

UNIVERSIDAD NACIONAL DE LOJA

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES
NO RENOVABLES**



Ingeniería en Sistemas

TÍTULO

**“PROCESAMIENTO DIGITAL DE IMÁGENES (PDI)
APOYADO EN UN ENTRENAMIENTO NEURO-DIFUSO,
PARA LA INSPECCIÓN DE OBJETOS DEFECTUOSOS”**

**Tesis previa a la Obtención del
título de Ingeniero en Sistemas**

Autor

Víctor Moisés Espinoza Ochoa

Director

Ing. Henry Francisco Cueva Bravo

**Loja - Ecuador
2014**



CERTIFICACIÓN

Sr. Ing. Henry Francisco Cueva Bravo

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA. DIRECTOR DE TRABAJO DE TITULACIÓN

CERTIFICA:

Que el presente Trabajo de Titulación elaborado previo la obtención del Título de Ingeniero en Sistemas, titulado: **“PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) APOYADO EN UN ENTRENAMIENTO NEURO-DIFUSO, PARA LA INSPECCIÓN DE OBJETOS DEFECTUOSOS”** realizado por el egresado **Víctor Moisés Espinoza Ochoa**, cumple con los requisitos establecidos por las normas generales para la graduación en la Universidad Nacional de Loja, tanto en aspectos de forma como de contenido; por lo cual autorizo su presentación para los fines pertinentes.

Loja, marzo de 2014

Ing. Henry Francisco Cueva Bravo

DIRECTOR DE TRABAJO DE TITULACIÓN



AUTORÍA

Yo Víctor Moisés Espinoza Ochoa declaro ser autor del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de titulación en el Repositorio Institucional-Biblioteca Virtual.

Autor: Víctor Moisés Espinoza Ochoa

Firma: Víctor Espinoza Ochoa

Cédula: 070452027-9

Fecha: 21 de mayo de 2014



CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo **Víctor Moisés Espinoza Ochoa** declaro ser autor del presente trabajo de titulación titulado: **“PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) APOYADO EN UN ENTRENAMIENTO NEURO-DIFUSO, PARA LA INSPECCIÓN DE OBJETOS DEFECTUOSOS”**, como requisito para optar al grado de Ingeniero de Sistemas; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 21 días de mes de mayo del dos mil catorce, firma el autor.

Firma: *Víctor Moisés Espinoza Ochoa*

Autor: Víctor Moisés Espinoza Ochoa

Cédula: 070452027-9

Dirección: Sucre 18-45 y Rocafuerte Correo Electrónico: soldier_ec@hotmail.com

Teléfono: 072579985

Celular: 0992949893

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Henry Francisco Cueva Bravo

Tribunal de Grado: Ing. Marco Augusto Ocampo Carpio, Mg. Sc.

Ing. Henry Patricio Paz Arias, Mg. Sc.

Ing. Luis Antonio Chamba Eras, Mg. Sc.



AGRADECIMIENTO

Con este proyecto termina una etapa de esfuerzo y trabajo, en la que ha habido buenos y malos momentos, pero que pese a esos malos momentos ha valido la pena vivir, y de la que sólo recordaré los buenos.

Quiero mostrar mi agradecimiento a todos los que han compartido conmigo esta etapa, a todos ellos que me han ayudado a superar todas las dificultades, que me han dado su apoyo y han tenido tanta paciencia.

A Jesucristo, mi Señor y Dios, por enseñarme el camino correcto de la vida, guiándome y fortaleciendo cada día con su Santo Espíritu.

A la Virgen María, por amarme y protegerme cada día y me permite estar rodeado de tantas personas maravillosas.

En especial quiero dar las gracias a mis padres que me han dado la oportunidad de existir, continúan rezando por mí y me inspiraron a seguir adelante para culminar mi meta. Al resto de mi familia, a mis abuelos, tíos (as), primos, etc. que siempre me han dado su apoyo.

A mis amigas y amigos, que siempre me apoyan y me dan ánimo cuando más lo necesito, además de sus enseñanzas de vida que me fortalecen como persona.

No puedo olvidarme de mis compañeros de universidad, con los que he pasado tantos ratos dentro y fuera de las aulas.

También quiero dar las gracias a mi director Ing. Henry Cueva Bravo, y al resto de docentes que me han proporcionado los conocimientos necesarios para mi formación.

A todos, GRACIAS.



DEDICATORIA

A Dios

Que permitió que este momento llegue
Justo en el tiempo y espacio que tenía que suceder.
Porque las cosas siempre pasan por algo
Y de esos momentos estamos hechos

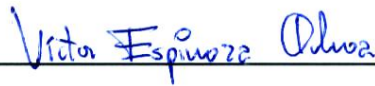
A mi padre; que ha sido mi ejemplo
y mi fortaleza en momentos difíciles
A mi madre; por siempre brindarme
su cariño y afecto cuando
más lo he necesitado
A mi familia; por su apoyo
incondicional

Víctor Espinoza Ochoa



CESIÓN DE DERECHOS

Víctor Moisés Espinoza Ochoa, autor intelectual de la presente Trabajo de Titulación, autorizo a la Universidad Nacional de Loja, al Área de Energía, las Industrias y los Recursos Naturales No Renovables y por ende a la carrera de Ingeniería en Sistemas hacer uso de la misma, con la finalidad que estime conveniente.



Víctor Moisés Espinoza Ochoa



A. TÍTULO

**“PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) APOYADO EN UN
ENTRENAMIENTO NEURO-DIFUSO, PARA LA INSPECCIÓN DE OBJETOS
DEFECTUOSOS”**



B. RESUMEN

El presente Trabajo de Titulación tiene como objeto aplicar y demostrar, que mediante el uso de teorías de control inteligentes, se logran controlar procesos industriales, cuando estos son altamente no lineales o cuando no se dispone de un modelo matemático adecuado por ser demasiado complejo o tienen una dinámica compleja.

Para ello, con el ánimo de favorecer y ampliar el entorno del conocimiento en estudiantes, docentes e investigadores en cuanto a técnicas de control inteligentes para procesos, se implementó y ejecutó el sistema control de inspección y clasificación de objetos defectuosos, específicamente en botellas de vidrio transparentes de 375 cm³, en el Laboratorio de Automatización y Control Industrial del AEIRNNR; basado en conocimiento en tiempo real, a partir de imágenes capturadas mediante el uso de una cámara. Todo lo anterior se consiguió previo el estudio de los procesos en una planta embotelladora de la región, en donde interviene el control de calidad del producto final.

El Trabajo de Titulación se centra en el área de procesamiento digital de imágenes (PDI) para dar lugar al desarrollo de un sistema de control por medio de la creación de un algoritmo, que permita la inspección y clasificación de objetos en imágenes como las botellas, esto se logró a través de un controlador Neuro-difuso debidamente entrenado y validado, mientras que la monitorización de los resultados se la realiza mediante la interfaz de usuario de forma visual y sonora.

Los tipos de investigación son: descriptiva y experimental, descriptiva porque se explica cómo se puede cuantificar características de las imágenes, con la ayuda de técnicas de procesamiento; se habla también que es de tipo experimental, porque se establece la manipulación de variables experimentales, en condiciones controladas, para describir una situación o acontecimiento particular. A su vez, según la fuente, se considera un estudio documental, puesto que se realizó sobre la base de documentos y revisión bibliográfica.

Para el desarrollo del sistema se utiliza conjuntamente las metodologías IDEAL y CommonKADS-RT, metodologías que son aplicadas en el campo de sistemas expertos basados en conocimiento.



SUMMARY

This research degree work has the objective to apply and to demonstrate that using intelligent control theories, it is possible to control the industrial process, when they are highly nonlinear, when there is not an appropriate mathematical model or there is a complex dynamic.

With the purpose of helping and expand the knowledge on students, educators and researchers in terms of intelligent control techniques for processes, it was implemented and applied a control system to inspect and classify flawed items, focus on 375 cc glass bottles, at the Automation and Industrial Control Laboratory located at AEIRNNR, this study was done in a real time, by capturing imagines with a camera. All mentioned above was done and achieved with a previous study on a bottling manufacture company, in this region, where the quality assurance is important in the final product.

This degree work is focused on the digital imagine processing to develop a control system using an algorithm that allows the inspection and classification of the faulty items by imagining like the bottles, this is being achieved by a Neuro-fuzzy controller, correctly applied and verified, while the results were monitored in a visual and auditory way.

The types of research used in this project are: descriptive and experimental, descriptive because it explains how the characteristics of the imagines are quantified, applying processing techniques. Also, it is experimental because the experimental variables are manipulated below controllable situations to describe particular situations. Besides, it is a documentary study because it was made from available data and bibliography information.

Finally, to develop the system it was used the IDEAL and CommonKADS-RT, methodologies which are applied in the field of experienced systems, based on knowledge.



ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	II
AUTORÍA.....	III
CARTA DE AUTORIZACIÓN.....	IV
AGRADECIMIENTO.....	V
DEDICATORIA.....	VI
CESIÓN DE DERECHOS.....	VII
A. TÍTULO.....	VIII
B. RESUMEN.....	IX
SUMMARY.....	X
ÍNDICE DE CONTENIDOS.....	XI
ÍNDICE DE FIGURAS.....	XIX
ÍNDICE DE TABLAS.....	XXVIII
C. INTRODUCCIÓN.....	1
D. REVISIÓN DE LITERATURA.....	4
CAPÍTULO I: TEORÍAS DE CONTROL.....	4
1.1. Aspectos generales.....	4
1.1.1. Sistema.....	4
1.1.1.1. Definición de Sistema de Control.....	5
1.1.1.2. Breve Historia.....	5
1.2. Control Clásico/Moderno e Inteligente.....	6
1.2.1. Control Inteligente.....	7
1.2.1.1. Control difuso. ("Fuzzy Control").....	9
1.2.1.2. Control Neuronal.....	11
1.2.1.3. Control Neurodifuso.....	13

CAPÍTULO II: SISTEMAS NEURO-DIFUSOS.....	15
2.1. Antecedentes.....	15
2.2. Introducción a las Redes Neuronales	15
2.2.1. La Neurona Biológica.....	16
2.2.2. Componentes de una Red Neuronal Artificial.....	16
2.2.2.1. El Proceso de Aprendizaje.....	18
2.3. Introducción a los Sistemas Difusos	18
2.3.1. Conceptos básicos.....	19
2.3.2. Función de membrecía.....	20
2.3.2.1. Funciones de Pertenencia Típicas.....	21
2.3.3. Razonamiento Difuso e Inferencia.....	25
2.3.4. Decodificando la salida: Defusificación	25
2.4. Introducción a los Sistemas Neuro-difusos	27
2.4.1. ANFIS	28
2.4.1.1. Estructuras de las ANFIS.....	29
CAPÍTULO III: VISIÓN ARTIFICIAL Y PROCESAMIENTO DIGITAL DE IMÁGENES	32
3.1. Visión Artificial	32
3.2. Procesamiento Digital de Imágenes.....	34
3.2.1. Aspectos Generales	34
3.2.1.1. Píxel	34
3.2.1.2. Imagen digital	35
3.2.1.3. Píxeles en referencia al Sistema Internacional de Medidas	36
3.2.1.4. Digitalización de Imágenes	38
3.2.1.5. Formatos numéricos	38
3.2.1.6. Clasificación de imágenes digitales	39
3.2.2. Espacios de Color	40



3.2.2.1.	Modelo RGB.....	41
3.2.3.	Detección de regiones basada en umbrales	41
3.2.3.1.	Método de segmentación de Otsu	42
3.2.4.	Detección de bordes	43
3.2.4.1.	Operador de canny.....	43
3.2.4.2.	Operadores de Prewitt, Sobel y Frei-Chen	44
3.2.5.	Operaciones morfológicas básicas.....	45
3.2.6.	Propiedades geométricas de los objetos.....	48
CAPÍTULO IV: METODOLOGÍAS PARA LA CONSTRUCCIÓN DE SISTEMAS BASADOS EN CONOCIMIENTO.....		49
4.1.	Antecedentes	49
4.2.	Metodología IDEAL	53
4.3.	Metodología CommonKADS.....	61
4.3.1.	CommonKADS-RT.....	63
CAPÍTULO V: HERRAMIENTAS DE PROGRAMACIÓN Y DESARROLLO		68
5.1.	MatLab	68
5.1.1.	Como hacer un programa en MATLAB	69
5.1.1.1.	El fichero *.m.....	70
5.1.1.2.	Las funciones	70
5.1.1.3.	Scripts	71
5.1.2.	ToolBoxes de MatLab	72
5.1.2.1	Image Processing Toolbox	73
5.1.2.2.	Fuzzy Logic Toolbox.....	74
5.1.3.	NetBeans - Java	76
5.1.4.	Herramienta visual para modelado UML - Enterprise	78
5.1.5	Frameworks y Librerías.....	80
5.1.5.1.	JMatLink.....	80

5.1.5.2.	Java Media Framework (JMF)	80
5.1.5.3.	Pentaho Reporting (JFreeReport).....	81
5.1.5.4.	JFreeChar.....	82
5.1.5.5.	Hibernate.....	83
5.1.5.5.1.	Mapas de objetos relacionales en ficheros XML	85
5.1.5.5.1.1.	Estructura del fichero XML	85
5.1.5.5.2.	Mapeando una colección	90
5.1.5.5.3.	Componentes.....	91
5.1.5.5.4.	Creación de Ficheros XML.....	92
5.1.5.5.5.	Ejemplo: mapeos	92
	Empleador/Empleado.....	92
5.1.6.	WebCam	94
E.	MATERIALES Y MÉTODOS.....	97
F.	RESULTADOS	106
1.	Modelo de la organización.....	106
1.1.	Condiciones iniciales	106
1.2.	Especificaciones del proyecto.....	107
2.	FASE 1: IDENTIFICACIÓN DE LA TAREA	109
	Etapas 1.1: Plan de requisitos y adquisición de conocimientos.....	109
1.1.1.	Proceso de Adquisición de Conocimientos.....	109
1.1.1.1.	Primeras reuniones y evaluación de viabilidad:	109
1.1.1.2.	Extracción de conocimientos:	110
1.1.1.3.	Educción de conocimientos con expertos:	110
1.1.1.4.	Educción de conocimientos:	110
	Etapas 1.2: Evaluación y selección de la tarea	110
1.2.1	Test de Viabilidad.....	110
	Etapas 1.3: Definición de las características del sistema	113



1.3.1	Requerimientos Funcionales	113
1.3.2.	Requerimientos No Funcionales.....	114
1.3.3	Casos de prueba o ensayo.....	114
1.3.4.	Recursos necesarios.....	115
1.3.4.1.	Recursos Humanos.....	115
1.3.4.2.	Recursos Técnicos y Tecnológicos	115
1.3.4.3.	Recursos Económicos.....	115
1.3.5.	Análisis de costo / beneficio	115
1.3.5.1.	Costo del Proyecto	115
1.3.5.1.1.	Costos tangibles	115
1.3.5.1.2.	Costos intangibles.....	116
1.3.5.2.	Beneficios del Proyecto	116
1.3.5.2.1.	Beneficios tangibles.....	116
1.3.5.2.2.	Beneficios intangibles.....	117
1.3.6.	Calendario o Cronograma de actividades.	117
3.	FASE 2: DESARROLLO DE LOS PROTOTIPOS	117
	Etapas 2.1: Concepción de la solución	117
2.1.1.	Modelado de Caso de Uso.....	117
	Etapas 2.2: Adquisición y Conceptualización de los conocimientos.....	119
2.2.1.	Diccionario de Términos	119
2.2.2.	Identificación de las relaciones de la aplicación	123
2.2.2.1.	Modelado Dinámico de la Aplicación	123
2.2.2.2.	Diagrama de Componentes.....	154
2.2.2.3.	Diagrama de paquetes	154
2.2.2.4.	Diagrama de Despliegue	155
2.2.2.5.	Modelo de tareas de alto nivel (TAN).....	155
2.2.2.6.	Modelo de tareas en tiempo real	157

2.2.2.7.	Modelo del conocimiento	159
2.2.2.7.1.	Factores o variables del proceso de control del micro mundo	159
2.2.2.7.2.	Diseño y Construcción del Banco de Pruebas.....	160
2.2.2.8.	Modelo de diseño	163
2.2.2.8.1.	Toma de muestras fotografías.....	163
2.2.2.8.2.	Desarrollo del algoritmo en tiempo diferido	163
2.2.3.	Identificación de los conocimientos tácticos.....	190
2.2.3.1.	Análisis de conocimientos tácticos.....	190
2.2.3.2.	Comprobación de los conocimientos tácticos	194
Etapa 2.3 Formalización de los conocimientos y definición de la arquitectura		195
2.3.1.	Motor de inferencia y base de conocimientos	195
2.3.2.	Codificación de Reglas y desarrollo del controlador difuso	196
2.3.2.1.	Preparación de Datos con la ayuda del software MatLab	196
2.3.2.2.	Construcción del sistema con Fuzzy Logic Toolbox: Aprendizaje e inferencia través de ANFIS.....	199
2.3.2.2.1.	Entrenando Anfis	208
2.3.2.2.2.	Evaluación del FIS.....	210
Etapa 2.4 Selección de la Herramienta e implementación.....		211
Etapa 2.5: Validación y evaluación del prototipo.....		211
4.	FASE 3: EJECUCIÓN DE LA CONSTRUCCIÓN DEL SISTEMA INTEGRADO.....	212
Etapa 3.1: Requisitos y diseño de la integración con otros sistemas		212
3.1.1.	Instalación de JAVA (jdk1.6.0_03 y jre1.6.0_03)	213
3.1.2.	Instalación de Java Media Framework (JMF)	216
3.1.2.1.	Configuración de JMF.....	218
3.1.3.	Instalación de MATLAB R2009A	222
3.1.4.	Instalación y Configuración de Gestor de Base de Datos: My SQL	
5.0.....		230



3.1.5.	Instalación de SQLyog 4.0	235
3.1.5.1.	Configurar SQLYOG para conectar a MYSQL.....	237
3.1.6.	Conexión de MATLAB con la Interfaz Gráfica del Sistema realizada en JAVA.....	240
3.1.7.	Instalación de ICOLab v1.0	241
3.1.8.	Verificar la variables de entorno PATH.....	244
	Etapa 3.2: Implementación y evaluación del Sistema Integrado.....	245
3.2.1.	Prueba de Usabilidad (Funcionalidad, Diseño y Presentación)	245
3.2.2.	Funcionalidad del sistema ICOLab: Resultados obtenidos en tiempo real.....	249
	Etapa 3.3: Aceptación del sistema por el usuario	251
	G. DISCUSIÓN	259
1.	DESARROLLO DE LA PROPUESTA ALTERNATIVA.....	259
2.	VALORACIÓN TÉCNICA – ECONÓMICA – AMBIENTAL	263
2.1.	Valoración Técnica Económica.....	263
2.2.	Valoración Ambiental.....	266
	H. CONCLUSIONES.....	267
	I. RECOMENDACIONES	268
	J. BIBLIOGRAFÍA	269
	K. ANEXOS	277
	Anexo 1: Observación sistemática en la embotelladora	278
	Anexo 2: Entrevista aplicada al Experto Operador de la planta Embotelladora	281
	Anexo 3: Entrevista aplicada al Experto en Control Automático	283
	Anexo 4: Entrevista aplicada al Experto en Procesamiento Digital de Imágenes... ..	285
	Anexo 5: Conjunto de Datos para el entrenamiento del FIS	287
	Anexo 6: Conjunto de Datos para la validación del FIS.....	291
	Anexo 7: Tablas de la Base de Datos del Sistema	294

Anexo 8: Solicitud para realizar pruebas de validación y encuestas	295
Anexo 9: Autorización para realizar pruebas de validación y encuestas	296
Anexo 10: Modelo de encuestas aplicadas a los administradores (Docentes)	297
Anexo 11: Modelo de encuestas aplicadas a los usuarios del sistema (Estudiantes)	299
Anexo 12: Certificación de Traducción del Resumen	303
Anexo 13: Cronograma de Actividades	304
Anexo 14: Licencia Creative Commons del Proyecto	307
Anexo 15: Anteproyecto de Tesis	308

ÍNDICE DE FIGURAS

Figura 1: Sistema limitado por una frontera real o imaginaria.	4
Figura 2: Representación a bloques de un sistema.	7
Figura 3: Diagrama de un sistema dinámico.	7
Figura 4: Esquema de un controlador difuso insertado en el esquema general de un control retroalimentado	9
Figura 5: Esquema de una neurona artificial.	11
Figura 6: Una capa de neuronas.	12
Figura 7: Funciones de activación comúnmente usadas.	12
Figura 8: Neurona Biológica.	16
Figura 9: Neurona Artificial	17
Figura 10: Función Singleton	21
Figura 11: Función Triangular	21
Figura 12: Función G	22
Figura 13: Aproximación lineal de la Función G	22
Figura 14: Función S	23
Figura 15: Función Gaussiana	23
Figura 16: Función Trapezoidal	24
Figura 17: Función Pseudo-Exponencial	24
Figura 18: Método Principio de la membresía Max	26
Figura 19: Método de defusificación Centroid.	26
Figura 20: Método de defusificación Weighted Average.	27
Figura 21: Arquitectura ANFIS	29
Figura 22: Ejemplos de aplicación de visión artificial.	33
Figura 23: Representación de un píxel	34
Figura 24: a) Imagen original; b) Estructura matricial de la imagen.	35
Figura 25: Imagen binaria.	39
Figura 26: Imagen en escalas de grises.	40
Figura 27: Imagen en color.	40
Figura 28: Distribución de colores en el cubo RGB	41
Figura 29: Imagen en niveles de gris	42
Figura 30: Histograma	42
Figura 31: A: umbral óptimo 97, B: umbral = 45, C: umbral = 140.	42

Figura 32: Imagen original e Imagen con detección de Canny	44
Figura 33: Máscaras de Convolución	44
Figura 34: A) Imagen Original, B) Filtro Sobel, C) Filtro Canny.	45
Figura 35: Elementos Estructurales A y B.....	45
Figura 36: Elemento Estructural N4	46
Figura 37: Elemento Estructural N8	46
Figura 38: Dilatación: A) imagen original, B) elemento estructurante N8, C imagen dilatada	46
Figura 39: Erosión con un elemento estructurante de 1x2	46
Figura 40: Erosión: A) imagen original, B) elemento estructurante de 3x3, C) imagen erosionada.....	47
Figura 41: Apertura: A) imagen original, B) imagen resultante después de aplicar la apertura.	47
Figura 42: Cerradura: A) imagen en niveles de gris, B) imagen en blanco y negro	48
Figura 43: Fases de la metodología IDEAL.....	53
Figura 44: Modelos definidos por CommonKADS	63
Figura 45: Modelos de CommonKADS-RT.....	64
Figura 46: Contenido del Modelo de Tareas de Tiempo Real	65
Figura 47: Estructura de la tarea de tiempo real a bajo nivel.....	67
Figura 48: Tipo de ficheros	70
Figura 49: Sintaxis de una función en MatLab.	70
Figura 50: Ejemplo de una Función en MatLab.....	71
Figura 51: Ejemplo de un script en MatLab con nombre ejem_script.m	71
Figura 52: Sistema de Inferencia Fuzzy	75
Figura 53: Previsualización de reporte con JFreeReport.....	81
Figura 54: Ejemplo de gráficos de pastel y barras con JFreeChart.	83
Figura 55: Arquitectura base	84
Figura 56: Fichero XML.....	85
Figura 57: Clase persistente	90
Figura 58: Clase Persona y su composición	91
Figura 59: Mapeo de la clase Persona.....	91
Figura 60: Ejemplo de Diseño Relacional	92
Figura 61: Ejemplo modelo de la relación entre Employer y Employee	93
Figura 62: Mapeo en XML.....	93



Figura 63: Esquema de tablas del modelo	94
Figura 64: Webcam Genius iSlim 1320.....	95
Figura 65: Escenario de proceso de envasado	106
Figura 66: Banco de Pruebas – Lado lateral Izquierdo	107
Figura 67: Banco de Pruebas – Lado lateral Derecho.....	107
Figura 68: Proceso que debe realizar el sistema controlador para cumplir con el objetivo final.....	108
Figura 69: Primera parte del Algoritmo: Procesamiento Digital de Imágenes.....	109
Figura 70: Segunda parte del Algoritmo: Evaluación del FIS (Controlador Difuso)....	109
Figura 71: Catálogo Actores	117
Figura 72: Diagrama de Casos de Uso	118
Figura 73: Modelo del Dominio	118
Figura 74: Modelo de la Base de Datos	119
Figura 75: Vista Acceso	123
Figura 76: Vista Ventana de trabajo - Sistema de control de Inspección y Clasificación	123
Figura 77: Vista Administración	123
Figura 78: Vista Administrar Personal.....	124
Figura 79: Vista Datos Personal	124
Figura 80: Vista Mensaje Datos Guardados Correctamente	124
Figura 81: Vista Mensaje de Campos Vacíos	124
Figura 82: Vista Mensaje de Error Campos Inválidos	125
Figura 83: Vista Mensaje de confirmación de eliminación.....	125
Figura 84: Vista Mensaje de no selección de Persona.....	125
Figura 85: DR_CU001 - Administrar Personal	128
Figura 86: DR_CU001 – Curso Alternativo A: No existe selección de persona.....	129
Figura 87: DR_CU001 – Curso Alternativo B: Campos obligatorios vacíos	129
Figura 88: DR_CU001 – Curso Alternativo D: Nueva Persona	129
Figura 89: DR_CU001 – Curso Alternativo D: Eliminar Persona	130
Figura 90: DS_CU001 - Administrar Personal.....	130
Figura 91: DS_CU001 – Curso Alternativo A: No existe selección de persona.....	131
Figura 92: DS_CU001 – Curso Alternativo B: Campos obligatorios vacíos	131
Figura 93: DS_CU001 – Curso Alternativo D: Nueva Persona	131
Figura 94: DR_CU001 – Curso Alternativo D: Eliminar Persona	132

Figura 95: Vista Administrar Usuario.....	132
Figura 96: Vista Mensaje Selección de registro de persona.....	132
Figura 97: Vista Ingreso Datos de Usuario.....	133
Figura 98: Vista Mensaje Datos Guardados Correctamente	133
Figura 99: Vista Mensaje de Campos Vacíos.....	133
Figura 100: Vista Mensaje de inhabilitar usuario.....	133
Figura 101: DR_CU002 - Administrar Usuario	136
Figura 102: DR_CU002 – Curso Alterno A: No existe selección de usuario	137
Figura 103: DR_CU002 – Curso Alterno B: Campos obligatorios vacíos	137
Figura 104: DR_CU002 – Curso Alterno C: Inhabilitar acceso de usuario.....	137
Figura 105: DS_CU002 - Administrar Usuario.....	138
Figura 106: DS_CU002 – Curso Alterno A: No existe selección de usuario	138
Figura 107: DR_CU002 – Curso Alterno B: Campos obligatorios vacíos	139
Figura 108: DS_CU002 – Curso Alterno C: Inhabilitar acceso de usuario.....	139
Figura 109: Vista Reporte Estadístico de Resultados	140
Figura 110: Vista Previa de Reporte	140
Figura 111: Vista Mensaje Fecha no válida.....	141
Figura 112: Vista Mensaje No hay Búsqueda	141
Figura 113: Vista Opciones de Impresión	141
Figura 114: DR_C003 – Consultar Reportes.....	144
Figura 115: DR_CU003 – Curso Alterno A: Fechas no válidas	144
Figura 116: DR_CU003 – Curso Alterno B: No existe búsqueda.....	145
Figura 117: DR_CU003 – Curso Alterno C: Disponibilidad de Impresora.....	145
Figura 118: DS_CU003 – Consultar Reporte	145
Figura 119: DS_CU003 – Curso Alterno A: Fechas no validas.....	146
Figura 120: DS_CU003 – Curso Alterno B: No hay Búsqueda	146
Figura 121: DS_CU003 – Curso Alterno C: Disponibilidad de Impresora	146
Figura 122: Vista Mensaje de Conexión con MatLab	147
Figura 123: Vista Control de Inspección y Clasificación	147
Figura 124: Vista Selección de nuevo directorio	147
Figura 125: Vista Mensaje Fallo de Conexión MatLab	148
Figura 126: Vista Mensaje Cámara no disponible	148
Figura 127: Opción Abrir Directorio	148
Figura 128: DR_CU004 – Operar Inspección.....	151



Figura 129: DR_CU004 – Curso Alterno A: Cámara no Disponible.....	151
Figura 130: DR_CU004 – Curso Alterno B: Conexión MatLab no Disponible.....	152
Figura 131: DR_CU004 – Curso Alterno C: Abrir Directorio.....	152
Figura 132: DS_CU004 – Operar Inspección.....	152
Figura 133: DS_CU004 – Cursos Alterno A: Cámara no disponible.....	153
Figura 134: DS_CU004 – Cursos Alterno B: Conexión MatLab no disponible.....	153
Figura 135: DS_CU004 – Cursos Alterno C: Abrir Carpeta.....	153
Figura 136: Diagrama de Componentes	154
Figura 137: Diagrama de Paquetes	154
Figura 138: Diagrama de Despliegue.....	155
Figura 139: Esquema general del comportamiento de los procesos en el sistema ...	155
Figura 140: Detalle de las actividades de la TAN 2 Inspeccionar Objeto	157
Figura 141: Tareas de alto nivel TAN.....	157
Figura 142: Diagrama de esquema de dominio para el sistema de inspección en el micro mundo.....	162
Figura 143: Diagrama de transición de estados del planificador	162
Figura 144: Imagen Original 640 x 480 (1,3M píxeles).....	166
Figura 145: Imagen Recortada	166
Figura 146: Imagen en Gris	167
Figura 147: Imagen Filtrada.....	167
Figura 148: Imágenes en gris extraídas del plano R, G y B respectivamente.	168
Figura 149: Imágenes Lógicas.....	169
Figura 150: Imagen Resultante.....	169
Figura 151: Imagen Diferencia de RGB.	170
Figura 152: Imagen Complemento entre Planos.....	171
Figura 153: Imagen Resultante después de la restricción.....	172
Figura 154: Imagen con Fallas de RGB	172
Figura 155: Bordes Detectados por operadores Canny y Sobel.....	173
Figura 156: Imagen con operaciones morfológicas.....	174
Figura 157: Imagen Diferencia RGB con defectos	175
Figura 158: Abertura y cerramiento por reconstrucción.....	175
Figura 159: Imagen binaria en bordes.	176
Figura 160: Imagen segmentada por morfología.....	177
Figura 161: Imagen final con fallas de bordes 1.....	177

Figura 162: Imagen Recortada.....	178
Figura 163: Imagen Segmentada por bordes 2	178
Figura 164: Imagen dilatada con bordes 2	179
Figura 165: Imagen rellenada	179
Figura 166: Imagen erosionada con bordes 2	180
Figura 167: Imagen final con fallas de bordes 2	181
Figura 168: Imagen sumada por bordes	182
Figura 169: Imagen final binaria.....	183
Figura 170: Imagen Original Inspeccionada.....	183
Figura 171: Command Window de MatLab	200
Figura 172: Ventana de Anfis Editor	200
Figura 173: Datos cargados en el Workspace de MatLab	201
Figura 174: Cargar datos de entrenamiento en Anfis	201
Figura 175: Datos de entrenamiento cargados (conjunto de círculos).....	202
Figura 176: Datos de validación cargados (Conjunto de +'s)	202
Figura 177: Ingreso y tipo de funciones de pertenencia.	203
Figura 178: Definición de las variables en la ventana FIS Editor.....	204
Figura 179: Editor de Funciones de Pertenencia, Membership Editor	205
Figura 180: Ventana Editor de Reglas, Rule Editor	205
Figura 181: Ventana Rule Viewer	206
Figura 182: Cargar el FIS.....	207
Figura 183: Estructura FIS	207
Figura 184: Entrenando Anfis	208
Figura 185: Prueba de datos de validación	209
Figura 186: Exportar el Controlador FIS.....	210
Figura 187: Archivo de instalación de Java 6.....	213
Figura 188: Licencia de Java SE Development Kit 6.....	213
Figura 189: Configuración personalizada del JDK de Java	213
Figura 190: Proceso de Instalación de JDK de Java	214
Figura 191: Configuración personalizada del JRE de Java	214
Figura 192: Proceso de Instalación de JRE de Java	214
Figura 193: Instalación completa de Java	215
Figura 194: Consola del sistema.....	215
Figura 195: Archivo de Instalación de Java Media Framework.....	216



Figura 196: Preparando proceso de Instalación de JMF	216
Figura 197: Contrato de Licencia JMF	216
Figura 198: Destino de Instalación.....	217
Figura 199: Selección de opciones de Instalación de JMF.....	217
Figura 200: Archivos Instalados.....	218
Figura 201: Instalación Finalizada	218
Figura 202: Ícono de JMF instalado.....	218
Figura 203: Ventana de Java Media Studio	219
Figura 204: Opción de Preferencia	219
Figura 205: Editor de Registro de JMF	219
Figura 206: Buscando Dispositivos de Captura	220
Figura 207: Dispositivos conectados.....	220
Figura 208: Dispositivos eliminados.....	220
Figura 209: Opción Captura.....	221
Figura 210: Seleccionar el Dispositivo de Captura.....	221
Figura 211: Imagen de video de la cámara seleccionada	221
Figura 212: Salir de la configuración de JMF	222
Figura 213: Configuración de Microsoft Visual C++ 2005 Redistributable.....	222
Figura 214: Opciones de Instalación de MatLab	223
Figura 215: Acuerdo de Licencia	223
Figura 216: Clave de Instalación.....	224
Figura 217: Tipo de Instalación.....	224
Figura 218: Carpeta de Instalación	225
Figura 219: Selección de Producto	226
Figura 220: Opciones de Instalación.....	226
Figura 221: Confirmación de Elecciones.....	227
Figura 222: Proceso de Instalación.....	227
Figura 223: Instalación completa	228
Figura 224: Activación de Matlab.....	228
Figura 225: Activación sin Internet.....	229
Figura 226: Selección del archivo de licencia	229
Figura 227: Archivo Seleccionado	230
Figura 228: Activación completa de MatLab	230
Figura 229: Archivo de instalación de MySQL	230

Figura 230: Asistente de configuración de MySQL 5.0.....	231
Figura 231: Tipo de Instalación	231
Figura 232: Configuración actual a instalar	231
Figura 233: Instalación en proceso	232
Figura 234: Registro de Cuenta de MySQL.....	232
Figura 235: Instalación completa	232
Figura 236: Configuración de la instancia del servidor de MySql	233
Figura 237: Tipo de configuración de la instancia de servidor.....	233
Figura 238: Opciones de servicio en Windows.....	233
Figura 239: Opciones de seguridad	234
Figura 240: Ejecutar las opciones de Instancia	234
Figura 241: Configuración del servicio de MySQL Finalizada	234
Figura 242: Archivo de Instalación de SQLyog	235
Figura 243: Asistente de configuración de SQLyog 4.0.....	235
Figura 244: Términos de Licencia de SQLyog 4.0.....	235
Figura 245: Directorio de Instalación.....	236
Figura 246: Proceso de Instalación	236
Figura 247: Instalación completa de SQLyog 4.0	236
Figura 248: Entorno de SQLyog.....	237
Figura 249: Nueva Conexión	237
Figura 250: Nombre de nueva conexión	238
Figura 251: Conexión con Host de MySQL	238
Figura 252: Guardar detalles de conexión.....	238
Figura 253: Creación de base de datos	239
Figura 254: Ejecutar sentencia.....	239
Figura 255: Base de Datos Creada.....	239
Figura 256: Salir de SQLyog.....	240
Figura 257: Archivo de comunicación con MATLAB.....	240
Figura 258: Código de conexión JMatLink	240
Figura 259: Instalación del archivo de librería JMatLink.....	241
Figura 260: Archivo de Instalación de ICOLab v1.0	241
Figura 261: Asistente de Instalación ICOLab v1.0.....	241
Figura 262: Carpeta de Instalación de ICOLab v1.0.....	242
Figura 263: Carpeta de Accesos Directos.....	242



Figura 264: Crear ícono	242
Figura 265: Configuraciones realizadas a instalar.....	243
Figura 266: Proceso de Instalación ICOLab v1.0.....	243
Figura 267: Instalación completa de ICOLab v1.0.....	243
Figura 268: Ventana Principal de ICOLab 1.0.....	244
Figura 269: Gráfica Estadística de los resultados de prueba de estudio	249
Figura 270: VALLTO AGROINDUSTRIA CIA. LTDA.....	278
Figura 271: Bandeja de Reciclado	278
Figura 272: Proceso de Lavado / Secado de Botellas.....	279
Figura 273: Proceso Llenado y Tapado de las Botellas	279
Figura 274: Producto Final.....	280
Figura 275: Empacado Final del Producto	280

ÍNDICE DE TABLAS

TABLA I: CUADRO COMPARATIVO ENTRE METODOLOGÍAS PARA SISTEMAS EXPERTOS	50
TABLA II: DESCRIPCIÓN DE LAS FASES DE LA METODOLOGÍA IDEAL	54
TABLA III: LISTA DE TOOLBOXES EN MATLAB	72
TABLA IV: DIAGRAMAS UML	80
TABLA V: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DEL PROBLEMA	111
TABLA VI: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DE PERSONAL	111
TABLA VII: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DE DESPLIEGUE	112
TABLA VIII: EVALUACIÓN DEL PROYECTO.....	112
TABLA IX: REQUERIMIENTOS FUNCIONALES.....	113
TABLA X: REQUERIMIENTOS NO FUNCIONALES.	114
TABLA XI: DICCIONARIO DE TÉRMINOS.....	119
TABLA XII: DESCRIPCIÓN DEL PROCESO EN FUNCIÓN DE LAS TAN QUE LO COMPONENTEN.....	156
TABLA XIII: FUNCIONES DEL ALGORITMO IMPLEMENTADAS EN MATLAB	164
TABLA XIV: DATOS EXTRAÍDOS CON EL ALGORITMO	184
TABLA XV: FUNCIONES UTILIZADAS EN EL ALGORITMO DE INSPECCIÓN	188
TABLA XVI: REGLAS POSIBLES PARA EL SISTEMA DE CONTROL FUZZY	191
TABLA XVII: REGLAS LINGÜÍSTICAS.....	194
TABLA XVIII: REGLAS LINGÜÍSTICAS CON SALIDA FUZZY	194
TABLA XIX: BASE DE CONOCIMIENTOS	195
TABLA XX: PARÁMETROS DEL SISTEMA DE INFERENCIA DIFUSO	195
TABLA XXI: VECTORES PARA EL ENTRENAMIENTO.....	196
TABLA XXII: VECTORES PARA LA VALIDACIÓN.	198
TABLA XXIII: CARACTERÍSTICAS MÍNIMAS DE HARDWARE	212
TABLA XXIV: SOFTWARE INSTALADO	212
TABLA XXV: ESQUEMA DE PRUEBAS.....	245
TABLA XXVI: POBLACIÓN DOCENTES	246
TABLA XXVII: POBLACIÓN ESTUDIANTES AEIRNNR	246



TABLA XXVIII: MUESTRA ESTRATIFICADA PROPORCIONAL DE LOS ESTUDIANTES	248
TABLA XXIX: MUESTRA DE DOCENTES	248
TABLA XXX: RESULTADOS DE LA PRUEBA DE ESTUDIO	249
TABLA XXXI: RECURSOS HUMANOS	264
TABLA XXXII: RECURSOS TÉCNICOS Y TECNOLÓGICOS	265
TABLA XXXIII: RECURSOS MATERIALES	265
TABLA XXXIV: RESUMEN DE RECURSOS	266



C. INTRODUCCIÓN

La Universidad Nacional de Loja dentro de sus objetivos conlleva la formación profesional para buscar soluciones a los problemas de la realidad que vive nuestra sociedad, por ello es indiscutible que la ciencia y la tecnología deben contribuir a elevar la calidad de vida de la población; aumentando la competitividad de la economía y disminuyendo los desequilibrios regionales.

El aumento de las necesidades de las instituciones, empresas e industrias de contar con sistemas de control automáticos, que permitan manejar sus recursos de forma eficaz y en busca de mejoras en sus procesos, han contribuido a la realización y perfeccionamiento de tecnologías para el desarrollo de Sistemas Basados en el Conocimiento (SBC). Estos sistemas tratan con problemas complejos en un ambiente o dominio dado y son utilizados como por ejemplo en problemas de diagnósticos, mantenimiento, monitorización, control de procesos, etc.

En este sentido el Laboratorio de Automatización y Control Industrial del Área de La Energía, Las Industrias y Los Recursos Naturales No Renovables de la U.N.L. brinda soporte a la ejecución de las experiencias didácticas de sistemas reales, que complementan la formación profesional del estudiante y por ello se propone soluciones innovadoras que optimicen el desempeño de sistemas de control y procesos, haciendo uso de tecnologías modernas y con criterio. Por consiguiente permite al estudiante tener un enfoque más amplio del campo profesional, en el cual podría desenvolverse a nivel de la industria, agroindustria, mecánica y otras áreas de automatización de procesos de una manera fácil y sostenible.

Luego de una investigación y análisis minucioso de las nuevas tecnologías, se determinó que en la actualidad los sistemas de control han asumido un papel cada vez más importante en el desarrollo y avance de la civilización moderna; es por ello que cada aspecto de las actividades de nuestra vida está afectado por algún tipo de sistema de control, mismos que se encuentran en la mayoría de los sectores de la industria.

Una de las herramientas con las que se puede realizar un sistema de control, es el procesamiento digital de imágenes, el cual posee un enorme valor científico-técnico, por los innumerables campos en los que puede ser aplicado, tales como: medicina,

astronomía, inspección industrial, biología, inteligencia artificial, seguridad, etc. Las ventajas del procesamiento digital de imágenes radican en la gran cantidad de información que puede ser obtenida, manipulada y procesada.

Otra de las herramientas que trabaja conjuntamente para el control de procesos de sistemas complejos no lineales, son las redes neuronales, la lógica difusa y la combinación de estas, las cuales han sido aceptadas de forma paulatina, debido a que buscan dar respuesta a aquellos problemas donde la toma de decisiones es muy parecida a la forma del razonamiento humano, basando sus fundamentos en la experiencia. Un sistema de inferencia neuro-difuso evita ecuaciones complejas y un tiempo de proceso extenso, características que típicamente son sinónimo de los algoritmos de control tradicionales, la lógica difusa permite ahorrar tiempo de respuesta y abordar procesos no lineales con relativa facilidad.

Con esta perspectiva el presente trabajo investigativo consta de cinco capítulos los mismos que se relatan a continuación.

El trabajo se inicia con la parte introductoria en la que se da a conocer las diferentes fases en la que se ha desarrollado el proyecto, comprende también las bases teóricas e información recogida respecto a las herramientas utilizadas, las mismas que son debidamente sustentadas gracias a la fase de fundamentación teórica realizada.

La metodología utilizada para establecer el camino a seguir con el fin de lograr conocimientos seguros, confiables y demostrables; y las técnicas aplicadas en el transcurso del proyecto se resumen en la fase de materiales y métodos.

En la fase de resultados se hace referencia a la propuesta alternativa, la cual plantea la solución del problema en el diseño de la herramienta implementada y en los detalles del respectivo funcionamiento, validación y las pruebas realizadas que respaldan el resultado obtenido.

Se hace referencia también a la fase de discusión en la que, se evalúa el objeto de investigación dando a conocer la manera en la que se llevaron a cabo cada uno de los objetivos específicos del proyecto, además se menciona la valoración técnica-económica-ambiental en la que se dan a conocer los recursos utilizados para el desarrollo del presente proyecto.



Finalmente se pone a consideración las conclusiones determinadas mediante el estudio de los resultados obtenidos del Trabajo de Titulación, las respectivas recomendaciones o sugerencias para el posterior mejoramiento del sistema y consideraciones a tomar en cuenta; así como también la bibliografía que hace referencia a los distintos textos consultados y los anexos en el que constan los recursos complementarios para el desarrollo del presente Trabajo de Titulación.

D. REVISIÓN DE LITERATURA

CAPÍTULO I: TEORÍAS DE CONTROL

1.1. Aspectos generales

1.1.1. Sistema

Un sistema se entiende como una unidad cuyos elementos interaccionan juntos, ya que continuamente se afectan unos a otros, de modo que operan hacia una meta común. Es algo que se percibe como una identidad que lo distingue de lo que la rodea, y que es capaz de mantener esa identidad a lo largo del tiempo y bajo entornos cambiantes [1].

Según Senge Peter, un sistema es una totalidad percibida cuyos elementos se “aglomeran” porque se afectan recíprocamente a lo largo del tiempo y operan con un propósito común. La palabra deriva del verbo griego *sunislánai* que originalmente significaba “causar una unión”. Como sugiere este origen, la estructura de un sistema incluye la percepción unificadora del observador [2].

Como ejemplos de sistemas se puede citar los organismos vivientes (incluidos los cuerpos humanos), la atmósfera, las enfermedades, los nichos ecológicos, las fábricas, las reacciones químicas, las entidades políticas, las comunidades, las industrias, las familias, los equipos y todas las organizaciones. Usted y su trabajo son elementos de muchos sistemas.

En pocas palabras, un ‘sistema’ (ver *Figura 1*) [2]; es un conjunto de elementos que interactúan entre sí y está limitado por una frontera (real o imaginaria).



Figura 1: Sistema limitado por una frontera real o imaginaria.

En un sistema físico, los elementos de interés son ciertas cantidades físicas, las cuales están relacionadas de acuerdo a principios basados sobre las leyes fundamentales de la física.

Mientras que en un proceso químico, las cantidades involucradas son el balance de masa y energía.

Por otra parte, en un sistema económico, se tiene la interacción de las fuerzas económicas (oferta y demanda).

La Ingeniería de Control, encara:

- El Control de Máquinas,
- La Automatización de Procesos Industriales,
- El Control de Procesos Económicos, etc.

1.1.1.1. Definición de Sistema de Control

Un sistema de control, es una interconexión de componentes que forman una configuración del sistema, la cual proporcionará una respuesta deseada del sistema.

1.1.1.2. Breve Historia

- A grandes rasgos, se presenta el desarrollo histórico de la teoría de control a nivel mundial.
- Herón de Alejandría, 1er siglo d.C., escribe el libro Pneumática, sobre un mecanismo de nivel de agua.
- James Watt, 1769, inventa el primer regulador centrífugo que controla la velocidad de una máquina de vapor.
- Pilzunov, 1765, inventa el regulador del nivel de agua a base de un flotador.
- Nyquist, 1930, inventa el sistema de realimentación en amplificadores telefónicos, en el laboratorio de la Bell Telephone, conjuntamente con Bode y Black.
- En la II guerra mundial, recibió un gran impulso la teoría de control, tanto en USA y Europa.
- Se utilizan métodos matemáticos y analíticos, la Ingeniería de Control llega a ser una disciplina completa.
- Después de la guerra, se amplía el uso de la Transformada de Laplace y el Plano de Frecuencia Compleja.
- En la década de los años 1950, se desarrolla y usa los métodos en el plano 's', con el enfoque de los lugares geométricos de las raíces, se emplean las computadoras analógicas y digitales.

