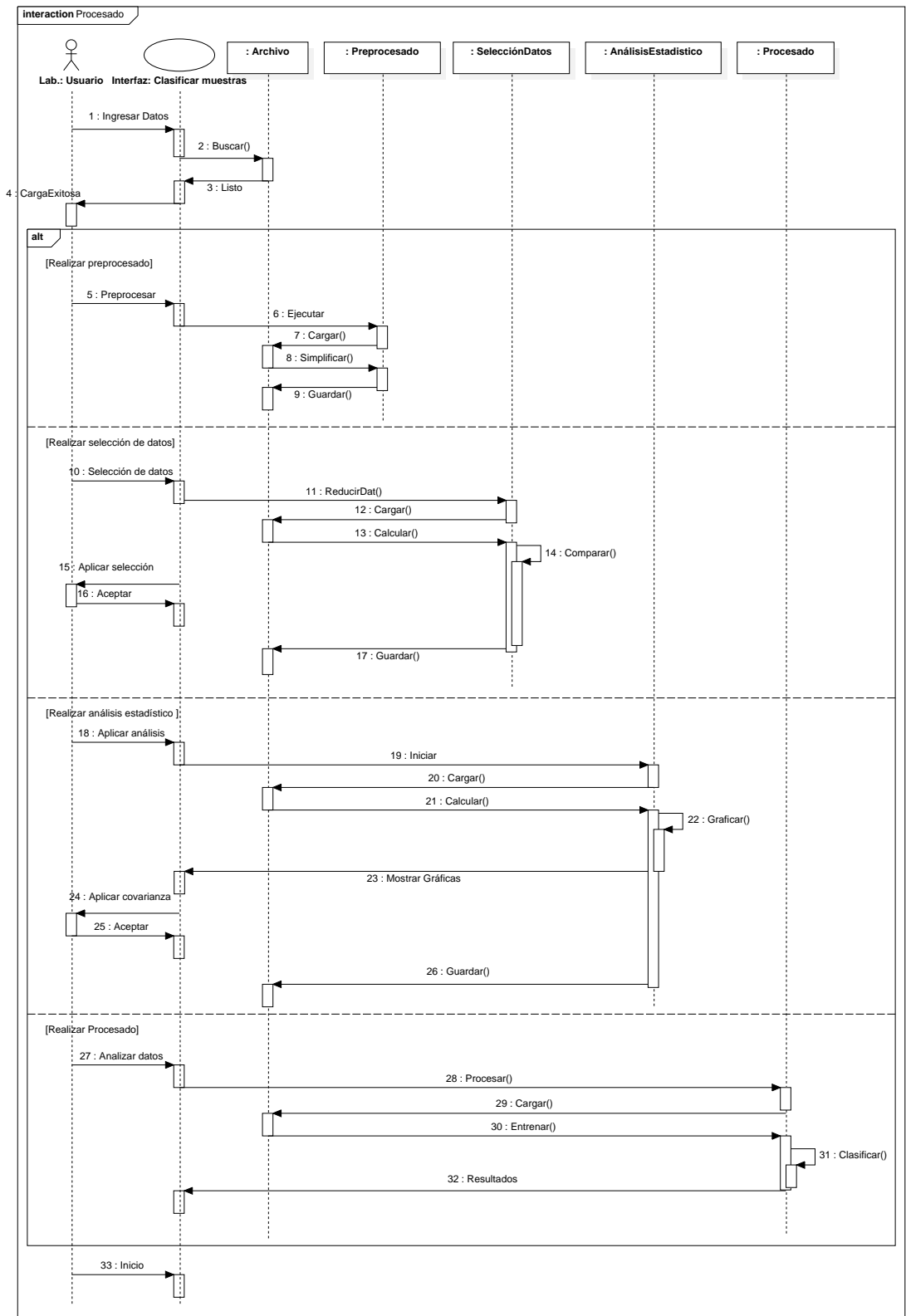
Si por el contrario, el usuario desea aplicar algún tipo de selección de datos, genera a través de la interfaz la función “ReducirDat” de la clase “SelecciónDatos”, la cual se comunica con la clase “Archivo” para cargar la información. Luego, se aplica el algoritmo elegido por el usuario, y se realiza la comparación correspondiente. El usuario tiene la opción de aplicar la selección, por lo tanto la clase “SelecciónDatos” interactúa con la clase “Archivo”.

Para realizar un análisis estadístico, el usuario solicita a través de la interfaz la función “Realizar AE” de la clase “AnálisisEstadístico”, la cual calcula y grafica después de cargar los datos por medio de la clase “Archivo”. La interfaz presenta al usuario la opción de aplicar los cambios generados por el objeto covarianza, si es el caso.

Por último, para realizar el procesamiento de los datos, el usuario llama la función “Analizar datos” del caso de uso “Interfaz” que a su vez invoca la función “procesar” a la clase “Procesado”. Luego, se cargan los datos correspondientes por medio de la clase “Archivo”, para entrenar las redes y clasificar los resultados. Esta información se envía a la interfaz para interactuar con el usuario.

El usuario siempre tiene la opción durante todo el proceso de salir por medio de la función “inicio” sobre el caso de uso “interfaz”, la cual se referencia a la plataforma principal de “Bnose” donde accede al procesamiento o a la adquisición.

Figura 36. Diagrama secuencia etapa de procesamiento.



4. PROPUESTA DE UN SERVICIO TIC BASADO EN NARIZ ELECTRÓNICA

A partir de la comprensión en detalle del concepto, funcionamiento y conformación de la nariz electrónica obtenida de los capítulos anteriores, ya sea desde un enfoque del ingeniero electrónico o bajo la ingeniería de software del UML, es pertinente pasar al punto de analizar un esquema de características básicas que permitirán describir y entender una propuesta innovadora como la de convertir este sistema multisensorial en un servicio TIC.

Como consecuencia surge el presente capítulo donde el lector perteneciente a cualquier área investigativa o profesional podrá analizar y determinar en qué parte del desarrollo de la solución juega un papel importante, entendiendo el por qué hace parte de este proceso y de qué manera puede verlo, con el fin de enfocar sus conocimientos, capacidades y fortalezas. Para esto se expone un enfoque general de la conformación de un servicio, por medio de un modelo de capas, el cual refleja en cada nivel los principales componentes y actores que permiten el desarrollo de una solución itinerante. Estas capas son posteriormente descritas mediante el empleo de representaciones directamente asociadas al lenguaje del ingeniero electrónico (modelo de capas) y del ingeniero de sistemas (MVC), de tal forma que sean complementarios entre sí, obteniendo una comunicación fluida y coherente entre todos, que permita llevar a cabo de manera rápida la implementación y desarrollo de soluciones convergentes. Adicionalmente, en complemento a la capa de cloud computing se presenta la evaluación de lenguajes de programación para la implementación de la lógica de negocio que conformará el servicio de nariz electrónica, teniendo en cuenta el entorno de origen de este procesamiento.

4.1. CARACTERÍSTICAS BÁSICAS DEL SERVICIO

Con el fin de definir aspectos importantes del proceso a llevar para la implementación de la nariz electrónica como servicio itinerante, se presenta el desarrollo de un esquema como el de la figura 37 en donde se puede ver específicamente cada una de las partes que conforman el servicio, a partir de lo cual se pueden designar tareas claras que conllevan a verdaderos avances. El diagrama se adapta a una idea de planteamiento del problema en donde se desarrolla una primera versión teniendo en cuenta una implementación rápida. Para esto, se decide conservar de la nariz electrónica actual (como clasificador multisensorial) el poder de cómputo (código de procesamiento) y llevarlo a un servidor administrado por RadioGis, implementándolo en un sistema embebido el proceso que se encargue de controlar las tareas de adquisición de muestras en el terminal de sensores y envío de datos al servidor; logrando controlar todo el proceso desde cualquier dispositivo con acceso a Internet y un navegador web.

En el modelo de capas del servicio se presenta desde la base hasta la punta la sociedad, tecnologías de usuario, acceso al medio, conectividad, *cloud computing* y aplicaciones como las necesidades a resolver para la implementación del servicio.

Figura 37. Modelo de capas para la propuesta de servicio.



UIS - Radiogis - Cemos - UP - Colciencias - Investigadores

A continuación se describe cada una de las capas mencionando las características más relevantes para el servicio.

4.1.1. Capa de sociedad: Usuario. Siendo la base de la pirámide, constituye la capa de mayor importancia, desde el usuario se construye el servicio, es decir que se deben analizar todas sus necesidades, dificultades y capacidades para desempeñar un agradable uso del servicio o aplicación, es por ello que para la versión más ambiciosa de la nariz electrónica se debe realizar un estudio detallado del usuario, pensando también en un avance por entregas de valor al cliente (Scrum).

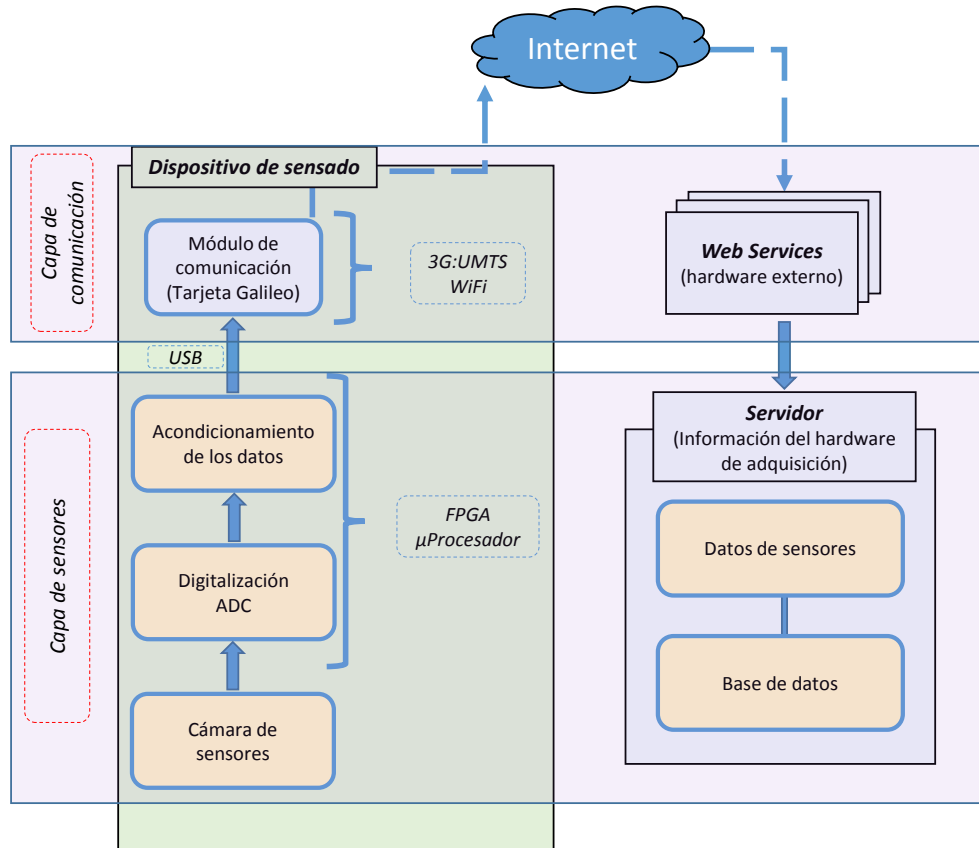
Tabla 8. Caracterización usuario para la primera versión del servicio.

<i>CARACTERÍSTICAS</i>	<i>DESCRIPCIÓN</i>
<i>Conocimientos</i>	<ul style="list-style-type: none"> ■ Manejo de aplicaciones Web ■ Tecnologías de procesamiento y reconocimiento de patrones
<i>Ambiente de trabajo</i>	<ul style="list-style-type: none"> ■ Dispositivo de sensado: Ambiente controlado ■ Dispositivo móvil: Lugar de interés

Como se observa en la tabla 8 el usuario de la primera versión está modelado como el usuario actual del sistema multisensorial, investigadores con interés en el reconocimiento de patrones de olor e identificación de los mismos, agregando la necesidad de un laboratorio remoto, es decir, la posibilidad de experimentar e integrar investigaciones con colegas en distintos puntos geográficos, pudiendo acceder a resultados de otras personas y muestras adquiridas por estos, siempre y cuando haya un previo acuerdo.

4.1.2. Capa de tecnologías de usuario: Dispositivos. Para la implementación del servicio se requiere de dos dispositivos físicos sobre los cuales tendrá control el usuario. El primero de ellos es un terminal de mando, el cual puede ser cualquier aparato de acceso a la red con opción de un navegador (como laptop, Smartphone, Tablet, etc), ya que la aplicación será desarrollada bajo el entorno web, teniendo en cuenta la facilidad de acceso y la independencia con el uso de hardware del dispositivo que la necesite. Para el segundo de ellos, es necesario la construcción de un hardware que cumpla con las especificaciones mostradas en la figura 38.

Figura 38. Esquema del dispositivo de sensado.



Los desarrolladores que se encuentran trabajando desde la perspectiva del software, encuentran a los sensores como una serie de datos de entrada, por lo tanto, se conforma la “Capa de sensores” aparte de la etapa de comunicación, la cual reúne al servidor como base de datos y el hardware de los sensores como el medio del que se pueden adquirir dichos datos.

Este hardware se compone por la cámara de adquisición que debe contener la matriz de sensores requerida para la identificación optimizada de la serie de componentes odoríficas de interés, además de un circuito de salida para tomar las señales de variaciones de voltaje y todo el sistema necesario que asegura el flujo de olores adecuado a través de los sensores (módulo neumático en el sistema actual). La etapa de digitalización puede ser implementada en un sistema embebido como una FPGA o microprocesador, completando el proceso de adquisición de datos.

En la capa de comunicación, se establecen los puentes que permiten acceder a la red de Internet o la base de datos. Es posible implementar una tarjeta que actúe como intermediario de comunicación para la transferencia de datos desde la capa de sensores (terminal sensorica) hacia Internet. Este módulo se puede desarrollar utilizando las potencialidades en IoT que tiene una tarjeta como la Intel Galileo, estableciendo el acceso por medio de Wi-Fi o 3G. Se ha pensado en utilizar esta alternativa gracias a los ejemplares de los que dispone el grupo RadioGis. En la capa de *cloud computing* el terminal de sensores se comunica con el servidor por medio

de un *web service*. Este debe gestionar las solicitudes de los múltiples usuarios que desean acceder a la capa de sensores, donde se encuentra el servidor que redirige los datos hacia la Base de datos para ser almacenados.

4.1.3. Capas de conectividad y acceso al medio: Red de comunicación y arquitectura del software. Cuando se habla de un servicio TIC es lógico pensar en la presencia de una forma de tener acceso a Internet, de aquí surge la idea de emplear la red WWAN (por sus siglas en inglés *Wireless Wide Area Network*), debido a las características de acceso y los propósitos de comunicación con un servidor remoto, además de la amplia cobertura que le permite ingresar en cualquier parte del mundo desde un dispositivo móvil, siendo esto, lo que define de la manera más básica la relación entre las capas de conectividad y el acceso a medio.

En estas capas se encuentran las empresas operadoras de redes móviles, las cuales se enfocan en desarrollar ideas de servicio que sirven de soporte y complemento para la entrega de valor asociada al servicio de nariz electrónica desarrollado, permitiendo así que estos se puedan ver como parte de la idea planteada. Sin embargo, para el desarrollo de la propuesta de este trabajo de grado, no es de alto interés profundizar en esa área, ya que para las primeras versiones del servicio no se tendrán mayor influencia de estas. Aún así no deja de ser interesante revisar el papel que cumplen estos entes y la forma como logran este fin, para ello, es pertinente referirse al trabajo realizado por la ingeniera Estefanía Figeroa titulado “Desarrollo de una Aplicación Prototipo para el Control de Ac” (Buitrago, 2014), donde se presenta con mayor detalle cada uno de los procesos que hacen posible la conexión de cualquier dispositivo a servidor, obteniéndose la portabilidad e incluso escalabilidad de sistemas innovadores.

4.1.4. Capa de *cloud computing* y aplicaciones: Patrón de Modelo Vista Controlador. En esta capa se puede ubicar principalmente al ingeniero de sistemas, el cual se enfoca en hacer la implementación de rutinas y algoritmos mediante los cuales se lleven a cabo, ya sea el almacenamiento, procesamiento o el uso y diseño de *Web Services*. Por esta razón, pensando en una visión más amable y entendible para el lector se explica dicha capa orientada a la propuesta de servicio TIC de la nariz, por medio del patrón Modelo Vista Controlador (MVC) mostrada en la figura 39, el cual es una filosofía de diseño para aplicaciones, que brinda una fácil evolución independiente de cada una de sus capas, o incluso módulos, e incrementa la reutilización y flexibilidad del sistema. (Mestras, 2010)

En este patrón la vista o interfaz gráfica de usuario interactúa con los controladores, mostrando en la capa de servicios las plataformas en las que el usuario puede depositar sus tareas y utilizar cada servicio para el desarrollo completo de su labor dentro de la plataforma web.