- El lanzamiento del Sputnik¹, da otro impulso a los sistemas de control.
- Liapunov y Minorsky, desarrollan métodos en el dominio del tiempo.
- R. Bellman (USA), desarrolla nuevas teorías del control óptimo.
- Se incrementa la automatización y control de procesos industriales por medios automáticos.
- Se optimiza la productividad, relación de la salida física a la entrada física, productividad del trabajador, producción real por horas de trabajo.
- Se implementa el control de procesos por computadora y se enfatiza en el desarrollo de Hardware y Software para tal fin.
- Desde hace dos décadas emerge el control inteligente, mediante:
 - ✓ El desarrollo de sistemas expertos y la inteligencia artificial.
 - ✓ El tratamiento de la información mediante las redes neuronales
 - ✓ La aplicación de la lógica difusa en el control moderno.

1.2. Control Clásico/Moderno e Inteligente

La teoría desarrollada para el control de procesos, desde el punto de vista clásico y moderno, tiene su base esencial en el conocimiento de la dinámica del proceso que se desea controlar. Esta dinámica normalmente se expresa haciendo uso de ecuaciones diferenciales ordinarias, y en el caso de sistemas lineales, se hace uso de la transformada de Laplace para obtener una representación matemática que relaciona la señal que se quiere controlar y la señal de entrada al sistema. Esta relación matemática se conoce como función de transferencia.

Desde la teoría clásica de control, considerando el caso más sencillo de un sistema lineal de una entrada y una salida, la dinámica se puede representar como en la *Figura 2* [3]. En esta figura se representa el bloque etiquetado como “Proceso” o “Planta”, que es el sistema que se desea controlar. A este sistema le llegan dos señales, una etiquetada como “Entrada de control” que será la señal que genera el controlador que se ha de diseñar y la señal etiquetada como “Entrada incierta” que puede representar cualquier señal indeseable externa al sistema y que se conoce también como “perturbación” o “ruido”. Finalmente la señal de “Salida” que será la señal que se desea que se comporte de una forma determinada. La señal de salida también se conoce como señal controlada [3].

¹ El Sputnik fue el primer satélite artificial de la historia, lanzado en 1957 por la Unión Soviética.

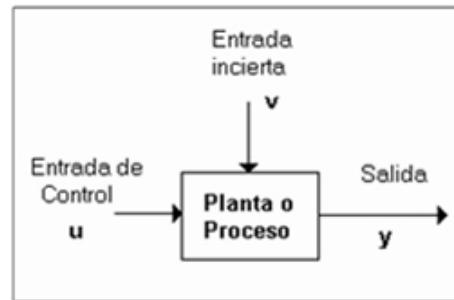


Figura 2: Representación a bloques de un sistema.

La representación anterior puede ser un poco más general si se hace uso de la teoría de control moderna, en donde la representación matemática utiliza el concepto denominado estado del sistema. Su representación más general se muestra en la Figura 3 [3]. De la planta ahora se observa una señal adicional denominada “Estado” x del sistema, que es la señal que proporcionará información más completa de la planta.

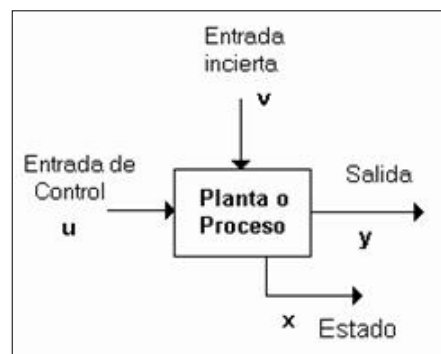


Figura 3: Diagrama de un sistema dinámico.

1.2.1. Control Inteligente

El incremento de las demandas tecnológicas en nuestros tiempos, ha generado sistemas muy complejos que requieren controladores altamente sofisticados para asegurar alto desempeño dentro de condiciones adversas. Estas y otras condiciones de control no se pueden cumplir con controladores convencionales, debido principalmente a la falta de conocimiento preciso acerca del proceso que se desea controlar. La adquisición de conocimiento adecuado del sistema en ocasiones es problemática o impráctica debido a la complejidad del sistema y al hecho de que la estructura y los parámetros en muchos sistemas cambian de manera significativa e impredecible con el tiempo. Es bajo estas condiciones en donde se utilizan las técnicas del control inteligente.

El control inteligente es una generalización del concepto de control y se puede ver como un campo dentro de la disciplina del control. El control inteligente es la disciplina donde los métodos de control se desarrollan para emular algunas características importantes del ser humano. Estas características incluyen adaptación y aprendizaje, planeación bajo gran incertidumbre y el trabajo con gran cantidad de datos.

Las metodologías de control inteligente están siendo aplicadas a la robótica, las comunicaciones, la manufactura, el control de tráfico, por mencionar algunas pocas. Las áreas donde se está realizando trabajo alrededor del control inteligente son: redes neuronales, control difuso, algoritmos genéticos, sistemas de planeación, sistemas expertos y sistemas híbridos (combinación de más de una de las técnicas anteriores).

Un sistema de control inteligente debe ser autónomo; esto significa que tiene el poder de autogobernarse. Existen varios grados de autonomía: un controlador totalmente autónomo debería tener la habilidad de aún reparar su propio hardware si uno de sus componentes falla. Un control fijo convencional como los mostrados en el tema anterior se considera con un bajo grado de autonomía, la autonomía es el objetivo en los sistemas de control complejos y los controladores inteligentes son una manera de lograrlo.

Los sistemas de control convencionales se diseñan usando los modelos matemáticos de sistemas físicos. Se selecciona un modelo matemático que captura el comportamiento de la dinámica de interés y entonces se aplican las técnicas de diseño, tal vez ayudados por sistemas CAD², para diseñar el modelo matemático apropiado del controlador. Luego se realiza el controlador ya sea en hardware o en software en el sistema físico. Este procedimiento puede llevar varias iteraciones hasta lograr el mejor comportamiento. El modelo matemático de la planta deberá ser “bastante simple” para que pueda ser analizado con técnicas matemáticas disponibles, y “bastante exacto” tal que describa los aspectos importantes y relevantes del comportamiento de la planta.

Aunque el proceso en donde se aplicarán estos controladores, en este trabajo de titulación, no es un proceso muy complejo, tiene algunos elementos no lineales que pueden ser un buen ejercicio para aplicar los esquemas de control inteligente. El objetivo de este trabajo radica esencialmente aplicar el esquema de control neuro-

² CAD (Computer Aided Design) o diseño asistido por computador

difuso, y sobre todo contribuir al conocimiento de estas técnicas de control para los estudiantes del Área de Energía.

1.2.1.1. Control difuso. (“Fuzzy³ Control”)

Para el diseño de un controlador difuso, como ya se mencionó anteriormente, no se requiere el modelo analítico completo del sistema dinámico. El resultado de este diseño es un controlador heurístico basado en conocimiento, utilizado para controlar un sistema complejo e indefinido.

En este apartado se explicará de manera breve la base necesaria para el diseño de un controlador difuso, así como su estructura fundamental, algunas características especiales a tomar en cuenta y algunas definiciones que serán necesarias para entender los conceptos aquí mencionados. Un controlador difuso es esencialmente un controlador no lineal.

Tomando como referencia el esquema de un control retroalimentado, el esquema de control difuso tiene ahora la estructura que se muestra en la *Figura 4* [3].

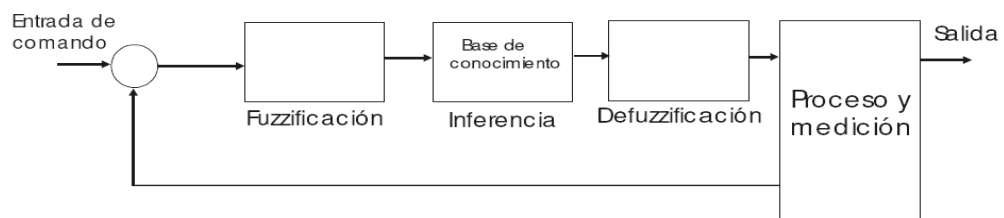


Figura 4: Esquema de un controlador difuso insertado en el esquema general de un control retroalimentado

En la *Figura 4* se aprecian tres bloques: uno etiquetado como Fuzzificación, otro como Defuzzificación y otro como Inferencia. La función involucrada en el primer bloque, la fuzzificación, tiene el objetivo de tomar el valor de la señal leída del proceso (en este esquema es la señal de error) y transformarla en una señal entendible por el sistema difuso. La función Defuzzificación tiene el objetivo de convertir el resultado del bloque etiquetado como inferencia, que es un resultado difuso, a un resultado entendible para el proceso. Finalmente el bloque etiquetado como Inferencia se encarga de realizar todo el razonamiento del sistema de control.

³ El término *Fuzzy* se suele traducir indistintamente como borroso o difuso.

Cuando se diseña un controlador difuso, es necesario determinar las variables de entrada y de salida. Las variables de salida representan el resultado de la operación que realizará el controlador y serán determinadas por el objetivo de control. La decisión de las variables de entrada depende de la situación en particular.

Generalmente, en sistemas de múltiples entradas y salidas (MIMO⁴), es difícil generar reglas de control. Existen aplicaciones exitosas del control difuso en sistemas de una entrada y una salida (SISO⁵). También se han aplicado en sistemas MIMO con buenos resultados.

El diseño de un controlador difuso involucra la construcción de reglas de control. En muchos casos, se pueden obtener estas reglas describiendo las acciones de los operadores del sistema de control en el formato IF-THEN. Sin embargo no es un método genérico para construir reglas de control, se pueden construir reglas IF-THEN no sólo de las acciones de los operadores sino de la respuesta característica del sistema a controlar. Una vez establecidas las reglas IF-THEN se puede realizar la estrategia de control usando razonamiento difuso. La estructura del controlador difuso es en sí misma la estructura del razonamiento difuso.

Las reglas difusas tienen la siguiente estructura:

IF (antecedente (1)) and (antecedente (2)) and and (antecedente (n))

THEN (consecuente (1))... (consecuente (n))

En el control difuso se distinguen también dos tipos de controladores, caracterizados esencialmente por la forma en que se definen los consecuentes, estos tipos de controladores denominados Mamdani y Takagi Sugeno, se definen de la forma siguiente:

- Control difuso tipo Mamdani. Un control difuso tipo Mamdani es aquel control que tiene reglas difusas cuya forma general es la siguiente:

$$r_k: \text{if } x_1 \text{ is } A_1 \text{ and } \dots \text{ and } x_n \text{ is } A_n \text{ then } y_1 \text{ is } B_1, \dots, y_m \text{ is } B_m$$

Un ejemplo particular de este tipo de control se puede escribir de la forma siguiente:

⁴ MIMO (Multiple Input Multiple Output)

⁵ SISO (Single Input Single Output)

*if el error es grande **and** el cambio del error es pequeño **then***

la señal de control es grande

- Control difuso tipo Takagi-Sugeno. Un control difuso de este tipo contiene reglas forma general es la siguiente:

$$r_k: \text{if } x_1 \text{ is } A_1 \text{ and } \dots \text{ and } x_n \text{ is } A_n \text{ then } y_1 = f_{1,k}(x_1, \dots, x_N), \dots, y_m = f_{m,k}(x_1, \dots, x_N)$$

En esta forma, se aprecia que el consecuente de la regla son funciones de las entradas al controlador.

Un controlador difuso tipo Takagi-Sugeno tiene como consecuente una o varias funciones que dependen directamente de las señales que se están leyendo del sistema o de las variables de estado. Un controlador tipo Mamdani tiene como consecuente un valor determinado con anterioridad y no una función.

1.2.1.2. Control Neuronal

Algunos autores retrataban a las redes neuronales artificiales hace dos décadas, como una forma de magia negra alternativa a los métodos racionales de la teoría de control. El reciente y considerable interés en control neuronal ha generado un número de propuestas y de esquemas diferentes para el diseño de sistemas de control que usan redes neuronales artificiales como se verá a continuación. Para aplicaciones en las industrias de procesos, lo más relevante tiene que ver con el uso de redes neuronales como identificadores de procesos o como sistemas que optimizan el funcionamiento de un controlador. En esta sección se describirán brevemente algunas de las propuestas de las redes neuronales artificiales usadas como controladores y que son llamados neurocontroladores.

El esquema más básico de una neurona artificial se muestra en la *Figura 5* [3].

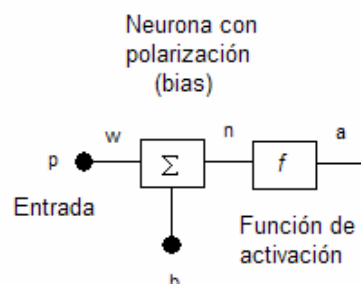


Figura 5: Esquema de una neurona artificial.

El valor de la salida de esta neurona está definido por la siguiente ecuación:

$$a = f(wp + b)$$

Donde p es el valor de la señal de entrada, b es un valor conocido como polarización (“bias”), w es el valor conocido como peso y es el que se ajusta en el proceso de entrenamiento y finalmente f es una función de activación. Una red neuronal artificial es la interconexión de varias neuronas, una capa de neuronas se muestra en la *Figura 6* [3].

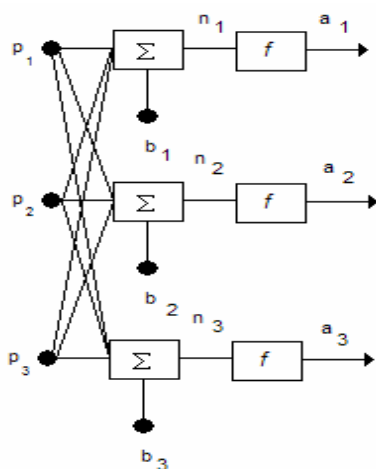


Figura 6: Una capa de neuronas.

Las funciones de activación más comúnmente usadas son: lineal, sigmoideal y tangente. En la *Figura 7* [3] se muestran los tres tipos de funciones mencionadas.

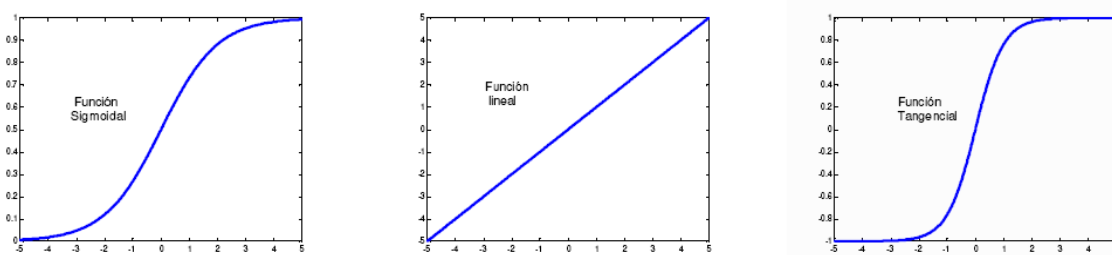


Figura 7: Funciones de activación comúnmente usadas.

Es importante hacer notar que una de las habilidades más importantes de la redes neuronales artificiales es su capacidad para aproximar funciones no - lineales. El entrenamiento de una red usando pares de datos (entrada – salida) de una planta o proceso con características no – lineales, se puede considerar como un problema de

aproximación de una función no – lineal. Una red con alimentación hacia adelante “feedforward” puede aproximar adecuadamente bien una función continua; la red puede tener una o varias capas ocultas y cada neurona puede utilizar, por ejemplo, la función de activación denominada sigmoideal

1.2.1.3. Control Neurodifuso

En los temas anteriores se ha descrito la forma en que se usan tanto las redes neuronales como los sistemas difusos para el control de procesos. El paso siguiente es describir ahora cómo utilizando estas dos técnicas se puede construir un sistema neurodifuso y los resultados pueden ser usados como un controlador.

Desde un primer punto de vista, un sistema neurodifuso es una combinación de redes neuronales y sistemas difusos de tal forma que la red neuronal o los algoritmos de aprendizaje de la red neuronal se usan para determinar parámetros del sistema difuso. Esto significa que la intención principal de un sistema neurodifuso es crear o mejorar un sistema difuso automáticamente por medio de los métodos o algoritmos de las redes neuronales. De manera inversa, una red neuronal difusa, es una red neuronal que usa métodos difusos para aprender más rápido o para funcionar mejor; en este caso la intención principal es mejorar la red neuronal.

Según el autor Lee la combinación de las redes neuronales y la lógica difusa recoge las ventajas de cada una de ellas. Las redes neuronales proporcionan el modelo de la estructura conexionista (tolerancia a fallas y propiedades de representación distribuida) con habilidades de aprendizaje a los sistemas de lógica difusa; y la lógica difusa proporciona a las redes neuronales una armazón estructural con reglas de representación del conocimiento y razonamiento de alto nivel SI - ENTONCES (“*if then rules*”).

Las diferentes combinaciones de redes neuronales y lógica difusa descritas por Nauck y Kruse, y también mencionadas por Lee, con una clasificación ligeramente diferente pero que al final de cuentas son las mismas, se pueden describir de la forma siguiente:

- **REDES NEURONALES DIFUSAS “FUZZY”**. En este tipo de combinación se usan métodos difusos para incrementar la capacidad de aprendizaje de una red neuronal. Por ejemplo, se puede hacer variar la razón de aprendizaje (“learning rate”) usando reglas difusas.

- **SISTEMAS NEURONAL/DIFUSO“FUZZY” CONCURRENTES.** En este tipo de sistemas, una red neuronal y un sistema difuso trabajan juntos en la misma tarea sin influenciarse el uno al otro. Usualmente la red neuronal se usa para preprocesar la entrada o post-procesar la salida del sistema difuso.
- **MODELOS NEURODIFUSOS COOPERATIVOS.** En este tipo de esquemas, una red neuronal se usa para determinar los parámetros de un sistema difuso (reglas, pesos y/o conjuntos difusos). Después de la fase de aprendizaje, el sistema difuso trabaja sin la red neuronal. Estos modelos de sistemas neurodifusos suelen ser de 4 clases:
 - ✓ Aprende conjuntos difusos fuera de línea.
 - ✓ Aprende reglas difusas fuera de línea.
 - ✓ Aprende conjuntos difusos en línea.
 - ✓ Aprende reglas difusas en línea.
- **MODELOS NEURODIFUSOS HÍBRIDOS.** En este caso, las redes neuronales y los sistemas difusos se combinan para hacer una arquitectura homogénea. El sistema se puede interpretar como una red neuronal especial con parámetros difusos o como un sistema difuso implementado en forma distribuida en paralelo. Como aplicación de estos tipos de modelos existen ya varios algoritmos desarrollados como son: ANFIS, FuNe, ARIC, NEFCLASS, NEFCON, NEFPROX.

En los siguientes capítulos se analiza más acerca de los sistemas Neurodifusos y el algoritmo ANFIS.

CAPÍTULO II: SISTEMAS NEURO-DIFUSOS

2.1. Antecedentes

Los sistemas Neuro-difusos forman parte de una nueva tecnología de computación llamada computación flexible (soft-computing), la cual tiene su origen en la teoría de la Inteligencia Artificial. La computación flexible engloba un conjunto de técnicas que tienen en común la robustez en el manejo de la información imprecisa e incierta que existe en los problemas relacionados con el mundo real (por ejemplo: reconocimiento de formas, clasificación, toma de decisiones, etc.) [4]. En algunos casos, las técnicas de computación flexible pueden ser combinadas para aprovechar sus ventajas individuales; algunas de estas técnicas son:

- La Lógica difusa.
- Las Redes Neuronales.
- Los Algoritmos Evolutivos o Algoritmos Genéticos.
- La Teoría del Caos.
- La Teoría del Aprendizaje.
- El Razonamiento Aproximado.

En los siguientes puntos se explicarán brevemente las técnicas de Redes Neuronales, Lógica Difusa y la combinación de ambas, por ser las de principal interés para el desarrollo de este trabajo de titulación.

2.2. Introducción a las Redes Neuronales

Las Redes Neuronales Artificiales (RNAs) son la implementación en hardware y/o software de modelos matemáticos idealizados de las neuronas biológicas, estas son interconectadas unas a otras y son distribuidas en capas de tal forma que emulan en forma simple la estructura neuronal de un cerebro. Cada modelo de neurona es capaz de realizar algún tipo de procesamiento a partir de estímulos de entrada y ofrecer una respuesta, por lo que las RNA en conjunto funcionan como redes de computación paralelas y distribuidas similares a los sistemas cerebrales biológicos [5]. Sin embargo, a diferencia de las computadoras convencionales, las cuales son programadas para realizar tareas específicas, las redes neuronales artificiales, tal como los sistemas cerebrales biológicos, deben ser entrenadas.

2.2.1. La Neurona Biológica

El sistema nervioso humano, incluido el cerebro, consiste en la interconexión de millones y millones de pequeñas unidades celulares llamadas neuronas, por ejemplo, sólo la corteza cerebral humana, una capa larga y plana de neuronas de alrededor 2 a 3 milímetros de ancho con un área superficial de aproximadamente 2,200 cm², contiene alrededor de cien mil millones (10¹¹) de neuronas [6].

En cuanto a su fisiología, las neuronas constan de un cuerpo celular o soma, en donde se aloja el núcleo de la célula. Del cuerpo de la célula salen ramificaciones de diversas fibras conocidas como dendritas y sale también una fibra más larga denominada axón. En su extremo, el axón también se ramifica en filamentos y sub-filamentos mediante los que establece conexión con las dendritas y los cuerpos de las células de otras neuronas; estas conexiones son conocidas como sinapsis (*Ver Figura 8*) [6]. La comunicación entre neuronas ocurre como resultado de la liberación, por parte de la neurona pre-sináptica, de sustancias químicas con propiedades eléctricas llamadas neurotransmisores y la subsecuente absorción de esas sustancias por parte de la neurona post-sináptica.

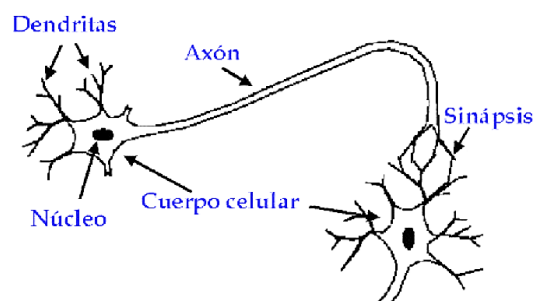


Figura 8: Neurona Biológica.

Aunque la operación de una neurona biológica no está totalmente esclarecida, la mayor parte de los biólogos concuerdan en que la neurona después de recibir entradas a través de sus dendritas, las procesa en su soma, transmite el resultado del cálculo a través de su axón y utiliza las conexiones sinápticas para transportar la señal desde el axón a miles de dendritas pertenecientes a otras neuronas.

2.2.2. Componentes de una Red Neuronal Artificial

Ciertamente existen computadoras que tienen la habilidad de realizar procesamiento paralelo, pero el gran número de procesadores que serían necesarios para imitar el

cerebro no puede ser alcanzado con la tecnología de hardware actual; además, la estructura interna de una computadora no puede ser cambiada mientras realiza una tarea. Estos hechos conducen a la utilización de un modelo idealizado de neurona biológica elemental para propósitos de simulación.

Aunque existen diferentes tipos de Redes Neuronales Artificiales todos ellos tienen casi los mismos componentes elementales. Como en el sistema nervioso biológico, una red neuronal artificial está constituida por neuronas que están unidas entre sí a través de conexiones, a las cuales se les asignan valores numéricos o pesos que representan el conocimiento de la Red Neuronal. Al cambiar los valores de los pesos se consigue imitar el cambio en la estructura de las conexiones sinápticas que ocurre durante el proceso de aprendizaje en la red neuronal biológica.

La siguiente figura muestra una neurona artificial idealizada.

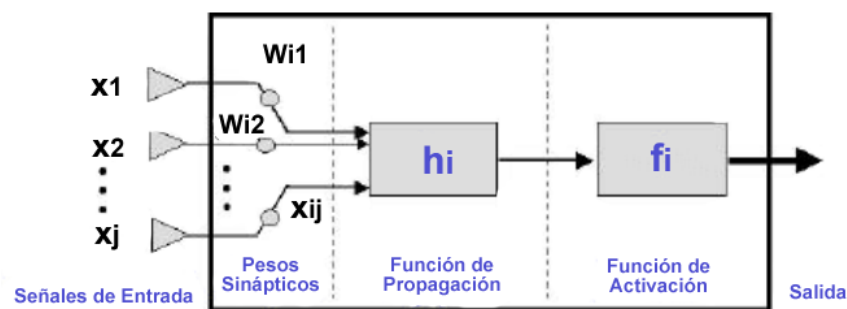


Figura 9: Neurona Artificial

Como se puede observar en la Figura 9 [6], la neurona artificial luce similar a su contraparte biológica.

La información que conforma un conjunto de entradas $X_j(t)$ es enviada a la neurona a través de sus conexiones con pesos sinápticos W_{ij} , donde el subíndice i representa a la neurona i . Esta entrada es procesada por una función de propagación (por ejemplo: $h_i(t) = (W_{ij} \cdot X_j)$). El resultado es comparado con un valor umbral θ por la función de activación $y_i(t) = f_i(h_i(t))$ que representa simultáneamente la salida de la neurona y su estado de activación. Sólo si la entrada excede el valor umbral, la neurona se activará, en caso contrario se inhibirá.

En una RNA las neuronas suelen estar agrupadas en capas. Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente, y cuyas

salidas tienen el mismo destino. Usualmente cada neurona de una capa está conectada a todas las neuronas de las capas anterior y posterior (excepto en la capa de entrada y en la capa de salida).

Las Redes Neuronales permiten resolver problemas que no pueden ser solucionados usando algoritmos convencionales. Tales son usualmente problemas de clasificación u optimización. Los diferentes dominios en los que las redes neuronales son utilizadas, incluyen:

- Asociación de patrones.
- Clasificación de patrones.
- Procesamiento de imágenes.
- Reconocimiento de voz.
- Problemas de optimización.
- Simulación.

2.2.2.1. El Proceso de Aprendizaje

Como ya se mencionó, biológicamente se suele aceptar que la información memorizada en el cerebro está más relacionada con los valores sinápticos de las conexiones entre las neuronas que con las neuronas mismas, es decir, el conocimiento se encuentra en las sinapsis y todo proceso de aprendizaje consiste en la creación, modificación y destrucción de estas conexiones entre las neuronas. De forma similar el aprendizaje en las RNA consiste en determinar un conjunto de pesos sinápticos que permita a la red realizar correctamente el tipo de procesamiento deseado, esto se logra a través del entrenamiento de la red. Una vez que la red neuronal ha sido correctamente entrenada será capaz de determinar la salida deseada para las entradas que se le presenten.

2.3. Introducción a los Sistemas Difusos

La vida real está llena de situaciones que requieren del razonamiento aproximado para manipular información cualitativa más que cuantitativa.

Un sistema difuso puede resolver problemas tal como lo haría un experto humano, problemas tales como controlar la presión y temperatura de una caldera en la industria, procesar y reconocer imágenes o controlar una lavadora de ropa son

situaciones que tienen en común el ser complejas y dinámicas y también que son más fácilmente caracterizadas por palabras que por expresiones matemáticas.

Mientras que los sistemas de lógica clásica no pueden actuar en problemas de este tipo:

- No hay ningún sistema para representar proposiciones en lenguaje natural cuando el significado es impreciso.
- Sí el significado puede representarse, no hay un mecanismo para poder evaluarlo y obtener una respuesta.

Hasta este momento ya tenemos una idea de lo que es la lógica difusa, pero ¿de dónde surge el término fuzzy?, ¿Quién lo introdujo?, veamos:

En la década de los años veinte del siglo pasado, J. Lukasiewicz desarrolló los principios de la lógica multivaluada, cuyos enunciados pueden tener valores de verdad comprendidos entre el 0 (falso) y el 1 (cierto) de la lógica binaria clásica. En 1965 **Lofti A. Zadeh**, aplicó la lógica multivaluada a la teoría de conjuntos, estableciendo la posibilidad de que los elementos pudieran tener diferentes grados de pertenencia a un conjunto. Zadeh introdujo el término fuzzy (difuso) y desarrollo un álgebra completa para los conjuntos difusos [7].

Desde sus inicios la teoría de conjuntos difusos causó controversias y debates, por lo que sus primeras aplicaciones prácticas surgieron hasta mediados de los años setenta.

2.3.1. Conceptos básicos

Comenzaremos a hablar de conjuntos tratando de ponernos de acuerdo acerca de lo que vamos a entender por conjunto. Esto no es nada simple ya que conjunto es un concepto bastante primitivo, análogo al concepto de punto o de recta. De ahora en adelante, para nosotros un conjunto será algo que puede o no tener elementos, pero con la propiedad de que se puede decir si un objeto dado cualquiera es o no elemento del conjunto en cuestión.

A los objetos (si los hay) que forman un conjunto, les llamamos "elementos" de dicho conjunto. En la teoría de conjuntos que estamos recordando, trabajaremos a veces

con varios conjuntos cuyos elementos son todos del mismo tipo, es decir, pertenecen todos a un mismo conglomerado de cosas al que podremos llamar conjunto universal.

Ahora bien, Sí A es un conjunto la proposición " x es un elemento de A " se denotará por $x \in A$. La negación de esta proposición es decir " x no es un elemento de A " se denotará por $x \notin A$.

Por ejemplo, sí A es el conjunto de los primeros tres números naturales pares, entonces $2 \in A$, $4 \in A$, $6 \in A$, $24 \notin A$, $9 \notin A$, son proposiciones verdaderas. Note que "*pertenencia*" es una relación que vincula cada elemento con un conjunto, no es una relación entre elementos de un conjunto.

Sí enumeramos todos los elementos de un conjunto, decimos que los hemos representado por *extensión* mientras que si enunciamos una propiedad definitoria de los elementos del conjunto, se dice que está representado por *comprensión*.

2.3.2. Función de membresía

Como ya se mencionó la *pertenencia* o *membresía* es una relación que vincula a cada elemento con un conjunto. En otras palabras, en un conjunto bien definido (lógica clásica), la pertenencia o no pertenencia de un elemento x a un conjunto A se describe mediante la función característica $\mu_A(x)$ donde [7]:

$$\mu_A = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Dicha función es llamada *función de membresía* o *función característica* de A y está definida para todos los elementos del universo. La función de membresía hace un mapeo de todo el universo U a su conjunto de evaluación de dos elementos $\{0, 1\}$, esto se escribe:

$$\mu_A(x): U \rightarrow \{0, 1\}$$

Con una identificación de $\{0, 1\}$ y $\{\text{verdadero, falso}\}$, esta función característica puede jugar un papel importante en la asignación de valores de verdad a proposiciones referentes al conjunto A .

2.3.2.1. Funciones de Pertenencia Típicas

Función Singleton: Sea a un punto del universo, la Función Singleton (solitaria) es aquella que toma valor 1 solo en a y 0 en cualquier otro punto (ver Figura 10) [8].

$$\mu(x) = \begin{cases} 1 & \text{si } x = a \\ 0 & \text{si } x \neq a \end{cases}$$

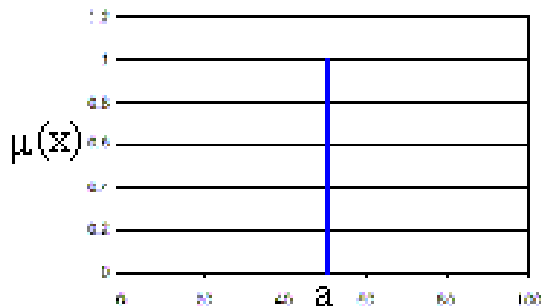


Figura 10: Función Singleton

Función Triangular: Definido por sus límites (inferior a y superior b), y el valor modal m , tal que $a < m < b$ (ver Figura 11) [8].

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x-a)/(m-a) & \text{si } x \in (a, m) \\ (b-x)/(b-m) & \text{si } x \in (m, b) \\ 0 & \text{si } x \geq b \end{cases}$$

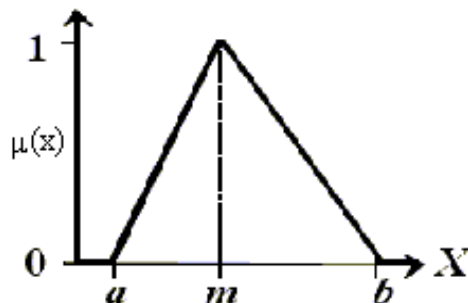


Figura 11: Función Triangular

Función G (gamma): Definida por su límite inferior a y el valor $k > 0$ (ver Figura 12) [8].



Figura 12: Función G

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ 1 - e^{-k(x-a)^2} & \text{si } x > a \end{cases}$$

Esta función se caracteriza por un rápido crecimiento a partir de a . Cuanto mayor es el valor de k , el crecimiento es más rápido aún. Nunca toman el valor 1, aunque tienen una asíntota horizontal en 1.

Se aproximan linealmente por (ver Figura 13) [8]:

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x-a)/(m-a) & \text{si } x \in (a, m) \\ 1 & \text{si } x \geq m \end{cases}$$

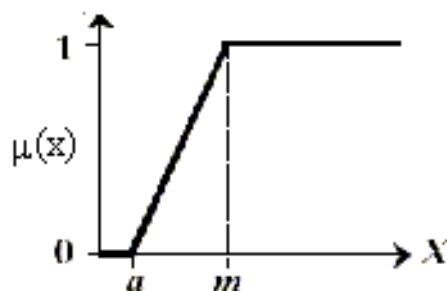


Figura 13: Aproximación lineal de la Función G

La función opuesta se denomina **Función L**.

Función S: Definida por sus límites inferior a y superior b , y el valor m , o punto de inflexión tal que $a < m < b$. Un valor típico es: $m = (a+b) / 2$ (ver Figura 14) [8].

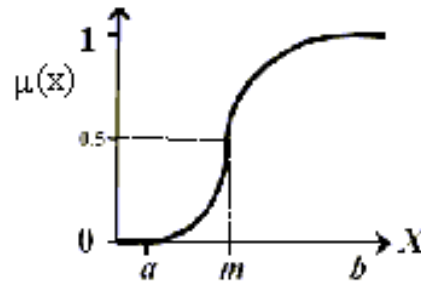


Figura 14: Función S

El crecimiento es más lento cuanto mayor sea la distancia $a-b$.

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ 2 \left(\frac{x-a}{b-a} \right)^2 & \text{si } x \in [a, m] \\ 1 - 2 \left(\frac{x-b}{b-a} \right)^2 & \text{si } x \in [m, b] \\ 1 & \text{si } x \geq b \end{cases}$$

Función Gaussiana: Definida por su valor medio m y el valor $k > 0$. Es la típica campana de Gauss. Cuanto mayor es k , más estrecha es la campana (ver Figura 15) [8].

$$\mu(x) = e^{-k(x-m)^2}$$

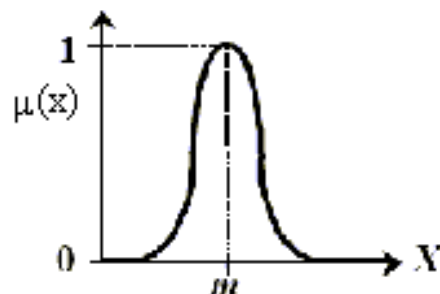


Figura 15: Función Gaussiana

Función Trapezoidal: Definida por sus límites inferior **a** y superior **d**, y los límites de su soporte, **b** y **c**, inferior y superior respectivamente (ver Figura 16) [8].

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \text{ o } x \geq d \\ (x-a)/(b-a) & \text{si } x \in [a, b] \\ 1 & \text{si } x \in [b, c] \\ (d-x)/(d-c) & \text{si } x \in (c, d) \end{cases}$$

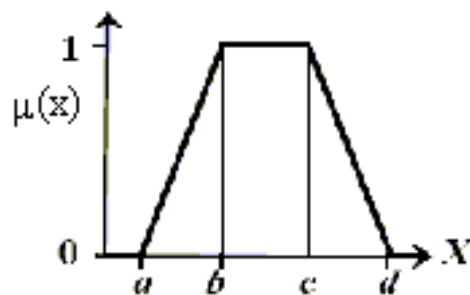


Figura 16: Función Trapezoidal

La función **Trapezoidal** se adapta bastante bien a la definición de cualquier concepto, con la ventaja de su fácil definición, representación y simplicidad de cálculos.

Función Pseudo-Exponencial: Definida por su valor medio **m** y el valor **k > 1**. Cuanto mayor es el valor de **k**, el crecimiento es más rápido aún y la “campana” es más estrecha (ver Figura 17) [8].

$$\mu(x) = \frac{1}{1 + k(x-m)^2}$$

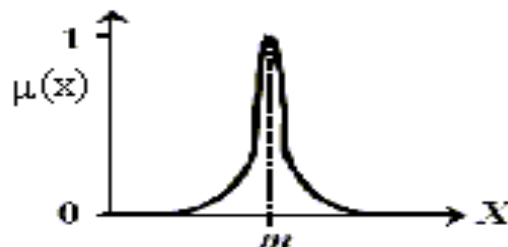


Figura 17: Función Pseudo-Exponencial

2.3.3. Razonamiento Difuso e Inferencia

El razonamiento difuso es un método que interpreta los valores de un vector de entrada y, basado en un conjunto de reglas, asigna valores al vector de salida [9]. Los tres conceptos que constituyen la base del razonamiento difuso son:

- Método de Implicación: Enlace funcional entre el grado de veracidad en las regiones difusas relacionadas por las proposiciones.
 - ✓ Mínimo: Valor lingüístico de salida truncado
 - ✓ Producto: Valor lingüístico de salida escalado
- Método de Agregación: Forma de combinar las regiones difusas relacionadas.
 - ✓ Máximo: $\text{Max } A, B, C = \text{Max}(\mu_A x, \mu_B x, \mu_C x)$
 - ✓ Suma: $R = \Sigma(\mu_A x, \mu_B x, \mu_C x, \dots)$
 - ✓ OR Probabilístico: $\text{ProbOR } A, B = \mu_A x + \mu_B x - \mu_A x \cdot \mu_B x$
- Método de Defusificación: Operación que recibe el resultado de la agregación y entrega un valor resultante esperado.
 - ✓ Centroide.
 - ✓ Bisector.
 - ✓ Media de Valores Máximos.
 - ✓ Mayor de los Máximos.
 - ✓ Menor de los Máximos

El proceso de inferencia difusa involucra funciones de membresía, operadores difusos y reglas difusas para relacionar el vector de entrada dado al vector de salida. Estas relaciones proporcionan la base para tomar decisiones o identificar patrones. Hay dos tipos de sistemas de inferencia difusa:

- El propuesto por Ebrahim Mamdani; y,
- El propuesto por Takagi, Sugeno y Kang.

2.3.4. Decodificando la salida: Defusificación

Como la entrada del proceso de defusificación es un conjunto fuzzy, este se encarga que salida sea un número real simple, es decir, convertir la salida difusa en un valor nítido [10], que es representada por la función de membresía de la acción de control.

Existen varios métodos para llevar a cabo estrategias de defusificación, pero los más utilizados son:

- **Principio de la membresía Max:** También conocido como el método de altura, este esquema se limita a funciones de salida de picos. Este método se da por la expresión algebraica:

$$\mu_c z^* \geq \mu_c z \quad \text{para todos } z \in Z$$

Donde z^* es el valor defusificado y se muestra gráficamente en la *Figura 18* [11].

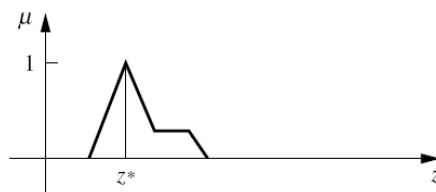


Figura 18: Método Principio de la membresía Max

- **Método Centroid:** Este procedimiento (también llamado centro de área, centro de gravedad) es la más prevalente y físicamente atractivo de todos los métodos de defuzzificación [11]. Este método viene dado por la expresión algebraica:

$$Z^* = \frac{\sum_{j=1}^q Z_j \mu_c(Z_j)}{\sum_{j=1}^q \mu_c(Z_j)}$$

Donde q es el número de niveles de cuantización de salida, Z_j es la suma de las salidas de control en el nivel de cuantización j y $\mu_c(Z_j)$ representa los valores de la función de membresía en c . En otras palabras, este método asigna el centro del área de la salida difusa final al valor defusificado. Este método se muestra en la *Figura 19* [11].

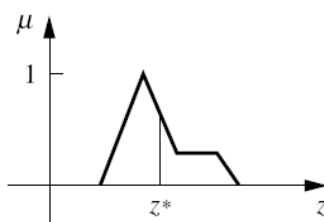


Figura 19: Método de defusificación Centroid.

- **Método Weighted Average:** El método de la media ponderada se la utiliza con mayor frecuencia en aplicaciones difusas, ya que es uno de los métodos más computacionalmente eficiente [11]. En este método la salida se obtiene mediante la media ponderada de la salida de cada uno del conjunto de reglas almacenadas en la base de conocimientos del sistema [12], puede ser expresada como:

$$Z^* = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i}$$

Donde Z^* es la salida defuzzificada, w_i es la pertenencia a la salida de cada regla, y z_i es el peso asociado a cada regla.

Este método se muestra en la *Figura 20* [11]. El método de la media ponderada se forma mediante la ponderación de cada función de la pertenencia a la salida por su respectivo valor máximo de membresía. Como un ejemplo, las dos funciones que se muestran en la *Figura 20* se traducirían en la siguiente forma general para el valor defusificado:

$$Z^* = \frac{a(0.5) + b(0.9)}{0.5 + 0.9}$$

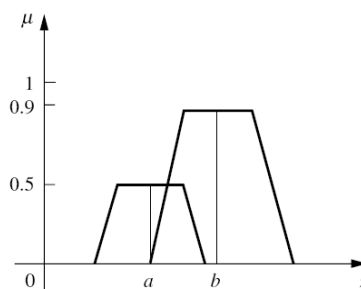


Figura 20: Método de defusificación Weighted Average.

2.4. Introducción a los Sistemas Neuro-difusos

Como se ha visto a lo largo del presente trabajo, la Lógica Difusa y las Redes Neuronales tienen propiedades computacionales particulares que las hacen adecuadas para ciertos problemas particulares y no para otros. Por ejemplo, mientras las redes neuronales ofrecen ventajas como el aprendizaje, adaptación, tolerancia a

fallas, paralelismo y generalización, no son buenas para explicar cómo han alcanzado sus decisiones. En cambio los sistemas difusos, los cuales razonan con información imprecisa a través de un mecanismo de inferencia bajo incertidumbre lingüística, son buenos explicando sus decisiones pero no pueden adquirir automáticamente las reglas que usan para tomarlas.

Los sistemas Neuro-Difusos combinan la capacidad de aprendizaje de las RNAs con el poder de interpretación lingüística de los sistemas de inferencia difusos, obteniéndose los siguientes resultados [7]:

- Aplicabilidad de los algoritmos de aprendizaje desarrollados para redes neuronales.
- Posibilidad de promover la integración de conocimiento (implícito que puede ser adquirido a través del aprendizaje y explícito que puede ser explicado y entendido).
- La posibilidad de extraer conocimiento para una base de reglas difusas a partir de un conjunto de datos.

Existen sistemas de desarrollo que han logrado unir la Lógica Difusa con las Redes Neuronales, por ejemplo se tiene: ANFIS

2.4.1. ANFIS

La arquitectura ANFIS (Sistema de Inferencia Neurodifuso Adaptivo) [13], sirve como base para construir un conjunto de reglas difusas si-entonces (if-then) con funciones de membresía apropiadas para generar los pares de datos de entrada-salida óptimos, es un tipo de red adaptiva funcionalmente equivalente a un Sistema de Inferencia Difuso tipo Sugeno. Su mecanismo de trabajo es similar al de una red neuronal solo que ajusta funciones de membresía en vez de pesos sinápticos. Al igual que en una red neuronal entrenada con el algoritmo de Retropropagación (Backpropagation), ANFIS utiliza este algoritmo para su entrenamiento, el ajuste de los valores en un sentido lo hace a través del método de optimización denominado Estimador de Mínimos Cuadrados, y de regreso ajusta con el método de Gradiente Descendiente, a este método de aprendizaje se le conoce como Aprendizaje Híbrido, pues combina ambos métodos de optimización. Mediante esto, se llega a una convergencia mucho más rápida que si solo se utiliza Gradiente Descendiente.

2.4.1.1. Estructuras de las ANFIS

El modelo ANFIS fue propuesto por Jang [14]. Por facilidad para su explicación, asumimos que el sistema de inferencia difusa bajo consideración tiene dos entradas x y y y una salida f . Para un modelo difuso Sugeno de primer orden, un conjunto de reglas y comunes son las reglas difusas si-entonces (if-then) siguientes:

Regla 1: if x es A_1 y y es B_1 , then $f_1 = p_1x + q_1x + r_1$,

Regla 2: if x es A_2 y y es B_2 , then $f_2 = p_2x + q_2x + r_2$.

La red ANFIS tiene una estructura de cinco capas, donde los nodos cuadrados representan nodos adaptivos y los circulares nodos fijos. La arquitectura ANFIS correspondiente se muestra en la *Figura 21* [14].

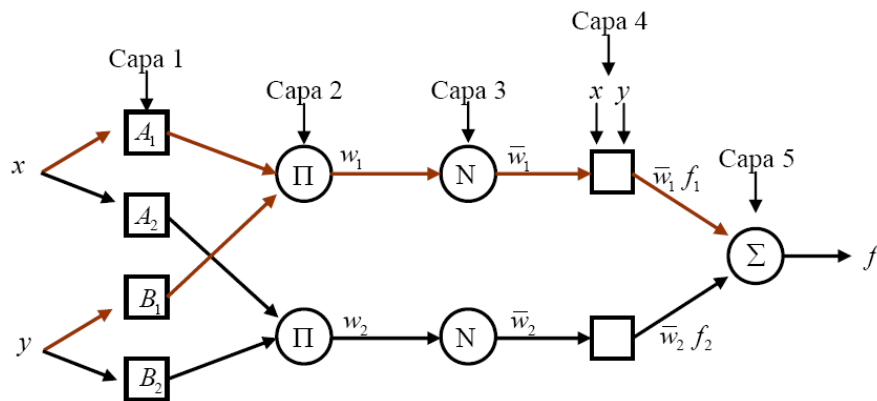


Figura 21: Arquitectura ANFIS

Capa 1

Cada nodo i en esta capa es un nodo adaptivo con una función:

$$O_{1,i} = \mu_{A_i} x, \quad \text{para } i = 1,2$$

$$O_{1,i} = \mu_{B_{i-2}} y, \quad \text{para } i = 3,4$$

Donde x o y son las entradas a los nodos, mientras que A_i y B_{i-2} son las variables lingüísticas asociadas con cada uno de ellos. Es decir, $O_{1,i}$ es el grado de membresía o pertenencia al conjunto difuso A , y da cuenta del grado de similitud de la entrada x (o) con las variables lingüísticas correspondientes. Por tanto, se representa mediante lo que se conoce como función de membresía o pertenencia al conjunto difuso. La

función de membresía tiene varias posibles representaciones aunque la más utilizada es la función campana.

Los parámetros de esta capa son los que aparecen en las cuatro funciones de membresía que se tienen y se denominan parámetros de premisa, ya que son los que describen la parte antecedente de las reglas If-Then.