La capa de servicios se comunica con la lógica de negocio para acceder a todo el potencial de cómputo que se encuentra en los servidores, encontrándose aquí todo el código necesario para llevar a cabo las operaciones y transformaciones solicitadas

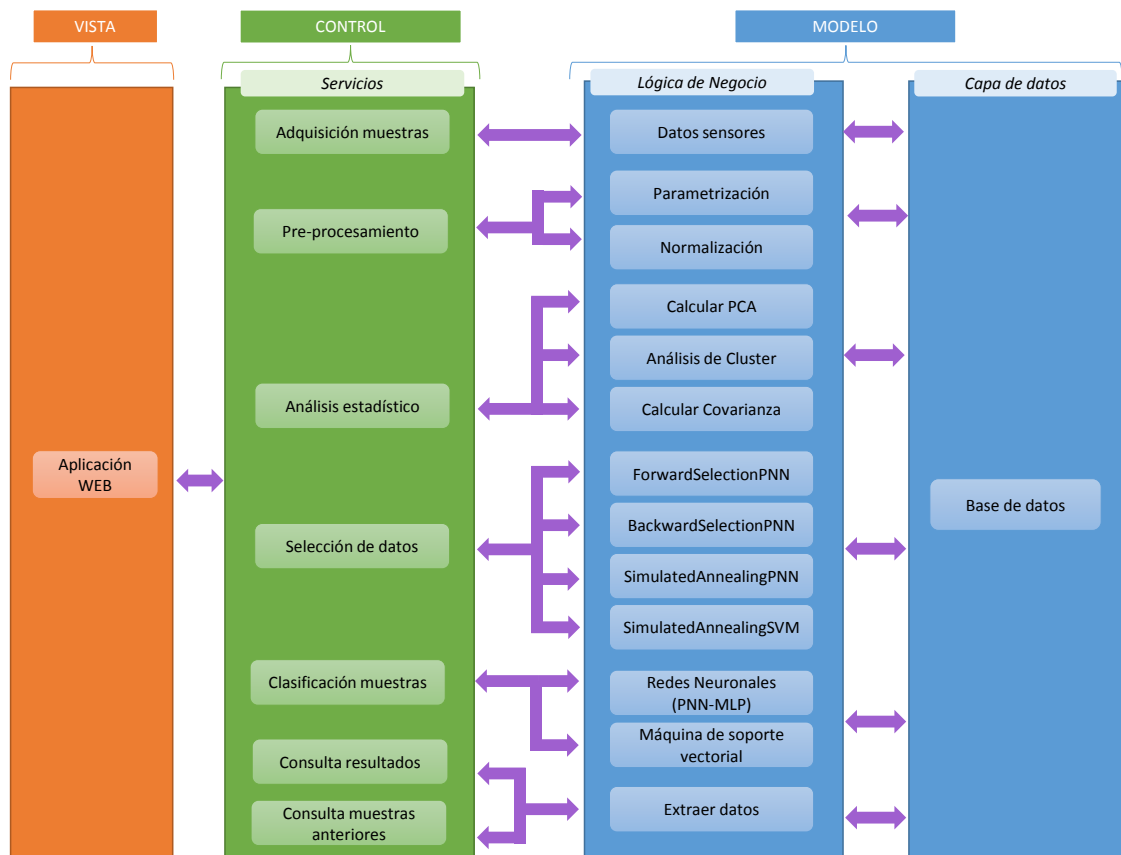
por el cliente tomando como insumo los datos almacenados en la capa de datos.

Específicamente, el acceso al servicio de adquisición de muestras, se relaciona directamente con el dispositivo de sensado el cual le entrega la serie de datos obtenidos de los sensores. La lógica de negocio correspondiente a este servicio, redirecciona los datos hacia la base de datos para su almacenamiento.

Adicionalmente se debe tener en cuenta que existen servicios implementados en la Plataforma de Servicios RadioGis que pueden ser reutilizados para la implementación, como lo es el registro y control de usuarios (llamado SGA User) y el control de datos en el servidor (llamado SGA data).

Como se puede observar cada capa puede ser configurada y rediseñada de acuerdo a las necesidades del usuario y servicio, permitiendo a los desarrolladores hacer mejoras graduales del servicio presentando cambios por capas, por ejemplo puede cambiar totalmente la capa de vista para una interacción más amable con el cliente sin necesidad de cambiar ninguno de los servicios prestados, o agregar nuevos servicios teniendo que cambiar solo un pequeño aspecto de la vista, siendo los cambios en la capa de Modelo los que generan mayor trabajo en el resto de capas.

Figura 39. Patrón MVC del servicio TIC basado en nariz electrónica.



Se puede implementar un servicio inicial como primer *sprint* en el modelo SCRUM para el desarrollo de un servicio TIC basado en nariz electrónica, que llevado a cabo bajo el patrón MVC y un modelamiento por capas permite la escalabilidad del mismo desde el uso de un laboratorio remoto hasta una aplicación específica que solucione algún problema a la sociedad, como el sabueso electrónico para la población militar, haciéndolo de forma integral y gratificante en cada entrega.

4.2. LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE LA LÓGICA DE NEGOCIO

La nariz electrónica en su forma más básica es la unión entre un hardware y un software. Bajo la idea del rediseño de la misma para las necesidades del servicio TIC, se hace indispensable poder comprender la diferencia más importante que se puede implementar, en donde el procesamiento actualmente realizado por medio de un computador va a ser llevado a cabo en la nube, a partir del concepto de *cloud computing*, el cual consiste básicamente en permitir a múltiples usuarios acceder al servicio de computación a través de cualquier dispositivo con acceso a Internet.

Para llevar a cabo este cambio es necesario tener en cuenta el lenguaje de programación asociado al código de procesamiento de la nariz electrónica, ya que de esto depende directamente el montaje del servicio en la nube. Por esta razón a continuación se hace un análisis de las alternativas de lenguajes de programación directamente asociados a la implementación de la lógica de negocio del servicio TIC.

4.2.1. MATLAB como alternativa de implementación. Actualmente el software asociado a la nariz electrónica se encuentra descrito en MATLAB, ante esto es preciso evaluar la posibilidad de llevar el montaje sin necesidad de ninguna traducción a otro entorno. Como alternativa inmediata se encuentra la posibilidad de emplear un servidor existente desarrollado por la empresa MathWorks en asociación con Amazon, en donde se podría emplear directamente el código existente obteniendo acceso al web service y mediante pago de un costo correspondiente al consumo, sin embargo el principal inconveniente de esta consideración radica en la no disponibilidad del servicio actualmente en el país (“MATLAB and Simulink”, s.f.).

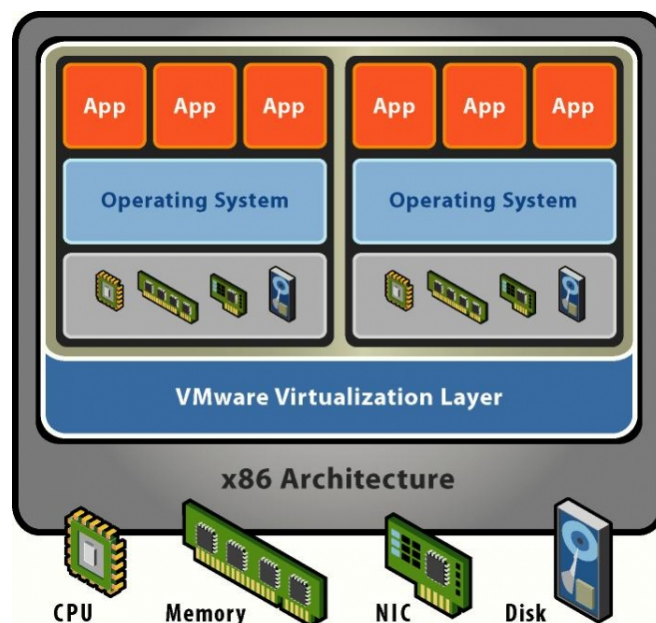
Ante esto y basados en la experiencia de implementación de servidores web por parte del equipo de trabajo del grupo RadioGis, se propone como alternativa la idea de implementar un servidor específico para el servicio de nariz electrónica empleando el concepto de máquinas virtuales en conjunto con MATLAB.

De manera sencilla como se puede ver en la figura 40 una máquina virtual consiste en un programa capaz de emular a un computador con cada uno de los recursos que este implica y posee la capacidad de ejecutar programas desarrollados para sistemas operativos estándar. Las principales ventajas que se resaltan es la posibilidad de crear múltiples máquinas virtuales sobre un mismo ordenador real, las cuales

entre si están totalmente aisladas, de esta forma el estado de una no influye sobre las otras, y la facilidad de gestión del servidor al ser independiente del hardware (Lescano, 2006; Márquez, 2011; Ledesma, 2013). La virtualización se asocia a la idea que se quiere para el servicio desde el punto de vista en donde se busca hacer el procesamiento en la nube, ya que mediante la primera se logra disminuir la cantidad de recursos físicos de la manera más eficiente, para así pasar al nivel en donde estas máquinas virtuales son accedidas como un servicio, permitiendo reducir el peso y costo del sistema actual.