Capa 2

Los nodos de esta capa son fijos, es decir, no tienen parámetros que haya que determinar de modo adaptivo. Se etiquetan mediante Π porque multiplican las salidas de la capa anterior, por lo que son nodos funcionales:

$$O_{2,i} = w_i = \mu_{A_i} x \times \mu_{B_i} y, i = 1,2.$$

Si se observa la expresión anterior, comprueba que la salida de cada nodo se calcula multiplicando las salidas de la capa anterior de acuerdo con las reglas si-entonces (if-then) de las que se parte, ya que $O_{2,1} = \mu_{A1} x \times \mu_{B1} y$ corresponde con la *regla 1*, y $O_{2,2} = \mu_{A2} x \times \mu_{B2} y$ con la *regla 2*. Por lo tanto, los nodos de esta capa representan el grado de verosimilitud que tiene cada una de las reglas.

Capa 3

Está formada por nodos fijos etiquetados como N en la *Figura 21*. La función de esta capa es realizar un promedio de las salidas de la capa anterior; de este modo consigue normalizar el grado de verosimilitud de cada regla:

$$O_{3,i} = w_i = \frac{w_i}{\sum_{i=1}^2 w_i} = \frac{w_i}{w_1 + w_2} \text{ para } i = 1,2.$$

Las salidas de esta capa se llaman “fuerzas de disparo normalizadas”.

Capa 4

Los nodos de esta capa son adaptivos y se describen mediante la siguiente función:

$$O_{4,i} = w_i f_i = w_i (p_i x + q_i x + r_i),$$

Donde w_i es el peso normalizado de la regla i y p_i, q_i, r_i es el conjunto de parámetros del consecuente porque aparecen en la parte consecuente de las reglas If-Then.

Capa 5

El único nodo presente en esta capa, es un nodo fijo denotado por Σ , el cual calcula la salida global como la suma de todas las señales entrantes:

$$O_{5,i} = Salida = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}.$$

Las salidas de esta capa se llaman “fuerzas de disparo normalizadas”. De esta forma, con esta arquitectura de cinco capas se implementa la red adaptiva que es funcionalmente equivalente al modelo difuso de Sugeno.

La estructura de esta red no es única, es posible combinar las capas 3 y 4 para obtener una red equivalente con 4 capas.

CAPÍTULO III: VISIÓN ARTIFICIAL Y PROCESAMIENTO DIGITAL DE IMÁGENES

3.1. Visión Artificial

La Visión Artificial es un subcampo de la Inteligencia Artificial. El propósito de la visión artificial es programar un ordenador para que entienda una escena o las características de una imagen [15].

Los objetivos típicos de la visión artificial incluyen:

- Detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes.
- Evaluación de los resultados.
- Registro de diferentes imágenes de una misma escena u objeto.
- Detección de movimiento de un objeto y su dirección y seguimiento del mismo en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena.
- Estimación de posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido.
- Detección de formas y su uso en la producción industrial y almacenaje.

Este proyecto está relacionado con la Visión Artificial ya que su objetivo es desarrollar una aplicación que sea capaz de reconocer ciertos objetos en imágenes capturado por una Webcam y lo traduzca en órdenes al ordenador, utilizando para ello funciones de procesamiento digital de la imagen.

La visión artificial es uno de los tipos de percepción que más se ha trabajado desde los comienzos de la Inteligencia Artificial como disciplina.

La visión de los humanos es una característica que típicamente se ha tratado de emular de manera artificial para poder luego construir dispositivos inteligentes con ciertas habilidades basadas en la visión. Luego, la visión artificial se puede definir como la capacidad de ver mediante dispositivos que procesan su información en un ordenador.

La visión artificial incluye muchas técnicas que son útiles para sí misma. Por ejemplo, el procesamiento de imágenes (que se refiere a la transformación, codificación y transmisión de las imágenes) y la extracción de patrones. Así, aunque la visión artificial se refiere ciertamente al procesamiento de imágenes, estas imágenes son solamente la materia prima de una ciencia mucho más amplia, la misma que se esfuerza en última instancia para emular las capacidades perceptivas del hombre, y quizás, para verter una luz sobre la manera por la cual él logra su interacción adaptativa y robusta con su ambiente. En la *Figura 22* [15] se muestran las diferentes aplicaciones de la Visión Artificial.

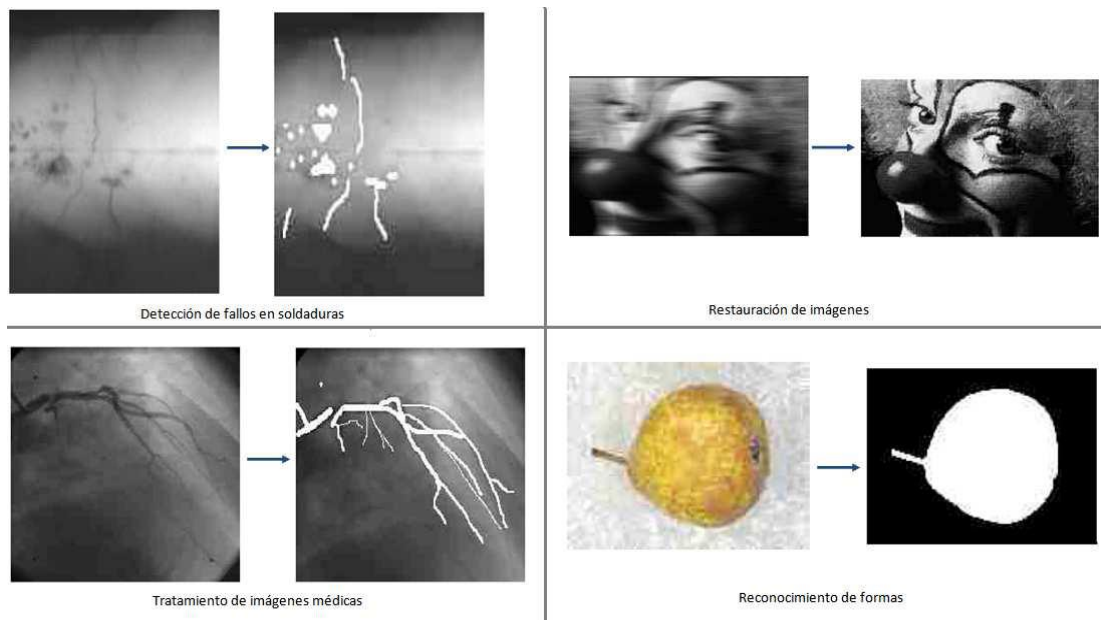


Figura 22: Ejemplos de aplicación de visión artificial.

La percepción del mundo actual abarca la detección y clasificación inmediata de múltiples objetos tanto animados como inanimados. Así la visión constituye una ventana al reconocimiento del mundo y el sentido con mayores posibilidades de que se dispone para conocer el mundo observándolo. La visión artificial, pretende realizar y simular tareas de este proceso mediante sistemas ópticos y computacionales en lugar de un ojo y cerebro de un ser vivo.

Las imágenes recogidas para su interpretación son imágenes digitales. Una imagen digital es una representación discreta, con valores en puntos regularmente muestreados y es una representación cuantificada, donde cada valor es un número entero.

La visión artificial incluye muchas técnicas, algunas de ellas son el procesamiento de imágenes y la extracción de patrones; otras más significativas tratan de conseguir una descripción geométrica útil: forma, perímetro, área para realizar modelos geométricos y con otra se realiza un análisis de color [15].

3.2. Procesamiento Digital de Imágenes

El procesamiento digital de imágenes está referido a la realización de transformaciones, restauración y mejoramiento de imágenes. El análisis consiste en la extracción de propiedades y características de las imágenes, así como la clasificación, identificación y reconocimiento de patrones [16].

3.2.1. Aspectos Generales

Referencias:

- $f(x,y)$: imagen digital
- q, p : píxeles
- S : un subconjunto de píxeles de $f(x,y)$

3.2.1.1 Píxel

Es la abreviatura de las palabras inglesas "*picture element*". Es el menor de los elementos de una imagen al que se puede aplicar individualmente un color o una intensidad o que se puede diferenciar de los otros mediante un determinado procedimiento [17], (Ver Figura 23) [18].

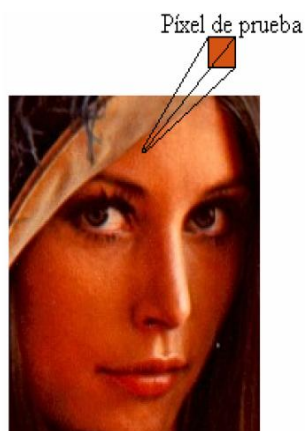


Figura 23: Representación de un píxel

3.2.1.2 Imagen digital

Una imagen digital es una imagen (función) $f(x,y)$ que ha sido convertida a una señal discreta tanto en coordenadas espaciales como en luminosidad. Esta puede ser considerada como una matriz cuyos índices de renglón y columna identifican un punto (un lugar en el espacio bidimensional) en la imagen y el correspondiente valor de elemento de matriz identifica el nivel de gris en aquel punto (ver Figura 24) [19]. Los elementos de estos arreglos digitales son llamados elementos de imagen o píxeles.

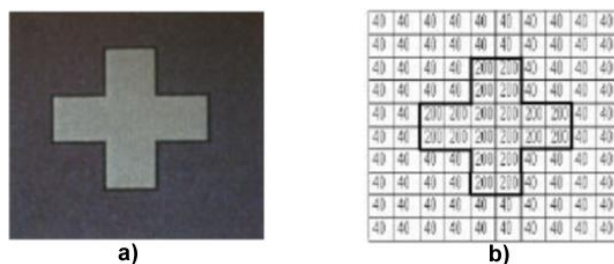


Figura 24: a) Imagen original; b) Estructura matricial de la imagen.

Principalmente, se tiene las siguientes características de una imagen:

- **Resolución**

El concepto de resolución es confuso de entender en imagen digital porque se utiliza refiriéndose a elementos diferentes:

1. La resolución en la impresión de imágenes digitales viene dada por el número de puntos impresos por unidad de superficie.
2. En una imagen digital la resolución depende del número de píxeles por unidad de superficie.
3. La resolución de pantalla de un monitor se expresa con la multiplicación del número de columnas y filas de píxeles. Las más habituales son 800x600 o 1024x768 píxeles de resolución.

Cuando utilizemos el término resolución de la imagen digital nos referimos a la cantidad de píxeles por unidad de superficie. Se suele utilizar como unidad la pulgada cuadrada (una pulgada es igual a 2,54 cm) y así nos encontramos que una imagen puede tener una resolución de 72x72 ppp (píxeles por pulgada), 150 ppp ó 300 ppp [20].

- **Resolución de la imagen**

La calidad de una imagen está directamente relacionada con la resolución que tiene: a mayor resolución mayor calidad de imagen. Si una imagen tiene una resolución de 200 ppp, en cada pulgada cuadrada nos encontramos la cantidad de 200 x 200 píxeles, lo que nos da un resultado de 40.000 píxeles que contienen la información digital de esa imagen. Si esa misma imagen tiene una resolución de 72 ppp, la información digital está contenida en 5.184 píxeles, con lo que la calidad de la imagen será menor: menor resolución, menor información y menor detalle [21].

Para poder comprender mejor este aspecto es necesario diferenciar los conceptos que estamos tratando: tamaño de la imagen, resolución de la imagen y tamaño del archivo.

El **tamaño de la imagen** depende de sus dimensiones (anchura y altura) medidas en píxeles u otras unidades: puntos, picas, milímetros, etc.

La **resolución de la imagen** la expresamos en píxeles por pulgada. A mayor resolución más píxeles caben en una pulgada y menor será el tamaño de los píxeles.

El **tamaño del archivo de la imagen**, como unidad de información digitalizada se expresa en bits. A mayor tamaño de imagen (más píxeles contiene la imagen), mayor tamaño del archivo.

- **La profundidad del color:** La profundidad del color de una imagen se refiere al número de niveles de gris (o número de colores) que se emplea para representar la imagen. Un valor aceptable podría corresponder con una profundidad de color de 8 bits, equivalente a 256 distintos niveles de gris.

3.2.1.3 Píxeles en referencia al Sistema Internacional de Medidas

En la tecnología informática y la imagen digital, un píxel o elemento de imagen es un punto físico en una imagen de trama, o el elemento direccionable más pequeña en un dispositivo de visualización. La dirección de cualquier píxel corresponde a sus coordenadas físicas en la pantalla. La coordenada *X* es la dirección horizontal y la *Y* es la dirección vertical de cualquier píxel o punto direccionable en una pantalla de visualización, es decir, Píxel (*px*) como unidad de medida en diseño es *la unidad física más pequeña de una imagen digital*.

Debido a lo abstracto de esta unidad usualmente en diseño se utiliza en conjunción con otras unidades de medida, como por ejemplo la expresión **píxel por pulgada (ppp)** relacionando el píxel con una dimensión sí cuantificar a como es la pulgada.

La unidad absoluta como *px* tiene un valor diferente dependiendo de la resolución de la pantalla y el tipo de computadora del usuario. Así, normalmente un sistema operativo Windows mantiene una equivalencia de 96px por pulgada y un Macintosh 72px por pulgada.

La conversión de píxeles a centímetros depende del DPI (dots per inch, puntos por pulgada) de los medios de visuales para mostrar una imagen.

DPI significa *Puntos Por Pulgada* e indica el número de puntos impresos (dots) por cada pulgada.

PPI significa *Píxeles Por Pulgada* e indica el número de píxeles por cada pulgada [22].

DPI y PPI son básicamente lo mismo, excepto que PPI (PPP en español) es la designación más correcta. DPI se aplica para algunos tipos de impresoras y métodos de impresión.

Pulgadas: La pulgada es la medida de longitud usada en Estados Unidos, Canadá y Gran Bretaña. La dimensión actual parte de la yarda internacional de 0,9114 m aceptada en 1959. Su expresión correcta en diseño es **in**. Y su correspondencia en milímetros en la siguiente [23]:

$$1 \text{ pulgada (inches, in)} = 25,4 \text{ mm (2,54 cm)}$$

$$1 \text{ pulgada (inches, in)} = 96 \text{ Píxeles (X)}$$

Hay 2,54 centímetros por pulgada, por lo que se sustituye con 96 píxeles por pulgada en la coordenada *X*.

La fórmula es bastante simple:

$$\text{centimetros} = \frac{\text{píxeles} * \text{centímetros por pulgada}}{\text{píxeles por pulgada (DPI)}}$$

$$\text{centimetros} = \frac{\text{píxeles} * 2,54}{96}$$

Entonces tenemos:

1 píxel (px) = 0,0264583 cm

1 píxel (px) = 0,000264583 m (0,264583 mm)

3.2.1.4. Digitalización de Imágenes

Generalmente los datos que describen una imagen gráfica son representados como un “Bitmap Data”. Un gráfico de tipo Bitmap Data es formado por un conjunto de valores numéricos especificando los colores de cada uno de los píxeles.

Un formato de archivo gráfico o imagen es el formato en el cual un Bitmap Data es grabado en un archivo. Los formatos de archivos gráficos más utilizados para objetivos computacionales son:

- BMP: Microsoft Windows Bitmap.
- PNG: Portable Network Graphic.
- TIFF: Tag Image File Format.
- JPG: Joint Picture Graphic;
- Entre otros tipos de formatos como el Portable Bitmap Utilities (PPM, PBM, PGM, PNM).

El espacio de almacenamiento que requiere una imagen depende del ancho de píxeles que contiene. Al especificar el ancho de píxel o “bit depth”, que es el tamaño de los píxeles en bits (o bytes), se determinan el número de colores que un píxel representa. Un píxel de 1 bit puede tener 2 colores, un píxel de 4 bits puede contener 16 colores y así sucesivamente hasta tener 16.000.000 de colores en un píxel de 24 bits [24].

3.2.1.5. Formatos numéricos

Numéricamente hablando, es posible trabajar con datos que tienen distintas escalas según el tipo de formato numérico que soporta el software utilizado. En el Software MATLAB [25] se cuenta con cuatro tipos de datos, a saber:

- 8 bits: establece un arreglo de número enteros en el rango [0; 255], con un total de valores: $2^8 = 256$

- 16 bits: establece un arreglo de número enteros en el rango [0; 65535], total de: $2^{16} = 65536$ valores.
- double: contiene un arreglo de valores enteros y no enteros en el rango [0; 1].
- logical: contiene un arreglo de ceros (0) y unos (1) únicamente.

El valor total de colores que un píxel se obtiene elevando el número dos (2) a la potencia correspondiente al número de “bits”, por ejemplo: $2^8 = 256$ niveles $2^{12} = 4.096$ niveles, $2^{16} = 65.536$ niveles, etc. A mayor cantidad de “bits” por píxel mayor será el número requerido para el almacenamiento en cantidad de Mega-bytes.

Debido a que el formato numérico de 8 bit es el más utilizado entonces la “profundidad del bit” más común será $8_{(\text{bit})} \times (\text{píxel}) = 256_{\text{Colores}}$, aquí cada píxel contiene valores en el rango de 0 a 255 como un índice en un gama de 256 colores [24].

3.2.1.6. Clasificación de imágenes digitales

Dependiendo del rango de los valores que pueda tomar cada píxel se distinguen los siguientes tipos de imágenes:

- **Imágenes binarias:** El rango está formado por los valores negro o blanco [0 1] únicamente. En la *Figura 25* [17] se muestra una imagen de este tipo.

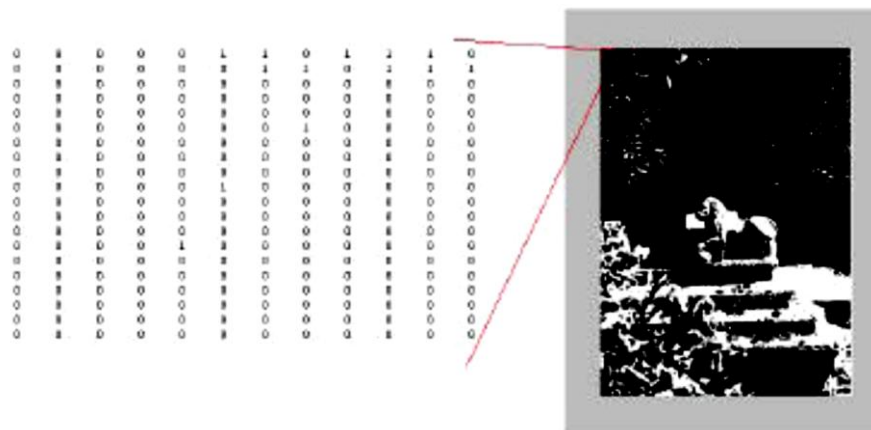


Figura 25: Imagen binaria.

- **Imágenes de intensidad:** También conocidas como imágenes en escala de grises (ver *Figura 26*) [17], existen hasta 256 niveles de grises, por lo que su rango se encuentra entre [0-255].

26	22	35	31	64	171	108	94	162	194
88	18	13	17	33	102	179	212	125	207
28	10	25	75	88	56	48	82	71	27
34	11	39	69	48	26	22	6	41	28
28	22	47	10	31	20	27	24	20	29
21	27	28	34	18	80	83	183	18	41
22	20	21	27	31	17	32	30	20	24
15	25	33	25	21	23	28	34	92	23
48	24	28	36	21	53	1	37	25	20
23	46	54	21	188	171	117	9	22	23
67	45	33	22	84	96	73	37	38	26
65	102	48	22	1	25	28	42	17	29
42	120	48	7	46	75	28	35	17	35



Figura 26: Imagen en escalas de grises.

- **Imágenes en color:** Todo color se puede componer a partir de tres componentes básicas. El contenido de cada píxel de la imagen es una terna de valores, un valor por cada componente de color básico (ver Figura 27) [17].

Azul													
31	27	28	36	27	175	196	85	164	195	158	51	25	21
41	35	18	23	43	107	178	214	125	209	220	188	95	20
29	19	46	84	75	44	53	88	75	101	113	188	65	27
45	22	48	81	59	35	31	32	46	51	13	14	26	25
41	36												
Verde													
23	22	29	31	31	88	171	182	88	188	203	130	78	
27	55	38	16	12	17	43	188	179	218	428	212	238	187
13	25	17	8	38	74	87	38	81	88	78	121	110	107
30	25	32	9	37	88	48	28	23	9	44	29	11	22
12	50	28	23	27	88	33	22	29	18	32	42	38	38
49	50	23	28										
Rojo													
50	67	29	31	30	25	56	32	62	165	165	77	154	155
21	109	28	30	38	22	15	18	35	97	169	204	113	129
19	40	48	41	23	11	27	74	82	31	41	75	63	69
51	26	38	94	23	11	27	74	82	31	41	75	63	69
80	26	38	94	23	11	27	74	82	31	41	75	63	69
37	27	79	96	26	21	45	53	27	19	20	17	25	53
55	29	42	104	15	21	20	27	32	29	56	128	12	26
82	50	28	48	15	21	14	28	29	11	25	65	12	21



Figura 27: Imagen en color.

3.2.2. Espacios de Color

Para la implementación del proyecto se utilizan imágenes en color y sus características, por lo que se aporta una pequeña introducción sobre los distintos espacios de color que pueden ser utilizados en el procesamiento de imágenes [17].

El color es una característica importante para la clasificación y descripción de objetos en la visión por computadora. Existen diferentes modelos de color como RGB, HSV, YCbCr, XYZ, CIE(Lab), se describirá solo el modelo RGB.

3.2.2.1. Modelo RGB

El modelo RGB (*Red, Green, Blue*), maneja 3 canales o planos, uno para el rojo, uno para el verde y uno para el color azul, la imagen resultante tiene tonalidades que surgen de las combinaciones entre los colores primarios. En esta codificación, por cada píxel se necesitan 24 bits, es decir 3 bytes, codificando cada color primario con 1 byte (*8 bits*) que pueden tomar 256 valores con un rango de 0 a 255. Los píxeles RGB son tridimensionales, $\text{píxelRGB} = f(R,G,B)$, donde el valor de f es la intensidad del color rojo, verde y azul en las coordenadas (x,y) del píxelRGB. De esta forma los colores se representan en coordenadas cartesianas dentro de un cubo unitario (ver *Figura 28*) [25].

Actualmente muchas cámaras a color utilizadas para adquirir imágenes digitales, utilizan el formato RGB. Esto convierte al modelo RGB en un modelo de gran importancia para el procesamiento de imágenes.

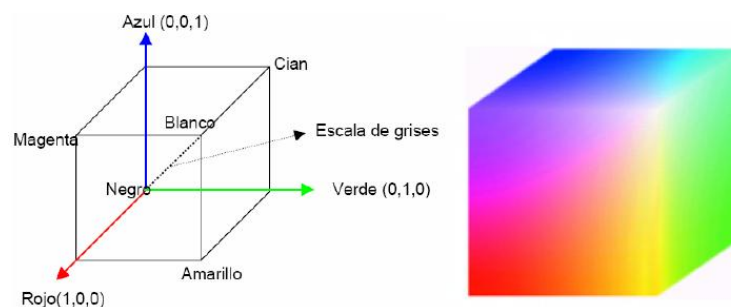


Figura 28: Distribución de colores en el cubo RGB

3.2.3. Detección de regiones basada en umbrales

Si se analiza el histograma de una imagen en niveles de gris que corresponde a una imagen con un objeto claro sobre fondo oscuro como la *Figura 29* [25], de tal forma que los píxeles del objeto y del fondo tienen los niveles de gris agrupados en dos modos dominantes, como se puede ver en el histograma de la *Figura 30* [25]; una forma de extraer los objetos del fondo es elegir un umbral " T " que separe dichos grupos. Cualquier punto (x, y) para el que $f(x, y) > T$ se asigna al objeto, en caso contrario al fondo; en otras palabras, se pueden separar los píxeles en dos regiones si se asigna con un valor de intensidad cero a todos los píxeles cercanos o iguales a cero (*fondo negro*) y con 255 al resto de los valores (*objeto*) [25].

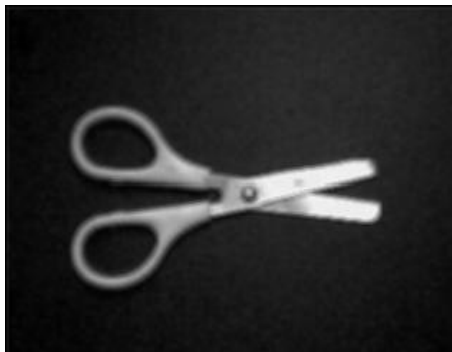


Figura 29: Imagen en niveles de gris

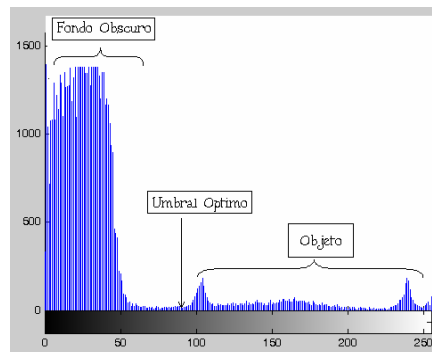


Figura 30: Histograma

Analizando el histograma, se deduce experimentalmente que un umbral óptimo que segmenta a la imagen en dos regiones es = 97. La primera región correspondiente al fondo y la otra al objeto. Al utilizar un umbral más pequeño, los puntos claros correspondientes al fondo se unen con el objeto a segmentar: por el contrario, si se toma un umbral más grande se corre el riesgo de tomar zonas del objeto que son un poco oscuras y unirse con el fondo (ver Figura 31) [25].



Figura 31: A: umbral óptimo 97, B: umbral = 45, C: umbral = 140

3.2.3.1. Método de segmentación de Otsu

La importancia del método de Otsu radica en que es automático, es decir, no necesita supervisión humana ni información previa de la imagen antes de su procesamiento. Este método se emplea cuando hay una clara diferencia entre los objetos a extraer respecto del fondo de la escena. En este método, se elige el umbral óptimo maximizando la varianza entre clases mediante una búsqueda exhaustiva.

Al aplicar un umbral T , la imagen en escala de grises, $f(x,y)$, quedará binarizada; etiquetando con "1" los píxeles correspondientes al objeto y con "0" aquellos que son del fondo [25].

3.2.4. Detección de bordes

Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc. [26].

Esto se realizará con la función **edge**, este efecto devuelve una matriz binaria compuesta de “unos” los cuales indican los bordes y de “ceros” que indican la parte negra o hueca de la imagen; **edge** permite definir los bordes como:

- Lugares donde la primera derivada de la intensidad es mayor en magnitud, que un umbral.
- Lugares donde la segunda derivada de la intensidad tiene un cruce por cero.

Estas derivadas forman parte importantísima para cumplir con la detección de bordes y con la ayuda de las siguientes técnicas: “*canny, sobel, prewit, roberts, zerocross, log*”.

En MatLab la función que detecta bordes es **edge** [27].

$$BW = edge(I, 'sobel', thresh, direction)$$

El parámetro *thresh* o *threshold* indica el umbral de binarización. Si se elige el umbral de binarización, hay que ser consciente de que la función *edge* normaliza la imagen antes de procesarla, llevándola al intervalo [0,1]. También divide las máscaras empleadas en los filtros por un factor, siendo de $\sqrt{2}$ para Roberts, 8 para Sobel y 6 para Prewitt [26].

3.2.4.1. Operador de canny

Básicamente, el filtro de detección de bordes de Canny consiste en un proceso multifase, en el que el usuario deberá introducir tres parámetros básicos: una desviación estándar y dos valores umbral. El resultado será una imagen binaria, en la que aparecerán en negro los píxeles correspondientes a los bordes y en blanco el resto de píxeles [28], como se puede apreciar en la *Figura 32* [29].

El objetivo de Canny era descubrir el algoritmo óptimo para la detección de bordes. Un detector óptimo significará una buena detección, el algoritmo debe marcar tantos bordes reales como sea posible, una buena localización, los bordes marcados deben estar lo más cerca posible del borde en la imagen real, y por último una mínima respuesta, es decir, un borde dado debe ser marcado sólo una vez y donde sea posible el ruido presente en la imagen no debería crear falsos bordes [29].



Figura 32: Imagen original e Imagen con detección de Canny

3.2.4.2. Operadores de Prewitt, Sobel y Frei-Chen

Los tres operadores pueden formularse de forma conjunta con las siguientes máscaras de convolución mostradas a continuación en la Figura 33 [26].

<p>Gradiente fila</p> $\frac{1}{2+K} \begin{bmatrix} 1 & 0 & -1 \\ K & 0 & -K \\ 1 & 0 & -1 \end{bmatrix}$	<p>Gradiente columna</p> $\frac{1}{2+K} \begin{bmatrix} -1 & -K & -1 \\ 0 & 1 & 0 \\ 1 & K & 1 \end{bmatrix}$
--	---

Figura 33: Máscaras de Convolución

En el operador Prewitt ($K=1$) se involucran a los vecinos de filas/columnas adyacentes para proporcionar mayor inmunidad al ruido.

El operador Sobel ($K=2$), se supone que es más sensible a los bordes diagonales que el de Prewitt aunque en la práctica hay poca diferencia entre ellos.

Frei-Chen ($K=2$), el gradiente es el mismo para bordes verticales, horizontales y diagonales.

En la *Figura 34* [26] se muestra los resultados al aplicar los operadores mencionados.



Figura 34: A) Imagen Original, B) Filtro Sobel, C) Filtro Canny.

3.2.5. Operaciones morfológicas básicas

Se utilizan para eliminar pequeños puntos aislados que no pertenecen al objeto en cuestión mediante ciertas técnicas de morfología matemática como son: *erosión*, *dilatación*, *apertura* y *cerradura*, en MatLab corresponden a las funciones *imrode*, *imdilate*, *imopen* e *imclose* [27].

Dilatación.- La dilatación $A \oplus B$ es el conjunto de puntos de todas las posibles sumas de pares de elementos uno de cada conjunto A y B :

$$A \oplus B = \bigcup_{\beta \in B} (A + \beta)$$

Por ejemplo, dados A y B en forma de vectores para los valores correspondientes a los unos cuyas coordenadas se definen con respecto a $[0,0]$ como se muestra en la *Figura 35* [25].

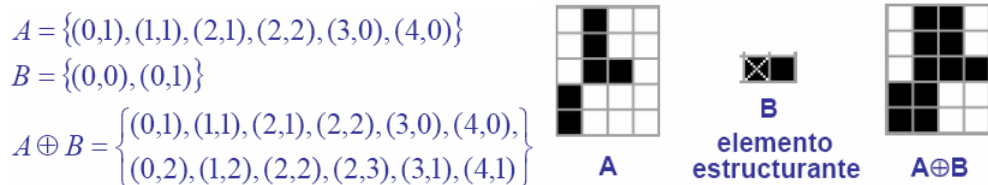


Figura 35: Elementos Estructurales A y B

En los conjuntos A y B , se considera que A es la imagen y B es el elemento estructural. El elemento estructural es en morfología matemática lo que la máscara o

núcleo de convolución⁶ es en los filtros lineales. Los elementos estructurales más comunes son los conjuntos que están 4-conectados(N₄), y 8-conectados(N₈), que se ilustran en la *Figura 36* y *Figura 37* [25].



Figura 36: Elemento Estructural N₄



Figura 37: Elemento Estructural N₈

La operación de dilatación hace que los objetos se expandan y la cantidad o la forma en que crezcan depende del elemento estructurante, como en la *Figura 38* [25].

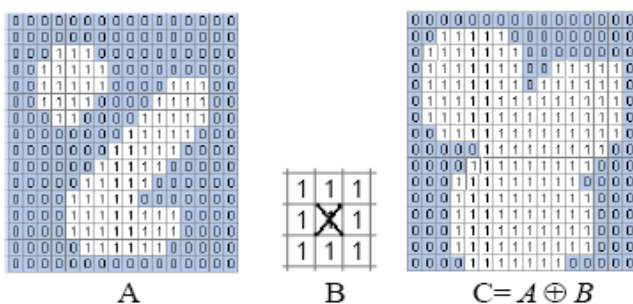


Figura 38: Dilatación: A) imagen original, B) elemento estructurante N₈, C imagen dilatada

Erosión.- Combina dos conjuntos usando la resta vectorial y es el dual de la dilatación:

$$A \ominus B = \bigcap_{\beta \in B} (A - \beta)$$

En otras palabras, son los puntos A para los cuales todas las posibles traslaciones definidas por B también están en A (ver *Figura 39* y *Figura 40*) [25].



Figura 39: Erosión con un elemento estructurante de 1x2

⁶ Convulsión: En matemáticas y, en particular, análisis funcional, una convulsión es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g .

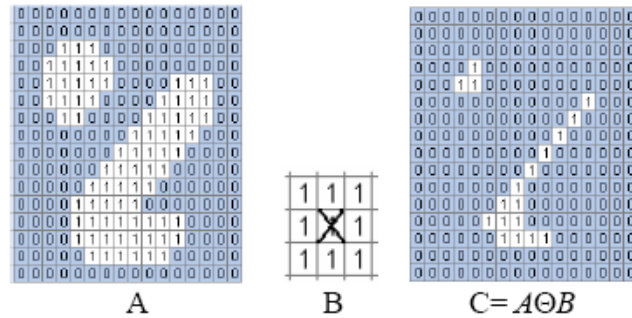


Figura 40: Erosión: A) imagen original, B) elemento estructurante de 3x3, C) imagen erosionada.

Apertura.- La apertura de A por B es una erosión de A por B seguida de la dilatación utilizando en ambas operaciones el mismo elemento estructurante.

$$A \circ B = A \ominus B \oplus B$$

Se utiliza para alisar el contorno de un objeto, rompe uniones pequeñas uniones entre objetos y elimina componentes ruidosas (*puntos blancos*). En la Figura 41 [25], se utiliza la apertura como un filtro de tamaño, de esta manera se eliminan los objetos pequeños menores a un elemento estructurante y también elimina el ruido contenido en la imagen.

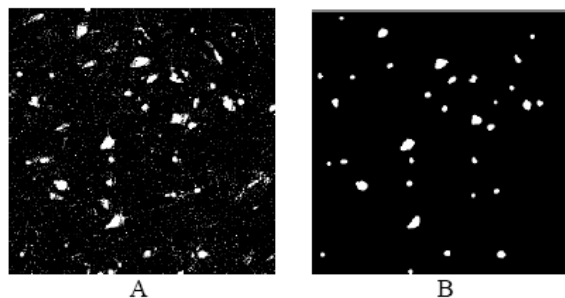


Figura 41: Apertura: A) imagen original, B) imagen resultante después de aplicar la apertura.

Cerradura.- La cerradura de A por B es la dilatación de A por B seguida de la erosión utilizando en ambas operaciones el mismo elemento estructurante.

$$A \bullet B = A \oplus B \ominus B$$

La Cerradura tiende a eliminar pequeños agujeros (*puntos negros*) del objeto, fusiona pequeños rompimientos y rellena pequeñas entradas al objeto como se muestra en la Figura 42 [25].

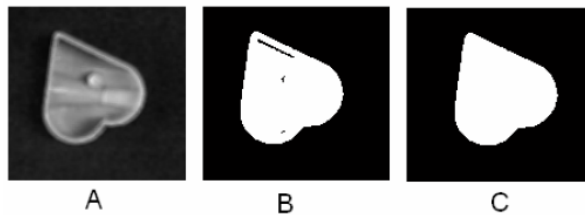


Figura 42: Cerradura: A) imagen en niveles de gris, B) imagen en blanco y negro
C) imagen resultante después de aplicar cerradura.

3.2.6. Propiedades geométricas de los objetos

Con las propiedades básicas (área, volumen, perímetro, centroide, etc.) de los objetos de diferentes figuras como círculos, rectángulos, cuadrados, triángulos, etc. [27], se puede plantear una solución para el reconocimiento de dichos cuerpos, lográndose esto a través de las diferentes funciones que ofrece MATLAB, como se procede continuación:

```
[M, NUM]=bwlabel(BW2, 8);  
stats = regionprops(M, 'all');
```

Donde M indica la matriz de BW2 que contiene la última imagen pre-procesada, donde los ceros indican el fondo de la imagen y diferentes conjuntos de números indican los distintos objetos que esta contiene [30].

NUM indica los números totales de los objetos que se encuentran en la figura, en este ejemplo se encuentran siete objetos. El numero 8 o 4 indica la conexión cercana de 8 o 4 objetos, o a su vez ninguno el cual indica por default el número 8.

La función **regionprops** se puede aplicar para diferentes aplicaciones como puede ser área, perímetro, centroide, etc.

A continuación definiciones de algunas propiedades.

- ✓ **Área:** Calcula el área en píxeles cuadrados de la región.
- ✓ **Centroide:** Posición del centroide de la región de algún objeto en mención.
- ✓ **Perímetro:** Calcula el perímetro (contorno) en píxeles lineales de la región.

CAPÍTULO IV: METODOLOGÍAS PARA LA CONSTRUCCIÓN DE SISTEMAS BASADOS EN CONOCIMIENTO

4.1. Antecedentes

Decía Nietzsche que "la riqueza más preciosa son los métodos". En este sentido, y para paliar una carencia notoria en la tecnología de la Ingeniería del Conocimiento, se presentan metodologías para construir sistemas expertos o sistemas basados en conocimiento (SBC) [31].

Entendemos por Metodología la aplicación coherente de un método, es decir, conjunto de operaciones ordenadas con objeto de alcanzar un resultado. Así pues, con las metodologías que aquí se presenta, se pretende definir, de una manera ordenada, las fases y etapas que debe cubrir un Ingeniero del Conocimiento, con el objeto de construir un SBC.

Hay que señalar, que estas metodologías son flexible de modo que siempre que el problema sea adecuado y esté perfectamente documentado, es posible saltar algún paso o etapa e incluso, aunque muy excepcionalmente, alguna fase. Es decir, las metodologías se presentan más como una facultad que se admite, que como una condición que se imponga. En otros términos, hay que verla más como guía [31].

En los años 80 se crearon algunos métodos que servían para modelar SBC y que se basaban en el nivel de conocimiento de Alan Newell [New82]. Este nivel permite describir el razonamiento en términos de los objetivos a ser alcanzados, las acciones necesarias para cumplir estos objetivos y el conocimiento necesario para ejecutar dichas acciones.

La ingeniería del conocimiento (IC) es un conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización del conocimiento (entendimiento, inteligencia o razón natural). Esta disciplina moderna puede ayudar a construir aplicaciones y sistemas orientados al aprendizaje apoyándonos con metodologías instruccionales y con tecnologías de computación y de telecomunicaciones.

En la TABLA I [32] se presentan los métodos y metodologías más usados o reconocidos para el desarrollo de SBC.

TABLA I: CUADRO COMPARATIVO ENTRE METODOLOGÍAS PARA SISTEMAS EXPERTOS

Características	METODOLOGÍA DE BUCHANAN (1983)	METODOLOGÍA MIKE (Model-based and Incremental Knowledge Engineering – IC Incremental y Basada en el Modelado) (1993)	METODOLOGÍA COMMONKADS (2000)	METODOLOGÍA IDEAL (I=Iniciar, D=Diagnosticar, E=Establecer, A=Actuar, L=Aprender) (Gómez et al, 1997)
Sistema Experto	Metodologías empleadas para el desarrollo de sistemas expertos.			
Metodología de Ciclo de Vida	Modelo de CASCADA	Modelo DESARROLLO INCREMENTAL y ESPIRAL	Modelo ESPIRAL	Modelo en tres dimensiones ESPIRAL TRONCO CONICO
Importancia	Fue la primera metodología para el desarrollo de los SBCs, dedicados al mundo de la industria	La importancia de MIKE radica en el cambio que propone pasar de las actividades de extracción del conocimiento hasta la fase de evaluación del sistema final.	Es una metodología completa para el desarrollo de SBCs. Cubre la gestión del proyecto, el análisis organizacional y los aspectos relativos a las Ingenierías de Software y Conocimiento relacionado con el desarrollo de los SBCs	Se ajusta a la tendencia del software actual, esto es: - Ser Reutilizable - Ser Integrable - Poseer Requisitos Abiertos - Diversidad de Modelos Computacionales.
Fases	Propone 5 fases: 1. Definición 2. Conceptualización 3. Formalización 4. Implementación 5. Verificación	Propone 4 fases: 1. Adquisición de conocimiento 2. Diseño 3. Implementación 4. Mantenimiento	Propone 4 etapas: 1. Planificación. 2. Monitorización 3. Revisión 4. Valoración de riesgos. A la vez propone dentro de cada uno los siguientes modelos: - Modelo de organización. - Modelo de tareas - Modelo de agentes - Modelo de conocimiento - Modelo de comunicaciones - Modelo de Diseño.	Propone 5 fases: Fase I. Identificación de la tarea. Fase II. Desarrollo de los prototipos. Fase III. Ejecución de la construcción del sistema integrado. Fase IV. Actuación para conseguir el mantenimiento perfectivo. Fase V. Lograr una adecuada transferencia tecnológica



Ventajas	<ol style="list-style-type: none">1. Identificación del proceso de adquisición de conocimiento como un cuello de botella de un SBC.2. Procesos bien definidos para poder pasar a las siguientes etapas.	<ol style="list-style-type: none">1. Se centra en la integración de prototipos y el desarrollo incremental.2. Representación sencilla del CommoKADS.	<ol style="list-style-type: none">1. Abarca desde los análisis iniciales que sirve para identificar el problema y establecer la idoneidad de la solución, hasta la implementación del mismo proporcionando un marco de trabajo donde llevar a cabo la gestión del proyecto.2. Orientado a la reutilización, tanto en el nivel de tareas como en el de conceptualización del dominio.	<ol style="list-style-type: none">1. Los requisitos están sometidos a constantes cambios y por ende el sistema también, por lo que como resultado se obtiene un sistema en constante evolución por lo que puede considerarse como un prototipo en constante perfeccionamiento.2. El desarrollo del sistema se basa en la filosofía del modelo en espiral en donde cada fase finaliza con un prototipo que conduce a la fase siguiente del ciclo.3. El mantenimiento perfecto involucra la incorporación sistemática de nuevos conocimientos (adquirido por el uso del sistema) típica cualidad de un SE. Los ejes de la base del cono representan el costo y el tiempo.4. Hace de gran importancia de los métodos de adquisición de conocimiento (redes bayesianas, redes neuronales, reglas, etc.)
Desventajas	<ol style="list-style-type: none">1. Inflexible, la configuración de las distintas etapas no se podía adaptar a situaciones concretas.2. Los requisitos iniciales deben estar bien definidos al inicio del proceso, lo que hacia difícil adaptarse a los cambios de dichos requisitos.	<ol style="list-style-type: none">1. Se centra sólo en el desarrollo del modelo de conocimiento.	<ol style="list-style-type: none">1. Por ser una metodología completa es muy amplia y compleja, en el peor de los casos difícil de controlar.2. Una metodología que tiene amplio desarrollo en Europa, no tanto en esta parte de América.3. Las herramientas de sistemas expertos, son dejadas de lado (redes bayesianas, redes neuronales, reglas, etc.)	<ol style="list-style-type: none">1. Hace una descomposición más simple de los niveles estratégicos y tácticos.2. Metodología en proceso de seguir implementándose.

Relación Ingeniería de Software con Ingeniería del Conocimiento	NO	SI	SI	SI
Diseño del Software	SI	SI	SI	SI
Pruebas del Software	SI	NO	SI	SI
Orientado a Objetos	NO	NO	SI	SI

Dadas las características del sistema que se desea construir, resulta adecuado adoptar las Metodologías IDEAL y CommonKads (CommonKads-RT) para el desarrollo del Sistema Basado en Conocimiento (SBC) de este trabajo de Titulación.

4.2. Metodología IDEAL

La metodología IDEAL fue propuesta por Gómez (1997) [32], en donde propone un ciclo de vida en espiral en tres dimensiones, y se ajusta a la tendencia del software actual, esto es:

- Ser Reutilizable.
- Ser Integrable.
- Poseer Requisitos Abiertos.
- Diversidad de Modelos Computacionales.

El desarrollo del sistema se basa en la filosofía del modelo en espiral en donde cada fase finaliza con un prototipo que conduce a la fase siguiente del ciclo. El mantenimiento perfecto involucra la incorporación sistemática de nuevos conocimientos (adquiridos por el uso del sistema).

Los requisitos están sometidos a constantes cambios y por ende el sistema también, por lo que como resultado se obtiene un sistema en constante evolución y puede considerarse como un prototipo en constante perfeccionamiento, mediante el agregado de nuevos marcos compuestos, mediante nuevas técnicas de descomposición del problema, mediante nuevas formas de documentación o estándares a los que debe ajustarse [33].

En la *Figura 43* se muestran las fases de desarrollo de la metodología IDEAL [34].

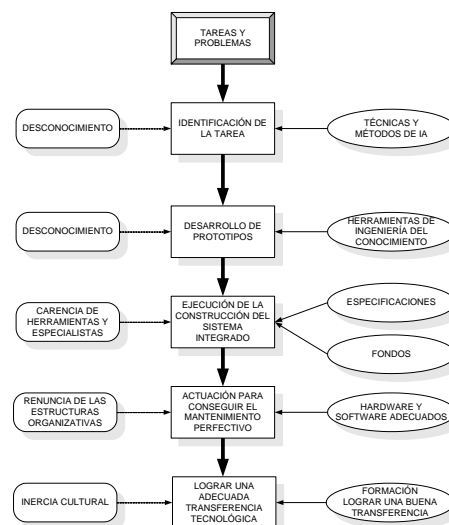


Figura 43: Fases de la metodología IDEAL

Se expone a continuación las fases y etapas que componen la metodología I.D.E.A.L. y que guiarán el desarrollo del sistema [32]:

TABLA II: DESCRIPCIÓN DE LAS FASES DE LA METODOLOGÍA IDEAL

FASE	ETAPA	DESCRIPCIÓN
<p>I. Identificación de la tarea</p>	<p>I.1. Plan de requisitos y adquisición de conocimientos</p>	<p>Lo primero que debe hacer el IC es tratar de identificar las necesidades del cliente describiendo, para ello los objetivos del sistema. Estos objetivos pueden ser: finalidades, de carácter filosófico; fines, de carácter cualitativo. Y, metas u objetivos a plazo fijo, de carácter cuantitativo que son las más interesantes. Además, debe determinar qué informaciones se van a obtener y suministrar funcionalidades a exigir y requisitos necesarios para todo ello, etc. Los parámetros fundamentales de este plan de requisitos son de acuerdo con Pressman, los siguientes:</p> <ul style="list-style-type: none"> • Fines específicos y generales del sistema. • Funcionamiento y rendimiento requeridos. • Fiabilidad y calidad. • Limitaciones de coste/tiempo. • Requisitos de fabricación. • Tecnología disponible. • Competencia. • Ampliaciones futuras. <p>Para confeccionar el plan de requisitos es necesario comenzar con la adquisición de conocimientos, entrevistándose con directivos, expertos y usuarios. Sin embargo, la adquisición profunda se llevará a cabo en la fase II.</p>
	<p>II.2. Evaluación y selección de la tarea</p>	<p>Esta etapa, que conforma el «estudio de viabilidad», se lleva a cabo realizando una evaluación de la tarea desde la perspectiva de la IC, y luego cuantificando dicha evaluación para</p>



FASE	ETAPA	DESCRIPCIÓN
		<p>ver qué grado de dificultad presenta la tarea. Existen tal y como se verá en los próximos capítulos, varias formas de llevar a cabo dicha evaluación. Esta etapa es fundamental para evitar a priori fallos detectados en la aplicación práctica de esta tecnología.</p>
	III.3. Definiciones de las características de la tarea	<p>Aquí, se establecen y eventualmente definen las características más relevantes asociadas con el desarrollo de la aplicación. En particular se dan:</p> <p>Una definición lo más formal posible de la aplicación, desde el punto de vista del sistema. Es decir, se pasa de una descripción informal de los requisitos del usuario a una especificación técnica completa emitida por el IC. Respecto a esto, hay que llevar a cabo una especificación inicial de los siguientes tipos de requisitos:</p> <ul style="list-style-type: none">• Funcionales: tipos de informaciones (datos, noticias y conocimientos) que se van a tratar operaciones a realizar sobre ellos, salidas deseadas, etc.• Operativos o de funcionamiento: estáticos que no varían con el tiempo y dinámicos que varían con el tiempo.• De interfaz: de usuarios, con otros productos y sistemas, etc.• De soporte: plataforma hardware y software de base requerido.• Criterios de éxito: que básicamente consisten en identificar las necesidades reales de los usuarios finales y decisores del sistema propuesto y definir el grado de satisfacción de dichas necesidades que debe cumplimentar el sistema.• Casos de prueba o juego de ensayo: que permita validar tanto el grado y la calidad de la experiencia del experto como las prestaciones del SE obtenido.• Recursos para desarrollar el SE: dentro de estos recursos, hay que especificar tanto los materiales (económicos, hardware, software, etc.), como humanos (expertos, IC, ingenieros

FASE	ETAPA	DESCRIPCIÓN
		<p>del software, programadores convencionales y de IA, etc.).</p> <ul style="list-style-type: none"> • Análisis de costes/beneficios y evaluación de riesgos: desglosados por conceptos de gastos (personal, material y varios) tipos de beneficios (tangibles, intangibles, concomitantes) y clases de riesgos (institucionales, económicos, estratégicos, tácticos, a corto, medio y largo plazo, etc.). • Hitos y calendario: que establece el plan de desarrollo del sistema, así como el programa para llevarlo a cabo. Realmente esta etapa configura la especificación del sistema. Mientras que en la etapa I.1, plan de requisitos, se describen unas mini-especificaciones que sirven de base para la evaluación de la tarea que se lleva a cabo en la etapa I.2; en la etapa I.3 se completa la especificación con los conocimientos iniciales que se tienen del sistema. <p>Con la definición de esta fase, los implicados en la misma, esto es, los IC, los expertos, usuarios y directivos, consiguen perfilar satisfactoriamente el ámbito del problema; definir coherentemente sus funcionalidades, rendimiento, e interfaces; analizar el entorno de la tarea y del riesgo de desarrollo del SE. Todo ello hace que el proyecto se justifique, y asegura que los IC y los clientes tengan la misma percepción de los objetivos del sistema.</p> <p>En cualquier caso, siempre hay que tener presente que las especificaciones iniciales de los SBC suelen ser inciertas por incompletas, imprecisas, inconsistentes o contradictorias por lo que su obtención real y completa exigirá el desarrollo de distintos prototipos.</p>
<p>II: Desarrollo de los distintos prototipos</p>	<p>II.1. Concepción de la solución:</p>	<p>Esta etapa tiene como objetivo producir un diseño general del sistema prototipo. Inicialmente, el IC y el experto estudian las especificaciones parciales del sistema y el plan</p>



FASE	ETAPA	DESCRIPCIÓN
		del proyecto obtenidos en la fase anterior y, en base a ellos, producen un diseño general.
	II.2. Adquisición y conceptualización de los conocimientos:	Aunque la adquisición de conocimientos es una actividad que impregna toda la INCO, desde que se inicia el estudio de viabilidad hasta que finaliza el uso del SE desarrollado, es en esta etapa donde adquiere todo su esplendor. Esta adquisición, en sus dos facetas de extracción de los conocimientos públicos de sus fuentes (libros, documentos, manuales de procedimientos, etc.) Y la educación ⁷ de los conocimientos privados de los expertos, se alterna cíclicamente con la etapa de conceptualización para modelizar el comportamiento del experto. En la conceptualización se plantean mapas conceptuales que permiten entender el ambiente o dominio del problema a partir de la información obtenida en la etapa de adquisición.
	II.3. Formalización de los conocimientos	Esta etapa presenta dos actividades fundamentales. Por un lado, se deben seleccionar los formalismos para representar en la máquina los conocimientos que conforman la conceptualización obtenida en la etapa anterior, y, por otro, se debe realizar un diseño detallado del SE. La formalización o representación interna de los conocimientos, se encuentra íntimamente ligada con los tipos de conocimientos más apropiados para su representación y las herramientas disponibles en su desarrollo. En lo que concierne a la actividad de diseño detallado del sistema, baste decir aquí que consiste en una estructura modular del sistema

⁷ Educación: Acción y efecto de educir, que significa obtener un resultado por medio de la relación o encontrar significado a algo relacionando a través de analogías de la información.

FASE	ETAPA	DESCRIPCIÓN
		<p>que incorpora todos los conceptos que participan en el prototipo. Esta actividad debe desarrollar la arquitectura general del prototipo, especificada en la etapa II.1, concepción de la solución. Es decir, en esta actividad hay que establecer los módulos que definen el motor de inferencias, la base de conocimientos, interfaces (de usuario y a otros sistemas), etc.</p>
	<p>II.4. Implementación</p>	<p>Si en la etapa anterior se seleccionó una herramienta de desarrollo adecuado y el problema se ajusta a ella y viceversa, la implementación es inmediata y automática. En otro caso, es necesario programar, al menos, parte del SBC. Pero esta cuestión es exactamente igual a cualquier otro software, con las dificultades y problemas que implican cualquier implementación.</p> <p>En todo caso, hay que dejar constancia aquí de que el uso de herramientas de desarrollo, a pesar de las facilidades que aportan, presenta algunos inconvenientes, en absoluto despreciables, como son:</p> <ul style="list-style-type: none"> • Dependencia. El prototipo construido queda ligado a la herramienta, sin que se genere un ejecutable independiente, limitando con ello su transportabilidad. • Eficiencia. Al quedar incorporada la herramienta al sistema, éste ocupa mucho espacio con una herramienta de la que sólo se utiliza una parte muy pequeña. • Gran tamaño, complejidad y coste. Por ser las buenas herramientas: grandes, complejas (tanto de aprender, como de manejar) y caras.
	<p>II.5. Validación y evaluación.</p>	<p>La fiabilidad de los resultados es, tal vez, el punto más sensible de todo SE y por tanto su punto crítico. Se trata de una de las tareas más difíciles</p>



FASE	ETAPA	DESCRIPCIÓN
		<p>dado que estos sistemas están contruidos para contextos en los que las decisiones son, en gran medida, discutibles. Sin embargo, existen cada vez más técnicas que permiten realizar esta validación de una forma razonablemente y satisfactoria. Para ello, se deben realizar las siguientes acciones, todas ellas independientes entre sí; y sin embargo, totalmente complementarias.</p> <p>Además, aquí se examina detalladamente la interfaz de usuario para ver si se ajusta a los deseos de expertos y usuarios finales tanto en su ergonomía como en las explicaciones que proporciona.</p>
	II.6. Definición de nuevos requisitos, especificaciones y diseño	<p>Como ya se ha mencionado, los SBC se construyen de forma incremental, generando primero un prototipo de investigación, que se convierte en un prototipo de campo para, finalmente, resultar un prototipo de operación. Esta etapa corresponde con la definición de los requisitos, especificaciones y diseño del siguiente prototipo, que para ser construido deberá pasarse, de nuevo, por las etapas II.1 a II.5. Esta fase acaba con la obtención del sistema experto completo.</p>
III: Ejecución de la construcción del sistema integrado	III.1. Requisitos y diseño de la integración con otros sistemas	<p>Consiste en el estudio y diseño de interfaces y puentes con otros sistemas hardware y software.</p>
	III.2. Implementación y evaluación de la integración	<p>Su fin es desarrollar, utilizando técnicas de IS, los requisitos de la etapa anterior. En, esta etapa se integra el SE con los otros sistemas hardware y software, para conseguir un sistema final.</p>

FASE	ETAPA	DESCRIPCIÓN
	III.3. Aceptación por el usuario del sistema final	Es la prueba última de aceptación por los expertos y usuarios finales, que debe satisfacer todas sus expectativas y exigencias, tanto en lo concerniente a su fiabilidad como eficiencia.
IV: Actuación para conseguir el mantenimiento perfecto	IV. I. Definir el mantenimiento del sistema global	Esta etapa emplea las técnicas de IS, definiendo el mantenimiento que se llevará a cabo igual que en cualquier otro tipo de sistema.
	IV.2. Definir el mantenimiento de las bases de conocimientos	Existen diversas técnicas para el mantenimiento de bases de conocimiento que se estudiarán en capítulos posteriores.
	IV.3. Adquisición de nuevos conocimientos	Diseñar protocolos para que cuando aparezcan nuevos conocimientos, en especial los correosos, puedan captarse y registrarse. Además habrá que establecer métodos para actualizar el sistema incorporando los conocimientos adquiridos.
Fase V: Lograr una adecuada transferencia tecnológica	V. I. Organizar la transferencia tecnológica	Organizar la transferencia tecnológica meticulosamente, mediante entrenamiento en sesiones de tutoría entre los diseñadores y los usuarios, que sirvan tanto para explicar el manejo del propio sistema como para manejar y entender la documentación del mismo.
	V.2. Completar la documentación del sistema	Completar la documentación del sistema desde el dossier técnico al manual del usuario, que deben incorporar todas las peculiaridades de su uso de una forma amigable para el usuario final a quien debe ir dirigido.

4.3. Metodología CommonKADS

CommonKADS (*Knowledge Analysis and Documentation System*) [35] es una metodología diseñada para el análisis y la construcción de sistemas basados en conocimiento (SBC) de forma análoga a los métodos empleados en ingeniería de software. Fue propuesta y desarrollada por un grupo de investigadores pertenecientes a diversos países de la comunidad Europea, a través del programa ESPRIT para la innovación y la aplicación de Tecnología Informática avanzada. El trabajo se comenzó en 1983 cuando había poco interés en tales metodologías. En ese momento, la construcción de sistemas de conocimiento estaba enmarcada bajo el paradigma de desarrollo por prototipos y de representación del conocimiento a través de reglas de producción, con hardware y software de propósito especial como máquinas LISP y PROLOG, herramientas especiales para sistemas expertos, etc.

Lo que se pretendía era crear un estándar para ingeniería del conocimiento y sistemas de conocimiento con el cual se pudieran construir sistemas industriales de calidad a gran escala, en una forma estructurada y controlada. En el desarrollo de CommonKADS han participado investigadores de diferentes áreas, de diferentes universidades europeas, e incluso empresas que han servido para ver su aplicación y validar lo establecido. Sobre esta metodología se han presentado varios artículos y ponencias en revistas y eventos especializados y se han escrito algunos libros, con el fin de que se conozca y aplique en la solución de problemas reales [36].

En CommonKADS podemos ver reflejadas tres ideas que han emergido, no solo de la experiencia en la Ingeniería del conocimiento, sino también en del campo de la Ingeniería del Software en general. Estas tres ideas se pueden concretar en tres conceptos:

- Modelado
- Reutilización
- Gestión del riesgo.

La metodología CommonKADS abarca todo el ciclo de desarrollo de software (puesto que se extiende no solamente a SBCs sino al software en general) mediante un número de modelos interrelacionados que capturan los principales rasgos del sistema y de su entorno.

El proceso de desarrollo de SBC consiste en rellenar un conjunto de plantillas de los modelos. Asociados a estas plantillas, CommonKADS define estados de los modelos que caracterizan hitos en el desarrollo de cada modelo. Estos estados permiten la gestión del proyecto, cuyo desarrollo se realiza de una forma cíclica dirigida por los riesgos.

Hay seis modelos definidos en CommonKADS, como se muestra en la *Figura 44* [36].

- **Modelo de la Organización (OM):** es una herramienta para analizar la organización en que el SBC va a ser introducido, y pretende descubrir problemas y oportunidades.
- **Modelo de Tarea (TM)** (Subpartes relevantes del proceso): describe a un nivel general las tareas que son realizadas o serán realizadas en el entorno organizativo en que se propone instalar el SBC y proporciona el marco para la distribución de tareas entre agentes.
- **Modelo de Agente (AM):** un agente es un ejecutor de una tarea. Puede ser humano, software o cualquier otra entidad capaz de realizar una tarea. Este modelo describe las competencias, características, autoridad y restricciones para actuar de los agentes.
- **Modelo de Comunicaciones (CM):** detalla el intercambio de información entre los diferentes agentes involucrados en la ejecución de las tareas descritas en el modelo de tarea.
- **Modelo del Conocimiento** (de Pericia o de Experiencia - EM): este es el corazón de la metodología CommonKADS y modela el conocimiento de resolución de problemas empleado por un agente para realizar una tarea. El modelo de la experiencia distingue entre el conocimiento de la aplicación y el conocimiento de resolución del problema. El conocimiento de la aplicación se divide en tres subniveles: nivel del dominio (conocimiento declarativo sobre el dominio), nivel de inferencia (una biblioteca de estructuras genéricas de inferencia) y nivel de tarea (orden de las inferencias).
- **Modelo de Diseño (DM):** mientras que los otros cinco modelos tratan del análisis del SBC, este modelo se utiliza para describir la arquitectura y el diseño técnico del SBC como paso previo a su implementación. En general produce la especificación técnica en términos de arquitectura, plataforma de implementación,

módulos de software, construcciones de representación, y mecanismos computacionales para la implementación de SBC.

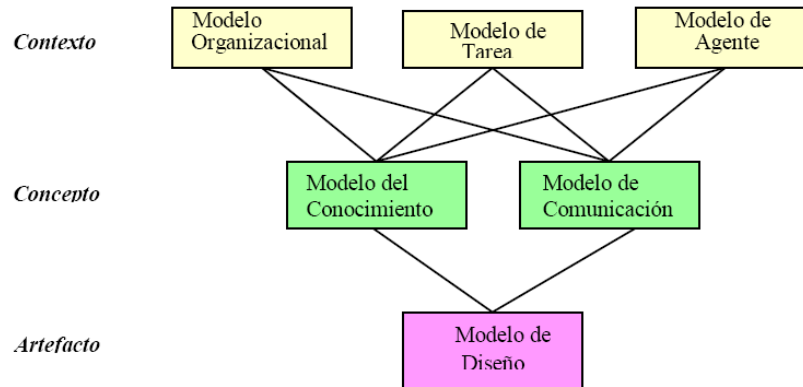


Figura 44: Modelos definidos por CommonKADS

4.3.1. CommonKADS-RT

El desarrollo de SBC de tiempo real son sistemas informáticos que manejan el conocimiento de un ambiente o dominio específico y garantizan una respuesta en un tiempo fijo. Para la creación de este tipo de sistemas, existe una metodología que propone las actividades a realizarse o las pautas a seguir y se conoce como CommonKADS-RT [37].

CommonKADS-RT permite seguir, en una forma comprensible y sencilla la construcción de un sistema basado en el conocimiento de tiempo real y se fundamenta en el desarrollo evolutivo y la orientación por los riesgos. En esta metodología se plantea que un sistema basado en el conocimiento de tiempo real se construye a través del desarrollo de siete modelos del problema o su solución. De estos modelos, los primeros cinco forman la fase de análisis y los restantes constituyen la fase de diseño.

CommonKADS-RT está basada en CommonKADS y RT-UML (Real-Time – del inglés, *Unified Modeling Language*) donde RT-UML es un planteamiento de una ampliación del lenguaje UML para poder hacer el modelado de las características de tiempo real. Entre las características generales de CommonKADS-RT están su usabilidad para la implementación de un SBC de tiempo real, las actividades específicas a realizarse para integrar este tipo de sistemas y las herramientas, conocimientos y utilidades necesarias para realizar estas actividades.

Modelos que integran a CommonKADS-RT

CommonKADS-RT toma como base los modelos de CommonKADS [38], incluyendo las características de los sistemas de tiempo real, agregándole los métodos que permiten especificar estos comportamientos temporales. En la *Figura 45* [38] se presenta el esbozo general de la propuesta.

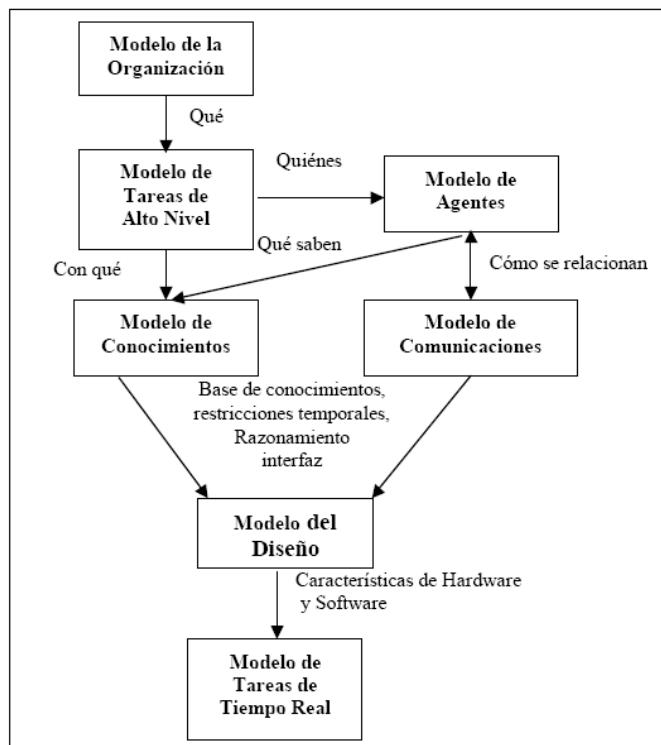


Figura 45: Modelos de CommonKADS-RT

La idea básica es que el desarrollo de un Sistema Basado en el Conocimiento de Tiempo Real se ve como la construcción de un número de modelos que juntos constituyen parte del producto que se debe entregar en el proyecto y en donde el nivel de elaboración de los modelos depende de cada proyecto.

Modelo de tareas de tiempo real (TTR)

El modelo *de TTR* [38] hace parte de la fase de diseño, en la cual se especifica una solución particular basada en los modelos del análisis.

Junto con el *modelo de diseño*, este modelo da la especificación técnica del sistema basado en el conocimiento de tiempo real en función de su arquitectura, la plataforma

de implantación, los módulos del software y los mecanismos computacionales necesarios para llevar a cabo las funciones determinadas en los *modelos de conocimiento* y de *comunicaciones*.

En términos de los componentes de un modelo en CommonKADS, los del *modelo de TTR*, son los siguientes:

- Bases o razones:
 - ✓ Debería servir para especificar las características particulares de una tarea de tiempo real.
 - ✓ Debería proporcionar las herramientas para modelar las tareas de tiempo real a través de diagramas apropiados.
 - ✓ Debería permitir la identificación de las tareas de tiempo real que tienen un comportamiento específico. Por ejemplo, cuándo las tareas son periódicas o no.
 - ✓ Debería proporcionar la información que se requiere para llevar a cabo un *test* de planificabilidad.
- Contenido del MTTR: El contenido de este modelo está dado por los elementos del modelo y sus relaciones que se pueden apreciar en la *Figura 46* [38].

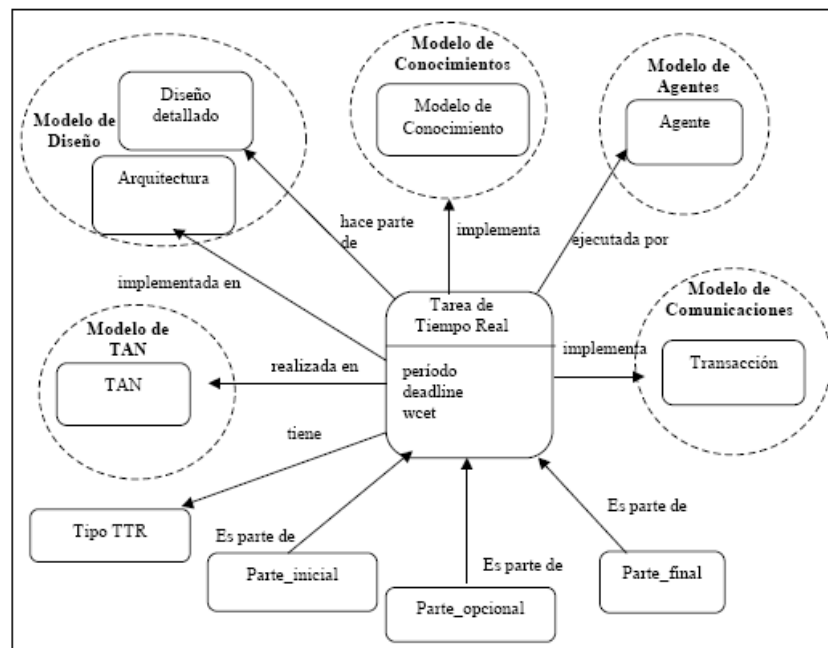


Figura 46: Contenido del Modelo de Tareas de Tiempo Real

- Estados del MTTR:

Los estados de este modelo están dados por las diferentes etapas en que él puede estar. De esta forma se han identificado los siguientes:

- Empieza con la identificación de los componentes de la TTR. En este estado se tipifican las partes iniciales, opcionales y finales de la tarea de tiempo real.
- Especificación de las variables temporales de los componentes. Se refiere a la determinación numérica (cantidad de tiempo) que tienen asociadas cada uno de los componentes de la tarea.

- Dependencias del MTTR:

Las dependencias se pueden observar en la figura anterior en donde se expresan tanto las dependencias entre los componentes del *modelo de tareas de tiempo real* como las que se refieren a las relaciones con los demás modelos.

- Criterios de calidad para el MTTR:

Los criterios de calidad para este modelo están dados por el cumplimiento o no de un estado posible, es decir, es la medida en que se logre alcanzar un estado. Por ejemplo, si ya se han definido los tiempos y plazos para cada una de las TTR, entonces se ha cumplido con un estado completo y así se ha definido que se asegura la calidad en él.

Confirmando lo presentado en el modelo anterior, la fase del diseño puede ser descompuesta en tres categorías básicas:

- El diseño de la arquitectura, en donde se detalla la estructura del software en términos de subsistemas, paquetes y tareas.
- El diseño mecánico, incluye el diseño de los mecanismos compuestos de clases que trabajan juntas para alcanzar un objetivo común.
- El diseño detallado para especificar las estructuras de datos primitivas a un nivel interno y los algoritmos particulares.

Así, el *modelo de TTR* integra el diseño de la arquitectura, particularmente en lo que se refiere a las tareas de tiempo real.

Este modelo es representado como una serie de tareas de tiempo real con una estructura que permite que varias tareas se relacionen entre sí. Una TTR será definida por sus restricciones temporales y sigue la estructura de un in-agent, el cual consta de un componente de percepción, un componente inteligente o cognitivo y un componente de acción. Estos componentes deben ser ejecutados en este orden y la ejecución de cada uno debe empezar después de la finalización del componente previo. Además, el in-agent se caracteriza por tener un comportamiento periódico y dispone de un plazo máximo de ejecución – *deadline* – para realizar las acciones que crea más oportunas. La *Figura 47* presenta esta estructura en forma gráfica [38].

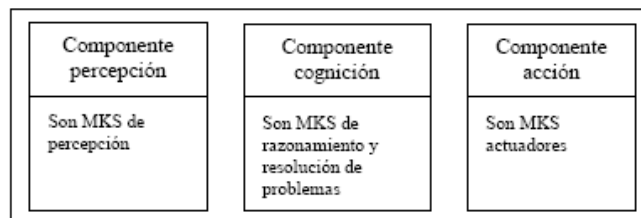


Figura 47: Estructura de la tarea de tiempo real a bajo nivel

Donde, un MKS es una entidad con conocimiento para resolver un problema o subproblema. Está constituido por una secuencia de niveles, cada uno de los cuales está definido por una secuencia de fuentes de conocimiento (KS) que contienen el conocimiento para resolver el problema en una forma diferente o para refinar la solución calculada por un nivel previo.

CAPÍTULO V: HERRAMIENTAS DE PROGRAMACIÓN Y DESARROLLO

5.1. MatLab

MatLab es un software matemático que ofrece un entorno de desarrollo (IDE) con un lenguaje de programación propio (lenguaje M) [39].

- ✓ Entre sus prestaciones básicas se hallan:
- ✓ Manipulación de matrices
- ✓ Representación de datos y funciones
- ✓ Implementación de algoritmos
- ✓ Creación de interfaces de usuario (GUI)
- ✓ Comunicación con programas en otros lenguajes y con otros dispositivos hardware

MatLab es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. MatLab integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían racionalmente, sin necesidad de hacer uso de la programación tradicional [40].

De forma coherente y sin ningún tipo de fisuras, integra los requisitos claves de un sistema de computación técnico: cálculo numérico, gráficos, herramientas para aplicaciones específicas y capacidad de ejecución en múltiples plataformas, para ello MatLab utiliza una serie de programas llamados ToolBoxes, los cuales se expondrán más adelante. Esta familia de productos proporciona al estudiante un medio de carácter único, para resolver los problemas más complejos y difíciles.

MatLab nace como una solución a la necesidad de mejores y más poderosas herramientas de cálculo para resolver problemas de cálculo complejos en los que es necesario aprovechar las amplias capacidades de proceso de datos de grandes computadores.

El nombre Matlab viene de "matrix laboratory" (laboratorio matricial). Matlab fue originalmente escrito para proveer acceso fácil al software matricial desarrollado por los proyectos LINPACK y EISPACK, que juntos representan el estado del arte y software para computación matricial.

Hoy Matlab es usado en una variedad de áreas de aplicación incluyendo procesamiento de señales e imágenes, diseño de sistemas de control, ingeniería financiera e investigación médica. La arquitectura abierta facilita usar Matlab y los productos que lo acompañan para explorar datos y crear herramientas personalizadas que proveen visiones profundas tempranas y ventajas competitivas.

Características de Matlab [41]:

- ✓ Cálculos intensivos desde un punto de vista numérico.
- ✓ Lenguaje de alto nivel basado en vectores, arrays y matrices.
- ✓ Colección muy útil de funciones de aplicación.
- ✓ Entorno de desarrollo para la gestión de código, archivos y datos
- ✓ Herramientas interactivas para exploración, diseño y resolución de problemas iterativos
- ✓ Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtraje, optimización e integración numérica
- ✓ Funciones gráficas bidimensionales y tridimensionales para visualización de datos
- ✓ Herramientas para crear interfaces gráficas de usuario personalizadas
- ✓ Funciones para integrar los algoritmos basados en MATLAB con aplicaciones y lenguajes externos, tales como C/C++, FORTRAN, Java, COM y Microsoft Excel.

5.1.1. Como hacer un programa en MATLAB

Es posible realizar un programa en Matlab tal como se hace en otros lenguajes como el Basic, Pascal, Java o el lenguaje C. Es necesario utilizar un editor para escribir el código.

- ✓ Para escribir código, se requiere crear un archivo **.m*. Para esto se necesita abrir new M.file en la ventana file.
- ✓ Ahora se escribe el código y se guarda el archivo utilizando la terminación archivo *.m*.

- ✓ Se puede correr el programa desde Matlab simplemente escribiendo el nombre del archivo que fue creado.
- ✓ Es posible abrir programas con la terminación *.m* desde Matlab, en el menú file, open M.file.

5.1.1.1. El fichero *.m*

Los ficheros o archivos con extensión (*.m*) son ficheros de texto sin formato (*ASCII*) y constituyen el sustento para la programación en MatLab [42].

Estos ficheros pueden crearse y/o modificarse con un editor de textos cualquiera aunque si se trabaja con MatLab lo mejor sería utilizar su propio editor de textos, que es a su vez el debugger.⁸

Existen dos tipos de ficheros *.m*, los ficheros de comandos (*scripts*) y las funciones como se muestra en *Figura 48* [42].

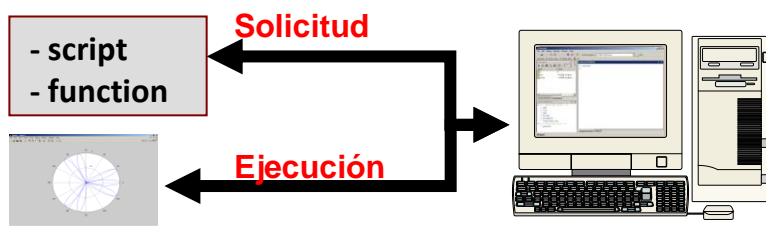


Figura 48: Tipo de ficheros

5.1.1.2. Las funciones

La implementación de funciones amplía grandemente el horizonte de posibilidades de MatLab.

Al igual que en otros lenguajes estas cuentan con el nombre, sus argumentos y sus valores de retorno, en la *Figura 49* [43] se puede apreciar su sintaxis.

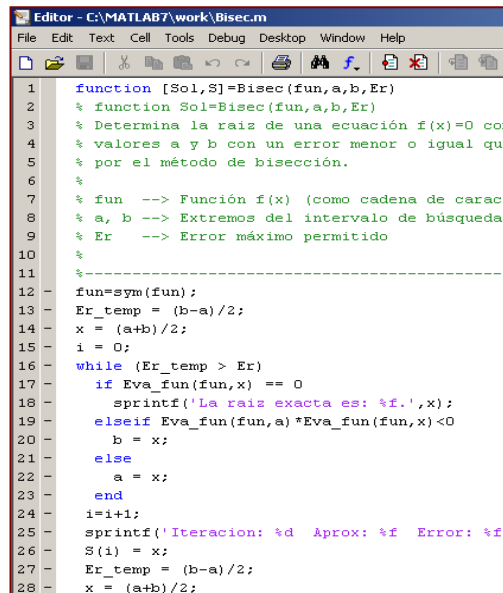
Declaración:

```
function [var_salida] = nombre_funcion(<argumentos_entrada>)
```

Figura 49: Sintaxis de una función en MatLab.

⁸ Debugger: Permite la ejecución controlada de un programa o un código, para seguir cada instrucción ejecutada y localizar errores.

Las funciones definidas en ficheros *.m se caracterizan porque la primera línea (no comentariada) comienza por la palabra “function”, si los parámetros de salida son más de uno, se sitúan entre corchetes y separados por comas, y si los parámetros de entrada son más de uno, se separan por comas, en la *Figura 50* [43] se puede apreciar esta sintaxis.

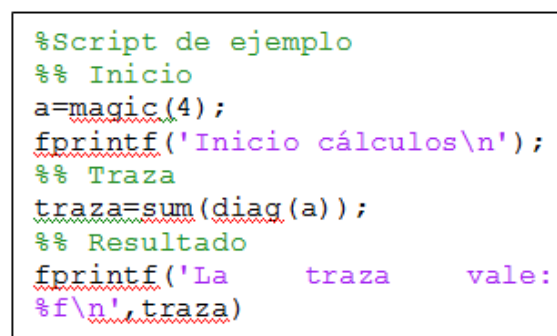


```
Editor - C:\MATLAB7\work\Bisec.m
File Edit Text Cell Tools Debug Desktop Window Help
1 function [Sol,S]=Bisec(fun,a,b,Er)
2 % function Sol=Bisec(fun,a,b,Er)
3 % Determina la raiz de una ecuación f(x)=0 com
4 % valores a y b con un error menor o igual que
5 % por el método de bisección.
6 %
7 % fun --> Función f(x) (como cadena de caract
8 % a, b --> Extremos del intervalo de búsqueda
9 % Er --> Error máximo permitido
10 %
11 %-----
12 fun=sym(fun);
13 Er_temp = (b-a)/2;
14 x = (a+b)/2;
15 i = 0;
16 while (Er_temp > Er)
17     if Eva_fun(fun,x) == 0
18         sprintf('La raiz exacta es: %f.',x);
19     elseif Eva_fun(fun,a)*Eva_fun(fun,x)<0
20         b = x;
21     else
22         a = x;
23     end
24     i=i+1;
25     sprintf('Iteracion: %d Aprox: %f Error: %f'
26     S(i) = x;
27     Er_temp = (b-a)/2;
28     x = (a+b)/2;
```

Figura 50: Ejemplo de una Función en MatLab.

5.1.1.3. Scripts

Un script es una secuencia de instrucciones de MatLab guardada en un archivo con extensión .m como se muestra [44].



```
%Script de ejemplo
%% Inicio
a=magic(4);
fprintf('Inicio cálculos\n');
%% Traza
traza=sum(diag(a));
%% Resultado
fprintf('La traza vale:
%f\n',traza)
```

Figura 51: Ejemplo de un script en MatLab con nombre ejem_script.m

Se ejecuta escribiendo su nombre: >> ejem_script

5.1.2. ToolBoxes de MatLab

Matlab dispone en la actualidad de una amplia variedad de programas de apoyo especializados, denominados Toolboxes⁹, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el toolbox de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neuronales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc. es un entorno de cálculo técnico, que se ha convertido en estándar de la industria, con capacidades no superadas en computación y visualización numérica. A continuación se muestra una lista de estas herramientas [45].

TABLA III: LISTA DE TOOLBOXES EN MATLAB

Lista de Toolboxes	
- Aerospace Toolbox	- Instrument Control Toolbox
- Bioinformatics Toolbox	- Mapping Toolbox
- Communications Toolbox	- Model-Based Calibration Toolbox
- Control System Toolbox	- Model Predictive Control Toolbox
- Curve Fitting Toolbox	- Neural Network Toolbox
- Data Acquisition Toolbox	- OPC Toolbox
- Database Toolbox	- Optimization Toolbox
- Datafeed Toolbox	- Parallel Computing Toolbox
- Econometrics Toolbox	- RF Toolbox
- Filter Design Toolbox	- Robust Control Toolbox
- Filter Design HDL Coder	- SIMULINK, SIMULINK Real Time Workshop
- Financial Toolbox	- Signal Processing Toolbox
- Financial Derivatives Toolbox	- Spreadsheet Link EX
- Fixed-Income Toolbox	- Statistics Toolbox
- Fixed-Point Toolbox	- Symbolic Math Toolbox
- Fuzzy Logic Toolbox	- System Identification Toolbox
- Global Optimization Toolbox	- Vehicle Network Toolbox
- Image Acquisition Toolbox	- Wavelet Toolbox
- Image Processing Toolbox	

⁹ ToolBox: (Caja de herramientas), es una colección especializada de archivos .m

En este proyecto se hizo uso de las funciones del Image Processing Toolbox (Toolbox de Procesamiento de Imágenes), Fuzzy Logic Toolbox (Toolbox de Lógica Difusa), Signal Processing Toolbox, Statistics Toolbox y Wavelet Toolbox.

5.1.2.1 Image Processing Toolbox

Este toolbox proporciona a MATLAB de un conjunto de funciones que amplía las capacidades del producto para realizar desarrollo de aplicaciones y de nuevos algoritmos en el campo del proceso y análisis de imágenes. El entorno matemático y de creación de MATLAB es ideal para el procesamiento de imágenes, ya que estas imágenes son, al fin y al cabo, matrices [46]. Este toolbox incorpora funciones para:

- Diseño de filtros y recuperación de imágenes.
- Mejora y retocado de imágenes.
- Análisis y estadística de imágenes.
- Definición de mapas de colores y modificación gráfica.
- Operaciones morfológicas, geométricas y de color.
- Transformaciones 2D.
- Transformación de imágenes
- Proceso de bloques

El proceso de imágenes es un campo de trabajo absolutamente crucial para aquellos colectivos e industrias que estén trabajando en áreas como diagnóstico médico, astronomía, geofísica, ciencias medioambientales, análisis de datos en laboratorios, inspección industrial, etc. Los programas actuales de procesamiento y análisis de imágenes se clasifican actualmente en dos categorías: librerías de bajo nivel para programadores profesionales y paquetes de aplicación con capacidades limitadas de personalización. Ambos tipos de aplicaciones están generalmente, pensados para tareas básicas de visualización de datos y 'rendering'. Sin embargo, muchos de ellos adolecen de la posibilidad de efectuar análisis numéricos de los mismos. El Image Processing Toolbox entra dentro de la categoría de familias de funciones que, desde el entorno de trabajo de MATLAB, permitirá al profesional efectuar una exploración exhaustiva y desde un punto de vista matemático de las imágenes y gráficos que se deseen tratar o analizar [46].

5.1.2.2. Fuzzy Logic Toolbox

Fuzzy Logic Toolbox es un conjunto de funciones integradas en el entorno de MATLAB[®]. Proporciona herramientas para crear y modificar los sistemas de inferencia difusos en el marco de MATLAB. También puede integrar sus sistemas difusos en simulaciones con software de Simulink[®]. Este toolbox se basa en gran medida de interfaz gráfica de usuario (GUI) para ayudar y facilitar los trabajos, aunque se puede trabajar totalmente desde la línea de comandos si lo prefiere [47].

El toolbox proporciona tres categorías de herramientas:

- Funciones de línea de comandos
- Gráfica, herramientas interactivas
- Bloques de Simulink y ejemplos.

La primera categoría de instrumentos se compone de las funciones que pueden llamar desde la línea de comandos o desde sus propias aplicaciones. Muchas de estas funciones son archivos *.M* que contienen una serie de instrucciones de MATLAB que implementan algoritmos especializados de la lógica difusa.

Se puede cambiar la manera en que trabaja cualquier función del toolbox copiando y cambiando el nombre del archivo *.M*, y luego, modificando la copia. También se puede ampliar toolbox mediante la adición de nuestros propios archivos.

En segundo lugar, el toolbox proporciona una serie de herramientas interactivas que le permiten acceder a muchas de las funciones a través de una interfaz gráfica de usuario (GUI). En conjunto, las herramientas GUI proporcionan un entorno para el diseño de sistemas difusos de inferencia, análisis y aplicación.

La tercera categoría de herramientas es un conjunto de bloques para su uso con Simulink. Estos están diseñados específicamente para la inferencia lógica de alta velocidad difusa en el entorno Simulink.

Lo que hace el toolbox tan poderoso es el hecho de que la mayor parte del razonamiento humano y la formación del concepto está relacionada con el uso de

reglas difusas. Al proporcionar un marco sistemático para la computación con reglas difusas, el toolbox en gran medida amplía el poder del razonamiento humano [47].¹⁰

Hay cinco principales herramientas de la GUI para la creación, edición, y la observación de los sistemas de inferencia difusos en el toolbox de Lógica Difusa: el Sistema de Inferencia Difuso o el Editor de FIS (Fuzzy Inference System), el editor de funciones de pertenencia, el editor de reglas, el Visor de Reglas, y el Visor de superficies como indica la *Figura 52* [48]. Estas interfaces gráficas están muy relacionadas entre sí, ya que los cambios que realice en el FIS con uno de ellos, puede afectar a lo que se ve en cualquiera de las interfaces gráficas de usuario abiertas.

Además de estos cinco interfaces gráficas, el toolbox incluye el editor de interfaz gráfica ANFIS, que se utiliza para la construcción y análisis de tipo Sugeno.

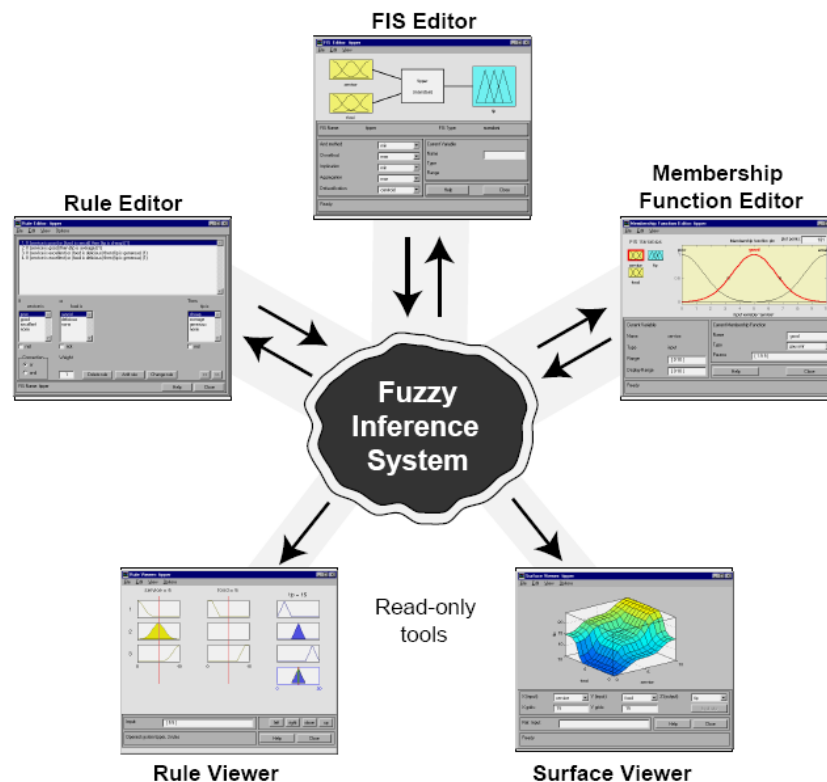


Figura 52: Sistema de Inferencia Fuzzy

¹⁰ The MathWorks, Inc., MATLAB Fuzzy Logic Toolbox. User's Guide (en inglés) 2, 69 p.

Un método eficaz desarrollado por Dr. Roger Jang para este propósito se llama ANFIS (Adaptive Neuro-Fuzzy Inference System). Este método es un componente importante del toolbox.

Restricciones del editor Anfis [49]:

- Solo trabaja con sistemas difusos Sugeno.
- Tiene una única salida, pero sus funciones de membresía pueden ser todas del mismo tipo o diferentes.
- Solo puede existir una regla inferencia que haga referencia a cada una de las funciones de pertenencia de la salida.
- Se debe tener peso unitario para cada regla.

5.1.3. NetBeans - Java

NetBeans se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma NetBeans [50].

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándose nuevos módulos.

Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

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

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndose al desarrollar enfocarse en la lógica específica de su aplicación.

Entre las características de la plataforma están:

- Administración de las interfaces de usuario.
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- Administración de ventanas.
- Framework basado en asistentes.

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio.

Ambos productos son de códigos abiertos y gratuitos, para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2 [51].

El lenguaje de programación que utiliza NetBeans es Java. Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. EL lenguaje en sí mismo toma mucha de su sintaxis de C y C++ pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria [52].

El lenguaje Java se creó con cinco objetivos principales:

- Debería utilizar la metodología de la programación orientada a objetos
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos
- Debería incluir por defecto soporte para trabajo en red
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura

- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Java es un lenguaje de programación con el que se realiza cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras [52].

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir, que si se hace un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc.

Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria [52].

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. Para lo que se interesa en este proyecto, con Java se puede interactuar con las aplicaciones en MatLab, con accesos a bases de datos y demás.

5.1.4. Herramienta visual para modelado UML - Enterprise

UML nació en durante los ochenta y principios de los noventa cuando Grady Booch, James Rumbaugh, e Ivar Jacobson trabajaban por separado en desarrollo de notaciones para el análisis y diseño de sistemas orientados a objetos. Los tres llegaron por separado a obtener bastante reconocimiento [53].

UML es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos [53].

Una vez que se conoce las herramientas se usan en cada momento las más adecuadas a las necesidades. Esto no es fácil, pues hay que saber para qué sirven y qué limitaciones tienen unas y otras para conocer su utilidad. Pero se puede alcanzar este conocimiento con un poco de práctica y sentido común.

Y así UML se convierte en una herramienta muy poderosa en el desarrollo de proyectos software, esta herramienta cuenta con una muy útil caja de herramientas las cuales son:

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones
- Diagrama de componentes
- Diagrama de distribución

Pues así como UML es una herramienta muy utilizada para la realización de modelados (análisis y diseño), esta es a su vez elaborada por herramientas automáticas para la creación de modelados como lo es Enterprise, una magnífica herramienta para la creación de diagramas en la elaboración de proyectos software.

Enterprise Architect (EA) es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad [54].

Enterprise Architect está soportado para los 13 diagramas de UML y más; algunos de estos diagramas se listan en la TABLA IV [54].

TABLA IV: DIAGRAMAS UML

Diagramas Estructurales	Diagramas de Comportamiento	Extendidos
<ul style="list-style-type: none"> • Clase • Objeto • Compuesto • Paquete • Componente • Despliegue 	<ul style="list-style-type: none"> • Casos de Uso • Comunicación • Secuencia • Descripción de la Interacción • Actividad • Estado • Tiempo 	<ul style="list-style-type: none"> • Análisis (actividad simple) • Personalizado (para requisitos, cambios)

5.1.5 Frameworks y Librerías

Los frameworks de Java en la práctica son conjuntos de librerías (API's) para desarrollar aplicaciones de diversa índole, es decir más librerías para su ejecución (o motor), y conjuntos de herramientas para facilitar tareas específicas.

5.1.5.1. JMatLink

Ha sido necesaria en este proyecto de titulación, la utilización de una librería desarrollada por Stefan Müller¹¹, denominada JMatLink que está basada en JNI (Java Native Interface). Esta librería conecta Java y MatLab y los métodos nativos hacen posible el uso del motor de cálculo de MatLab dentro de las aplicaciones Java, Applets y Servlets [55].

Esta librería se ha utilizado en NetBeans, es decir, se ha programado con Java y con diferentes métodos es posible realizar llamadas a funciones de MatLab y recibir valores como resultado de la función.

Mediante esta librería lo que se pretende es que una vez el interfaz programado en Java, se haga una llamada a las funciones programadas con MatLab las cuales son las encargadas de tratar las imágenes capturadas inicialmente por Java. Una vez trata las imágenes y extrae sus características, devuelve un valor al programa Java y éste, dependiendo del valor devuelto, realizará una u otra opción programada.

5.1.5.2. Java Media Framework (JMF)

El Java™ Media Framework (JMF) es una interfaz de programación de aplicaciones (API) para la incorporación de datos media tales como audio y vídeo en las

¹¹ <http://jmatlink.sourceforge.net/developers.php>

aplicaciones y applets Java. Se ha diseñado específicamente para aprovechar las características de la plataforma Java.

El JMF 1.0 API (el Java Media Player API) permite a los programadores desarrollar programas en Java para la reproducción “time-based media”. El JMF 2.0 API amplía el marco de la prestación de apoyo para la captura y almacenamiento de datos media, controlando el tipo de procesamiento que se realiza durante la reproducción, y realizar un tratamiento personalizado en los flujos de datos media. Además, JMF 2.0 define un plug-in¹² API que permite a los desarrolladores y proveedores de tecnología avanzada para personalizar más fácilmente y extender la funcionalidad de JMF [56].

La última implementación de este API de Sun es JMF 2.1.1e que incluye varias mejoras a los componentes de RTP/RTSP¹³ (Recepción – Transmisión).

5.1.5.3. Pentaho Reporting (JFreeReport)

JFreeReport es una librería libre de la clase Java™ para la generación de reportes. Los datos de JFreeReport son proporcionados a través de la interfaz TableModel de Swing¹⁴, y formateado de acuerdo con un archivo de definición de reportes basado en XML. Los reportes se pueden previsualizar (*Ver Figura 53*) [57] en la pantalla, envíos a la impresora o guardar en varios formatos, incluyendo PDF, HTML, Excel, CSV y texto sin formato [57].