La idea en general consiste en el planteamiento de dos posibilidades: primero hacer el montaje de un servidor basado en múltiples máquinas virtuales con soporte del software MATLAB, sin embargo implica gastos asociados a licencias por cada una de las máquinas desarrolladas sobre las cuales se hará ejecución del programa; la otra opción consiste en no hacer la instalación de todo el software sino solo la del compilador asociado a MATLAB con el fin de aprovechar la cualidades de ejecutable que permite llevarse a cabo basado en el código existente descrito como una interfaz gráfica de usuario, en donde la inversión principal en software sería debida a la adquisición de una única licencia de MATLAB con las toolboxes necesarias, así como la licencia del compilador, siendo esta posibilidad la más viable al no presentar gastos influenciados por el número de usuarios a atender, además de ser factible para el desarrollo de una primera versión aprovechando la licencia estudiantil adquirida por la universidad para nivel investigativo, sin embargo se debe tener presente un costo por licencia aproximado 6'733.597 pesos en adición al precio base de cada una de las toolbox necesarias en un futuro para poder brindar el servicio (“MATLAB The Language of Technical Computing”, s.f.).

Figura 40. Esquema concepto general de máquina virtual

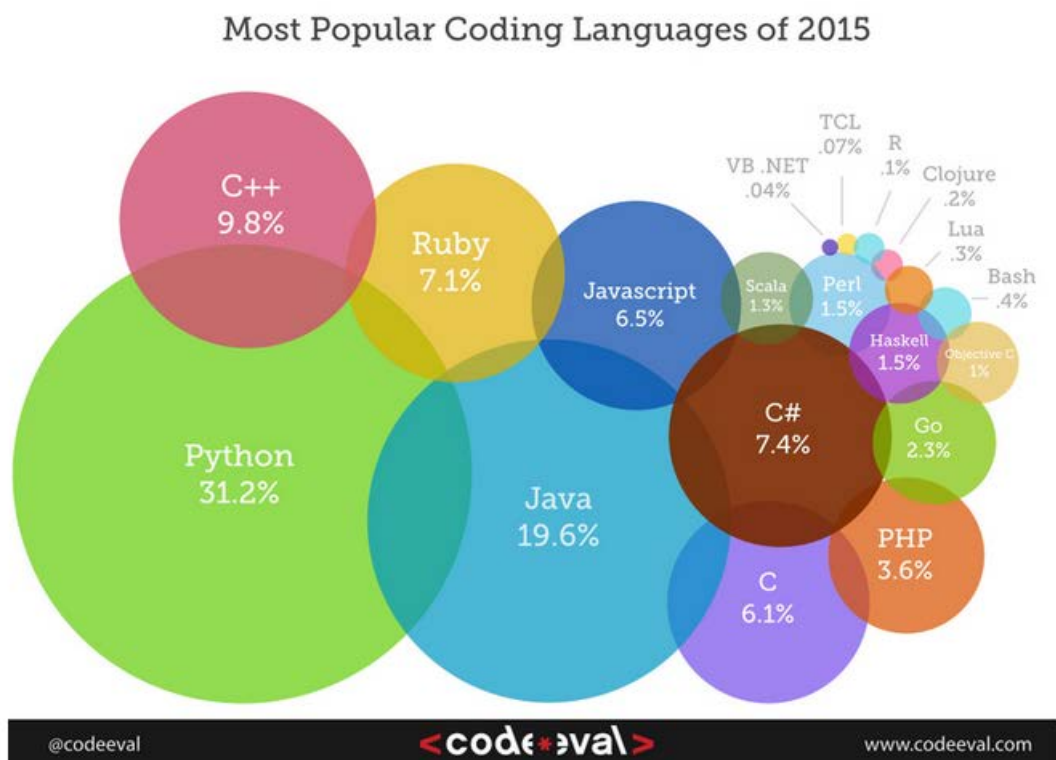


Tomado de (“VMware and x86 virtualization, guides and Tutorials”, 2005)

4.2.2. Lenguajes de programación orientados a objetos. En el ámbito de desarrollo de aplicaciones directamente relacionadas con la idea de servicio TIC de la nariz electrónica, se encuentra la posibilidad de realizar una traducción del código de procesamiento existente a un lenguaje orientado objetos, debido a que este tipo de enfoque ha venido tomando gran aceptación, en la medida que brinda ventajas como extensibilidad, evolución, y mantenimiento del código, aspectos importantes para la estructuración y generación de un servicio que se encuentra en condiciones de mejora continua para la satisfacción del cliente (Aguilar, 1998).

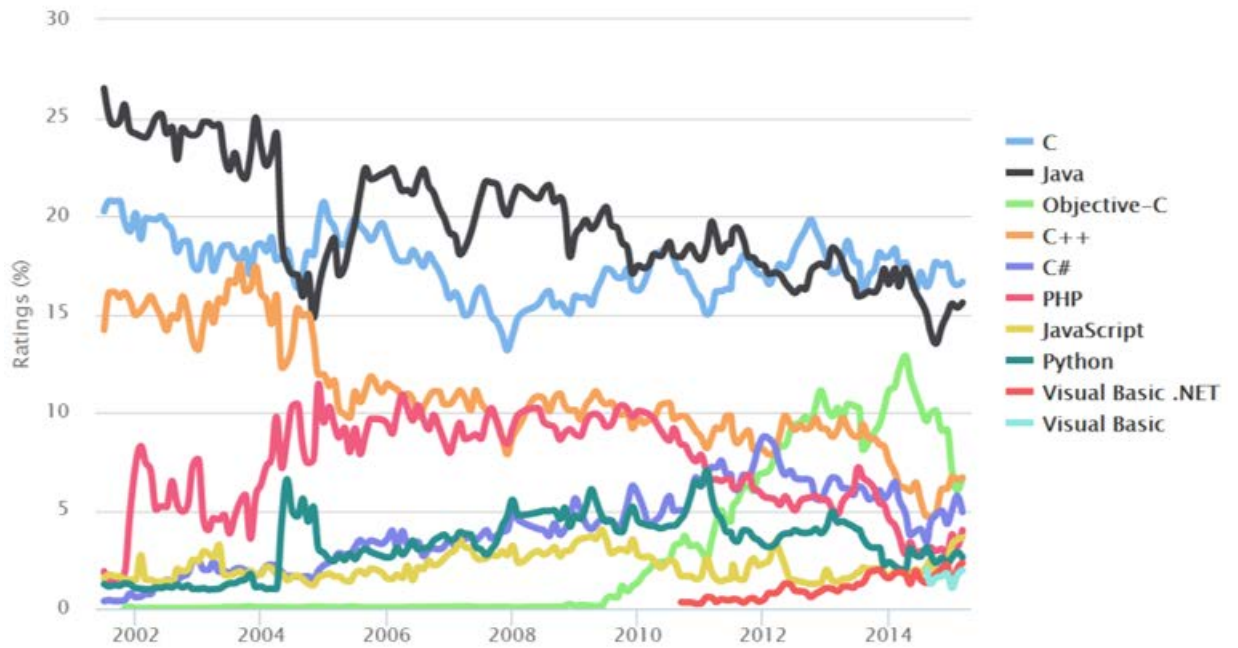
En la medida en que se desea evaluar la posibilidad de lenguaje más adecuada para las necesidades y requerimientos propios de la solución, se inicia por analizar el índice de popularidad entre los desarrolladores y el índice de calidad de software registrado en los últimos tiempos sobre lenguajes de programación orientados a objetos, por medio de los gráficos estadísticos mostrados en la figura 41 y figura 42 de la plataforma codeeval (“Most Popular Coding Languages of 2015”, 2015) y la empresa TIOBE software (“TIOBE Index for March”, 2015) respectivamente, dando como resultado los tres lenguajes a ser analizados en el transcurso de esta sección, los cuales son Java, Objective-C y Python.

Figura 41. Gráfico estadístico de la popularidad lenguajes de programación por CodeEval



Tomado de (“Most Popular Coding Languages of 2015”, 2015)

Figura 42. Gráfico estadístico del índice en la calidad del software por TIOBE



Tomado de (“TIOBE Index for March”, 2015)

Como estrategia para la evaluación de los lenguajes se realiza una comparación mediante la tabla 9, bajo la revisión de algunas de las principales características entre las que se encuentran: el paradigma haciendo referencia al tipo de lenguaje, las implicaciones de la traducción desde MATLAB del código existente al lenguaje, complejidad en la sintaxis, el tipo de licencia requerida para su uso, las aplicaciones más comunes y el soporte de librerías asociadas al reconocimiento de patrones. A partir de esta revisión se da la propuesta de lenguaje más adecuada para la implementación de la solución.

Tabla 9. Comparación lenguajes de programación orientados a objetos

Aspectos	Python	Java	Objective C
Paradigma	Imperativo e interpretado. Uso de programación orientado a objetos y algunas utilidades de la programación funcional	Se basa en el concepto de programación orientada a objetos	Orientado a objetos, creado como un superconjunto orientado a C para un mayor acercamiento al lenguaje de máquina.

Aspectos	Python	Java	Objective C
Traducción desde MATLAB	Sintaxis similar a MATLAB en cuanto a la funcionalidad de operaciones matriciales (NUMPY). Compatibilidad con algunas librerías	Es necesario realizar una reprogramación con lógica orientada a objetos	Mediante MATLAB Coder se puede realizar la conversión a lenguaje de programación C, y este se puede integrar a Objective C.
Complejidad de sintaxis	Al ser un lenguaje de muy alto nivel utiliza una sintaxis sencilla y cercana al usuario para una rápida implementación de las aplicaciones desarrolladas (consecuencia de un lenguaje interpretado)	Derivada del lenguaje de programación C++, por lo que sus reglas de sintaxis son muy parecidas. Sin embargo la estructura del lenguaje se basa en paquetes	Lenguaje de bajo nivel con funcionalidades orientada a objetos con sintaxis idéntica a C y SmallTalk.
Librerías de reconocimiento de patrones	Acceso a OpenCV, librería de reconocimiento de patrones para C. Para redes neuronales Con-x, para máquinas de soporte vectorial LibSVM. Análisis estadístico por medio de RPY	Acceso a OpenCV, librería de reconocimiento de patrones para C	Acceso a OpenCV, librería de reconocimiento de patrones para C. Librerías para redes neuronales de código libre como FANN, MLPNeuralNet (licencia BSD), SNN-NeuralNet (licencia MIT).

Aspectos	Python	Java	Objective C
Aplicaciones	Aplicaciones Web, infraestructura en la nube. Desde el 2008, por medio de la incorporación de librerías se utiliza en el ámbito investigativo como alternativa de MATLAB	Contiene ByteCodes que emula bajo máquinas virtuales. Uso para <i>applets</i> , programas embebidos en Web, entre otros	Como lenguaje de programación en MAC OS X, IOS y GNUStep.
Costos de licencias	Licencia de código abierto y gratuito denominado Python Software Foundation License	Licencia tipo GNU GPL / Java Community Process	Código abierto, gratuito bajo licencia GPL.

Como se puede observar en la tabla comparativa, cada uno de estos tres lenguajes brinda la ventaja para el desarrollador de poseer licencia de código abierto, igualmente cada uno se basa en la programación orientada a objetos, sin embargo para el caso de Python también se puede ver que tiene un enfoque de programación funcional y es un lenguaje imperativo, siendo así el más completo en cuanto a enfoques de programación. Otro aspecto de gran importancia para el servicio es tener presente que el código existente está en MATLAB, por esta razón cuando se evalúa la facilidad de traducción se descarta Java como opción para una primera versión, a diferencia de los otros dos lenguajes que permiten una traducción más directa. Al revisar aspectos de sintaxis es un poco más complicado tener un criterio de selección de cual es mejor, sin embargo basados en la información obtenida y lo que mejor se adapta al código existente se podría pensar en la sintaxis de Python, ya que es un lenguaje de alto nivel con similitudes a la sintaxis original, a diferencia de ObjectiveC que es casi el mismo lenguaje de C, así como el de Java que se parece a C++. Como requisito para el objetivo de desarrollo del servicio en base al código existente se busca el lenguaje con más existencia y compatibilidad de librerías asociadas al tema de reconocimiento de patrones, ya que de esto depende críticamente el correcto desempeño del código funcional de la nariz electrónica, ante esto los tres lenguajes permiten usar librerías como OpenCV directamente asociadas con el tema, sin embargo Python se destaca por poseer compatibilidad con librerías de MATLAB que son empleadas en el código base. Finalmente se hace revisión de las aplicaciones en las que se destaca cada uno de estos lenguajes, en especial Python y Java en aplicaciones web, directamente relacionada con la idea de implementación de servicio TIC.

Siendo consecuentes con el respectivo análisis y la revisión hecha sobre los potenciales

aspectos en cada uno de los lenguajes, se encuentra a Python como el lenguaje de programación basado en la filosofía de programación orientada a objetos más adecuado para la traducción e implementación del código de la nariz electrónica existente en pro del servicio, debido a que es un lenguaje con propiedades similares a las de MATLAB y compatibilidad de librerías asociadas al tema de reconocimiento de patrones empleadas.

5. CONCLUSIONES

A partir del desarrollo y finalización del presente trabajo de grado, se obtienen las siguientes conclusiones.

Gracias a nuestro trabajo se cuenta con una caracterización de la nariz que consiste en una descripción de hardware y una descripción a nivel de software hecha mediante UML que facilita al ingeniero de sistemas comprender cómo está organizado el software actual, como insumo para la implementación de un servicio.

Los componentes básicos de una nariz electrónica, como lo son el módulo de sensores y el sistema de inteligencia artificial, permiten ver a esta bajo las mismas fases funcionales de un servicio TIC en la medida en que se le adapte un canal de comunicación que le otorgue acceso a la red.

El lenguaje UML y los diagramas diseñados sirven como herramienta para replantear el sistema de reconocimiento de patrones hacia una filosofía de programación orientada a objetos, que permita un desarrollo fluido de futuras mejoras en el sistema y el servicio mediante entregas que generen una evolución gradual.

El dispositivo hardware que caracteriza la propuesta de servicio está conformado sólo por los sensores, acondicionamiento de señales y módulo de comunicación con la nube, careciendo de cualquier fase de procesamiento, lo cual permite que se obtengan vastas bases de datos individuales y colectivos de información de muestras de olor en el servidor, generando que el sistema de reconocimiento de patrones evolucione y sea robusto para las necesidades de diferentes clientes futuros del servicio nariz electrónica.

Con el fin de convertir la nariz electrónica en un servicio itinerante y de realizar una primera versión en el ámbito investigativo, es posible implementar el servicio haciendo uso del procesamiento en MATLAB desde una máquina virtual.

En busca de la evolución en la prestación del servicio de nariz electrónica a múltiples usuarios, la escalabilidad del mismo y una óptima gestión del servidor, el uso de un lenguaje de programación más cercano al entorno de desarrollo de aplicaciones en la nube como Python permite la implementación del procesamiento de datos a partir de una conversión desde MATLAB, ya sea utilizando la misma estructura del código actual o aprovechando las ventajas de una programación orientada a objetos. Por otro lado, Python es un lenguaje que empieza a tomar parte en el proceso de formación de estudiantes de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones de la UIS.

La tendencia de este proyecto es la disolución de la línea que divide a los ingenieros de sistemas con los electrónicos. Es posible que surja un nuevo tipo de ingeniero, el convergente.

6. RECOMENDACIONES PARA TRABAJOS FUTUROS

Según las observaciones obtenidas durante el desarrollo de este proyecto de grado, se realizan las siguientes recomendaciones para futuros trabajos y/o continuidad del tema.

El reconocimiento de patrones del sistema multisensorial actual debe complementarse con un módulo que tome como base la clasificación de olores y avance hasta lograr la identificación de los volátiles, mejorando las capacidades del sistema como nariz electrónica.

Es pertinente verificar que la primera versión de la nariz electrónica como servicio contenga como mínimo las mismas capacidades del sistema actual, con el fin de garantizar la mejora continua y la conformación de un servicio estable.

Con el fin de continuar con el diseño, desarrollo e implementación del servicio de nariz electrónica se recomienda el empleo de metodologías ágiles como Scrum, ya que ayuda a enfocarse en la base de la plataforma de servicios (sociedad), la productividad, flexibilidad y reducción de tiempo en la entrega de valor.

Para la prestación del servicio es importante prestar atención al sistema químico, debido a que cada aplicación o usuario que requiera el servicio presenta necesidades distintas, por ende el sistema de sensado debe ser adecuado de acuerdo a esas exigencias (capacidad de concentración óptima, cantidad y tipo de sensores acordes a la composición química de las muestras tratadas) para no presentar problemas de poca capacidad de identificación o sobrestimar recursos para la prestación del servicio.