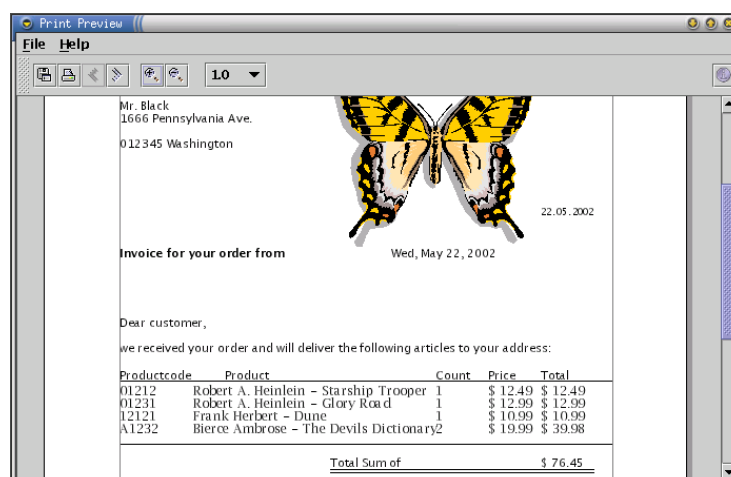


Figura 53: Previsualización de reporte con JFreeReport.

¹² Un **plug-in** es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

¹³ RTP/RTSP: Proveen los mecanismos necesarios para la transmisión de audio y video a través de la red.

¹⁴ Se utiliza para obtener datos de las tablas creadas con la clase JTable de Swing.

JFreeReport, ha sido rebautizado como Pentaho Reporting, es un potente motor de reporting que permite el acceso a diversos formatos de datos en diferentes fuentes, incluyendo bases de datos relacionales como MySQL, OLAP¹⁵ y XML. Crea informes en formatos populares: PDF, html, texto y Excel.

Pentaho ha diseñado un asistente que facilita la implementación paso a paso de este nuevo producto suyo que puede descargarse gratuitamente bajo licencia de código abierto. La dinámica es tan sencilla como descargarlo y, siguiendo las instrucciones, conectarse a una fuente de datos, elegir un diseño y los formatos de salida deseados. Además, también ha desarrollado una conexión entre JFreeReport y Mondrian, su servidor OLAP [58].

Ahora, los clientes de Pentaho pueden elegir el despliegue de reporting que se ajuste a sus necesidades.

5.1.5.4. JFreeChar

JFreeChart es una librería libre 100% de gráficos de Java que hace que sea fácil para los desarrolladores para mostrar gráficos estadísticos de calidad profesional en sus aplicaciones. JFreeChart soporta gráficos circulares (2D y 3D), gráficos de barras (horizontal y vertical, regular y apilados), de líneas, gráficos de dispersión, gráficos de series de tiempo, gráficos de alto-bajo-de apertura y cierre, gráficos de velas, diagramas de Gantt, gráficos combinados, termómetros, relojes y mucho más; como se puede ver en la *Figura 54* [59]. Además JFreeChart se puede utilizar en aplicaciones, applets, servlets y JSP [59].

Este proyecto es mantenido por David Gilbert. JFreeChart incluye un conjunto de características como:

- Una API consistente y bien documentada, soportando a una amplia gama de tipos de gráficos;
- Un diseño flexible fácil de extender, y se dirige tanto del lado del servidor y las aplicaciones del lado del cliente;

¹⁵ OLAP: Es el acrónimo en inglés de procesamiento analítico en línea (*On-Line Analytical Processing*). Es una solución utilizada en el campo de la llamada Inteligencia empresarial (o *Business Intelligence*) cuyo objetivo es agilizar la consulta de grandes cantidades de datos.

- Soporte para muchos tipos de salida, incluyendo los componentes Swing, archivos de imagen (incluyendo PNG y JPEG) y gráficos vectoriales formatos de archivo (incluyendo PDF, EPS y SVG);
- JFreeChart es "open source" o, más específicamente software libre. Se distribuye bajo los términos de la GNU Licencia Pública General Menor (LGPL), que permite el uso de las aplicaciones propietarias.

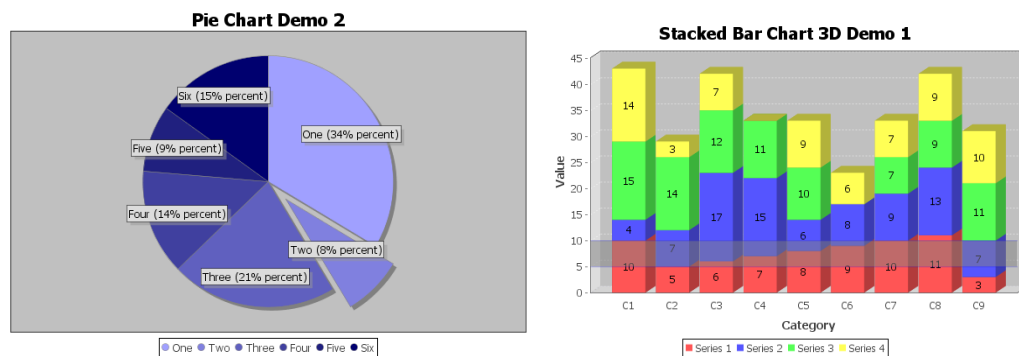


Figura 54: Ejemplo de gráficos de pastel y barras con JFreeChart.

5.1.5.5. Hibernate

Hibernate [60] es una capa de persistencia objeto/relacional y un generador de sentencias *sql*. Te permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada podremos generar BBDD en cualquiera de los entornos soportados: Oracle, DB2, MySql, etc. Y lo más importante de todo, es **open source**, lo que supone, entre otras cosas, que no tenemos que pagar nada por adquirirlo.

Uno de los posibles procesos de desarrollo consiste en, una vez tengamos el diseño de datos realizado, mapear este a ficheros XML siguiendo la DTD de mapeo de Hibernate. Desde estos podremos generar el código de nuestros objetos persistentes en clases Java y también crear BBDD independientemente del entorno escogido.

Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. Una posible configuración básica de Hibernate es la siguiente (ver Figura 55) [60].

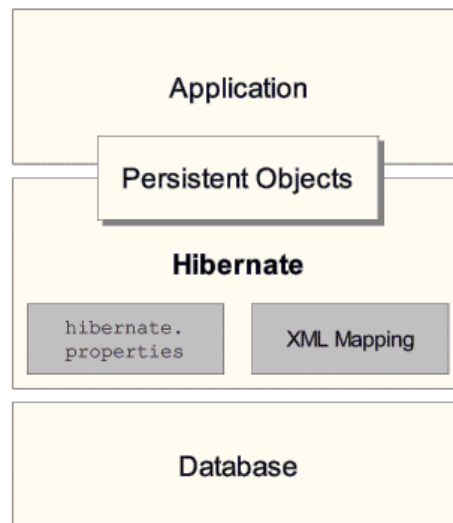


Figura 55: Arquitectura base

Podemos observar como Hibernate utiliza la BBDD y la configuración de los datos para proporcionar servicios y objetos persistentes a la aplicación que se encuentre justo por arriba de él.

El trabajar con software orientado a objetos y con una base de datos relacional puede llegar a ser engorroso y puede llegar a tomar bastante tiempo en los entornos empresariales de hoy en día. Hibernate es una herramienta de mapeo objeto/relacional para entornos Java. El término de mapeo objeto/relacional (ORM) se refiere a la técnica de mapear una representación de datos desde un modelo de objeto a un modelo de datos relacionales con un esquema basado en SQL.

Hibernate no solamente se ocupa del mapeo desde las clases Java a las tablas de las bases de datos (y desde los tipos de datos de Java a los tipos de datos de SQL), sino que también facilita la consulta y recuperación de datos. Esto puede reducir de manera importante el tiempo de desarrollo que se tomaría con el manejo de datos de forma manual en SQL y JDBC.

La meta de Hibernate es aliviar el trabajo del desarrollador en 95% de la persistencia de datos comunes relacionados con tareas de programación. Es posible que Hibernate no sea la mejor solución para aquellas aplicaciones centralizadas en datos que sólomente utilizan los procedimientos almacenados para implementar la lógica empresarial en la base de datos.

Hibernate es mucho más útil con modelos de dominio orientados a objetos y con lógica empresarial middle-tier con base en Java. Sin embargo, Hibernate ciertamente le puede ayudar a eliminar o a encapsular código SQL específico del vendedor y le ayudará con la tarea común de traducción del grupo de resultados desde una representación tabular a un grafo de objetos [61].

5.1.5.5.1. Mapas de objetos relacionales en ficheros XML

Objetivos: Conseguir desde el diseño de un objeto relacional un fichero XML bien formado de acuerdo a la especificación Hibernate. Este trabajo consistirá en *traducir* las propiedades, relaciones y componentes al formato XML de Hibernate.

A partir de ahora se asume que objeto persistente, registro de tabla en BBDD y objeto relacional son la misma entidad.

5.1.5.5.1.1. Estructura del fichero XML

El esquema general de un fichero XML de mapeo puede ser como el de *Figura 56* [61].

```
<Encabezado XML>
  <Declaración de la DTD>
  <class - Definición de la clase persistente>
    <id - Identificador>
      <generator - Clase de apoyo a la generación
automática de OID's>
    <component - Componentes, son las columnas de la tabla>
      .
    <one-to-many / many-to-many - Posibles relaciones con
otras entidades persistentes>
    .
  .
```

Figura 56: Fichero XML

A continuación se detallan las características, parámetros y definición de las etiquetas arriba utilizadas así como de algunas otras que serán de utilidad a la hora de pasar el esquema relacional a ficheros de mapeo XML.

- **Declaración de la DTD.** El documento DTD que usaremos en nuestros ficheros XML se encuentra en cada distribución de Hibernate en el propio **.jar** o en el directorio **src**.

Elemento Raíz **<hibernate-mapping>**. Dentro de él se declaran las clases de los objetos persistentes. Aunque es posible declarar más de un elemento **<class>** en un

mismo fichero XML, no debería hacerse ya que aporta una mayor claridad a la aplicación realizar un documento XML por clase de objeto persistente.

Atributos del elemento class

• **<class>** Este es el *tag* donde declaramos la clase persistente. Una clase persistente equivale a una tabla en la base de datos, y un registro o línea de esta tabla es un objeto persistente de esta clase. Entre sus posibles atributos se destacan:

- ✓ *name*: Nombre completo de la clase o interface persistente. Se debe incluir el package dentro del nombre.
- ✓ *table*: Nombre de la tabla en la BBDD referenciada. En esta tabla se realiza las operaciones de transacciones de datos. Se guardarán, modificarán o borrarán registros según la lógica de negocio de la aplicación.
- ✓ *discriminator-value*: Permite diferenciar dos sub-clases. Utilizado para el polimorfismo.
- ✓ *proxy*: Nombre de la clase Proxy cuando esta sea requerida.

Atributos del elemento id

• **<id>** Permite definir el identificador del objeto. Se corresponderá con la clave principal de la tabla en la BBDD. Es interesante definir en este momento lo que será para la aplicación un OID (Identificador de Objeto). Se tiene que asignar identificadores únicos a los objetos persistentes, en un primer diseño se podría estar tentados a asumir un dato con significado dentro de la capa de negocios del propio objeto fuese el identificador, pero esta no sería una buena elección.

Imaginemos una tabla de personas con su clave primaria N.I.F.. Si el tamaño, estructura o composición del campo cambiase deberíamos realizar este cambio en cada una de las tablas relacionadas con la nuestra y eventualmente en todo el código de la aplicación.

Utilizando OID's (Identificadores de Objetos) tanto a nivel de código como en BBDD simplificamos mucho la complejidad de la aplicación y se puede programar partes de la misma como código genérico.

El único problema en la utilización de OID es determinar el nivel al cual los identificadores han de ser únicos. Puede ser a nivel de clase, jerarquía de clases o

para toda la aplicación, la elección de uno u otro dependerá del tamaño del esquema relacional.

- ✓ *name*: Nombre lógico del identificador.
- ✓ *column*: Columna de la tabla asociada en la cual se almacena su valor.
- ✓ *type*: Tipo de dato.
- ✓ *unsaved-value* ("any|none|null|id_value"): Valor que contendrá el identificador de la clase si el objeto persistente todavía no se ha guardado en la BBDD.
- ✓ *generator*: clase que se utilizar para generar los oid's. Si requiere de algún parámetro este se informa utilizando el elemento `<parameter name="nombre del parámetro">`.

Generadores de OID's presentes en Hibernate

• **Hibernate** proporciona clases que generan automáticamente estos OID evitando al programador recurrir a trucos como coger la fecha/hora del sistema. Entre ellas caben destacar:

- ✓ *vm*: Genera identificadores de tipo long, short o int. Que serán únicos dentro de una JVM.
- ✓ *sequence*: Utiliza el generador de secuencias de las bases de datos DB2, PostgreSQL, Oracle, SAP DB, McKoi o un generador en Interbase. El tipo puede ser long, short o int.
- ✓ *hilo*: Utiliza un algoritmo hi/lo para generar identificadores del tipo long, short o int. Estos OID's son únicos para la base de datos en la cual se generan. En realidad solo se trata de un contador en una tabla que se crea en la BBDD. El nombre y la columna de la tabla a utilizar son pasados como parámetros, y lo único que hace es incrementar/decrementar el contador de la columna con cada nueva creación de un nuevo registro. Así, si por ejemplo decimos tener un identificador único por clase de objetos persistentes, deberíamos pasar como parámetro tabla *table_OID* y como columna el nombre de la clase *myclass_OID*.
- ✓ *uuid.string*: Algoritmo UUID para generar códigos ASCII de 16 caracteres.
- ✓ *assigned*: Por si después de todo queremos que los identificadores los gestione la propia aplicación.

Atributos del elemento discriminator

• **<discriminator>** Cuando una clase declara un discriminador es necesaria una columna en la tabla que contendrá el valor de la marca del discriminador. Los conjuntos de valores que puede tomar este campo son definidos en la cada una de las clases o sub-clases a través de la propiedad `<discriminator-value>`. Los tipos permitidos son `string`, `character`, `integer`, `byte`, `short`, `boolean`, `yes_no`, `true_false`.

- ✓ *column*: El nombre de la columna del discriminador en la tabla.
- ✓ *type*: El tipo del discriminador.

Atributos del elemento property

<property> Declara una propiedad persistente de la clase, que se corresponde con una columna.

- ✓ *name*: Nombre lógico de la propiedad.
- ✓ *column*: Columna en la tabla.
- ✓ *type*: Indica el tipo de los datos almacenados. Se puede ver más adelante en Tipos de datos en Hibernate.

• **Tipos de relaciones. (Componentes y Colecciones)** En todo diseño relacional los objetos se referencian unos a otros a través de relaciones, las típicas son: uno a uno 1-1, uno a muchos 1-n, muchos a muchos n-m, muchos a uno n-1. De los cuatro tipos, dos de ellas n-m y 1-n son colecciones de objetos las cuales tendrán su propio y extenso apartado, mientras que a las relaciones 1-1 y n-1 son en realidad componentes de un objeto persistente cuyo tipo es otro objeto persistente.

Atributos de las relaciones n-1

- **<many-to-one>** La relación n-1 necesita en la tabla un identificador de referencia, el ejemplo clásico es la relación entre padre - hijos. Un hijo necesita un identificador en su tabla para indicar cuál es su padre. Pero en objetos en realidad no es un identificador si no el propio objeto padre, por lo tanto el componente n-1 es en realidad el propio objeto padre y no simplemente su identificador. (Aunque en la tabla se guarde el identificador).

- ✓ *name*: El nombre de la propiedad.

- ✓ *column*: Columna de la tabla donde se guardara el identificador del objeto asociado.
- ✓ *class*: Nombre de la clase asociada. Hay que escribir todo el package.
- ✓ *cascade* ("all|none|save-update|delete"): Especifica que operaciones se realizaran en cascada desde el objeto padre.

Atributos de las relaciones 1-1

- **<one-to-one>** Asociación entre dos clases persistentes, en la cual no es necesaria otra columna extra. Los OID's de las dos clases serán idénticos.

- ✓ *name*: El nombre de la propiedad.
- ✓ *class*: La clase persistente del objeto asociado.
- ✓ *cascade* ("all|none|save-update|delete"): Operaciones en cascada a partir de la asociación.
- ✓ *constrained* ("true"|"false").

Tipos de datos en Hibernate.

- *Son tipos básicos*: integer, long, short, float, double, character, byte, boolean, yes_no, true_false.
- *string*: Mapea un java.lang.String a un VARCHAR en la base de datos.
- *date, time, timestamp*: Tipos que mapean un java.util.Date y sus subclases a los tipo SQL: DATE, TIME, TIMESTAMP.
- *calendar, calendar_date*: Desde java.util.Calendar mapea los tipos SQL TIMESTAMP y DATE
- *big_decimal*: Tipo para NUMERIC desde java.math.BigDecimal.
- *locale, timezone, currency*: Tipos desde las clases java.util.Locale, java.util.TimeZone y java.util.Currency. Se corresponden con un VARCHAR. Las instancias de Locale y Currency son mapeadas a sus respectivos códigos ISO y las de TimeZone a su ID.
- *class*: Guarda en un VARCHAR el nombre completo de la clase referenciada.
- *binary*: Mapea un array de bytes al apropiado tipo de SQL.
- *serializable*: Desde una clase que implementa el interface Serializable al tipo binario SQL.
- *clob, blob*: Mapean clases del tipo java.sql.Clob y java.sql.Blob

Tipos enumerados persistentes (Persistente Enum Types): Un tipo enumerado es una clase de java que contiene la enumeración a utilizar (ej: Meses, Días de la semana, etc.). Estas clases han de implementar el interface **net.sf.hibernate.PersistentEnum** y definir las operaciones **toInt()** y **fromInt()**. El tipo enumerado persistente es simplemente el nombre de la clase completo. Ejemplo de clase persistente se puede apreciar en la *Figura 57* [61].

```
package com.hecsua.enumerations;
import net.sf.hibernate.PersistentEnum
public class Meses implements PersistentEnum {
    private final int code;
    private Meses(int code) {
        this.code = code;
    }
    public static final Meses Enero = new Meses(0);
    public static final Meses Febrero = new Meses(1);
    public static final Meses Marzo = new Meses(2);
    public int toInt() {return code;}
    public static Meses fromInt(int code){
        case 0: return Enero;
        case 1: return Febrero;
        case 2: return Marzo;
        default: throw new RuntimeException("Mes no valido.");
    }
}
```

Figura 57: Clase persistente

5.1.5.5.2. Mapeando una colección

Las colecciones son declaradas utilizando <set>, <list>, <map>, <bag>, <array> y <primitive-array>. Los posibles parámetros y sus valores son:

- *name*: El nombre lógico de la propiedad. Es útil poner un nombre que nos recuerde el rol de la colección (ej: Administradores, MultasSinPagar, etc.)
- *table*: Nombre de la tabla de la colección.
- *lazy* ("*true*"/*false*): Permite el uso de inicialización "lazy". Este tipo de inicialización hace que los objetos de la colección sean solicitados en demanda y no se carguen todos a la vez. Esto es especialmente útil para optimizar búsquedas, etc.
- *inverse*: Señala esta colección como el fin de una asociación bidireccional. Utilizada en relaciones many-to-many sobre todo.
- *cascade*: Permite las operaciones en cascada hacia las entidades hijas.
- *sort*: Especifica una colección con una ordenación natural o con una clase comparadora dada.

- *order-by*: Columna\s de la tabla que definen el orden de iteración. Puede ser ascendente o descendente.

5.1.5.5.3. Componentes

Un buen diseño relacional necesitará de la composición de objetos. El nombre de la persona, la dirección, una localidad etc. son todas estructuras de datos que componen objetos más grandes. Una componente en Hibernate es un objeto persistente contenido dentro de la misma tabla de su propietario. ej.: Clásico ejemplo del nombre/dirección de una persona, como se explica en la *Figura 58* [61]:

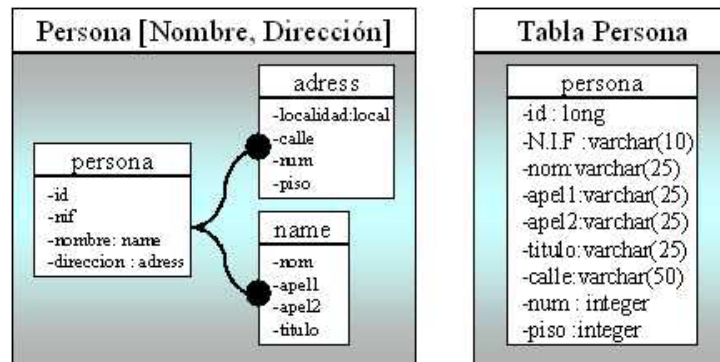


Figura 58: Clase Persona y su composición

El diseño del primer recuadro aparte de ser mucho más claro es muy sencillo de mapear, procedemos de la siguiente manera:

```
<class name="com.hecsua.bean.Person" table="persona">
  <id name="id" column="id" type="long">
    <gen...
  </id>
  <property name="nif" column="nif" type="string" length="10"/>
  <component name="nombre" class="com.hecsua.bean.Name">
    <property name="nom" column="nombre"
type="string" length="25"/>
    <property name="apell" column="apell"
type="string" length="25"/>
    <property name="apel2" column="apel2"
type="string" length="25"/>
    <property name="titulo" column="titulo"
type="string" length="25"/>
  </component>
  <component name="direccion"
class="com.hecsua.bean.Adress">
    <property name="calle" column="calle"
type="string" length="50"/>
    <property name="num" column="numero"
type="integer"/>
    <property name="piso" column="piso"
type="integer"/>
    <componente name="localidad" />
    ...
  </component>
</class>
```

Figura 59: Mapeo de la clase Persona

La tabla persona tiene que contener los elementos correspondientes a los objetos name y adress. Así como al resto de las propiedades de persona. Los tipos de estas propiedades pueden ser cualquier tipo de los soportados por Hibernate, incluso más componentes, colecciones y relaciones. Por último destacar que cuando un componente es nulo todas sus propiedades lo son a la hora de guardarlo en la BBDD. En el caso contrario cuando cargamos un objeto a memoria, si todas las propiedades del elemento son nulas el elemento es nulo.

5.1.5.5.4. Creación de Ficheros XML

El siguiente paso es aplicar todo lo anteriormente explicado en un ejemplo. Partiendo del siguiente diseño, crearemos los ficheros XML de acuerdo a las especificaciones anteriormente explicadas:

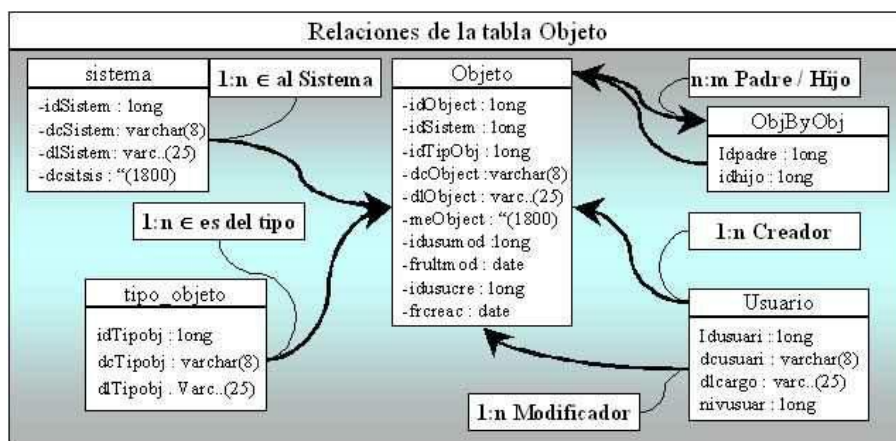


Figura 60: Ejemplo de Diseño Relacional

5.1.5.5.5. Ejemplo: mapeos

Empleador/Empleado

El siguiente modelo de la relación entre Employer y Employee [61] utiliza una clase de entidad (Employment) para representar la asociación. Puede hacer esto cuando podría haber más de un período de empleo para los dos mismos participantes. Se utilizan componentes para modelar los valores monetarios y los nombres de los empleados.

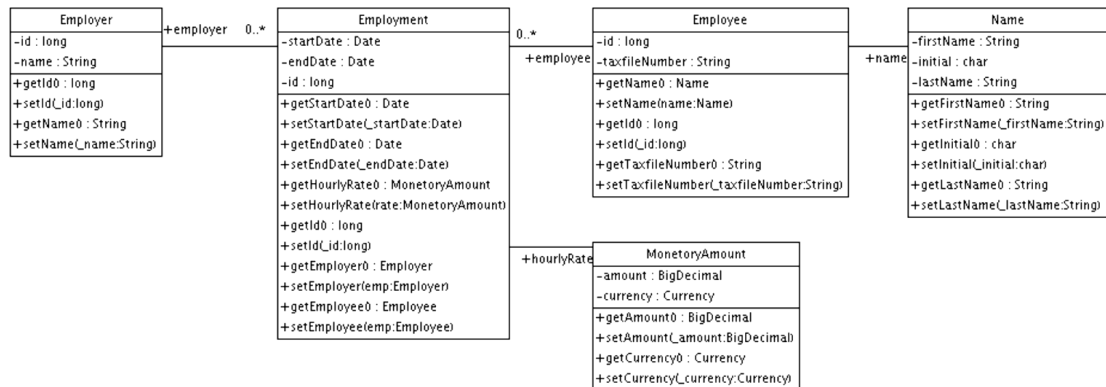


Figura 61: Ejemplo modelo de la relación entre Employer y Employee

El siguiente documento de mapeo (ver Figura 62) [61] representa estas relaciones de manera correcta:

```
<hibernate-mapping>

  <class name="Employer" table="employers">
    <id name="id">
      <generator class="sequence">
        <param name="sequence">
          >employer_id_seq</param>
        </generator>
      </id>
      <property name="name"/>
    </class>

    <class name="Employment" table="employment_periods">
      <id name="id">
        <generator class="sequence">
          <param name="sequence">
            >employment_id_seq</param>
          </generator>
        </id>
        <property name="startDate" column="start_date"/>
        <property name="endDate" column="end_date"/>

        <component name="hourlyRate" class="MonetaryAmount">
          <property name="amount">
            <column name="hourly_rate" sql-type="NUMERIC(12, 2)"/>
          </property>
          <property name="currency" length="12"/>
        </component>

        <many-to-one name="employer" column="employer_id" not-null="true"/>
        <many-to-one name="employee" column="employee_id" not-null="true"/>
      </class>

    <class name="Employee" table="employees">
      <id name="id">
        <generator class="sequence">
          <param name="sequence">
            >employee_id_seq</param>
          </generator>
        </id>
        <property name="taxfileNumber"/>
        <component name="name" class="Name">
          <property name="firstName"/>
          <property name="initial"/>
          <property name="lastName"/>
        </component>
      </class>
</hibernate-mapping>
>
```

Figura 62: Mapeo en XML

El esquema de tablas (ver Figura 63) [61] es el siguiente:

```
create table employers (  
  id BIGINT not null,  
  name VARCHAR(255),  
  primary key (id)  
)  
  
create table employment_periods (  
  id BIGINT not null,  
  hourly_rate NUMERIC(12, 2),  
  currency VARCHAR(12),  
  employee_id BIGINT not null,  
  employer_id BIGINT not null,  
  end_date TIMESTAMP,  
  start_date TIMESTAMP,  
  primary key (id)  
)  
  
create table employees (  
  id BIGINT not null,  
  firstName VARCHAR(255),  
  initial CHAR(1),  
  lastName VARCHAR(255),  
  taxfileNumber VARCHAR(255),  
  primary key (id)  
)  
  
alter table employment_periods  
  add constraint employment_periodsFK0 foreign key (employer_id) references employers  
alter table employment_periods  
  add constraint employment_periodsFK1 foreign key (employee_id) references employees  
create sequence employee_id_seq  
create sequence employment_id_seq  
create sequence employer_id_seq
```

Figura 63: Esquema de tablas del modelo

5.1.6. WebCam

Para el desarrollo del proyecto es necesaria la utilización de una webcam para la captura de imágenes.

Una cámara web o webcam es una pequeña cámara digital conectada a una computadora, la cual puede capturar imágenes y transmitir las en tiempo real [62].

En 1991, se puso en marcha la primera webcam localizada en el departamento de Ciencia y Computación de la Universidad de Cambridge, y desde entonces, se ha propagado a hogares, negocios, oficinas e incluso a calles públicas. Las webcam se suelen usar para uso personal en casas privadas, principalmente para contactar con familiares lejanos, monitorizar casas desde el trabajo o desde otra parte remotamente. Puede ser para vigilar una mascota o por temas de seguridad. En realidad, los usos que se le pueden dar son innumerables [63].

El funcionamiento de una webcam es muy simple: una cámara de video toma imágenes y las pasa a un ordenador que las traduce a lenguaje binario para luego hacer uso de estas para un determinado propósito.

Las cámaras web normalmente están formadas por una lente, un sensor de imagen y la circuitería necesaria para manejarlos.

Existen distintos tipos de lentes, siendo las lentes plásticas las más comunes. Los sensores de imagen pueden ser CCD (*charge coupled device*) o CMOS (*complementary metal oxide semiconductor*). Este último suele ser el habitual en cámaras de bajo coste, aunque eso no signifique necesariamente que cualquier cámara CCD sea mejor que cualquiera CMOS. Las cámaras web para usuarios medios suelen ofrecer una resolución VGA (640x480) con una tasa de unos 30 fotogramas por segundo, si bien en la actualidad están ofreciendo resoluciones medias de 1 a 1,3 MP, actualmente las cámaras de gama alta cuentan con 3, 5, 8 y hasta 10 megapíxeles y son de alta definición.

En la *Figura 64* [64] se muestra la webcam utilizada para la realización de este proyecto. Se puede utilizar cualquier tipo de cámara, sólo tiene que estar instalada en el ordenador.



Figura 64: Webcam Genius iSlim 1320.

La WebCam iSlim 1320 viene equipada con una alta velocidad de conexión USB 2,0 para poder disfrutar de 1.3 mega píxeles de resolución y videos, además viene con micrófono y facilidad de movilizarla al gusto del usuario. Sus características son [64]:

- Sensor de imágenes: 1.3M píxel CMOS
- Interfaz: USB 2.0

- Formato de archivo: JPEG/WMV
- Resolución: 1280 x 1024, 1280 x 960, 800 x 600, 640 x 480, 352 x 288, 320 x 240, 176 x 144, 160 x 120 píxeles
- Resolución de Video: CIF/VGA: 30 fps 1.3M: 15 fps
- Angulo de vista: giro de 360 grados, arriba y abajo de 74 grados
- Conexión: Plug & Play
- Sistema operativo Windows Vista/XP SP2, Linux

E. MATERIALES Y MÉTODOS

Para llevar a cabo el desarrollo del sistema, fue necesario seguir un esquema metodológico, basado en el uso de métodos y técnicas encargadas de facilitar la recolección de datos y de guiar correctamente el presente Trabajo de Titulación.

1. METODOLOGÍA DE DESARROLLO DEL SISTEMA BASADO EN CONOCIMIENTO

Una vez analizado las metodologías (*Ver Sección Revisión de Literatura - Capítulo IV*) existentes y a criterios más importantes en intento de buscar la solución que mejor se adapte al escenario específico que comprende el actual trabajo, se pudo determinar el uso conjunto de las metodologías IDEAL y CommonKADS-RT.

La Metodología IDEAL comenzó identificando la **tarea** de control, que fue tratada con la ingeniería del conocimiento. Todo el entendimiento se basó en esta cuestión, haciendo abstracción de entendimiento del ambiente o dominio del negocio. Su método está orientado al modelado del sistema basado en el conocimiento.

En el mismo sentido de la Ingeniería del Conocimiento, se propuso la Metodología CommonKADS-RT, para la resolución y modelado de las tareas o proceso de control en tiempo real; determinando la condición inicial del sistema. Además con la ayuda de sus modelos se pudo incluirla en la fase II y III de la Metodología IDEAL.

Tomando en cuentas las fases, etapas y los modelos de metodologías empleadas, se realiza su descripción a continuación.

- **Modelo de organización**

En este modelo se analizó y determinó la planta (sistema) a controlar, conjuntamente con las variables de entrada y salida que la componen, estableciendo las condiciones iniciales, especificaciones y procesos que se realizaron para dicho control en áreas interesantes para sistemas inteligentes (*Ver Sección Resultados apartado 1.1. Condiciones Iniciales y 1.2. Especificaciones del Proyecto*); las técnicas de recolección de información que se utilizaron para estudiar la planta fue la observación directa (*Ver Anexo 1*) y la entrevista al operador de la planta (*ver Anexo 2*). Todo lo expuesto anteriormente estuvo dentro de los lineamientos y objetivos del Trabajo de Titulación;

así se dio paso a la FASE I de la Metodología IDEAL para el análisis más profundo del sistema.

- **Fase I: Identificación de la Tarea**

En esta fase se definieron las características del sistema y se especificaron los requerimientos para su solución. Esta fase se divide en 3 etapas:

Etapa 1: Plan de requisitos y Adquisición de Conocimientos

En esta etapa se realizaron las actividades de recolección del conocimiento mediante entrevistas a los expertos (*Ver Anexo 2, Anexo 3 y Anexo 4*), luego se interpretaron y analizaron para proporcionar una visión de las teorías en la organización del conocimiento y estrategias de solución de problemas de control (*Ver Sección Revisión de Literatura - Capítulo I: Teorías de Control, Capítulo II: Visión Artificial y Procesamiento Digital de Imágenes, Capítulo V apartado 5.1.2 ToolBoxes de MatLab*).

Etapa 2: Evaluación y Selección de la Tarea

En esta etapa se realizó un test de viabilidad para determinar si el proyecto tendrá éxito o no; por tan razón, se trabajó con una lista de ítems que debe cumplir el proyecto, estos incluyen características del sistema de control (*ver Sección Resultados - apartado 1.2.1. Test de Viabilidad - TABLA V: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DEL PROBLEMA*), características de la gente involucrada del proyecto (*ver Sección Resultados - apartado 1.2.1. Test de Viabilidad - TABLA VI: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DE PERSONAL*), y asuntos de despliegue (*ver Sección Resultados - apartado 1.2.1. Test de Viabilidad - TABLA VII: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DE DESPLIEGUE*).

Una vez efectuado este test, se determinó que el proyecto reúne características, condiciones técnicas y operativas que aseguraron el cumplimiento de sus metas y objetivos.

Etapa 3: Definición de las características de la tarea

En esta etapa se definieron las características desde el punto de vista de las tareas del sistema de control, para ello se realizó una especificación inicial de los requisitos funcionales y no funcionales (operativos, de interfaz y de soporte) (*ver Sección*

Resultados - apartado 1.3.1. Requerimientos Funcionales y apartado 1.3.2. Requerimientos No Funcionales). Esto permitió identificar necesidades reales de los usuarios finales, ingenieros en conocimiento y expertos. Además se especificaron los recursos materiales y humanos para desarrollar el sistema de control (*ver Sección Discusión – apartado 2. Valoración Técnica Económica Ambiental*), y por último se determinó el plan de desarrollo del sistema (*ver Anexo 13 Cronograma de Actividades*).

- **Fase II: Desarrollo de los Prototipos**

En esta fase se desarrollaron los diferentes prototipos de carácter arquitectónico, que permiten definir y refinar las especificaciones o requerimientos del sistema, esto se realizó en forma gradual hasta conseguir las especificaciones exactas.

Etapa 1: Concepción de la solución

Una vez identificados y perfeccionados los requerimientos en la fase a anterior, se diseñaron los diagramas que representan las funciones con que se estructura el sistema, para lo cual se elaboraron los diagramas que corresponden a los actores de sistema con sus receptoras funciones (*ver Sección Resultados – apartado 2.1.1. Modelado de Caso de Uso*), el diagrama de las relaciones entre todas las entidades comprendidas en el ámbito del dominio del sistema (*ver Sección Resultados – apartado 2.1.2. Modelo del Dominio*), el diagrama de la estructura lógica de la base de datos (*ver Sección Resultados – apartado 2.1.2. Modelo de la Base de Datos*).

Etapa 2: Adquisición de Conocimientos y Conceptualización de Conocimientos

En esta etapa se adquirió el conocimiento, iniciando con la extracción de conocimientos públicos de sus fuentes (libros, documentos webs y proyectos de tesis) mediante la conceptualización de la misma (*ver Sección Resultados – apartado 2.2.1. Diccionario de Términos*), de igual manera se formalizó la educación de conocimientos privados del experto, previo al filtrado de toda la información, tomando como referencia las tareas más importantes. Para entender el ambiente o dominio del sistema de control se diseñaron las relaciones del mismo, mediante la elaboración de la descripción de los casos de uso, prototipo rápido de la interfaz gráfica, diagramas de robustez, diagramas de secuencia, diagrama de componentes, diagrama de paquetes

y despliegue (ver *Sección Resultados – apartado 2.2.2.1. Modelado Dinámico de la Aplicación*).

Para modelar el prototipo del conocimiento del requerimiento más importante del sistema de control propuesto en este Trabajo de Titulación, se hizo uso del Modelo de Tareas de Alto Nivel, Modelo de Tareas en Tiempo Real, Modelo del Conocimientos, Modelo del Diseño de la Metodología CommonKADS-RT.

Modelo de Tareas de Alto Nivel (TAN)

En este modelo se especificaron las características de la tarea (Caso de uso) que más requiere del conocimiento en el sistema de control, denominada Operar Inspección.

El comportamiento global del proceso de tareas del sistema de control de inspección, se modeló mediante diagramas que especificaron las técnicas y herramientas inteligentes que se usaron en el sistema (ver *Sección Resultados – apartado 2.2.2.5 Modelo de tareas de alto nivel (TAN) - Figura 139*). Por consiguiente se elaboró el formulario donde se describieron los procesos de las tareas y sus acciones para cumplir su objetivo (ver *Sección Resultados – apartado 2.2.2.5 Modelo de tareas de alto nivel (TAN) - TABLA XII*).

Modelo de Tareas en Tiempo Real

En este modelo se diagramaron las tareas en tiempo real, es decir todas las actividades u operaciones técnicas de acuerdo a las teorías de control inteligente, teniendo en cuenta sus componentes de percepción, cognitivos y de acción, estas tareas fueron desarrollados en los lenguajes de programación para proporcionar una respuesta rápida al entorno (ver *Sección Resultados – apartado 2.2.2.6. Modelo de tareas en tiempo real - Figura 141: Tareas de alto nivel TAN*).

Modelo del Conocimientos

En este modelo se mostró el conocimiento llevado al sistema para solucionar las TAN incluyendo las especificación temporales.

Por ello, se analizaron las variables que influyen directa e indirectamente en el dominio del sistema de inspección (ver *Sección Resultados – apartado 2.2.2.7.1. Factores o variables del proceso de control del micro mundo*). Para tener controladas algunas de

estas variables y además para trabajar en condiciones deseables, se diseñó un banco de pruebas (ver *Figura 66* y *Figura 67*), simulando el ambiente o dominio del micro mundo para el proceso de control de inspección. Con base en lo anterior, además se diseñó el diagrama de conocimientos del sistema de inspección (ver *Sección Resultados – apartado 2.2.2.7. Modelo del conocimiento - Figura 142: Diagrama de esquema de dominio para el sistema de inspección en el micro mundo*) y en cuanto al conocimiento de las tareas se diseñó el diagrama del planificador de las mismas (ver *Sección Resultados – apartado 2.2.2.7. Modelo del conocimiento - Figura 143: Diagrama de transición de estados del planificador*).

Modelo del Diseño

Basándose en los requerimientos y especificación de los anteriores modelos y etapas, este modelo ayudó a realizar la especificación técnica para la implementación del sistema.

Para desarrollar la aplicación, fue necesario utilizar el lenguaje de programación matemática MATLAB, con esto se dio paso a la creación y codificación del algoritmo que permitió percibir una muestra significativa de imágenes tomadas a través una cámara, analizarlas mediante técnicas de procesamiento digital de imágenes y extraer cuatro características (variables) que representaron el patrón de información de la imagen para su comparación, interpretación y razonamiento (ver *Sección Resultados – apartado 2.2.2.8.2. Desarrollo del algoritmo en tiempo diferido*).

Continuando en la Etapa 2, se procedió a la identificación los conocimientos extraídos y calculados mediante el algoritmo mencionado, para modelizar el comportamiento del experto. De este modo, se realizó una matriz de reglas (ver *Sección Resultados – apartado 2.2.3.1. Análisis de conocimientos tácticos - TABLA XVI*) y se determinó la estrategia de control de lógica difusa, esta nos permitió manejar y representar las variables de forma lingüística asignando una respuesta deseada a su comportamiento (ver *Sección Resultados – apartado 2.2.3.1. Análisis de conocimientos tácticos - TABLA XVII y TABLA XVIII*).

Etapas 3: Formalización de Conocimientos y Definición de la Arquitectura

En esta etapa se representaron los formalismos para representar los conocimientos de la etapa anterior, por ello se estableció la base de conocimientos (ver *Sección*

Resultados – apartado 2.3.1. Motor de inferencia y base de conocimientos - TABLA XIX), asimismo se recurrió a la utilización de la herramienta de software para implementar la estrategia de control inteligente y su modelo estructural, la misma que reunió mejor las necesidades para el sistema, en cuanto a velocidad de procesamiento, sencillez y robustez. Dicha herramienta fue el Toolbox de Lógica Fuzzy de MATLAB, que mediante el uso del modelo ANFIS, se diseñó el Sistema Controlador de Inferencia Difuso (FIS) (ver Sección Resultados – apartado 2.3.1. Motor de inferencia y base de conocimientos - TABLA XX).

Para construir el FIS se utilizaron datos extraídos del algoritmo desarrollado anteriormente para el entrenamiento y validación (ver Sección Resultados – apartado 2.3.2.1. Preparación de datos con la ayuda del software MatLab). Adicionalmente se inicializaron los parámetros del FIS como: las funciones de pertenencia y las reglas de inferencia a partir de los datos dados por el entrenamiento (Pares de datos de entrada/salida) (ver Sección Resultados – apartado 2.3.2.2. Construcción del sistema con Fuzzy Logic Toolbox: Aprendizaje e Inferencia a través de ANFIS).

Etapas 4: Selección de la Herramienta e Implementación

Debido a que en etapa anterior se utilizó adecuadamente la herramienta de construcción de FIS, la implementación de esta es automática, pero adicionalmente se utilizó el lenguaje de programación JAVA conjuntamente con el IDE NetBeans para la codificación de las clases y atributos de los modelos basados en los diagramas ya elaborados, permitiendo la construcción de la interfaz gráfica y otras funciones requeridas, además se estableció la conexión entre JAVA y MATLAB mediante la librería JMatLink proporcionando al sistema una funcionalidad potente y estable.

Etapas 5: Validación y Evaluación del Prototipo

En esta etapa se probaron y evaluaron los prototipos construidos anteriormente de forma independiente como: el algoritmo de inspección de las imágenes, el FIS y la interfaz gráfica del usuario, esto nos permitió comparar resultados e identificar errores para poder corregirlos de acuerdo a las necesidades de las tareas establecidas.

- **Fase III: Ejecución de la Construcción del Sistema Integrado**

Esta fase consta de tres etapas que se describen a continuación:

Etapa 1: Requisitos y Diseño de la Integración con otros Sistemas

En esta etapa se evaluaron y estudiaron todas las herramientas de soporte de software y hardware, en el cual fue programado, configurado e interactúa todo sistema de control (ver *Sección Resultados – apartado 3.1. Requisitos y diseño de la integración con otros sistemas - TABLA XXIII: CARACTERÍSTICAS MÍNIMAS DE HARDWARE y TABLA XXIV: SOFTWARE INSTALADO*).

Etapa 2: Implementación y Evaluación de la Integración

Conjuntamente con la etapa anterior se implementó la integración de todas las herramientas y los prototipos inteligentes desarrollados, consiguiendo el sistema final de control denominado ICOLab, posteriormente se procedió a realizar las pruebas de aceptabilidad con los posibles usuarios, esto permitió validar la accesibilidad, navegabilidad, usabilidad y funcionalidad de sistema.

Estas pruebas se llevaron a efecto en el Laboratorio de Automatización y Control Industrial del AEIRNNR, lugar donde se encuentra implementado el sistema ICOLab, para ello se eligió una muestra de la población de estudiantes y docentes del AEIRNNR, a los cuales se les aplicó el método de la encuesta para obtener su grado de satisfacción frente al sistema (ver *Sección Resultados – apartado 3.2.1. Pruebas de Usabilidad (Funcionalidad, Diseño y Presentación)*).

Adicionalmente, se efectuaron las pruebas de funcionalidad de ICOLab en tiempo real analizando una muestra de imágenes de botellas reales en el escenario establecido, estas pruebas dieron como resultado una eficiencia global del sistema del 95%, valor considerado como aceptable y eficiente para este tipo de sistemas (ver *Sección Resultados – apartado 3.2.2. Funcionalidad del sistema ICOLab: Resultados obtenidos en tiempo real*).

Etapa 3: Aceptación de sistema por el usuario

Luego de haberse aplicado las encuestas, se determinó que existe una gran satisfacción por parte de los expertos, usuarios y del ingeniero del conocimiento por la eficiencia y fiabilidad del sistema (ver *Sección Resultados – apartado 3.3. Aceptación del sistema por el usuario – Análisis de Prueba: Administrador y Análisis de Prueba: Estudiantes*).

2. MÉTODOS DE INVESTIGACIÓN

En cuanto a los métodos de trabajo científico se han utilizado los siguientes:

- **Método Documental:** La investigación basada en documentos se dedica a reunir, seleccionar y analizar datos que están en forma de “documentos” producidos por la sociedad para estudiar un fenómeno determinado [65]. Es decir, al que se puede acudir como fuentes o referencia en cualquier momento o lugar sin que se altere su naturaleza o sentido, para que aporte información o rinda cuenta de una realidad o acontecimiento.

Con este método se analizó la información escrita sobre el tema objeto de estudio, de tal manera que se realizó sobre la base de documentos o revisión bibliográfica (Tesis de Magister, Tesis Doctorales, Publicaciones Científicas, Revistas de Universidades, Libros, etc.).

- **Método Descriptivo:** Tipo de investigación que describe de modo sistemático las características de una población, situación o área de interés. Aquí los investigadores recogen los datos sobre la base de una hipótesis o teoría, exponen y resumen la información de manera cuidadosa y luego analizan minuciosamente los resultados, a fin de extraer generalizaciones significativas que contribuyan al conocimiento [66].

Con esto se trata de explicar cómo podemos cuantificar las imágenes, con la ayuda de técnicas del procesamiento de imágenes como la binarización, umbralización, segmentación y otros patrones aplicados.

- **Método experimental:** En la investigación de enfoque experimental el investigador manipula una o más variables de estudio, para controlar el aumento o disminución de esas variables y su efecto en las conductas observadas. Dicho de otra forma, un experimento consiste en hacer un cambio en el valor de una variable (variable independiente) y observar su efecto en otra variable (variable dependiente). Esto se lleva a cabo en condiciones rigurosamente controladas, con el fin de describir de qué modo o por qué causa se produce una situación o acontecimiento particular [67].

Esto llevo a construir un banco de pruebas y tomar varias muestras de imágenes para su posterior análisis, en el banco de pruebas se controlan varios factores como la luminosidad, distancia, posición del objeto, entre otros.

3. TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

Durante el desarrollo del presente trabajo de titulación fue necesario utilizar diversas fuentes de información para la obtención de la información de los actores que participan en este campo: como los operadores de la planta embotelladora, ingenieros en control automático y visión artificial (procesamiento digital de imágenes), además de las personas que hacen uso del Laboratorio de Automatización y Control Industrial como estudiantes y docentes.