BIBLIOGRAFÍA

- Acevedo, C. M. D. (2013, Enero-Abril). Optimización de una nariz electrónica a través de un método de selección de variables. *Revista el hombre y la máquina*(41), 56-63.
- Aguilar, L. J. (1998). *Programación orientada a objetos* (U. P. de Salamanca, Ed.). McGraw-Hill.
- Annan, K. (Ed.). (2003). *Discurso inaugural de la primera fase de la wsis*.
- Bautista, J. C. (2014). *Matemáticas y programación en python*. Autor.
- Betancourt, G. A. (2005, Abril). Las máquinas de soporte vectorial (svms). *Scientia et Technica Año XI*(27), 121-167.
- Boothe, H. (2013). What's the difference between virtualization and cloud computing? *virtualcommand go beyond the cloud*. (<http://www.virtualcommand.com/virtualization-cloud-computing-difference/> [citado en marzo de 2015])
- Buitrago, E. F. (2014). Desarrollo de una aplicación prototipo para el control de ac. *Universidad del Valle*. (diapositivas de resumen <https://prezi.com/zgp9cnjj0mu5/desarrollo-de-una-aplicacion-prototipo-para-el-control-de-ac/> [citado en abril de 2015])
- Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.
- Cadena, C. J. M. . P. L. S. M., J. A. (2008). Aplicación de redes neuronales probabilísticas en la detección de fallas incipientes en transformadores. *Scientia et Technica*.
- Celia Clemente, V. C. A. G., Lara Fajardo (Ed.). (2007). *Python*.
- Centro de investigación científica y de desarrollo en tecnologías de la información y las comunicaciones (centrotic). (2012). *CentroTIC, Universidad Industrial de Santander*. (<https://sites.google.com/site/centroticuis/> [citado en abril de 2015])
- Cosino Distante, P. S., Nicola Ancona. (2003). Support vector machines for olfactory signals recognition. *Sensors and Actuators B*, 88, 30-39.
- De matlab a c de manera sencilla. (2013). *MathWorks*. (<http://www.mathworks.com/videos/matlab-to-c-made-easy-82035.html> [citado en enero de 2015])
- Desarrollar modelos para el diseño de software. (2014). *Microsoft Developer Network*. (<https://msdn.microsoft.com/es-es/library/dd409436.aspx> [citado en octubre de 2014])
- Discurso sobre el almacenamiento en la nube. (2007). *IEEE Computer Society*.
- Durán, C. (2005). *Diseño y optimización de los subsistemas de un sistema de olfato electrónico para aplicaciones agroalimentarias e industriales* (U. R. i Virgili, Ed.). Departament D'Enginyeria electrònica elèctrica i automàtica.
- Figaro gas sensors 1-series 8-series. (2013). *Figaro an ISO9001 company - Product catalogue*.
- Florez, C. H. (2007). *Diseño de una nariz electrónica como discriminador de olores utilizando algoritmos genéticos y redes neuronales artificiales* (U. R. Palma, Ed.). Escuela profesional de ingeniería electrónica.

- Govett, D. (2013). SNeuralnet. *GitHub*. (<https://github.com/devongovett/SNNeuralNet> [citado en enero de 2015])
- Grupo de investigación radiogis. (2010). *RadioGis, Universidad Industrial de Santander*. (<https://sites.google.com/a/radiogis.uis.edu.co/> [citado en febrero de 2015])
- Gualdrón, O. E. (2005). *Desarrollo de diferentes métodos de selección de variables para sistemas multisensoriales* (U. R. i Virgili, Ed.). Departament D'Enginyeria electrònica elèctrica i automàtica.
- Hernández, E. (2012). El lenguaje unificado de modelado (uml). *DISCA Departamento de informática de sistemas y computadores de la UPV*.
- Hernández, F. L. (2008). El lenguaje objective-c para programadores c++ y java. *MacProgramadores*.
- Hernández, J. C. (2010). Implementación de aplicaciones y herramientas de post-procesamiento de imágenes médicas vía web. *Universitat Autònoma de Barcelona*.
- Hillegass, A., y Ward, M. (2013). *Objective-c programming: The big nerd ranch guide* (B. N. Ranch, Ed.). Pearson Technology Group.
- Iveth Moreno, R. G. F. M. A. J., Rony Caballero. (2009). La nariz electrónico: Estado del arte. *Revista Iberoamericana de Automática e informática industrial*.
- James Rumbaugh, I. J. (2000). *El lenguaje unificado de modelado. manual de referencia* (A. Otero, Ed.). ADDISON WESLEY.
- Jesús Brezmes, C. M. D. (2007). Últimas tendencias en el diseño de sistemas de olfato electrónico. *Revista colombiana de tecnologías de avanzada*.
- Jesus Brezmes, X. C. (2002). *Diseño de una nariz electrónica para la determinación no destructiva del grado de maduración de la fruta* (U. P. de Catalunya, Ed.). Departament de teoria del senyal i comunicacions.
- J. Lozano, M. A. I. S. M. G. A. M. M. C., P. Santos. (2010). Sistema de nariz electrónica para el análisis de la evolución del vino controlada por pc. *Comité español de automática*.
- Joachims, T. (2013). Svmlight support vector machine. *Cornell University, Department of Computer Science*. (<http://svmlight.joachims.org/> [citado en enero de 2015])
- Julian W. Gardner, P. N. B. (1994). A brief history of electronic noses. *Sensors and actuators b*, 18, 19, 211-220.
- Ledesma, A. C. (2013). *El uso de programas de cómputo en los cursos de la carrera de actuaría en la facultad de ciencias, unam* (U. N. A. de México, Ed.).
- Lescano, W. S. (2006). Máquinas virtuales(vmware, virtual pc,sandbox. qué son y para qué sirven. *aprenderaprogramar.com:Didáctica y divulgación de la programación*. (<http://aprenderaprogramar.com/> [citado en marzo de 2015])
- L. Nolle, A. A. H. J. A. W., D.A. Armstrong. (2002, April). Simulated annealing and genetic algorithms applied to finishing mill optimisation for hot rolling of wide steel strip. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 6(2), 104-111.
- Luis A. Munoz, H. J. V., Luis E. Mendoza. (2013). Segmentación de imágenes de resonancia magnética irm utilizando ls-svm y análisis multiresolución wavelet. *TecnoLógicas*, 681-693.

- Mark Berenson, D. L. (1987). *Estadística para administración y economía. conceptos y aplicaciones*. Editora Interamericana de México.
- Martínez, E. (2007). Historia de la itu. *Eveliux.com: Artículos sobre redes, telecomunicaciones y tecnologías de la información*. (<http://www.eveliux.com/mx/curso/historia-de-la-itu.html> [citado en octubre de 2014])
- Masso, M. D. (2014). Selección de variables en problemas anchos con alta correlación. *Universidad Nacional de Rosario*.
- Matlab and simulink. (s.f.). *awsmarketplace*. (<https://aws.amazon.com/marketplace/pp/B00F009JDY> [citado en enero de 2015])
- Matlab the language of technical computing. (s.f.). *MathWorks*. (<http://www.mathworks.com/products/matlab/> [citado en enero de 2015])
- Mestras, J. P. (2010). Estructura de las aplicaciones orientadas a objetos - el patrón modelo-vista-controlador (mvc). *Universidad Complutense Madrid*. (diapositivas de resumen <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf> [citado en abril de 2015])
- Most popular coding languages of 2015. (2015). *Codeeval*. (<http://blog.codeeval.com/> [citado en enero de 2015])
- Márquez, A. (2011). *Virtualización de servidores* (U. P. de Catalunya (UPC), Ed.). Facultad de Informática de Barcelona.
- MSc, A. G. D. (Ed.). (2015). *Desarrollo de aplicaciones móviles*.
- Ni usb-621x specifications. (2009). *National Instruments*.
- Ortega, H. (Ed.). (2013). *Centro de investigación científica y tecnológica en tecnologías de la información y las comunicaciones-centro tic*.
- Paloma Teresita Gutiérrez Rosas, M. D. H. R. A. J. R. L., José Antonio Vázquez López. (2013). Aplicación de la red neuronal probabilística para la clasificación de productos conforme a sus especificaciones. *Innovation in Engineering, Technology and Education for Competitiveness and Prosperity*.
- Pavlov, M. (2013). *Mlpneuralnet*. *GitHub*. (<https://github.com/nikolaypavlov/MLPNeuralNet> [citado en enero de 2015])
- Plan vive colombia digital. (2010). *Ministerio de tecnologías de la información y las comunicaciones MINTIC*. (<http://www.mintic.gov.co/portal/vivedigital/612/w3-channel.html> [citado en octubre de 2014])
- Pérez Ramírez, F. C. H., F. O. (2007). Las redes neuronales y la evaluación del riesgo de crédito. *Revista Ingenierías Universidad de Medellín*.
- Roca, J. M. (2014). ¿qué es el internet de las cosas? *Informe TIC plus*. (<http://www.informeticplus.com/que-es-el-internet-de-las-cosas> [citado en febrero de 2015])
- Rojas, L. D. (2010). *Diseño y desarrollo de un equipo multisensorial basado en sensores de gases químicos para la validación de diferentes métodos de selección de variables* (U. de Pamplona, Ed.). Departamento de ingeniería eléctrica, electrónica, sistemas y telecomunicaciones.
- Segura, B. (2012). *Fann*. *PortalProgramas*.

- (<http://www.portalprogramas.com/fann/> [citado en enero de 2015])
- S. Kirkpatrick, M. P. V., C. D. Gelatt. (1983, Mayo). Optimization by simulated annealing. *Science*, 220(4598), 671– 680.
- Software de comunicaciones, iphneos. (s.f.). *Universidad Carlos III de Madrid*. (<https://sites.google.com/site/swcuc3m/home/iphone/3-desarrollo-de-aplicaciones/3-2-librerias-basicas/3-2-1-objective-c> [citado en enero de 2015])
- Suykens, J., y Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters* 9, 293-300.
- Tiobe index for march. (2015). *TIOBE SOFTWARE*. (<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> [citado en enero de 2015])
- Trejo, J. A. R. (2006). Las maquinas de vectores de soporte para identificación en línea. *Centro de investigación de estudios avanzados del instituto politécnico nacional*.
- Useche, P. J. L. (2009). Reconocimiento de patrones utilizando máquinas de soporte vectorial (msv). *Universidad Nacional Abierta*.
- Vapnik, V. N. (1999, september). An overview of statistical learning theory. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 10(5), 988-999.
- Vmware and x86 virutalization, guides and tutorials. (2005). *Tacharena community*. (<http://forums.techarena.in/guides-tutorials/1104460.htm> [citado en marzo de 2015])
- Walter F. Wilkens, J. D. H. (2006). An electronic analog for the olfactory processes. *Annals of the New York academi of sciences*.

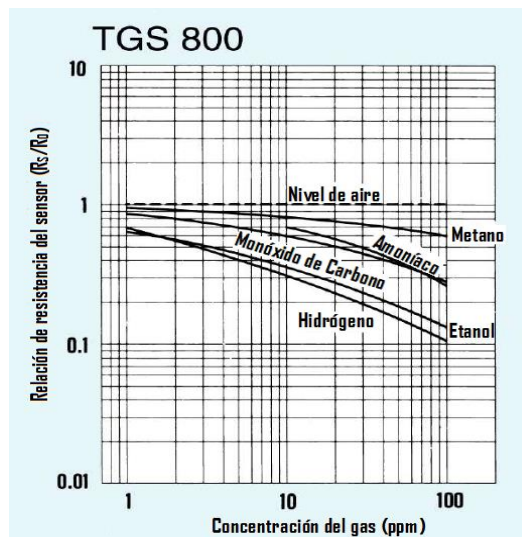
ANEXOS

ANEXO A. Curvas de caracterización de sensibilidad de los sensores Figaro 8

Sensor TGS 800R: Este sensor tiene una aplicación dirigida al control del aire, es decir, reacciona ante presencia de aire contaminado con elementos odoríficos tales como tabaco, gasolina, etc. El TGS 800R, tiene una curva característica de sensibilidad mostrada en la figura 43, con un R_o medido bajo condiciones de aire limpio.

En la curva de sensibilidad se observa que el sensor no presenta reacción alguna ante distintas concentraciones del aire limpio ($R_s = R_o$), cuando existe presencia de componentes distintos responde con una disminución en la resistencia R_s a medida que aumenta dicha concentración. Específicamente el sensor responde más rápido ante la presencia del amoníaco que ante al metano.

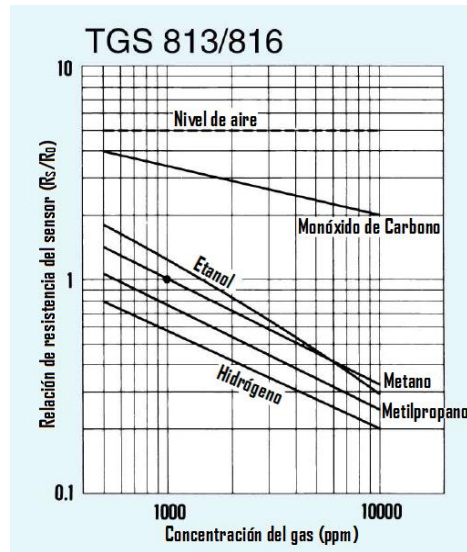
Figura 43. Curvas de caracterización de sensibilidad del TGS 800R.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 813: Este tipo de sensor es capaz de reaccionar ante cualquier tipo de Hidrocarburos, por lo tanto, su resistencia cambia ante la presencia de gases combustibles. Suele ser usado como detector de fugas gases domésticos, vehículos, o incluso detectores portátiles de gas. La gráfica de la figura 44 muestra la curva característica de la sensibilidad sensor, con un R_o ante 1000 ppm de metano.

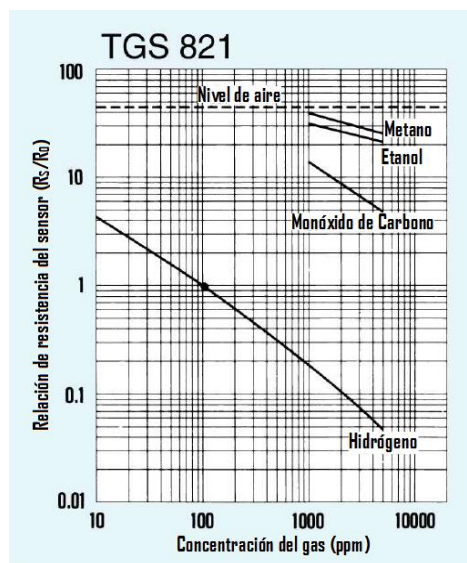
Figura 44. Curvas de caracterización de sensibilidad del TGS 813.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

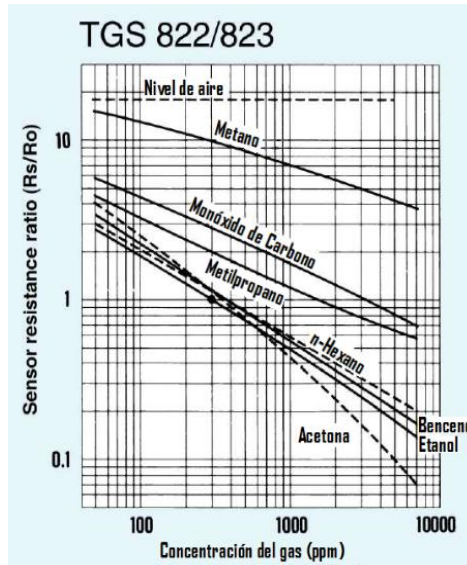
Sensor TGS 821: El TGS 821 es un sensor con alta sensibilidad ante el hidrógeno, es decir, es capaz de detectar gases combustibles. Este puede reaccionar ante concentraciones tan bajas como 50 ppm de Hidrógeno, haciéndolo ideal para aplicaciones industriales. La gráfica de la figura 45 muestra la curva característica de la sensibilidad sensor, con un R_0 ante 100 ppm de hidrógeno.

Figura 45. Curvas de caracterización de sensibilidad del TGS 821.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

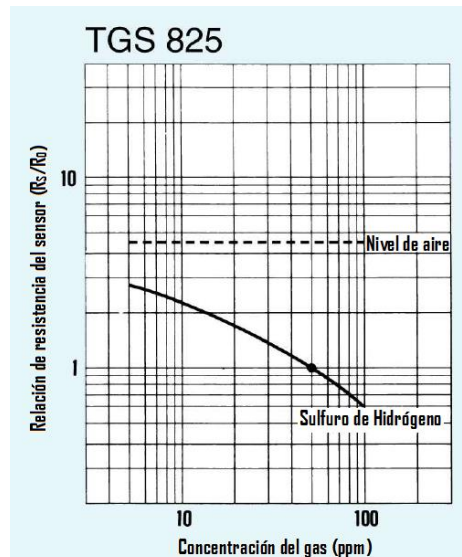
Figura 46. Curvas de caracterización de sensibilidad del TGS 822.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 822: El TGS 822 tiene una alta sensibilidad ante gases combustibles de vapores de alcohol, como el mismo alcohol, el tolueno, xileno, entre otros. La gráfica de la figura 46 muestra la curva característica de la sensibilidad sensor, con un R_o ante 300ppm de etanol.

Figura 47. Curvas de caracterización de sensibilidad del TGS 825.