- **Observación directa:** Esta técnica consistió en observar el comportamiento de la planta embotelladora, sus procesos y actividades para obtener el producto terminado, con el fin de determinar sus variables y las personas (expertos) que intervienen en dicho procesamiento.
- **Técnica de la Entrevista:** Con el objetivo de obtener mayor información sobre el funcionamiento y problemas se planteó una entrevista al operador de la planta embotelladora, dicha entrevista nos permitió conocer los factores de estudio para el proceso del control implementado. También se desarrolló una entrevista al experto en control automático con el fin de adquirir conocimiento de técnicas de control inteligentes para procesos de sistemas de comportamientos no lineales. Finalmente se aplicó una entrevista al experto en visión artificial para entender cómo se pueden aplicar y manipular las técnicas de procesamiento en imágenes, además de herramientas tecnológicas en la que se pudieron implementar dichas técnicas.

Esta técnica permitió identificar piezas claves del conocimiento, conceptos y reglas del campo del control de procesos inteligentes y así explorar alternativas de resolución para el desarrollo del sistema ICOLab.

- **Técnica de la Encuesta:** Se aplicó esta técnica a los posibles usuarios, con el fin recoger la opinión sobre aspectos del sistema ICOLab y medir el grado de satisfacción dichos usuarios. Estas encuestas se realizaron en el Laboratorio de Automatización y Control Industrial del AEIRNNR, mediante una presentación y demostración del funcionamiento del sistema en tiempo real.

F. RESULTADOS

A continuación se detallan las fases realizadas correspondientes a las metodologías mencionadas en la sección E. *Materiales y Métodos*.

1. Modelo de la organización

1.1. Condiciones iniciales

En este proyecto se tomó como escenario a la inspección visual y clasificación de botellas de 375 cm³, que hace un operario después del lavado de cada una de ellas dentro de las etapas del proceso de producción de envasado de una embotelladora (ver Figura 65). El proceso de clasificación se ha estudiado con los expertos y se han encontrado algunos inconvenientes, es aquí donde se requiera una inspección visual automatizada, el sistema de control en tiempo real propone desarrollar un procedimiento novedoso basado en Lógica Fuzzy, para clasificar las botellas limpias de las sucias.



Figura 65: Escenario de proceso de envasado

Para esto se escoge el software MatLab, versión R2009a como herramienta en la elaboración del algoritmo, porque su especialidad son las matrices, lo cual se podrá apreciar más adelante en este trabajo. El Trabajo de Titulación se centra en el área de procesamiento digital de imágenes (PDI) para dar lugar a la creación de un algoritmo que permita extraer características de dichas imágenes y mediante el modelado y entrenamiento de un sistema neuro-difuso detectar las botellas de vidrio transparentes

que contengan objetos extraños, y realizar un aporte teniendo en cuenta las prácticas, experiencias, y resultados obtenidos para incrementar el conocimiento en docentes, estudiantes e investigadores, incentivando el desarrollo de aplicaciones que mejoren las condiciones de vida y trabajo de la región.

1.2. Especificaciones del proyecto

Las imágenes son tomadas con una cámara digital webcam en el ‘banco de pruebas’ desarrollado por la persona autora de este trabajo de titulación. La imagen es capturada desde la interfaz gráfica de Java, haciendo uso del paquete de librería de Java Media Framework, En la *Figura 66* y *Figura 67* se observa el Banco de Pruebas que tiene forma de paralelepípedo.



Figura 66: Banco de Pruebas – Lado lateral Izquierdo



Figura 67: Banco de Pruebas – Lado lateral Derecho

En el presente Trabajo de Titulación se hace uso de motor trifásico de inducción “Jaula de Ardilla”¹⁶ CA en lazo cerrado que accionar una banda transportadora pequeña, que funciona por pequeños intervalos de tiempo y con muy baja resistencia. Este motor está instalado en el Laboratorio de Automatización y Control, y se utiliza para prácticas académicas de los estudiantes del Área de Energía.

¹⁶ Motor eléctrico es un dispositivo capaz de transformar la energía eléctrica en energía mecánica, es decir, puede producir movimiento al convertir en trabajo la energía eléctrica proveniente de la red o almacenada en un banco de baterías.

Por otro lado para la captura de la imagen, la botella (Ejemplo: La Cañita y Cantaclaro en presentación de 375 cm³) siempre está en la misma posición o cercana a ella, se puede o no presentar movimiento.

Existe toda una variedad de objetos frecuentemente encontrados en las botellas, solo se trabajará con cinco de ellos; estos son: anillos de las tapas, plástico de seguridad de las tapas, colillas de cigarrillo, etiquetas e impurezas.

Como herramienta de diseño e implementación, se utiliza el ToolBox de Procesamiento de Imágenes del software MATLAB R2009a.

Proceso a realizar

El proceso en pasos está dado en la siguiente secuencia:

- Inicialmente se debe activar cámara y la conexiones de software; y hacer un chequeo de su estado, Este procedimiento ya viene predefinido en el sistema
- Una vez listo, el sistema debe comenzar la inspección del objeto, Para esto debe hacer una serie de tareas que más adelante se definirán.
- Luego de obtener el resultado, debe identificarlo de acuerdo a los valores que se ha entrenado.
- Posteriormente, el sistema visualiza el resultado.

El proceso general se representa en el siguiente diagrama (ver Figura 68):

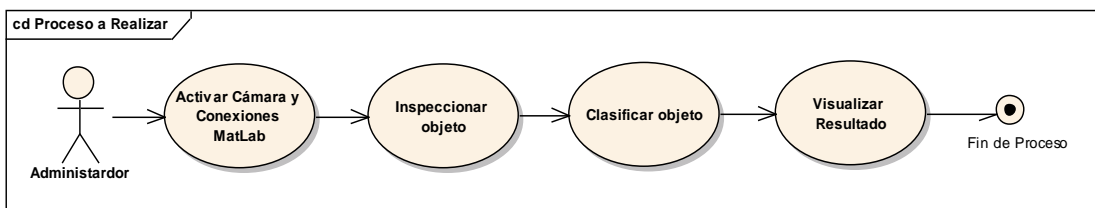


Figura 68: Proceso que debe realizar el sistema controlador para cumplir con el objetivo final

Como se puede observar, en este proceso sólo interviene el actor humano para definir las condiciones iniciales o para iniciar la ejecución del sistema, Pero durante el desarrollo de la tarea, es el sistema en si autónomo en su realización.

En las Figura 69 y Figura 70 se puede apreciar de forma más forma gráfica el funcionamiento del algoritmo a realizar para dicho objetivo.

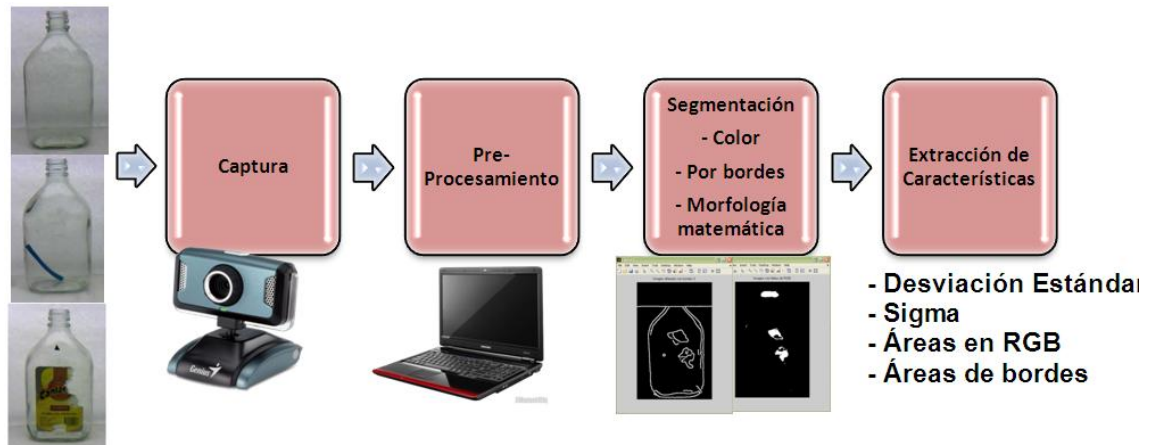


Figura 69: Primera parte del Algoritmo: Procesamiento Digital de Imágenes

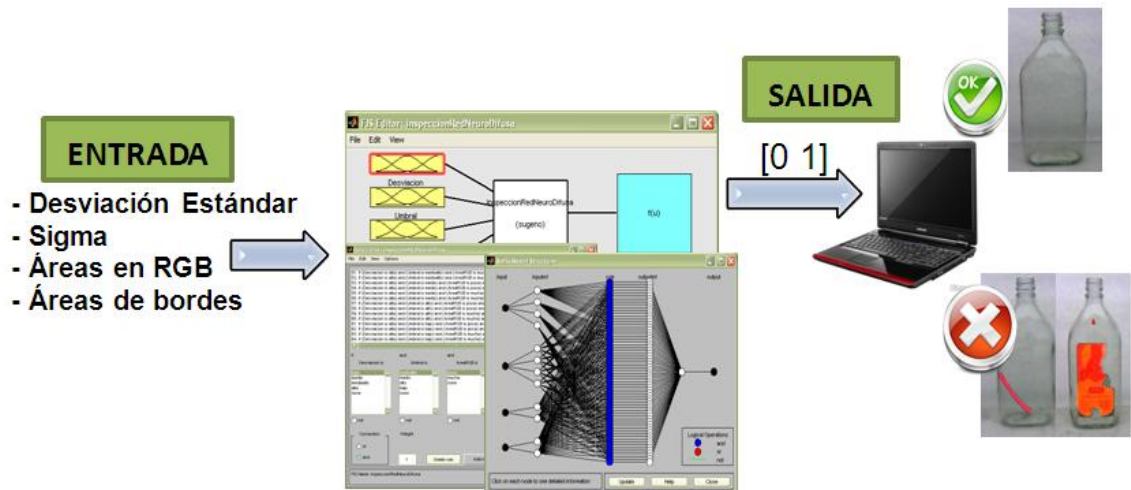


Figura 70: Segunda parte del Algoritmo: Evaluación del FIS (Controlador Difuso)

2. FASE 1: IDENTIFICACIÓN DE LA TAREA

Etapa 1.1: Plan de requisitos y adquisición de conocimientos

1.1.1. Proceso de Adquisición de Conocimientos

1.1.1.1. Primeras reuniones y evaluación de viabilidad: Estas reuniones con el o los expertos han permitido lo siguiente:

- Determinar los objetivos del Sistema Experto.
- Precisar el alcance y ámbito del proyecto.
- Determinar la problemática del proyecto.
- Conocer las necesidades o requerimientos.

- Recopilar información del tema.
- Comprometer la participación del experto principal en la preparación y evaluación de las sesiones.
- Detectar posibles candidatos a Experto para realizar el estudio de viabilidad.

1.1.1.2. Extracción de conocimientos: Las sesiones de extracción de conocimientos han permitido lo siguiente:

- Revisar la bibliografía sugerida y/o otorgada por el o los expertos.
- Familiarizarse con la terminología del tema.
- Obtener información por medio de exposiciones y conferencias.

1.1.1.3. Educción de conocimientos con expertos: Estas reuniones con el grupo de expertos han permitido lo siguiente:

- Analizar la información obtenida.
- Detallar las características del proyecto que se van a evaluar.
- Delimitar las funciones y necesidades de un primer prototipo del SE con la información obtenida hasta el momento.

1.1.1.4. Educción de conocimientos: Este proceso han permitido lo siguiente:

- Analizar la consistencia de la información obtenida en el proceso de educación de conocimientos con el grupo de expertos.
- Corroborar el grado de correspondencia entre el modelo mental del experto y la representación obtenida.

Etapa 1.2: Evaluación y selección de la tarea

1.2.1 Test de Viabilidad

Según Beckman (1991) para determinar la viabilidad de un sistema experto se forma una lista de requerimientos para considerar, luego a cada tema se le asigna un peso (entre 0 y 10) que refleja la importancia de cada tema durante la evaluación de un proyecto, los números (entre 0 y 10) son atribuidos a cada tema que refleja el grado de creencia en el tema. Este valor es luego multiplicado por el valor del tema para establecer un puntaje por el tema. Todos los puntajes son luego añadidos y divididos por la suma de los pesos del tema. Este número es limitado entre 0 y 10, y

proporciona una estimación de determinación de viabilidad del proyecto, Este Test se ve reflejado en la TABLA V, TABLA VI, TABLA VII y TABLA VIII.

TABLA V: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DEL PROBLEMA

ASUNTOS DE VIABILIDAD DEL PROBLEMA			
PUNTAJE = PESO * VALOR			ASUNTO
56	7	8	Conocimiento experto necesitado
72	9	8	Los pasos de solución de problema son definibles
56	7	8	Conocimiento simbólico usado
64	8	8	Heurísticas usadas
90	10	9	El problema es solucionable
0	8	0	Existen sistemas exitosos
81	9	9	El problema es bien enfocado
42	6	7	El problema es razonablemente complejo
56	7	8	El problema es estable
54	9	6	Conocimiento incierto utilizado
35	5	7	No determinístico
48	6	8	Solución más de una recomendación
654	91		
Puntaje Total	Peso Total		$Viabilidad\ del\ Problema = \frac{Puntaje\ Total}{Peso\ Total} = 7,19$

Fuente: Durkin, J. 1994 "Expert Systems: Design and Development".

TABLA VI: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DE PERSONAL

ASUNTOS DE VIABILIDAD DEL PERSONAL			
PUNTAJE = PESO * VALOR			ASUNTO
EXPERTO DEL DOMINIO			
70	7	10	El experto puede comunicar el conocimiento
72	9	8	El experto puede dedicar tiempo
70	7	10	El experto es cooperativo
212	23		
Puntaje Total	Peso Total		$Viabilidad\ del\ experto = \frac{Puntaje\ Total}{Peso\ Total} = 9,22$
INGENIERO DEL CONOCIMIENTO			
64	8	8	Buenas habilidades de comunicación
48	8	6	Puede relacionar el problema al software
45	9	5	Tiene destrezas de programación de SE
81	9	9	Puede dedicar tiempo
238	34		
Puntaje Total	Peso Total		$Viab.\ del\ ing.\ en\ conocimiento = \frac{Puntaje\ Total}{Peso\ Total} = 7$

USUARIO FINAL			
48	6	8	El usuario final puede dedicar tiempo
49	7	7	El usuario final es receptivo al cambio
70	7	10	El usuario final es cooperativo
167	20		
Puntaje Total	Peso Total		$Viab. del usuario final = \frac{Puntaje Total}{Peso Total} = 8,35$

Fuente: Durkin, J. 1994 "Expert Systems: Design and Development".

TABLA VII: FORMULARIO DE DETERMINACIÓN DE VIABILIDAD DE DESPLIEGUE

ASUNTOS DE VIABILIDAD DEL DESPLIEGUE			
PUNTAJE = PESO * VALOR			ASUNTO
56	7	8	El sistema puede ser introducido fácilmente
72	9	8	El sistema puede ser mantenido
49	7	7	El sistema no tiene una ruta crítica
72	9	8	El sistema puede ser integrado con recursos existentes
56	7	8	Entrenamiento disponible
305	39		
Puntaje Total	Peso Total		$Viabilidad del Despliegue = \frac{Puntaje Total}{Peso Total} = 7,82$

Fuente: Durkin, J. 1994 "Expert Systems: Design and Development".

Para ilustrar, asumir que se evaluó un proyecto candidato que resulto de los siguientes puntajes para cada categoría:

TABLA VIII: EVALUACIÓN DEL PROYECTO

CATEGORÍA	PUNTAJE TOTAL	PESO TOTAL
Problema	654	91
Personal	617	77
Despliegue	305	39
Total	1576	207

$$Viabilidad del Proyecto = \frac{1576}{207} = 7,61$$

Etapa 1.3: Definición de las características del sistema

1.3.1 Requerimientos Funcionales

El sistema permitirá:

TABLA IX: REQUERIMIENTOS FUNCIONALES.

Código	Descripción	Categoría	Aceptado
RF001	• Al administrador ingresar al sistema a través de su cuenta de administrador con su respectivo ID y contraseña.	Evidente	()
RF002	• Al Administrador visualizar la lista de empleados y usuarios.	Evidente	()
RF003	• Al administrador ingresar datos de nuevo personal.	Evidente	()
RF004	• Al administrador modificar datos del personal.	Evidente	()
RF005	• Al administrador eliminar datos del personal.	Evidente	()
RF006	• Al administrador crear un Usuario con su respectivo ID, contraseña y permiso de acceso al sistema	Evidente	()
RF007	• Al administrador modificar el ID Usuario contraseña y permiso de acceso al sistema.	Evidente	()
RF008	• Al administrador inhabilitar el ID Usuario y clave de otro administrador.	Evidente	()
RF009	• Al administrador operar el control de inspección de la imagen (objeto botella).	Oculto	()
RF010	• Al administrador visualizar el resultado de la inspección y clasificación de la imagen (objeto botella).	Evidente	()
RF011	• Al administrador conectar la cámara en saco de fallo.	Evidente	()
RF012	• Al administrador conectar la MatLab en saco de fallo.	Evidente	()
RF013	• Al administrador cambiar y/o crear el directorio donde se guardan las imágenes.	Evidente	()
RF014	• Al administrador abrir el directorio donde se guardan las imágenes.	Evidente	()
RF015	• Al usuario operar el control de inspección de la imagen (objeto botella).	Oculto	()
RF016	• Al usuario visualizar el resultado de la inspección y clasificación de la imagen (objeto botella).	Evidente	()

RF017	• Al usuario conectar la cámara en caso de fallo.	Evidente	()
RF018	• Al usuario conectar la MatLab en caso de fallo.	Evidente	()
RF019	• Al usuario cambiar y/o crear el directorio donde se guardan las imágenes.	Evidente	()
RF020	• Al usuario abrir el directorio donde se guardan las imágenes.	Evidente	()
RF021	• Al administrador consultar e imprimir un reporte estadístico de los resultados acuerdo a las fechas seleccionadas.	Evidente	()
RF022	• Al usuario consultar e imprimir un reporte estadístico de los resultados acuerdo a las fechas seleccionadas.	Evidente	()

1.3.2. Requerimientos No Funcionales

TABLA X: REQUERIMIENTOS NO FUNCIONALES.

Código	Descripción	Aceptado
RNF001	• El sistema constara de una interfaz gráfica amigable y fácil de usar	()
RNF002	• El sistema será desarrollado bajo la plataforma Java y MatLab.	()
RNF003	• El sistema será mono plataforma.	()
RNF004	• El sistema será multiusuario: Administrador y Usuario.	()
RNF005	• El sistema funcionara con el reloj del computador respectivamente actualizado.	()
RNF006	• El sistema funcionará con un gestor de base de datos MySQL	()

1.3.3 Casos de prueba o ensayo

Debemos de tener en cuenta que hoy en día los responsables expertos de todo el mundo industrializado reconocen que la alta calidad del producto de software se traduce en ahorro de costo y en una mejora general; por lo que se debe calificar que el desarrollador del presente proyecto han adquirido una experiencia relativamente buena en cuanto al desarrollo de sistemas expertos.

También se visitó y realizó la observación en una embotelladora del cantón Loja, relacionada con los procesos y tratamientos con botellas para así tener un amplio

conocimiento de dicho tema. Lo cual nos permitió profundizar en el mismo y poder desarrollar nuestro Sistema Experto de control basado en conocimiento (Ver Anexo 1).

Dichas pruebas efectuadas serán tomadas como referencias para sus primeros ensayos en el Laboratorio de Automatización y Control del AEIRNNR y así determinar la fiabilidad del sistema experto.

1.3.4. Recursos necesarios

1.3.4.1. Recursos Humanos

Los datos se pueden verificar en la TABLA XXXI: *RECURSOS HUMANOS*.

1.3.4.2. Recursos Técnicos y Tecnológicos

Los datos se pueden verificar en la TABLA XXXII: *RECURSOS TÉCNICOS Y TECNOLÓGICOS*.

1.3.4.3. Recursos Económicos

- Todo el desarrollo del proyecto es financiado por el realizador del Trabajo de Titulación.

1.3.5. Análisis de costo / beneficio

Cuando se analizan los sistemas se deben tener en cuenta los beneficios y costos tangibles e intangibles.

1.3.5.1. Costo del Proyecto

Los costos principales del proyecto son establecidos por los gastos de trabajo y software. La cuenta de gastos de trabajo para el tiempo gastado en el proyecto es por el ingeniero del conocimiento, el experto de dominio y el usuario final. La opción del software de desarrollo del sistema experto está basada en la naturaleza del problema y las facilidades del computador y/o recursos de la organización.

1.3.5.1.1. Costos tangibles

- Costo de equipos y computadoras \$ 2568,99
- Costo de materiales de oficina y servicios \$ 279,00

- Costo del tiempo de los analistas \$ 0.00
- Costo del tiempo del programador \$ 2880,00

1.3.5.1.2. Costos intangibles

- Toma de decisiones ineficaz debido a la información inoportuna o inaccesible.
- Insatisfacción del usuario.
- Deterioro de la imagen institucional.

1.3.5.2. Beneficios del Proyecto

El beneficio de desarrollar un sistema experto puede ser medido en una de las cuatro maneras: productividad mejorada, costos bajos, calidad mejorada o un asunto muy tangible pero importante - imagen mejorada.

- Productividad mejorada
 - ✓ Mejores Decisiones
 - ✓ Decisiones más rápidas
 - ✓ Propaga especialización
- Costos más bajos
 - ✓ Reduce costos de trabajo
 - ✓ Mejora uso de material
- Calidad mejorada
 - ✓ Producto de mejor calidad
 - ✓ Servicios superiores
 - ✓ Proporciona entrenamiento
- Imagen mejorada
 - ✓ Innovador

1.3.5.2.1. Beneficios tangibles

- Acceso a la información en forma más oportuna.
- Disminución en el tiempo empleado para cumplir tareas específicas.
- Aumento de la velocidad de procesamiento de la información.

1.3.5.2.2. Beneficios intangibles

- Mejorar el proceso de toma de decisión con respecto a la clasificación de los objetos.
- Ser más competitivo en el campo de control de procesos y control inteligentes.
- Mantener una buena imagen e incrementar la satisfacción de la institución, estudiantes, usuarios, etc.

1.3.6. Calendario o Cronograma de actividades.

El cronograma de actividades se lo puede apreciar en el Anexo 13.

3. FASE 2: DESARROLLO DE LOS PROTOTIPOS

Etapa 2.1: Concepción de la solución

2.1.1. Modelado de Caso de Uso

Se presentan los diagramas de casos de uso del sistema (ver Figura 71) obtenidos durante el proceso de especificación de requisitos, los cuales permiten mostrar a alto nivel las funcionalidades que el sistema realiza.

Catálogo de Actores

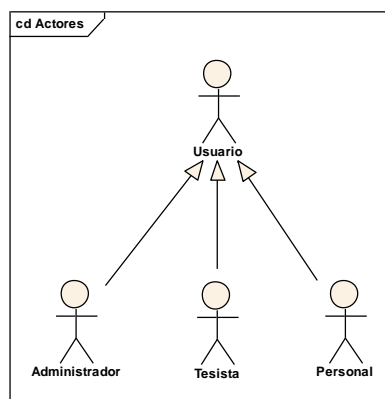


Figura 71: Catálogo Actores

Modelo de Caso de Uso General

Administrador: Es el usuario encargado de realizar el mantenimiento de personal y usuario del sistema, prácticamente el control total del sistema (ver Figura 72).

Usuario (Personal): Son aquellos usuarios que interactúan con el sistema. Es el usuario que puede realizar el control de los objetos de las botellas, también puede visualizar la evaluación de la inspección de dicho control y obtener reportes hasta la fecha del último control (ver Figura 72).

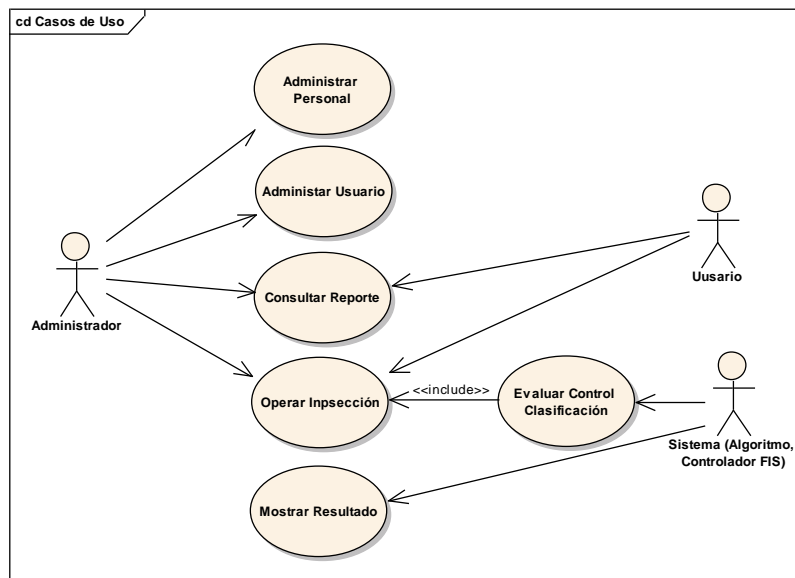


Figura 72: Diagrama de Casos de Uso

2.1.2. Modelo del Dominio

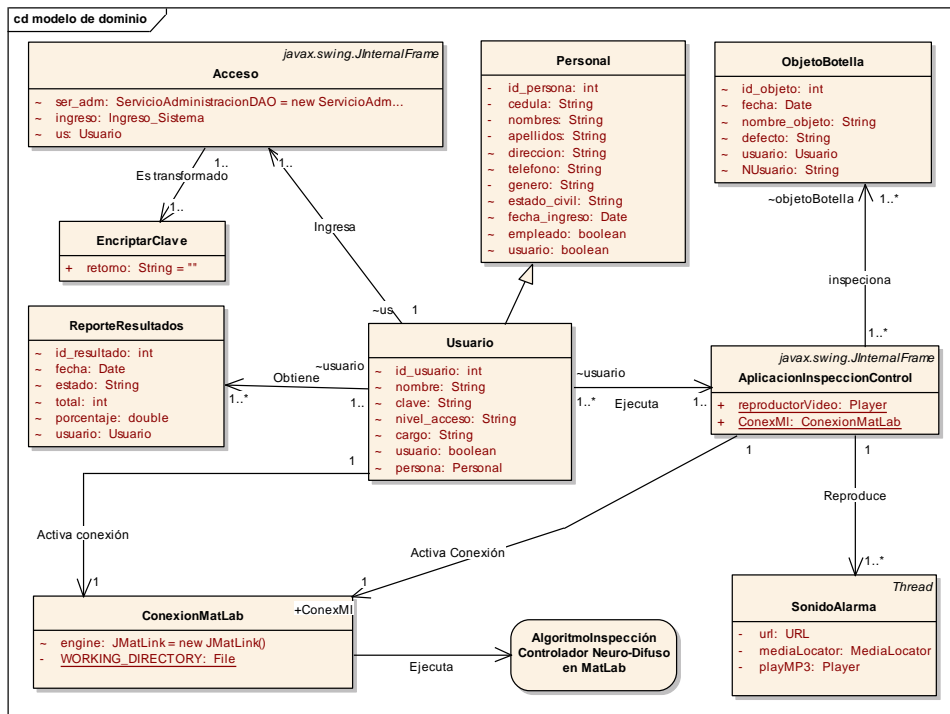


Figura 73: Modelo del Dominio

2.1.2. Modelo de la Base de Datos

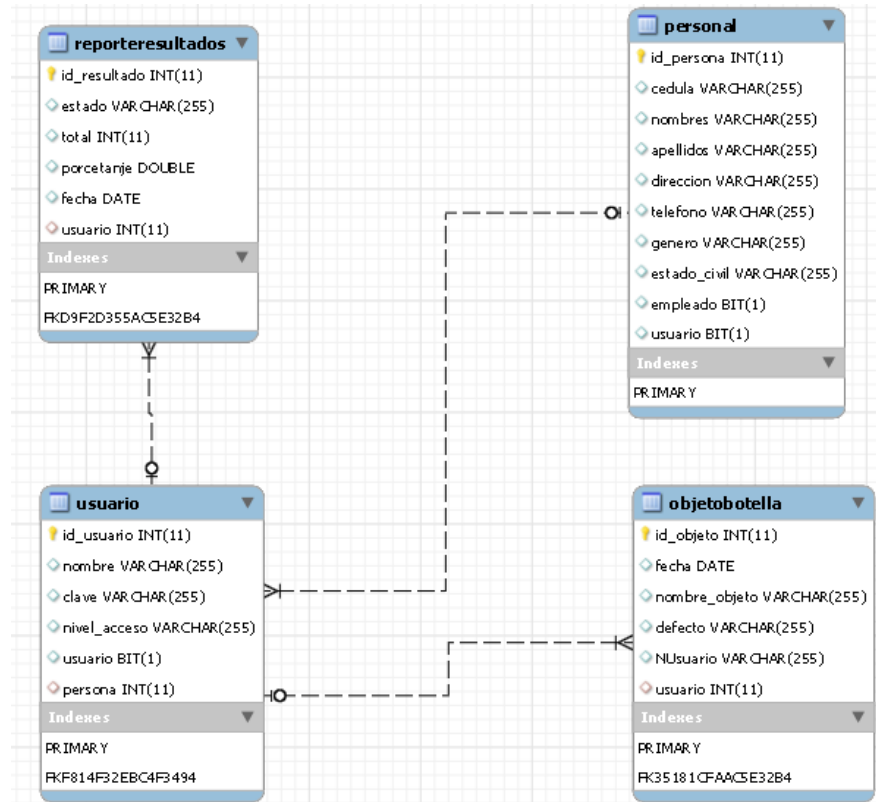


Figura 74: Modelo de la Base de Datos

Etapa 2.2: Adquisición y Conceptualización de los conocimientos

2.2.1. Diccionario de Términos

TABLA XI: DICCIONARIO DE TÉRMINOS.

Término	Categoría	Comentario
Administrador (actor)	Término	Persona responsable de la administración de la información que se maneja en el sistema y otras actividades operativas.
Administrar Personal	Caso de Uso	Descripción del proceso en el cual el usuario administrador registra los datos del personal (Operario, etc.).
Administrar Usuario	Caso de Uso	Descripción del proceso en el cual el usuario administrador registra los datos de acceso y asigna el cargo a los usuarios del sistema
Base de conocimientos	Término	Es la parte del sistema experto que contiene el conocimiento sobre el ambiente o dominio. Hay que obtener el conocimiento del experto y

		codificarlo en la base de conocimientos. Una forma clásica de representar el conocimiento en un sistema experto son las reglas.
Base de datos	Término	Es una colección estructurada de tablas que contienen datos.
Control	Término	Manipulación de un conjunto de las magnitudes o acciones para obtener una respuesta deseada
Control Difuso	Término	Sistema experto en tiempo real que implementa una parte de la forma de operar de una persona o de un proceso con experiencia en ingeniería.
DAO (Data Access Object)	Término Sigla	Es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo
Hibernate	Librería	Es una capa de persistencia objeto/relacional y un generador de sentencias sql. Te permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos.
In-Agent	Término	Es una entidad interna que posee el conocimiento necesario para solucionar un problema en particular, a distintos niveles de abstracción y el método que se aplica depende del tiempo que tenga disponible el sistema para razonar antes de proporcionar una respuesta al entorno.
Inspección	Término	Examinar y medir las características de calidad de un producto,
Java	Lenguaje de Programación	Se refiere al lenguaje de programación Java, que fue diseñado para usar con la plataforma Java
Java Media Framework (JMF)	Framework	Interfaz de programación de aplicaciones (API) para la incorporación de datos media tales como audio y vídeo en las aplicaciones y applets Java.
JDBC (Java Database Connectivity)	Término sigla	Permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.



JFreeChar	Librería	JFreeChart es una librería libre 100% de gráficos de Java que hace que sea fácil para los desarrolladores para mostrar gráficos estadísticos de calidad profesional en sus aplicaciones.
JMatLink	Librería	Librería basada en JNI (Java Native Interface) que conecta Java y Matlab. Los métodos nativos permiten utilizar diferentes funciones de Matlab desde una aplicación desarrollada en el lenguaje de programación Java.
MatLab	Programa	Es un software matemático que ofrece un entorno de desarrollo (IDE) con un lenguaje de programación propio (lenguaje M)
Modelo Vista Controlador (MVC)	Término	Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
Motor de inferencia	Término	El sistema experto modela el proceso de razonamiento humano con un módulo conocido como el motor de inferencia. Dicho motor de inferencia trabaja con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos. Contrasta los hechos particulares de la base de hechos con el conocimiento contenido en la base de conocimientos para obtener conclusiones acerca del problema.
MySql	Programa	Es un sistema de gestión de bases de datos relacional multihilo y multiusuario
NetBeans	Programa	Es una plataforma que permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos.
Operador (actor)	Término	Persona responsable de operar (manipular) la aplicación de inspección de control de objetos mediante el sistema y que tiene acceso restringido a la modificación de la información del mismo.
Pentaho Reporting (JFreeReport)	Librería	Librería libre de la clase Java para la generación de reportes.
Personal	Término	Personal administrativo u operativo que labora en una institución u organización.
Regla	Término	Es una estructura condicional que relaciona

		lógicamente la información contenida en la parte del antecedente con otra información contenida en la parte del consecuente.
Reportes	Término	Es una descripción escrita y grafica de las acciones efectuadas en el sistema durante un periodo de tiempo.
Resultado	Término	Respuesta del análisis realizado por medio de la aplicación de inspección de control.
Proceso	Término	Grupo de elementos en interacción que le dan al conjunto (o sistema) determinadas cualidades dinámicas o temporales.
Proceso y sistema	Término	Proceso es un sistema, pero en el que sus elementos pueden ser solo materia, energía o información.
Sistema	Término	Conjunto de medios tales como: personas, materiales, equipos, software, instalaciones, datos, etc. integrados de tal forma que puedan desarrollar una determinada función en respuesta a una necesidad concreta.
Sistema de Control	Término	Conjunto de dispositivos o herramientas que actúan juntos para lograr un objetivo de control.
Sistema Experto	Término	Son aquellos programas que se realizan haciendo explicito el conocimiento en ellos, que tienen información específica de un ambiente o dominio concreto y que realizan una tarea relativa a este dominio.
Tarea	Término	Serie de actividades u operaciones que interactúan entre sí y con el entorno, y que se ejecutan continuamente. Cada una de estas operaciones modela una pequeña parte del sistema.
Teoría de Control	Término	Estudia el desempeño de los sistemas dinámicos. Cuando una o más variables de salida del sistema deben seguir una salida deseada (referencia)
Usuario	Término	Son aquellos usuarios que interactúan con el sistema. Es el usuario que puede realizar el control de los objetos de las botellas, también puede visualizar la evaluación de la inspección de dicho control y obtener reportes hasta la fecha del último control.

2.2.2. Identificación de las relaciones de la aplicación

2.2.2.1. Modelado Dinámico de la Aplicación

Descripción de Casos de Uso

PANTALLAS DE CASO DE USO: Administrar Personal

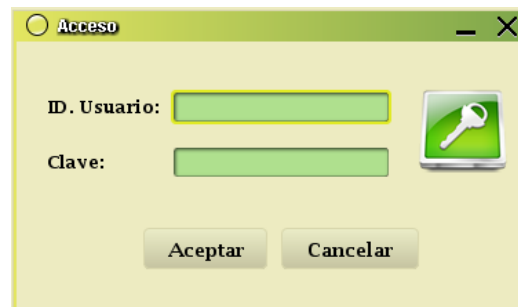


Figura 75: Vista Acceso

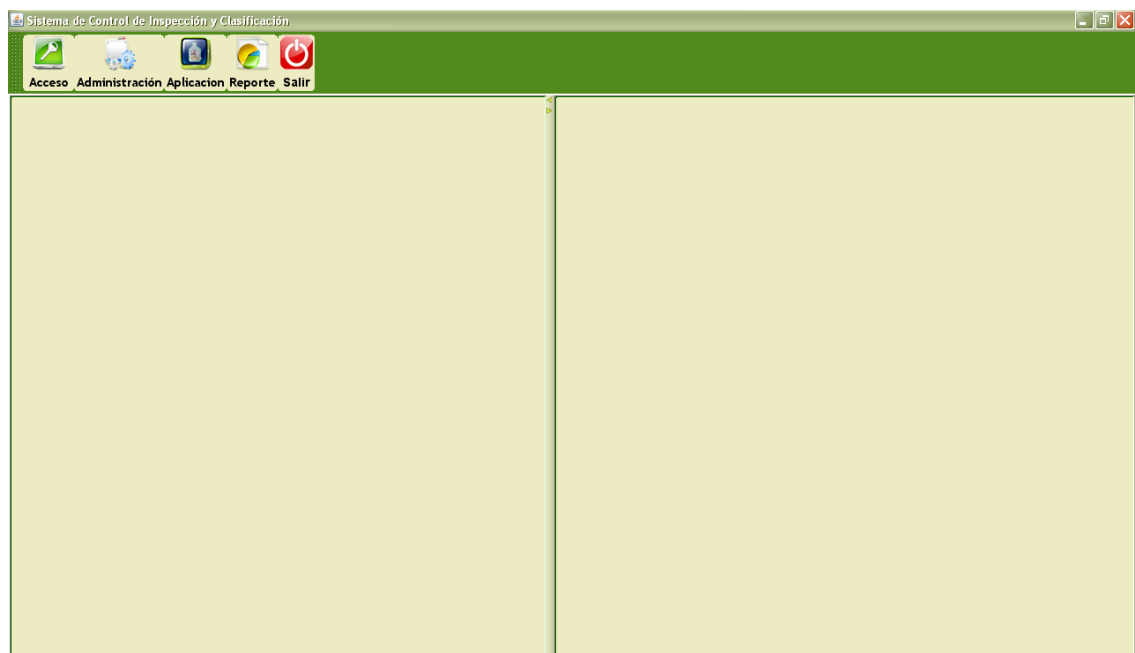


Figura 76: Vista Ventana de trabajo - Sistema de control de Inspección y Clasificación

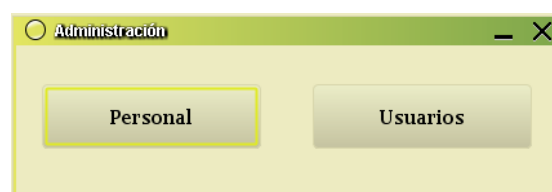


Figura 77: Vista Administración



Figura 78: Vista Administrar Personal

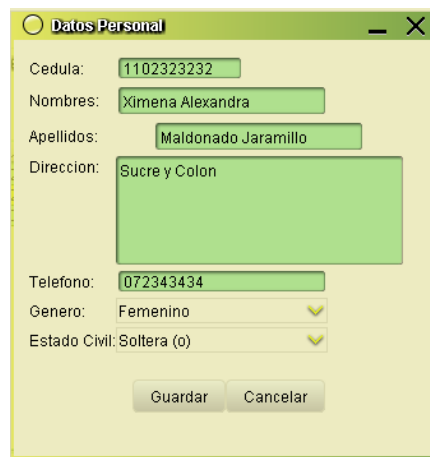


Figura 79: Vista Datos Personal

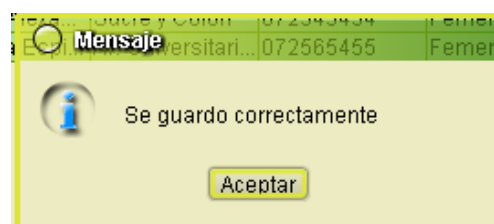


Figura 80: Vista Mensaje Datos Guardados Correctamente

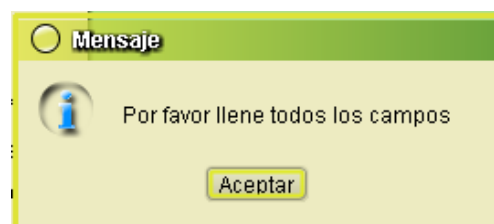


Figura 81: Vista Mensaje de Campos Vacios

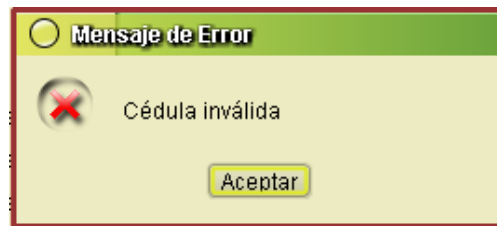


Figura 82: Vista Mensaje de Error Campos Inválidos

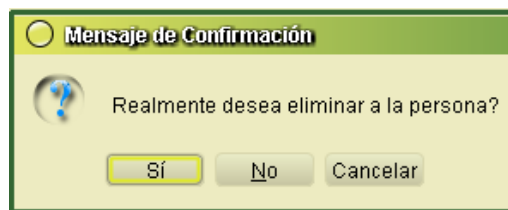


Figura 83: Vista Mensaje de confirmación de eliminación

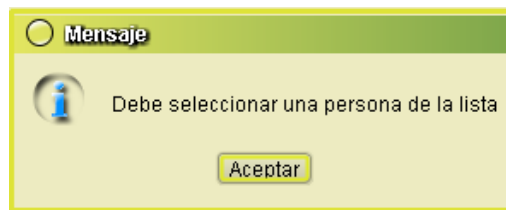


Figura 84: Vista Mensaje de no selección de Persona

CASOS DE USO: Administrar Personal

Nombre C. U.: Administrar Personal		Cod. C. U.: CU001
Objetivos / Propósitos:		
<ul style="list-style-type: none"> • Visualizar lista de personal • Registrar o crear nuevas personas. • Modificar datos del personal existente. • Eliminar datos del personal existente. 		
Descripción:		
<ul style="list-style-type: none"> • El administrador puede visualizar listar de personal existente para modificar o eliminar sus datos, también se podrá crear nuevas personas. 		
Actor (es):	Referencia de Requerimientos:	
<ul style="list-style-type: none"> • Administrador 	<ul style="list-style-type: none"> • RF002, RF003, RF004, RF005 	

Tipo C. U.: Sistema Primario**Precondiciones:**

- El administrador haya ingresado al sistema *Figura 75: Vista Acceso*.
- El administrador haya activado la opción de barra [Administración] de la *Figura 76: Vista Ventana de trabajo - Sistema de control de Inspección y Clasificación*
- El administrador haya activado la opción [Personal] de la *Figura 77: Vista Administración* y mostrado la *Figura 78: Vista Administrar Personal*

Post-Condiciones:

- Ver lista de personal
- Seleccionar una persona
- Crear nuevo personal
- Modificar personal existente
- Eliminar personal existente

CURSO NORMAL DE EVENTOS

1. El Administrador elige la opción [Ver Lista] de la *Figura 78: Vista Administrar Personal*
2. El sistema presenta el personal existente en la tabla [Lista de Personal] de la *Figura 78: Vista Administrar Personal*
3. El Administrador selecciona el registro de la persona que desea modificar en la tabla [Lista de Personal] de la *Figura 78: Vista Administrar Personal*
4. El Administrador elige la opción [Modificar] de la *Figura 78: Vista Administrar Personal*
5. El sistema obtiene los datos de la tabla [Lista de Personal].
6. El sistema muestra la *Figura 79: Vista Datos Personal* con la información de la persona seleccionada.
7. El Administrador modifica los datos que desea de la persona en la *Figura 79: Vista Datos Personal*
8. El Administrador elige la opción [Guardar] en la *Figura 79: Vista Datos Personal*
9. El sistema valida que la información este completa en la *Figura 79: Vista Datos Personal*
10. El sistema actualiza y guarda la información de la persona.

11. El sistema muestra un mensaje de confirmación (*Figura 80*) que la información se ha guardado correctamente y cierra la pantalla de la *Figura 79: Vista Datos Personal*

12. C.U finaliza.

CURSO ALTERNO DE EVENTOS

A. NO EXISTE SELECCIÓN DE PERSONA

A.5. El sistema muestra un mensaje de información (*Figura 84*) indicando que se debe seleccionar un registro de la tabla [Lista de Personal] de la *Figura 78: Vista Administrar Personal*.

A.6. El C.U. continúa en el paso 3 del Curso Normal de Eventos

B. CAMPOS OBLIGATORIOS VACÍOS

B.10. El sistema presenta un mensaje de información (*Figura 81*) indicando que algún(os) de los campos se encuentran vacíos en la *Figura 79: Vista Datos Personal*.

B.11. El C. U. continúa en el paso 7 del Curso Normal de Eventos.

C. CAMPOS A GUARDAR INVÁLIDOS

C.10. El sistema presenta un mensaje de error (*Figura 82*) indicando que los datos ingresados son inválidos según su tipo en la *Figura 79: Vista Datos Personal*.

C.11. El C. U. continúa en el paso 7 del Curso Normal de Eventos.

D. NUEVA PERSONA

D.1. El Administrador elige la opción [Nuevo] de la de la *Figura 78: Vista Administrar Personal*.

D.2. El sistema muestra la *Figura 79: Vista Datos Personal*

D.3. El Administrador ingresa la información en la *Figura 79: Vista Datos Personal*

D.4. El Administrador elige la opción [Guardar] en la *Figura 79: Vista Datos Personal*

D.5. El sistema valida que la información esté correcta.

D.6. El sistema guarda la información de la persona.

D.7. El sistema muestra un mensaje de confirmación (*Figura 80*) que la información se ha guardado correctamente y cierra la pantalla de la *Figura 79: Vista Datos Personal*

D.8. El C.U. finaliza.

E. ELIMINAR PERSONA

- E.1. El Administrador selecciona el registro de la persona que desea eliminar de la tabla [Lista de Personal] de la *Figura 78: Vista Administrar Personal*.
- E.2. El Administrador elige la opción [Eliminar] de la *Figura 78: Vista Administrar Personal*.
- E.3. El sistema muestra un mensaje de confirmación de eliminación (*Figura 83*).
- E.4. El Administrador elige la opción [Si].
- E.5. El sistema obtiene los datos de la tabla y los elimina.
- E.6. El C.U. finaliza.