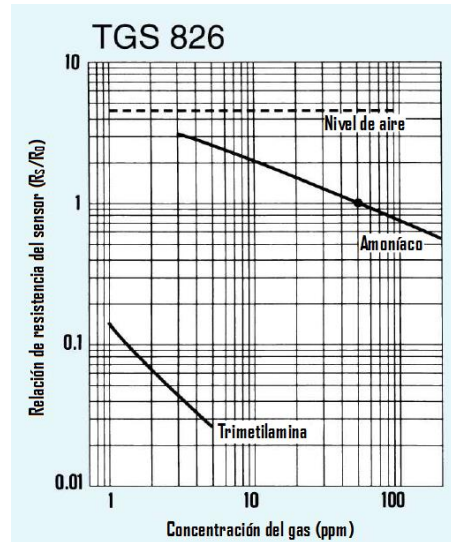


Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 825: El TGS 825 es un sensor con alta sensibilidad al sulfuro de hidrógeno, ya que puede detectar concentraciones de este volátil bajas como 5 ppm,

lo cual lo hace ideal en aplicación como la detección de fugas de gas. En la figura 47 se observa la gráfica de sensibilidad con un R_o ante 50 ppm de sulfuro de hidrógeno.

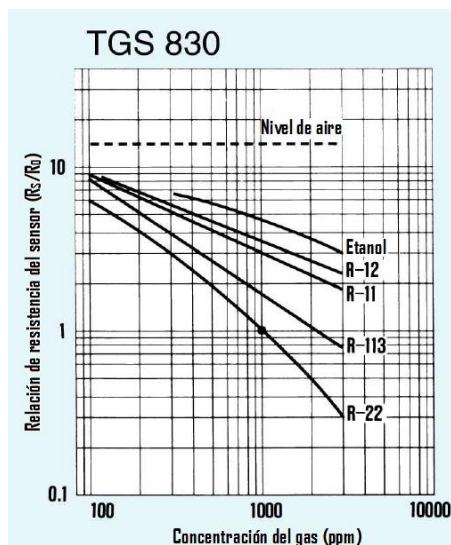
Figura 48. Curvas de caracterización de sensibilidad del TGS 826.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 826: El TGS 826 es un sensor con alta sensibilidad al gas amoníaco, ya que puede detectar concentraciones de este volátil bajas como 30 ppm, lo cual lo hace ideal en aplicaciones de seguridad críticas como la detección de fugas de amoníaco en sistemas de refrigeración y en el campo agrícola. En la figura 48 se observa la gráfica de sensibilidad con un R_o ante 50 ppm de amoníaco.

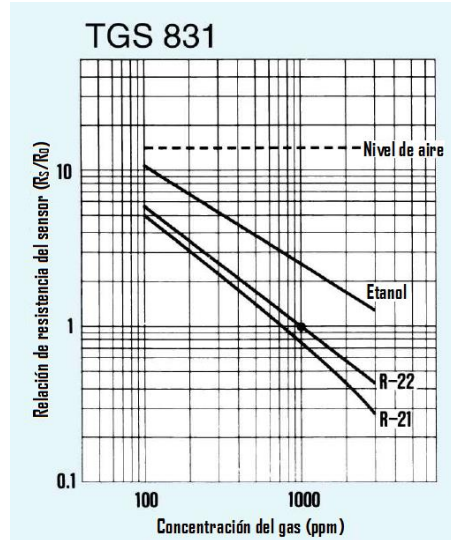
Figura 49. Curvas de caracterización de sensibilidad del TGS 830.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

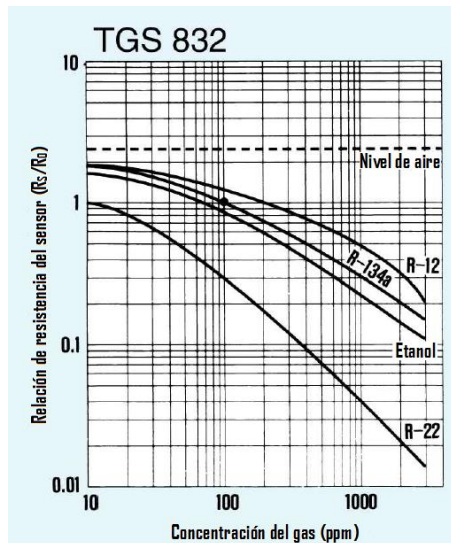
Sensor TGS 830: El TGS 830 es un sensor con alta sensibilidad al clorofluorocarbono, más específicamente al R-22, así como al R-11, R-12 y R-113, presenta baja sensibilidad al hidrógeno y vapores de alcohol lo que le permite una buena selectividad, y a su vez lo hace ideal en aplicaciones como la detección de fugas de refrigerantes. En la figura 49 se observa la gráfica de sensibilidad con un R_o ante 1000 ppm de R-22.

Figura 50. Curvas de caracterización de sensibilidad del TGS 831.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Figura 51. Curvas de caracterización de sensibilidad del TGS 832.

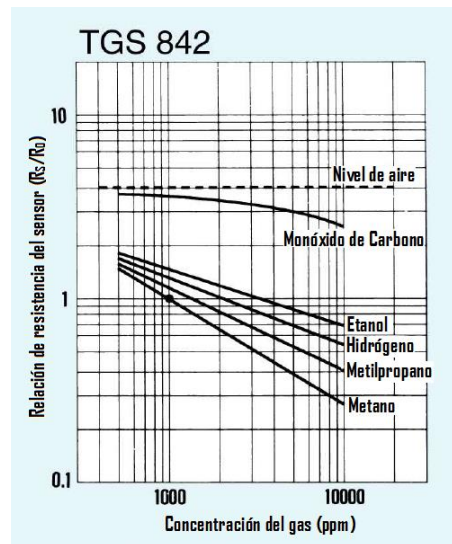


Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 831: El TGS 831 es un sensor con alta sensibilidad al clorofluorocarbono, más específicamente al R-21 y R-22, presenta una buena reproducibilidad en la medición y estabilidad, lo hace ideal en aplicaciones como la detección de fugas de refrigerantes. En la figura 50 se observa la gráfica de sensibilidad con un R_o ante 1000 ppm de R-22.

Sensor TGS 832: El TGS 832 es sensible al refrigerante R134a, siendo utilizado comúnmente para detección de fugas en sistemas de aire acondicionado y refrigeradores, adicionalmente el 832 puede detectar el R12, R22 y etanol. En la figura 51 se observa la gráfica de sensibilidad con un R_o ante 100 ppm de R-134a.

Figura 52. Curvas de caracterización de sensibilidad del TGS 842.

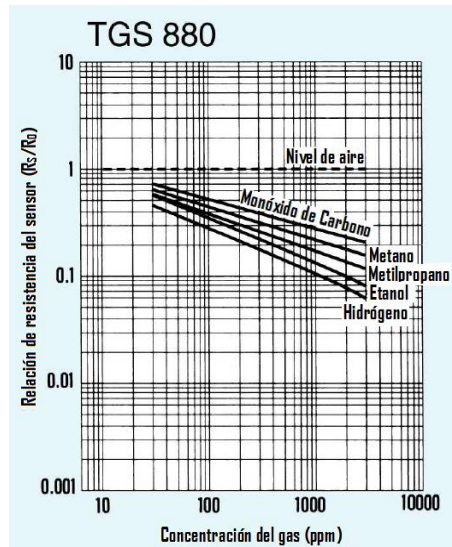


Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 842: El TGS 842 es altamente sensible al gas metano, respondiendo también ante la presencia de vapores de alcohol con alta concentración, por esto es comúnmente utilizado en alarmas domésticas. En la figura 52 se observa la gráfica de sensibilidad con un R_o ante 3500 ppm de metano.

Sensor TGS 880: El TGS 880 está diseñado para la detección de vapores de alimentos (alcoholes). En la figura 53 se observa la curva característica de sensibilidad con un R_o medido bajo condiciones de aire limpio.

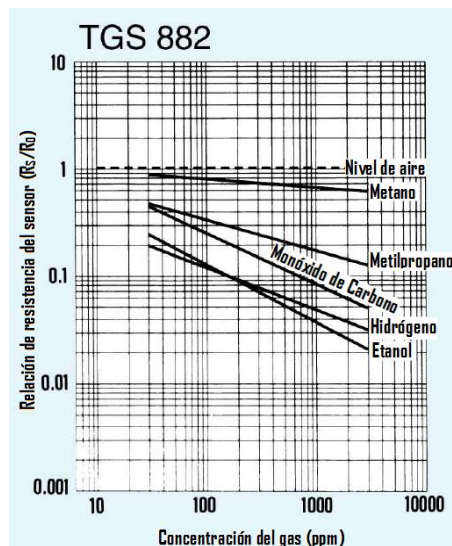
Figura 53. Curvas de caracterización de sensibilidad del TGS 880.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)

Sensor TGS 882: Este sensor es utilizado para la detección de gases de alcohol al igual que el TGS 880 sin embargo la variación de resistencia ante los mismos gases es diferente para cada sensor. En la figura 54 se observa la curva característica de sensibilidad con un R_0 medido bajo condiciones de aire limpio.

Figura 54. Curvas de caracterización de sensibilidad del TGS 882.



Tomado de (“Figaro Gas Sensors 1-Series 8-Series”, 2013)