DIAGRAMAS DE ROBUSTEZ DEL CASO DE USO CU001: Administrar Empleado

CURSO NORMAL DE EVENTOS

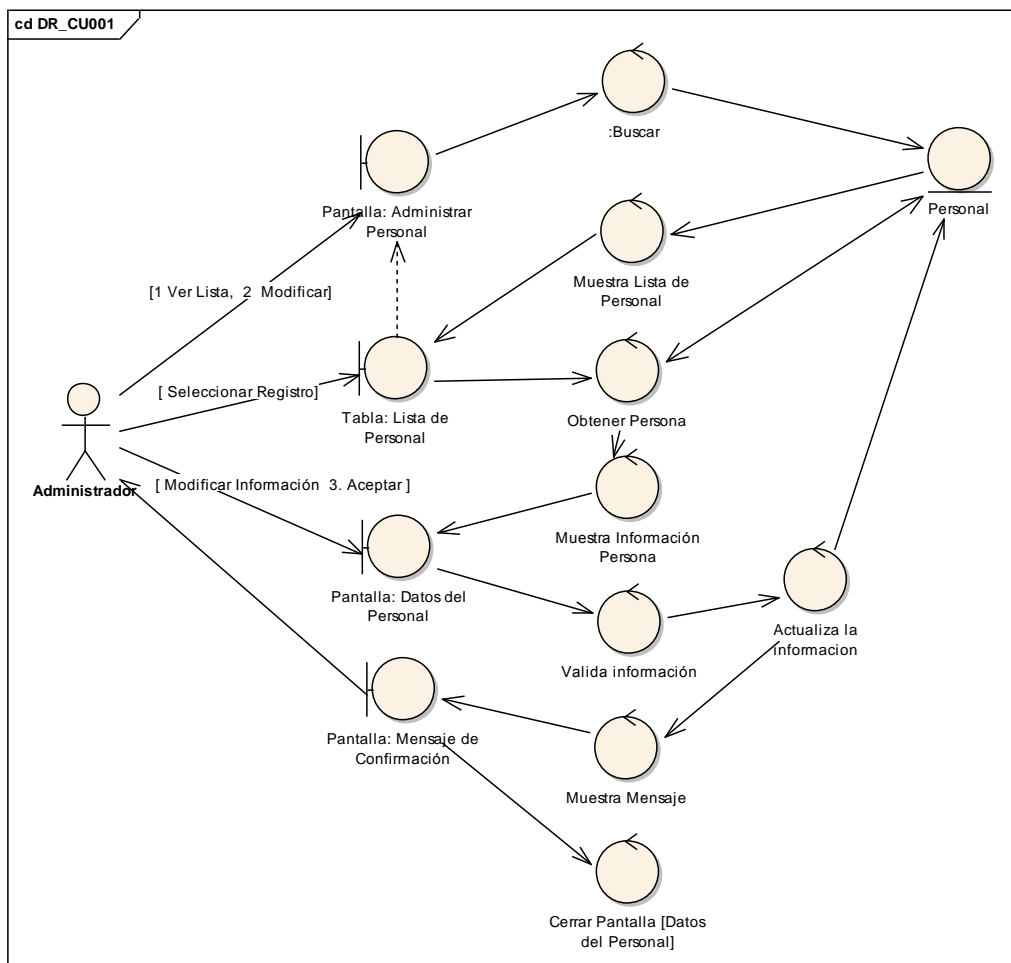


Figura 85: DR_CU001 - Administrar Personal

CURSO ALTERNO DE EVENTOS

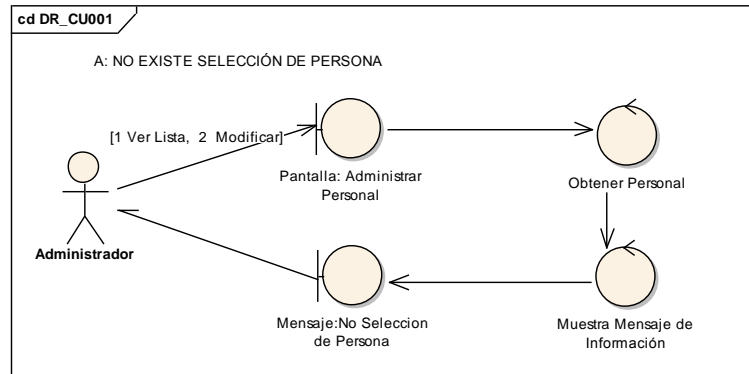


Figura 86: DR_CU001 – Curso Alterno A: No existe selección de persona

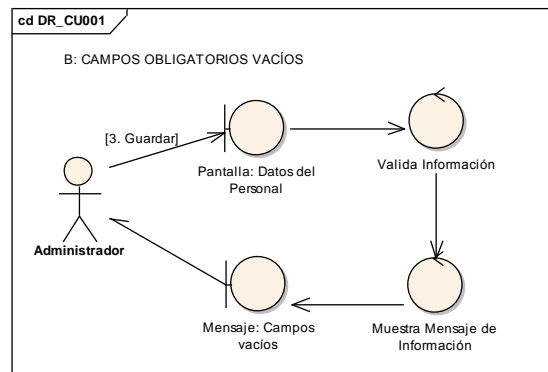


Figura 87: DR_CU001 – Curso Alterno B: Campos obligatorios vacíos

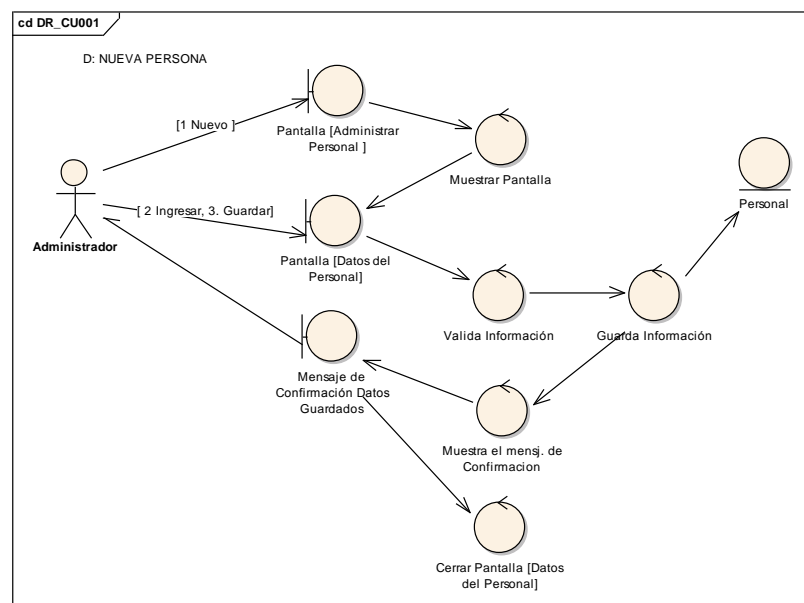


Figura 88: DR_CU001 – Curso Alterno D: Nueva Persona

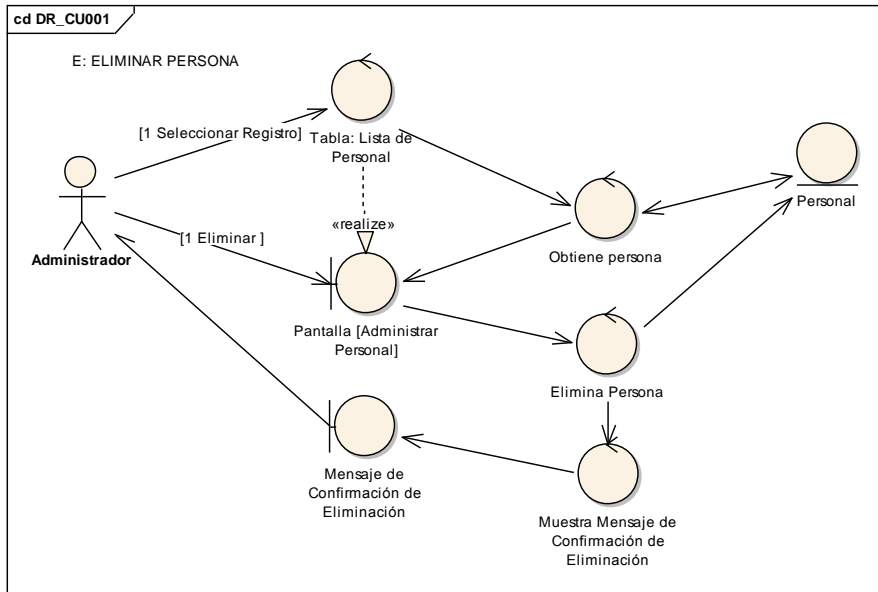


Figura 89: DR_CU001 – Curso Alterno D: Eliminar Persona

DIAGRAMAS DE SECUENCIA DEL CASO DE USO CU001: Administrar Empleado

CURSO NORMAL DE EVENTOS

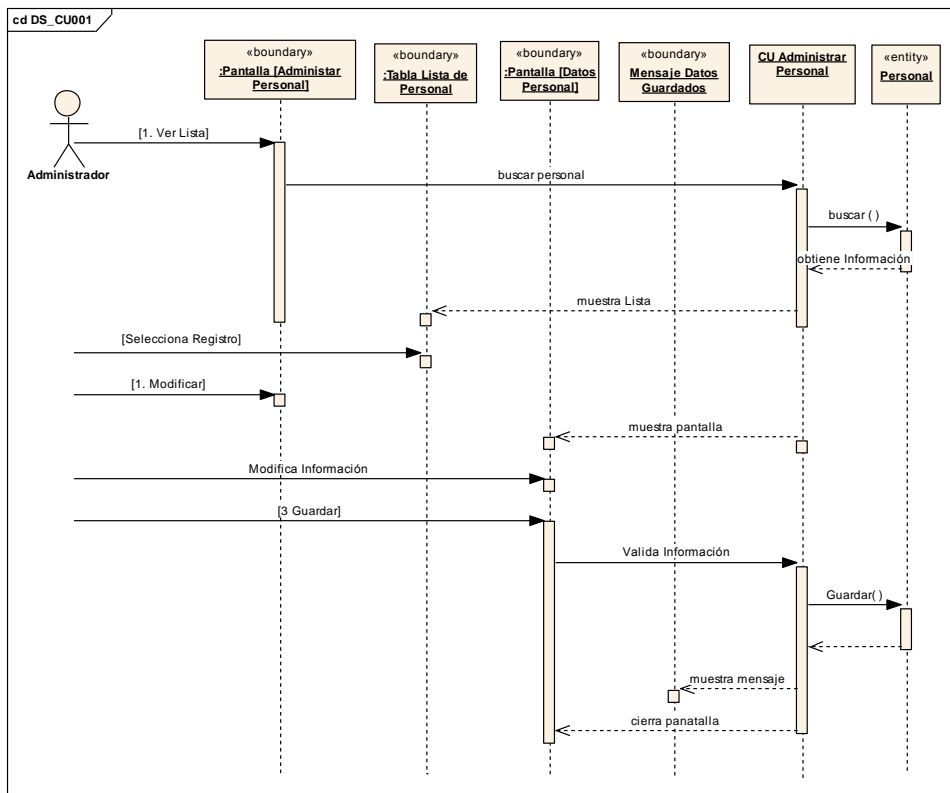


Figura 90: DS_CU001 - Administrar Personal

CURSO ALTERNO DE EVENTOS

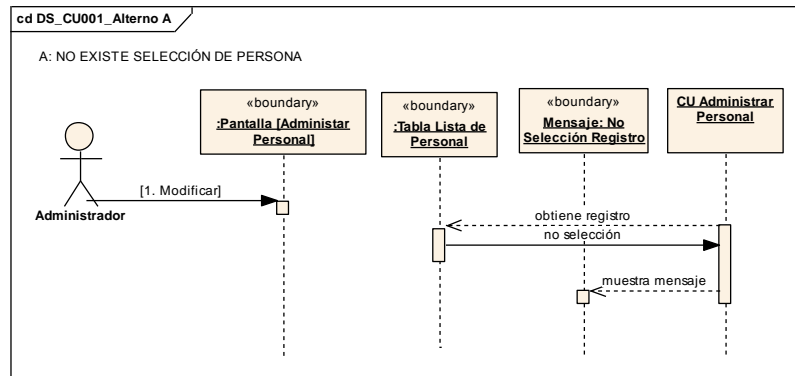


Figura 91: DS_CU001 – Curso Alterno A: No existe selección de persona

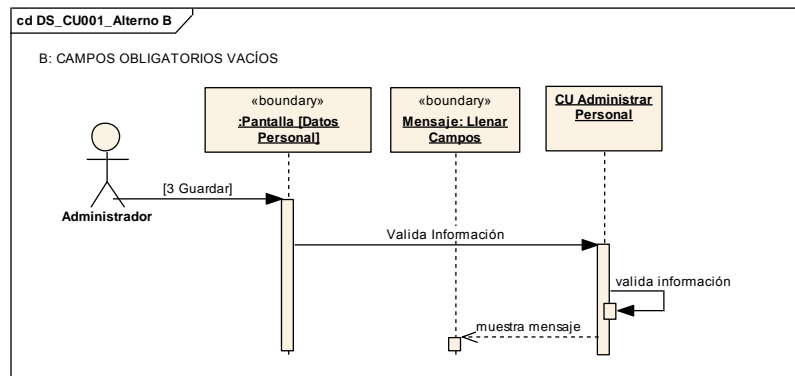


Figura 92: DS_CU001 – Curso Alterno B: Campos obligatorios vacíos

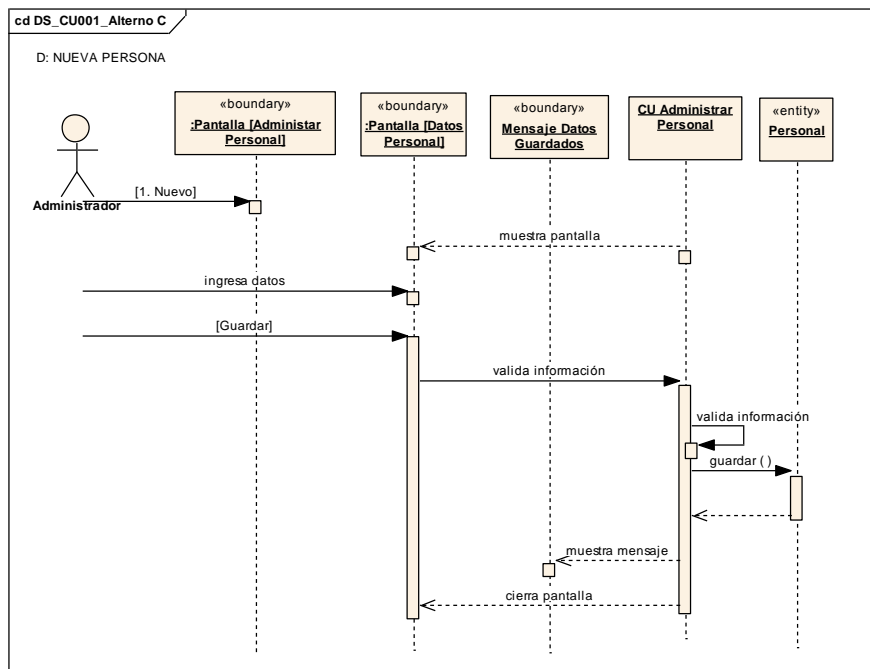


Figura 93: DS_CU001 – Curso Alterno D: Nueva Persona

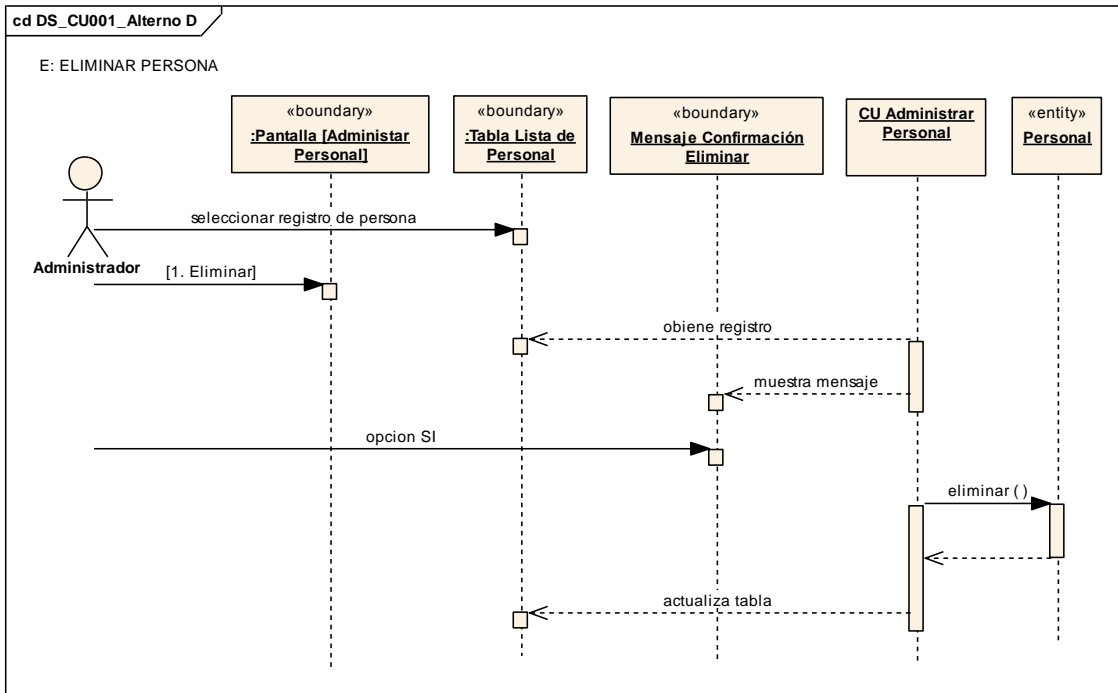


Figura 94: DR_CU001 – Curso Alterno D: Eliminar Persona

PANTALLAS DE CASO DE USO: Administrar Usuario

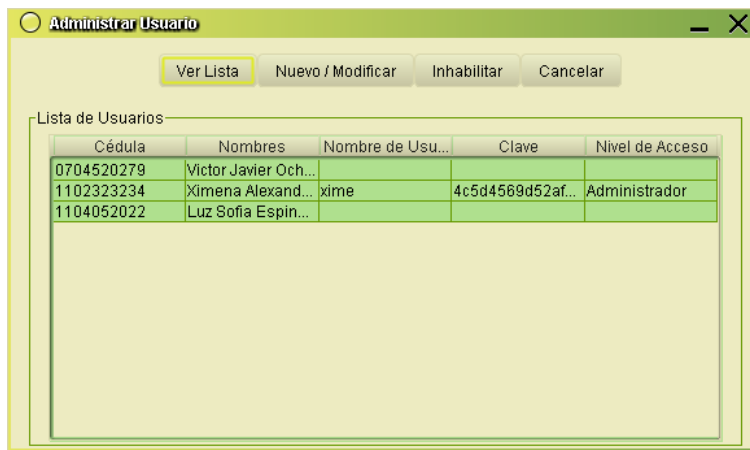


Figura 95: Vista Administrar Usuario

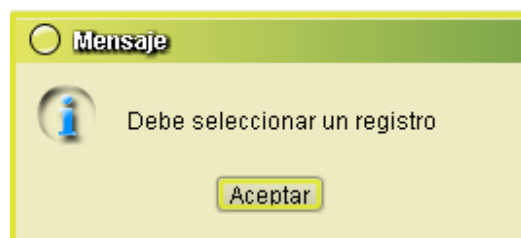


Figura 96: Vista Mensaje Selección de registro de persona

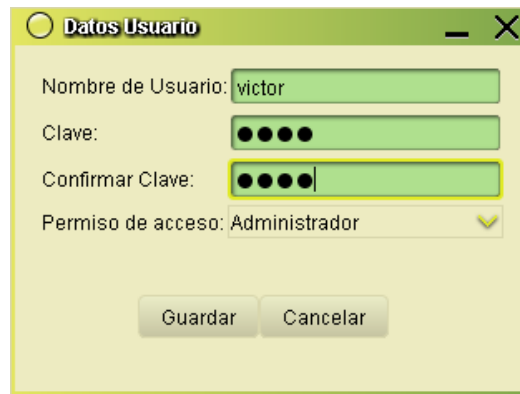


Figura 97: Vista Ingreso Datos de Usuario

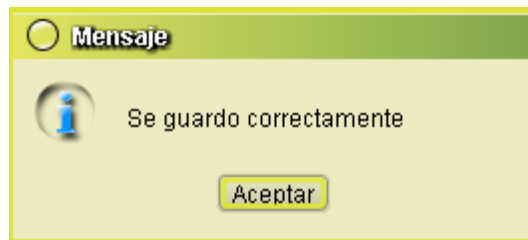


Figura 98: Vista Mensaje Datos Guardados Correctamente

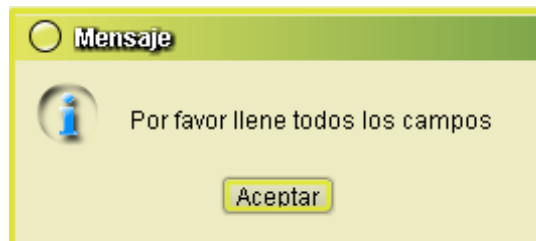


Figura 99: Vista Mensaje de Campos Vacíos

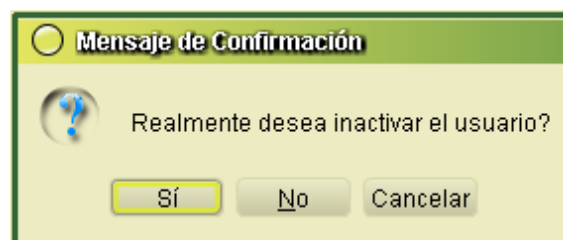


Figura 100: Vista Mensaje de inhabilitar usuario

CASOS DE USO: Administrar Usuario

Nombre C. U.: Administrar Usuario		Cod. C. U.: CU002
Objetivos / Propósitos:		
<ul style="list-style-type: none"> • Visualizar lista de usuarios • Registrar o crear nuevo usuario a partir de una persona • Modificar datos de usuario existente. • Asignar permisos de usuario. • Inhabilitar acceso de usuario. 		
Descripción:		
<ul style="list-style-type: none"> • El administrador puede visualizar lista de usuarios existentes para modificar sus datos e inhabilitar los usuarios, también se podrá crear nuevos usuarios y asignar permisos de acceso a las distintas opciones del sistema. 		
Actor (es):	Referencia de Requerimientos:	
<ul style="list-style-type: none"> • Administrador 	<ul style="list-style-type: none"> • RF002, RF003, RF004, RF005 	
Tipo C. U.: Sistema Primario		
Precondiciones:		
<ul style="list-style-type: none"> • El administrador haya ingresado al sistema <i>Figura 75: Vista Acceso</i>. • El administrador haya activado la opción de barra [Administración] de la <i>Figura 76: Vista Ventana de trabajo - Sistema de control de Inspección y Clasificación</i> • El administrador haya activado la opción [Usuarios] de la <i>Figura 77: Vista Administración</i> y mostrado la <i>Figura 95: Vista Administrar Usuario</i>. 		
Post-Condiciones:		
<ul style="list-style-type: none"> • Visualizar lista de usuarios • Seleccionar una persona • Registrar o crear nuevo usuario • Modificar datos de usuario existente • Asignar permisos de usuario • Inhabilitar acceso de usuario 		



CURSO NORMAL DE EVENTOS

1. El Administrador elige la opción [Ver Lista] de la *Figura 95: Vista Administrar Usuario*.
2. El sistema presenta los usuarios existentes en la tabla [Lista de Usuarios] de la *Figura 95: Vista Administrar Usuario*.
3. El Administrador selecciona el registro del usuario que desea modificar en la tabla [Lista de Usuarios] de la *Figura 95: Vista Administrar Usuario*.
4. El Administrador elige la opción [Nuevo / Modificar] de la *Figura 95: Vista Administrar Usuario*.
5. El sistema obtiene los datos referenciales del usuario de la tabla [Lista de Usuarios].
6. El sistema muestra la *Figura 97: Vista Ingreso Datos de Usuario* con la información del usuario seleccionado.
7. El Administrador ingresa o modifica los datos del usuario y asimismo asigna el permiso de acceso que tendrá en el sistema en la *Figura 97: Vista Ingreso Datos de Usuario*.
8. El Administrador elige la opción [Guardar] en la *Figura 97: Vista Ingreso Datos de Usuario*.
9. El sistema valida que la información este completa en la *Figura 97: Vista Ingreso Datos de Usuario*.
10. El sistema guarda y actualiza la información del usuario.
11. El sistema muestra un mensaje de confirmación (ver *Figura 98*) que la información se ha guardado correctamente y cierra la pantalla de la *Figura 97: Vista Ingreso Datos de Usuario*.
12. C.U finaliza.

CURSO ALTERNO DE EVENTOS

A. NO EXISTE SELECCIÓN DE REGISTRO DE USUARIO

A.5. El sistema muestra un mensaje de información (ver *Figura 96*) indicando que se debe seleccionar un registro de la tabla [Lista de Usuarios] de la *Figura 95: Vista Administrar Usuario*.

A.6. El C.U continúa en el paso 3 del Curso Normal de Eventos

B. CAMPOS OBLIGATORIOS VACÍOS

B.10. El sistema presenta un mensaje de información (ver *Figura 99*) indicando

que algún(os) de los campos se encuentran vacíos en la *Figura 97: Vista Ingreso Datos de Usuario*

B.11. El C. U. continúa en el paso 7 del Curso Normal de Eventos.

C. INHABILITAR ACCESO DE USUARIO

C.1. El Administrador selecciona el registro del usuario que desea inhabilitar de la tabla [Lista de Usuarios] de la *Figura 95: Vista Administrar Usuario*.

C.2. El Administrador elige la opción [Inhabilitar] de la *Figura 95: Vista Administrar Usuario*.

C.3. El sistema muestra un mensaje de confirmación de inactivación del usuario (Figura 100).

C.4. El Administrador elige la opción [Si].

C.5. El sistema obtiene los datos de la tabla y los elimina los permisos de acceso.

C.6. El C.U finaliza.

DIAGRAMAS DE ROBUSTEZ DEL CASO DE USO CU002: Administrar Usuario

CURSO NORMAL DE EVENTOS

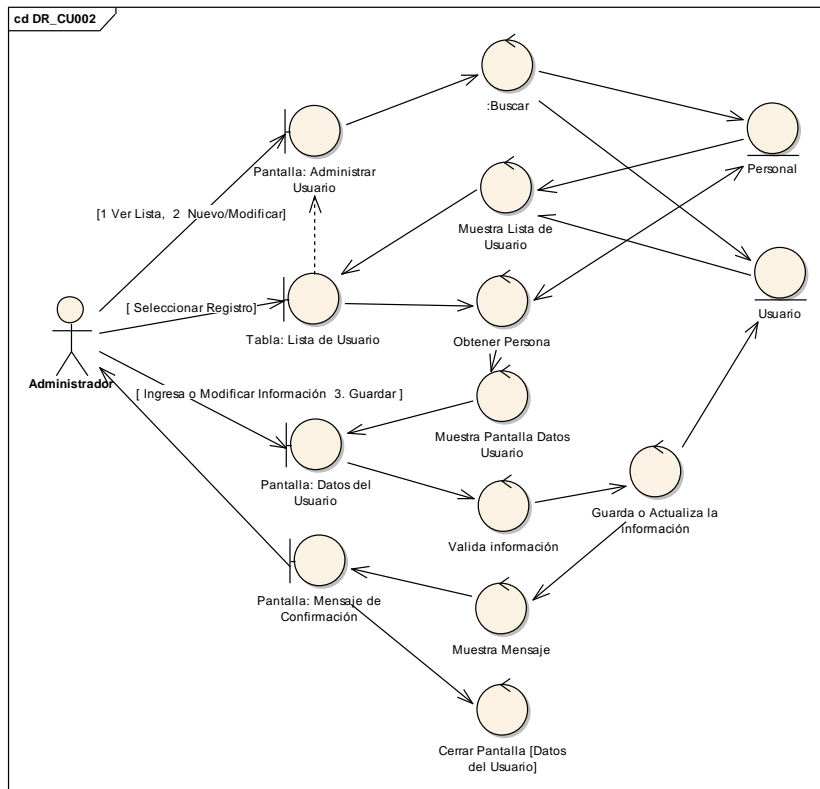


Figura 101: DR_CU002 - Administrar Usuario

CURSO ALTERNO DE EVENTOS

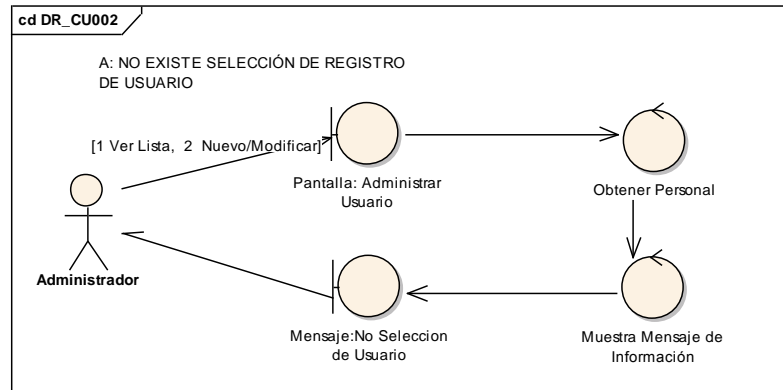


Figura 102: DR_CU002 – Curso Alterno A: No existe selección de usuario

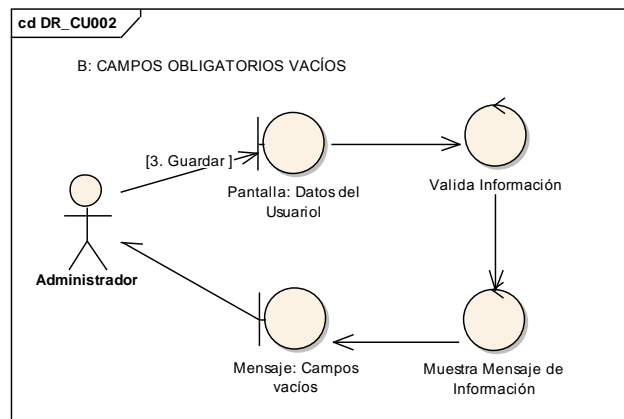


Figura 103: DR_CU002 – Curso Alterno B: Campos obligatorios vacíos

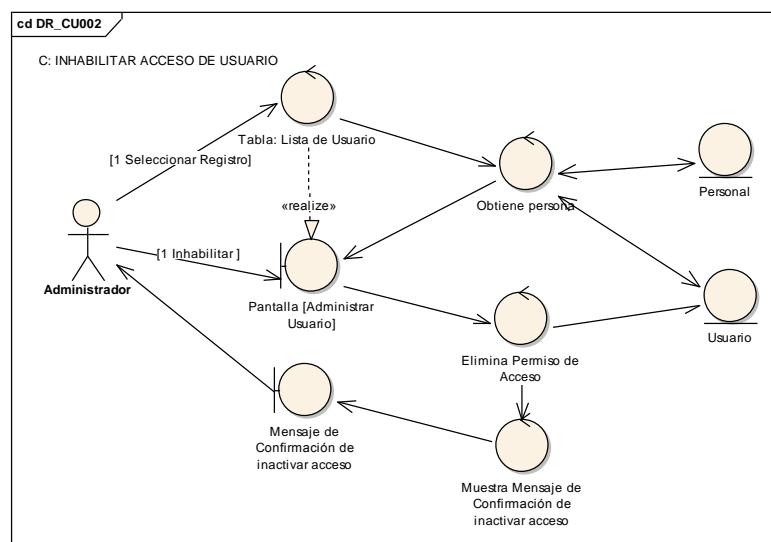


Figura 104: DR_CU002 – Curso Alterno C: Inhabilitar acceso de usuario

DIAGRAMAS DE SECUENCIA DEL CASO DE USO CU002: Administrar Usuario

CURSO NORMAL DE EVENTOS

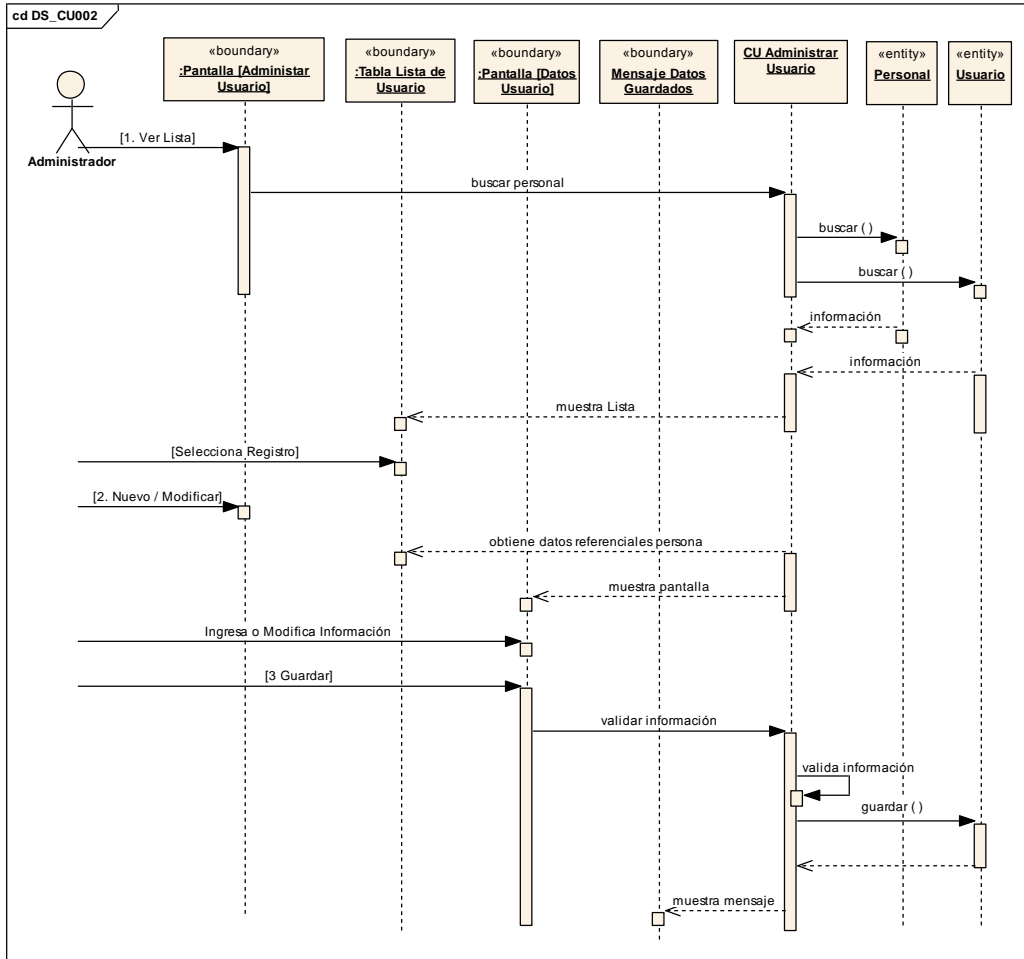


Figura 105: DS_CU002 - Administrar Usuario

CURSO ALTERNO DE EVENTOS

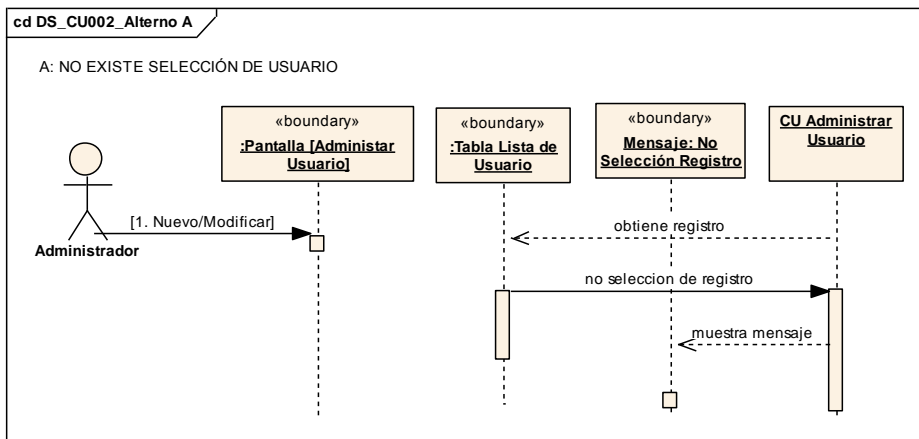


Figura 106: DS_CU002 – Curso Alternativo A: No existe selección de usuario

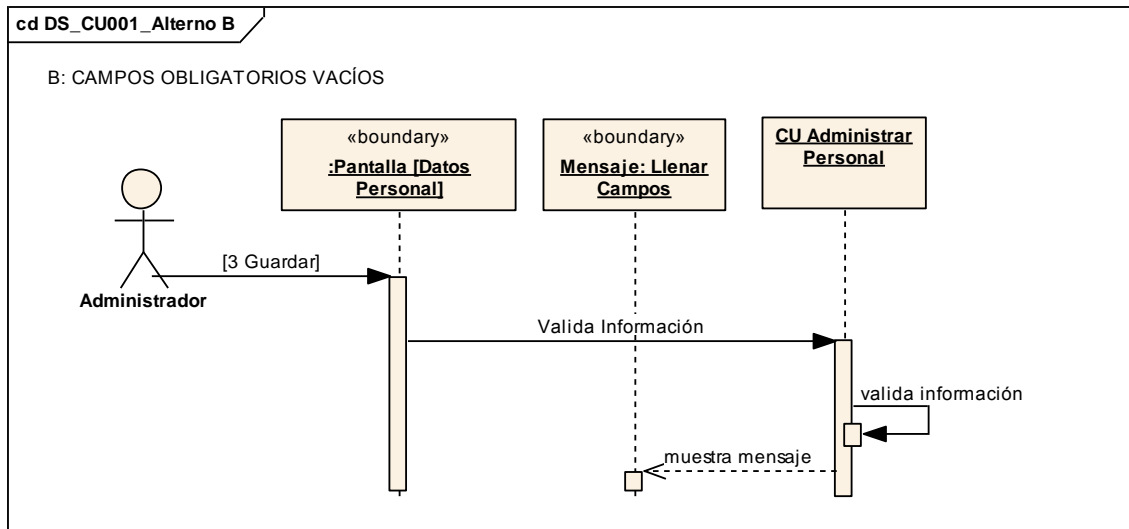


Figura 107: DR_CU002 – Curso Alterno B: Campos obligatorios vacíos

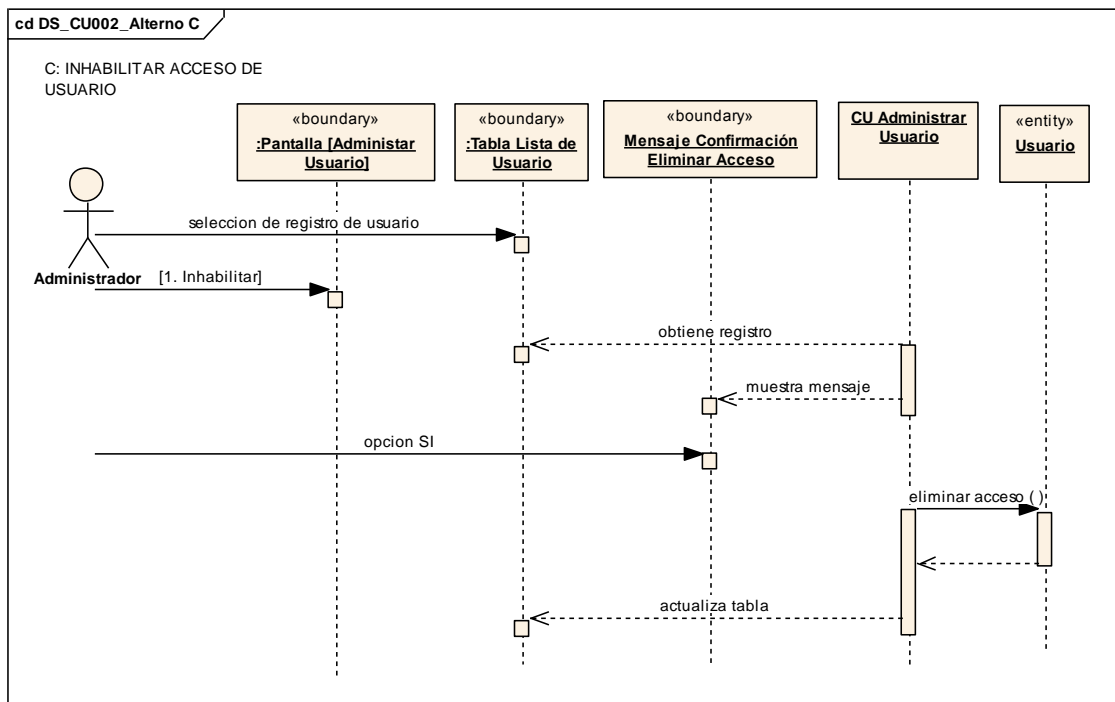


Figura 108: DS_CU002 – Curso Alterno C: Inhabilitar acceso de usuario

PANTALLAS DE CASO DE USO: Consultar Reportes



Figura 109: Vista Reporte Estadístico de Resultados

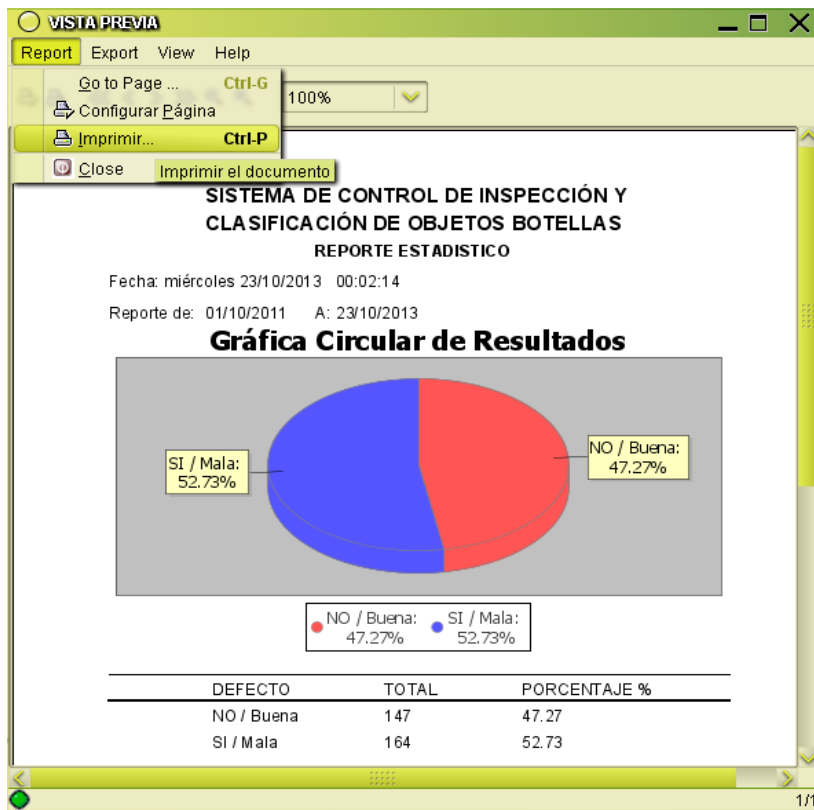


Figura 110: Vista Previa de Reporte

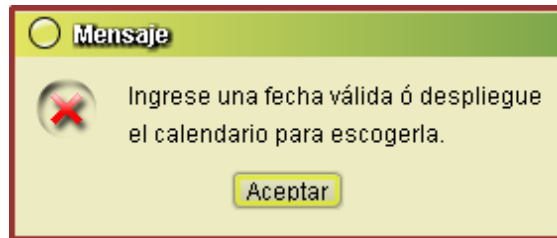


Figura 111: Vista Mensaje Fecha no válida

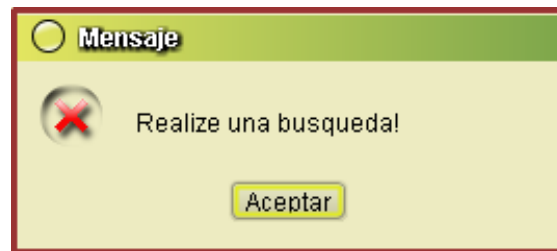


Figura 112: Vista Mensaje No hay Búsqueda

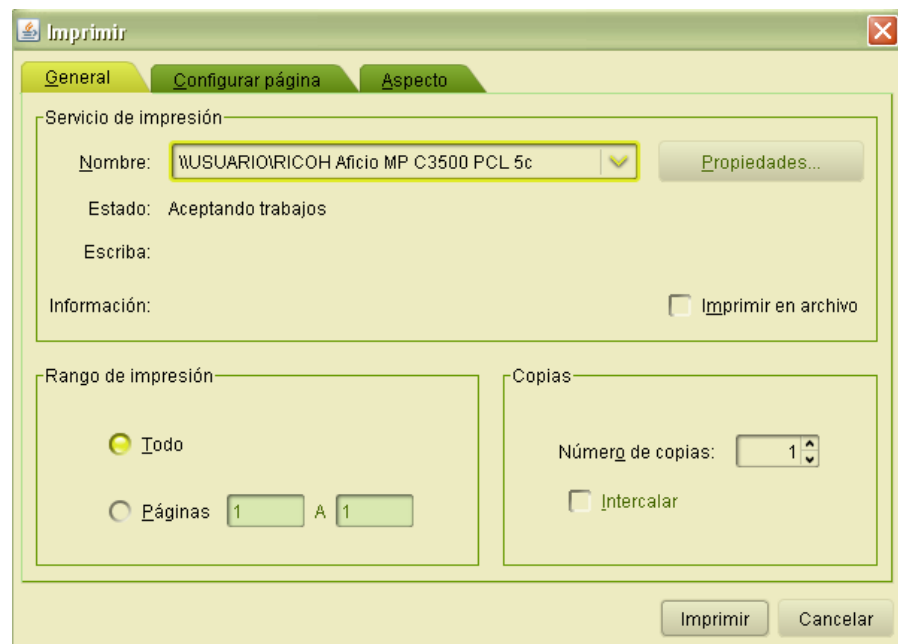


Figura 113: Vista Opciones de Impresión

CASOS DE USO: Consultar Reportes

Nombre C. U.: Consultar Reportes		Cod. C. U.: CU003
Objetivos / Propósitos:		
<ul style="list-style-type: none"> Realizar consulta estadística sobre los resultados de los objetos botellas inspeccionadas. Imprimir el reporte de consulta. 		
Descripción:		
<ul style="list-style-type: none"> El usuario o administrador que desee obtener información relacionada con los resultados de las inspecciones a los objetos botellas, deberán hacer una consulta por el método de búsqueda (por fecha). Además puede visualizar una gráfica estadística y una tabla de los mismos. 		
Actor (es):	Referencia de Requerimientos:	
<ul style="list-style-type: none"> Administrador Usuario 	<ul style="list-style-type: none"> RF002, RF003, RF004, RF005 	
Tipo C. U.: Sistema Primario		
Precondiciones:		
<ul style="list-style-type: none"> El Usuario o administrador haya ingresado al sistema <i>Figura 75: Vista Acceso</i>. El Usuario o administrador haya activado la opción de barra [Reporte] de la <i>Figura 76: Vista Ventana de trabajo - Sistema de control de Inspección y Clasificación</i> 		
Post-Condiciones:		
<ul style="list-style-type: none"> Realizar consulta estadística sobre los objetos botellas inspeccionadas. Visualizar una gráfica estadística y una tabla de los resultados. Imprimir el reporte de resultados. 		
CURSO NORMAL DE EVENTOS		
<ol style="list-style-type: none"> El Usuario o Administrador ingresa o escoge el intervalo de fechas deseadas desplegando el calendario de la <i>Figura 109: Vista Reporte Estadístico de Resultados</i>. 		



2. El Usuario o Administrador elige la opción [Buscar] de la *Figura 109: Vista Reporte Estadístico de Resultados*.
3. El sistema obtiene los datos y presenta una lista de Resultados.
4. El sistema presenta una Grafica Circular de Resultados
5. El Usuario o Administrador elige la opción [Vista Previa/Imprimir] de la *Figura 109: Vista Reporte Estadístico de Resultados*.
6. El sistema muestra una página con los datos correspondiente de la consulta como la *Figura 110: Vista Previa de Reporte*
7. El Usuario o Administrador elige la opción de menú [Report] y luego la opción [Imprimir] de la *Figura 110: Vista Previa de Reporte*
8. El sistema responde a la petición del Usuario o Administrador mostrando la pantalla de opciones de impresión en la *Figura 113: Vista Opciones de Impresión*
9. El Usuario o Administrador puede realizar la configuración deseada y elegir la opción [Imprimir] en la *Figura 113: Vista Opciones de Impresión*.
10. El sistema verifica la disponibilidad de la impresora.
11. El sistema imprime el reporte correspondiente.
12. El C. U. finaliza

CURSO ALTERNO DE EVENTOS

A. FECHAS NO VÁLIDAS

A.2. El sistema muestra un mensaje de error (Figura 111) indicando que se debe seleccionar fechas correctas.

A.3. El C.U continúa en el paso 1 del Curso Normal de Eventos

B. NO EXISTE BÚSQUEDA

B.6. El sistema presenta un mensaje de error (Figura 112) indicando que no se ha realizado ninguna búsqueda.

B.7. El C. U. continúa en el paso 1 del Curso Normal de Eventos.

C. DISPONIBILIDAD DE LA IMPRESORA

C.10. El sistema presenta un mensaje de error indicando que la impresora no es disponible (apagada o desconectada).

C.11. El C. U. continúa en el paso 9 del Curso Normal de Eventos

DIAGRAMAS DE ROBUSTEZ DEL CASO DE USO CU002: Administrar Usuario

CURSO NORMAL DE EVENTOS

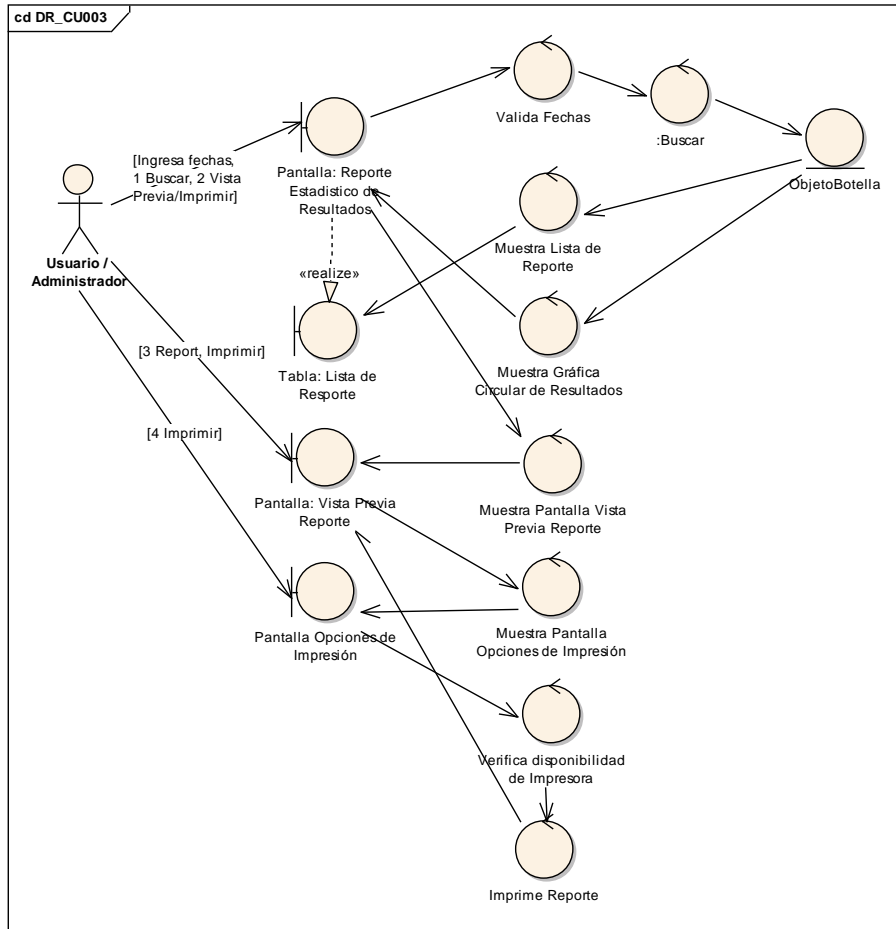


Figura 114: DR_C003 – Consultar Reportes

CURSO ALTERNO DE EVENTOS

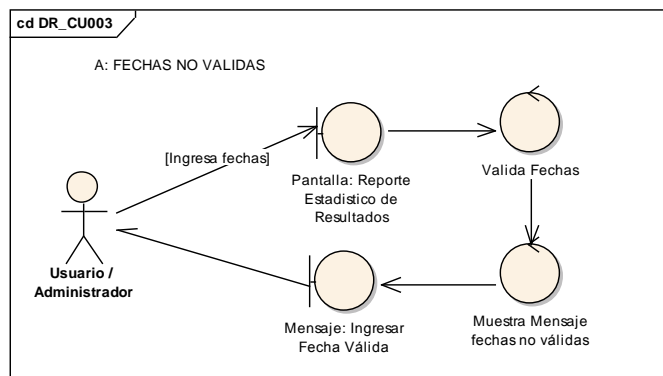


Figura 115: DR_CU003 – Curso Alterno A: Fechas no válidas

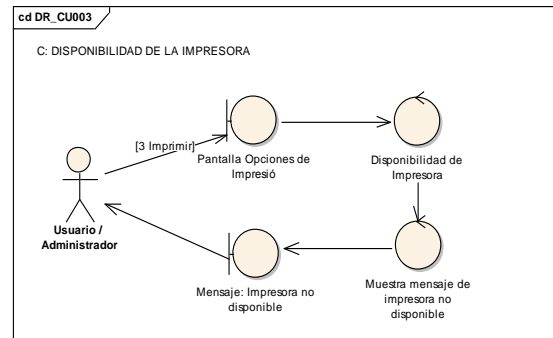
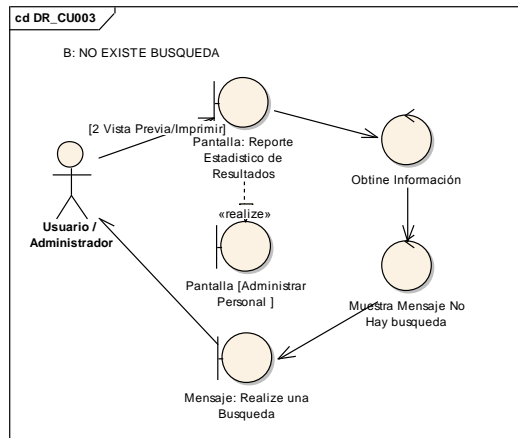


Figura 117: DR_CU003 – Curso Alterno C: Disponibilidad de Impresora

Figura 116: DR_CU003 – Curso Alterno B: No existe búsqueda

DIAGRAMAS DE SECUENCIA DEL CASO DE USO CU003: Consultar Reporte

CURSO NORMAL DE EVENTOS

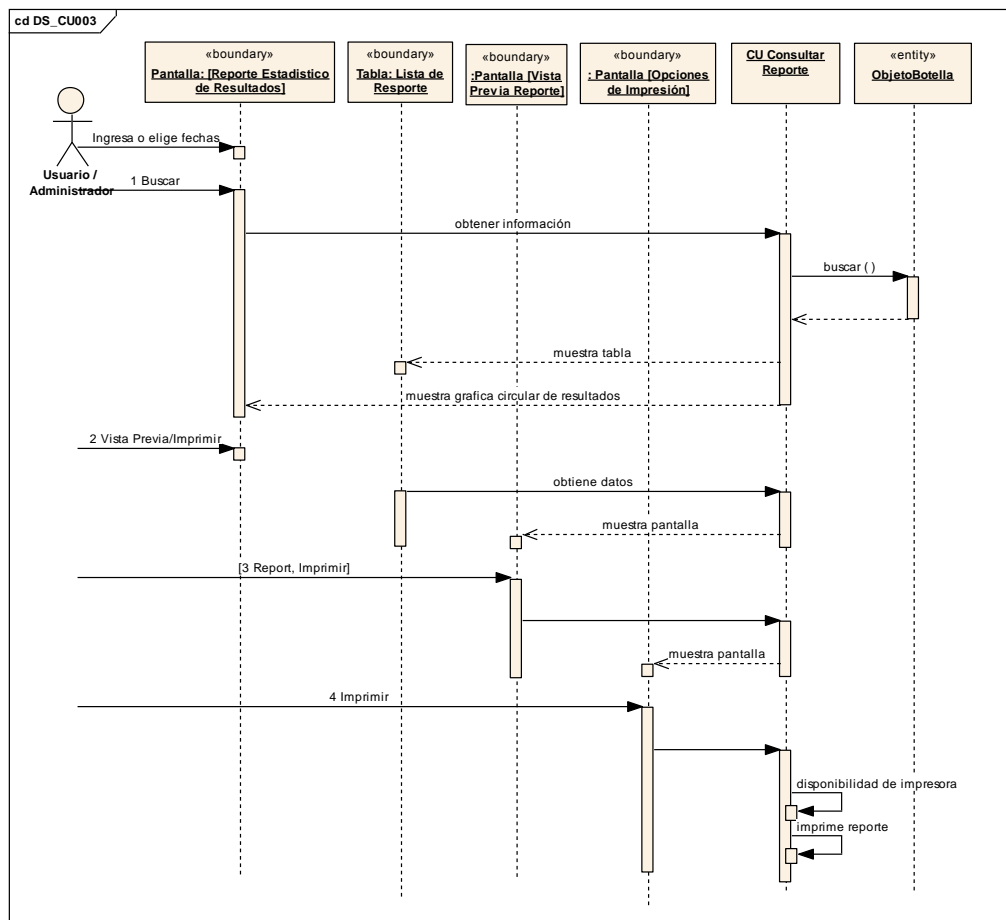


Figura 118: DS_CU003 – Consultar Reporte

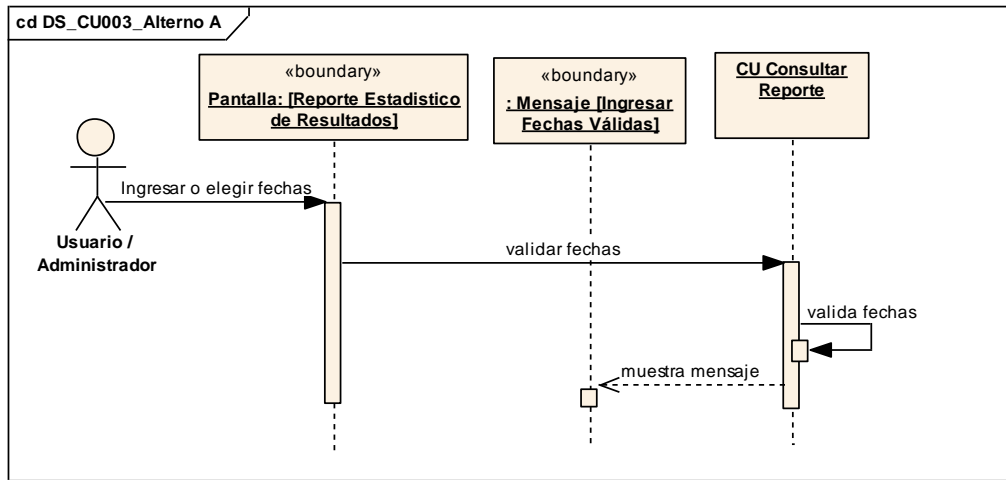


Figura 119: DS_CU003 – Curso Alterno A: Fechas no validas

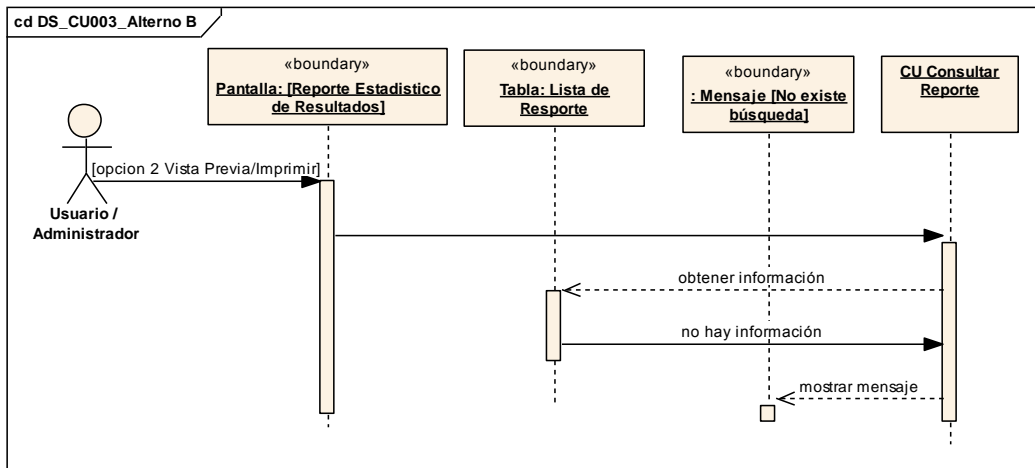


Figura 120: DS_CU003 – Curso Alterno B: No hay Búsqueda

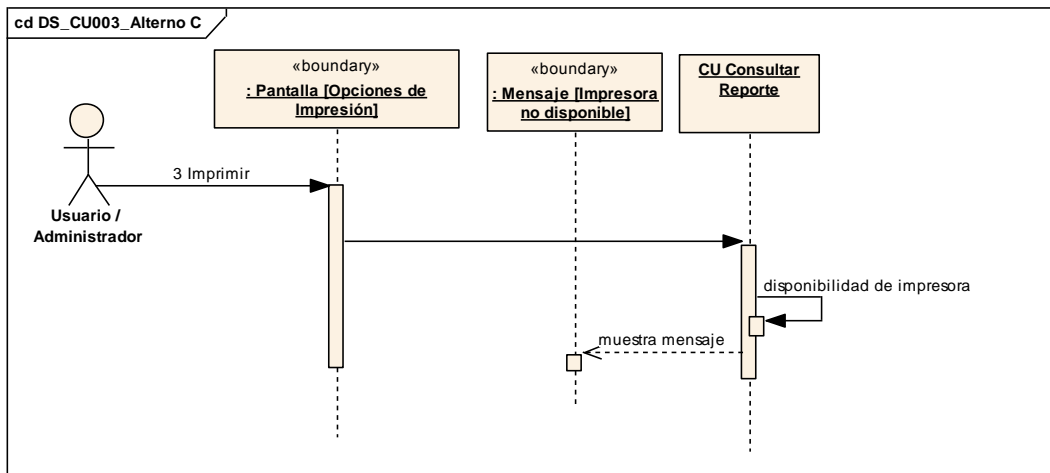


Figura 121: DS_CU003 – Curso Alterno C: Disponibilidad de Impresora

PANTALLAS DE CASO DE USO: Operar Inspección

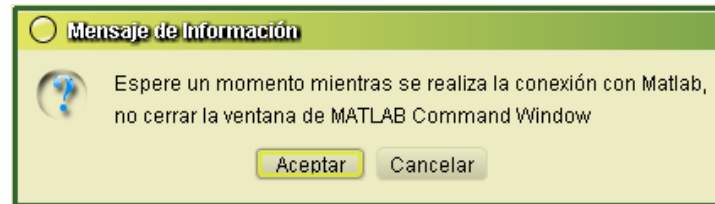


Figura 122: Vista Mensaje de Conexión con MatLab

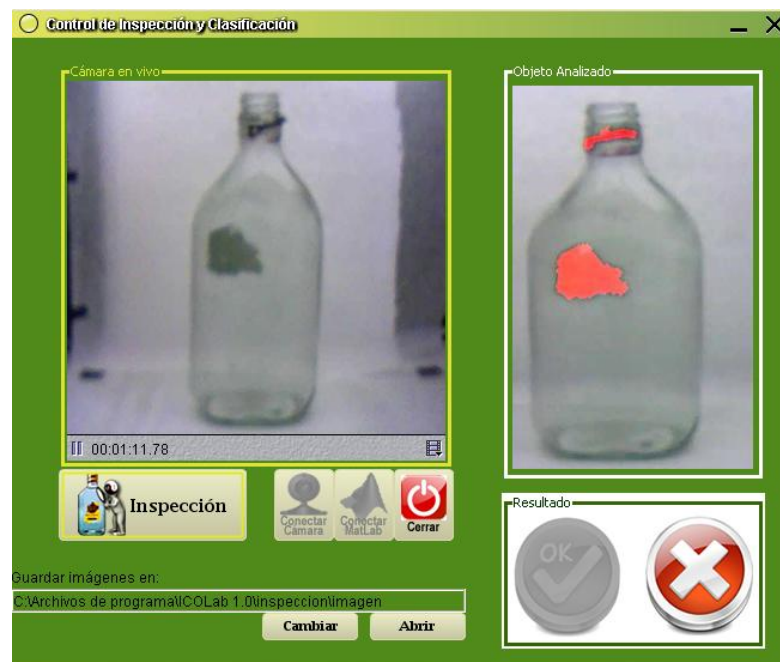


Figura 123: Vista Control de Inspección y Clasificación

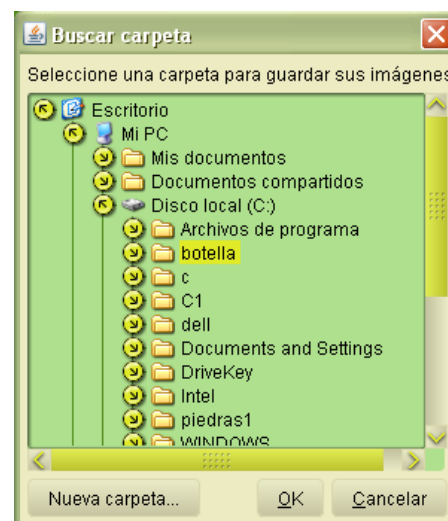


Figura 124: Vista Selección de nuevo directorio

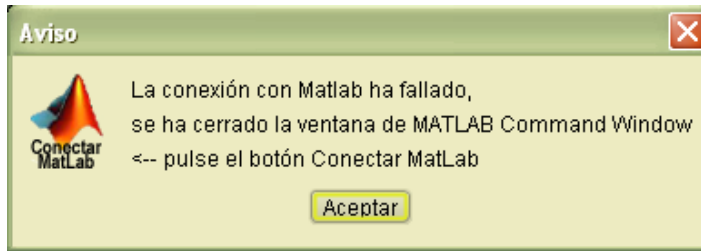


Figura 125: Vista Mensaje Fallo de Conexión MatLab

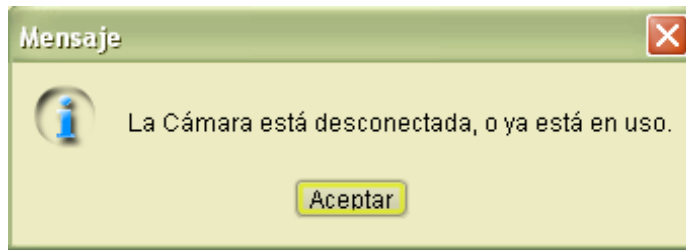


Figura 126: Vista Mensaje Cámara no disponible

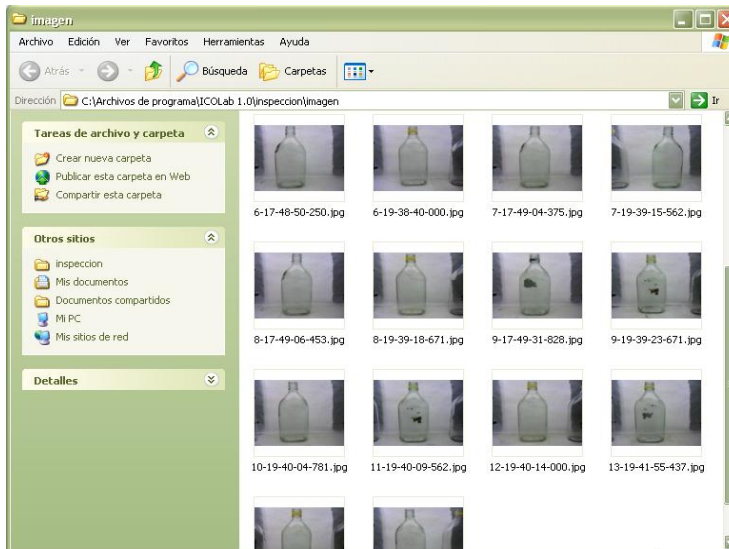


Figura 127: Opción Abrir Directorio

CASOS DE USO: Operar Inspección

Nombre C. U.: Operar Inspección	Cod. C. U.: CU004
Objetivos / Propósitos:	
<ul style="list-style-type: none"> Realizar el control de inspección del objeto botella de acuerdo al escenario elegido (micro mundo). Presentar el resultado de la inspección de forma visual y sonora. 	

Descripción: <ul style="list-style-type: none">• El usuario o administrador puede operar la inspección de los objetos botellas de forma automática en tiempo real y asimismo apreciar su resultado o clasificación de forma visual y sonora. Además puede cambiar reconectar los recursos como la cámara y MatLab; y cambiar el ubicación o directorio donde se guardan los archivos de las imágenes de las botellas.• NOTA: La tarea de Inspección utiliza un algoritmo de inspección y el controlador neuro-difuso, los cuales se describen más adelante mediante pasos y etapas de tiempo real basadas en las metodologías descritas anteriormente.	
Actor (es): <ul style="list-style-type: none">• Administrador• Usuario	Referencia de Requerimientos: <ul style="list-style-type: none">• RF002, RF003, RF004, RF005
Tipo C. U.: Sistema Primario	
Precondiciones: <ul style="list-style-type: none">• El Usuario o administrador haya ingresado al sistema <i>Figura 75: Vista Acceso.</i>• El Usuario o administrador haya activado la opción de barra [Inspección] de la <i>Figura 76: Vista Ventana de trabajo - Sistema de control de Inspección y Clasificación</i>• El Usuario o administrador haya aceptado la condición del mensaje <i>Figura 122: Vista Mensaje de Conexión con MatLab.</i>• El Sistema haya activado la <i>Figura 123: Vista Control de Inspección y Clasificación</i>• El Sistema haya activado las conexiones de la Cámara y MatLab. en la <i>Figura 123: Vista Control de Inspección y Clasificación</i>	
Post-Condiciones: <ul style="list-style-type: none">• Realizar el control de inspección del objeto botella de acuerdo al escenario elegido (micro mundo).• Presentar el resultado de la inspección de forma visual y sonora.• Conectar cámara si no está disponible.• Conectar MatLab si no está disponible.	

- Cambiar directorio donde guardar imágenes.
- Abrir directorio donde guardar imágenes.

CURSO NORMAL DE EVENTOS

1. El Usuario o Administrador elige la opción [Cambiar] de la *Figura 123: Vista Control de Inspección y Clasificación*.
2. El sistema muestra la *Figura 124: Vista Selección de nuevo directorio*.
3. El Usuario o Administrador selecciona o crea una carpeta donde guardar los archivos de las imágenes a inspeccionar en la *Figura 124: Vista Selección de nuevo directorio*.
4. El Usuario o Administrador elige la opción [Inspección] de la *Figura 123: Vista Control de Inspección y Clasificación*.
5. El sistema visualiza la imagen del la inspección con el defecto encontrado en el panel [Objeto Analizado] de la *Figura 123: Vista Control de Inspección y Clasificación*.
6. El sistema presenta el resultado de clasificación de la inspección de forma visual y sonora en el panel [Resultado] de la *Figura 123: Vista Control de Inspección y Clasificación*.
7. El C. U. finaliza

CURSO ALTERNO DE EVENTOS

A. CÁMARA NO DISPONIBLE

- A.4. El sistema muestra el mensaje de información *Figura 126: Vista Mensaje Cámara no disponible*.
- A.5. El Usuario o Administrador elige la opción [Conectar Cámara] de la *Figura 123: Vista Control de Inspección y Clasificación*.
- A.6. El sistema conecta la cámara.
- A.7. El C.U. continúa en el paso 4 del Curso Normal de Eventos.

B. CONEXIÓN MATLAB NO DISPONIBLE

- B.4. El sistema muestra el mensaje de información *Figura 125: Vista Mensaje Fallo de Conexión MatLab*.
- B.5. El Usuario o Administrador elige la opción [Conectar MatLab] de la *Figura 123: Vista Control de Inspección y Clasificación*.
- B.6. El sistema realiza la conexión con MatLab.

B.7. El C.U continúa en el paso 4 del Curso Normal de Eventos.

C. ABRIR DIRECTORIO

C.1 El Usuario o Administrador elige la opción [Abrir] de la *Figura 123: Vista Control de Inspección y Clasificación*.

C.2. El sistema abre la carpeta donde se encuentran guardados los archivos de imágenes de los objetos inspeccionados *Figura 127: Opción Abrir Directorio*.

C.3. El C. U. Finaliza

DIAGRAMAS DE ROBUSTEZ DEL CASO DE USO CU004: Operar Inspección

CURSO NORMAL DE EVENTOS

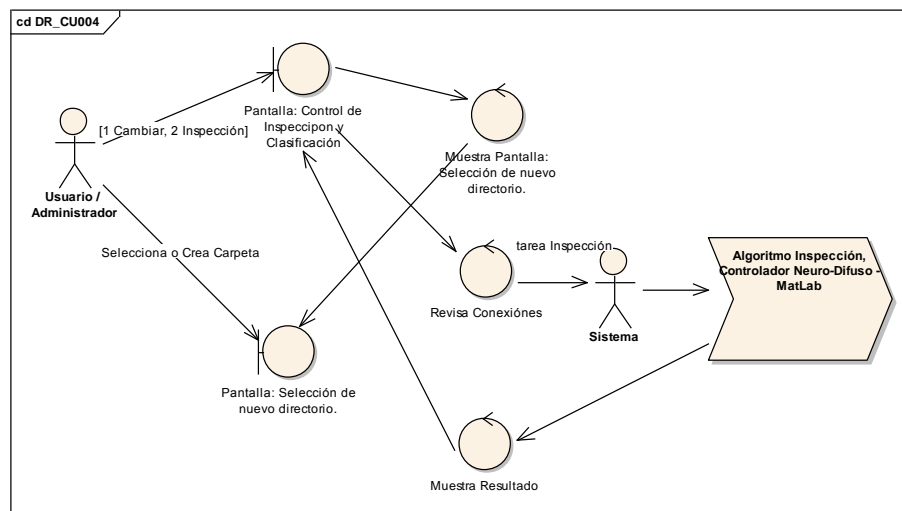


Figura 128: DR_CU004 – Operar Inspección

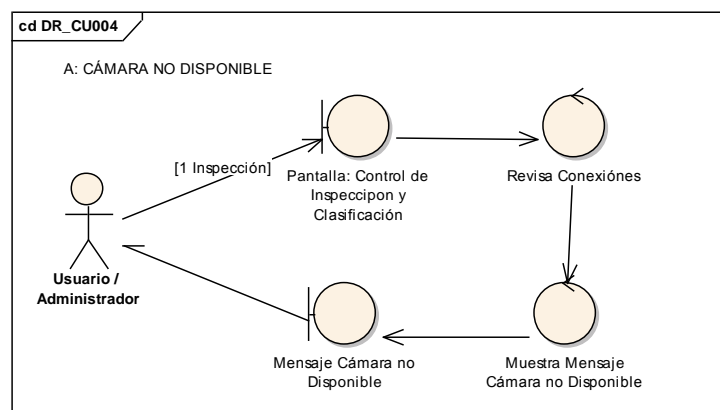


Figura 129: DR_CU004 – Curso Alterno A: Cámara no Disponible

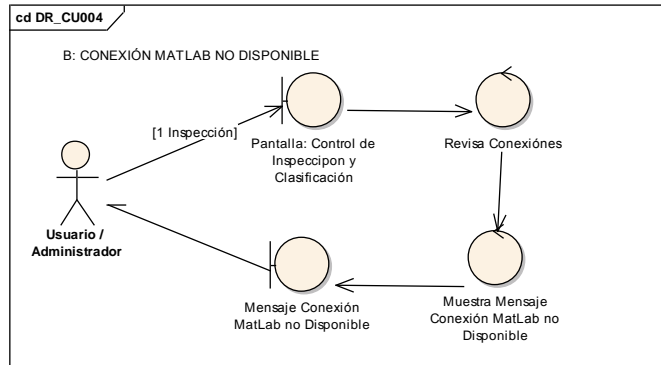


Figura 130: DR_CU004 – Curso Alterno B: Conexión MatLab no Disponible

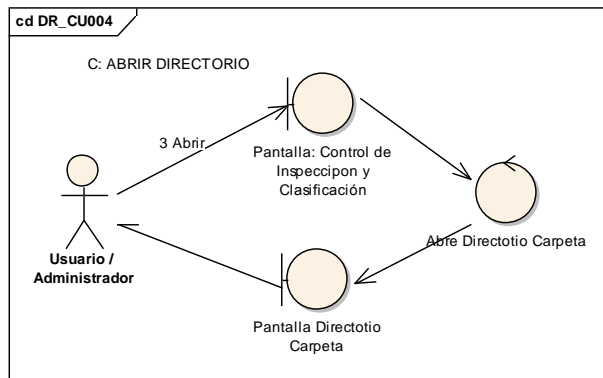


Figura 131: DR_CU004 – Curso Alterno C: Abrir Directorio

DIAGRAMAS DE SECUENCIA DEL CASO DE USO CU004: Operar Inspección

CURSO NORMAL DE EVENTOS

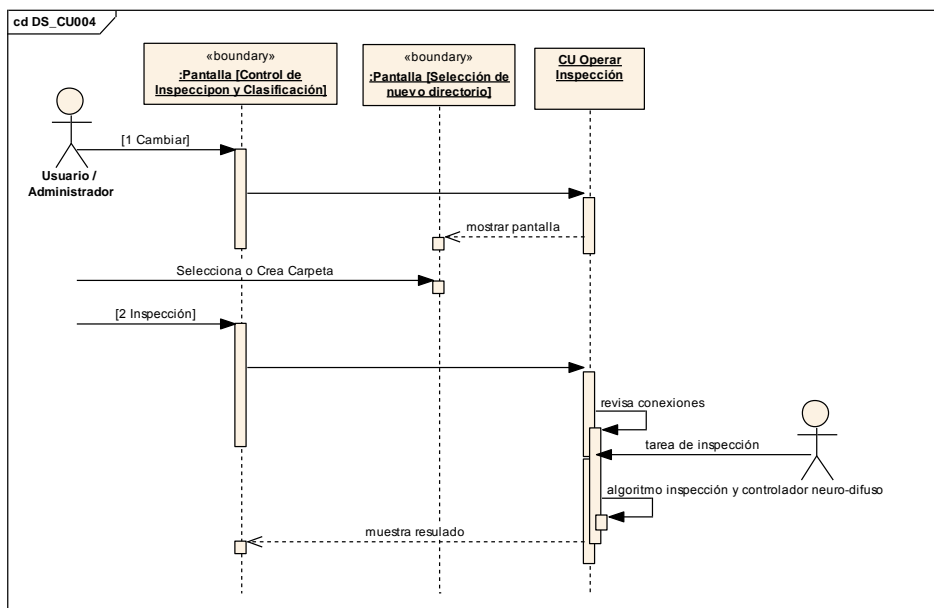


Figura 132: DS_CU004 – Operar Inspección

CURSO ALTERNO DE EVENTOS

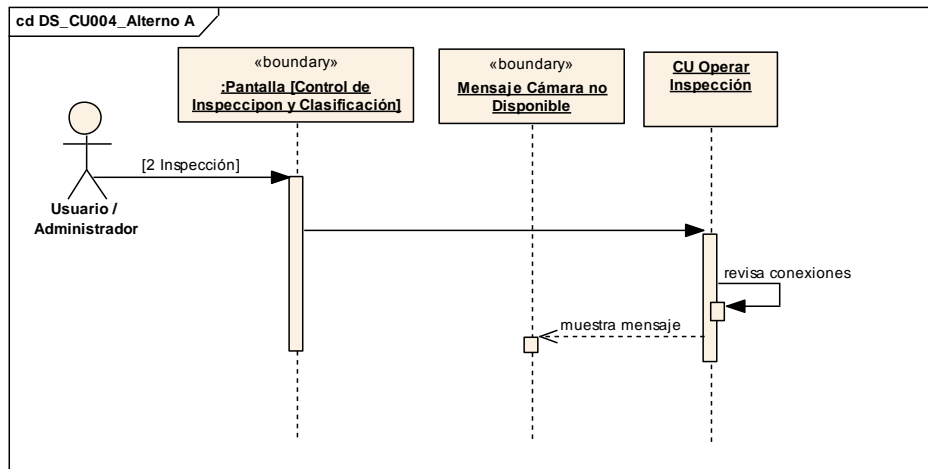


Figura 133: DS_CU004 – Cursos Alterno A: Cámara no disponible

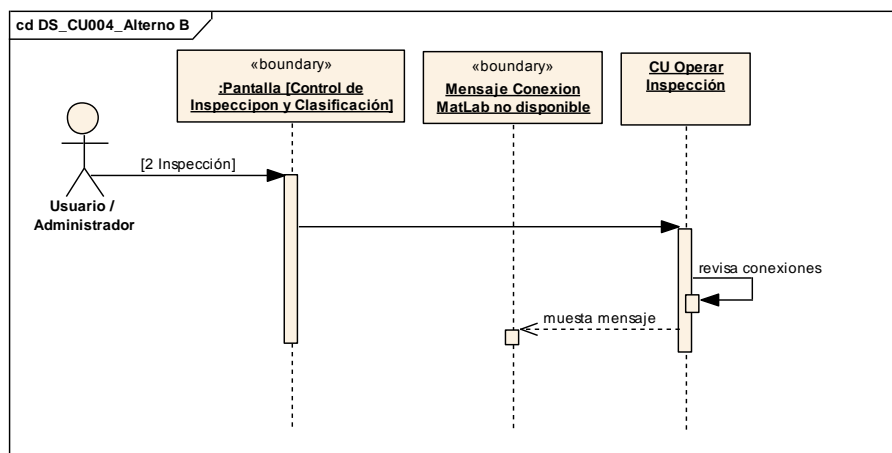


Figura 134: DS_CU004 – Cursos Alterno B: Conexión MatLab no disponible

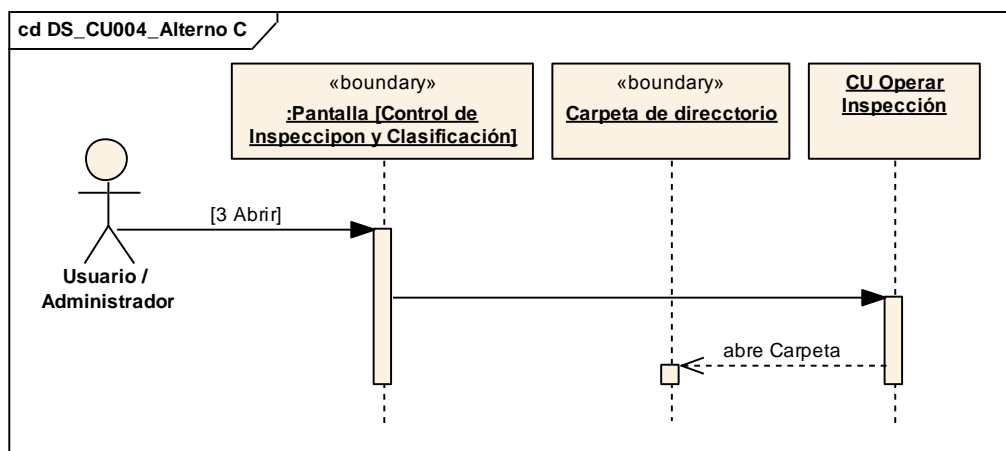


Figura 135: DS_CU004 – Cursos Alterno C: Abrir Carpeta

2.2.2.2. Diagrama de Componentes

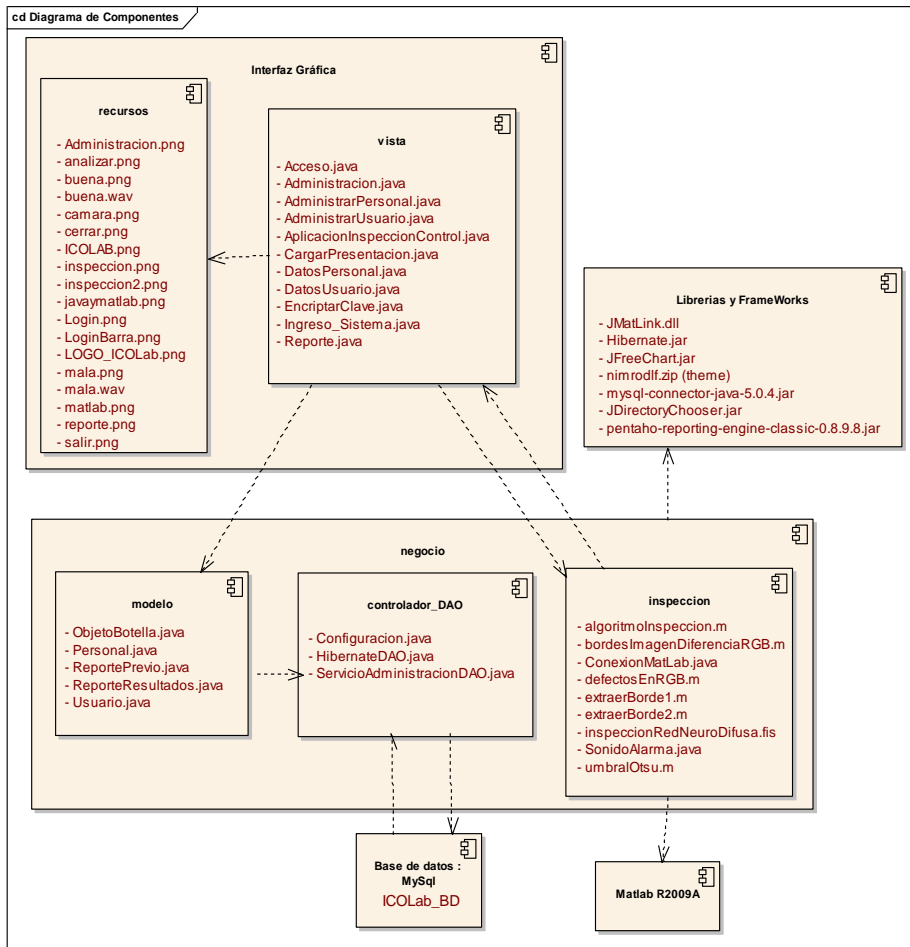


Figura 136: Diagrama de Componentes

2.2.2.3. Diagrama de paquetes

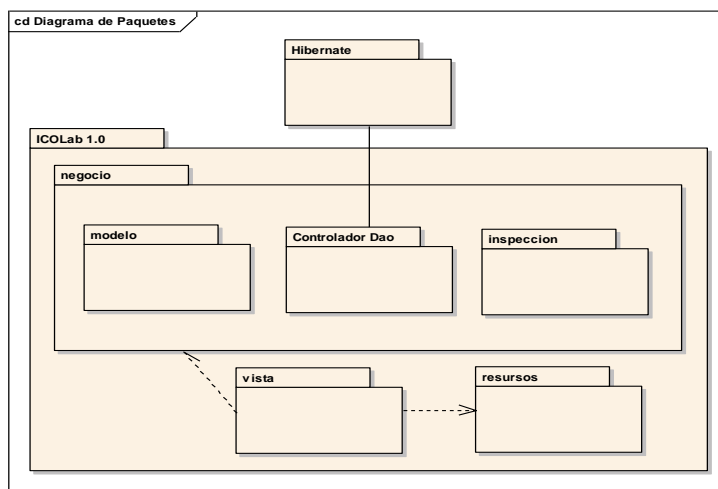


Figura 137: Diagrama de Paquetes

2.2.2.4. Diagrama de Despliegue

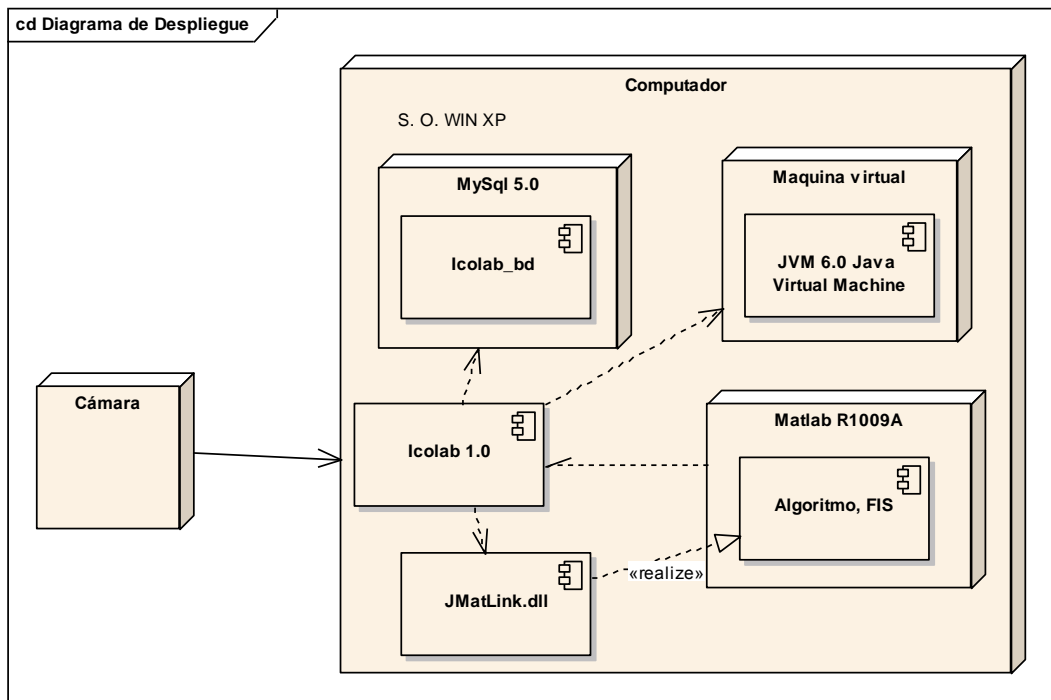


Figura 138: Diagrama de Despliegue

2.2.2.5. Modelo de tareas de alto nivel (TAN)

Tarea de Inspección del Caso de Uso CU004: Operar Inspección

El esquema general del comportamiento de los procesos de las tareas en el sistema, para la inspección y clasificación de imágenes se presenta en la *Figura 139*.

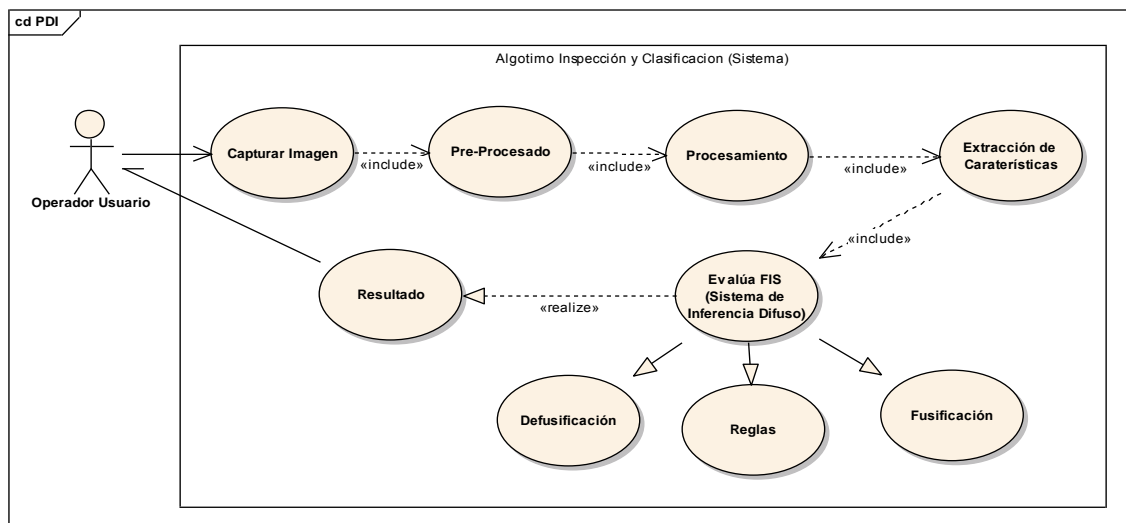


Figura 139: Esquema general del comportamiento de los procesos en el sistema

Este modelo describe los procesos asignados al sistema en proceso de control de **inspección**. Ver TABLA XII.

TABLA XII: DESCRIPCIÓN DEL PROCESO EN FUNCIÓN DE LAS TAN QUE LO COMPONEN

Ident TAN.	Nombre TAN	Objetivo de la TAN	Tipo de TAN	Importancia de la TAN para el proceso	¿Es intensiva en conocimientos ?	Conocimiento involucrado	¿Puede tener restricciones temporales?
ActConexiones	Activación de cámara conexiones con MatLab	Establecer las condiciones del micro mundo y verificación		Es importante pero no requiere de conocimientos	Nada		Si, Se manejan unos tiempos de conexión dependiendo del hardware.
InspecObj	Inspección del objeto botella	Encontrar y extraer características (valores) de la imagen de la botella	Planeación Scheduling Monitorización	Es fundamental para planificar las siguientes acciones	Alto	Planificación de acciones, interpretación del micro mundo, conocimientos en procesamiento digital de imágenes	Si, Hay tareas que serán periódicas, que tendrán unos deadlines y también un wctet
ClasifObj	Clasificar objeto botella	Reconocimiento y obtener resultados	Clasificación	La clave es aplicar lógica neuro - difusa.	Alto	Conocimiento lingüístico, Base de conocimiento, Reglas de Inferencia	Igual que la anterior
VisualResult	Visualizar Resultado	Presentar la imagen de la botella inspeccionada	Monitorización	Es necesario tener información y tomar decisiones	Alto	Identificación del objeto inspeccionado.	Igual que la anterior

La TAN InspecObj tiene que incluir actividades con restricciones temporales para controlar algunas de las acciones del proceso de control. Por ejemplo, dentro de esta tarea es necesario hacer un plan de las acciones a realizar, de tal forma que se define una subtarea llamada *PlanAcción*, con un período de 2 segundos, para significar que esta TAN puede ser repetida o realizada cada 2 segundos (Ver Figura 140).

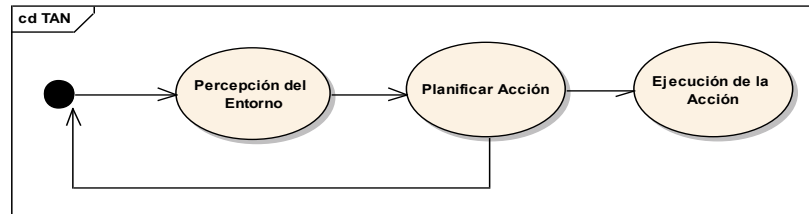


Figura 140: Detalle de las actividades de la TAN 2 Inspeccionar Objeto

2.2.2.6. Modelo de tareas en tiempo real

La construcción de un sistema de tiempo real está formada por una serie de actividades u operaciones que interactúan entre sí y con el entorno, y que se ejecutan continuamente. Cada una de estas operaciones modela una pequeña parte del sistema que comúnmente se llama *tarea*.

Es importante decir que cuando se tienen sistemas que requieren manejo del tiempo se deben considerar áreas de la informática que pueden afectar el cumplimiento de las propiedades del sistema. Es así como el hardware, el lenguaje de programación, el sistema operativo, la arquitectura del sistema y las metodologías de desarrollo juegan un papel trascendental dentro de estos sistemas. Para este caso se pueden apreciar las siguientes relaciones (ver Figura 141).

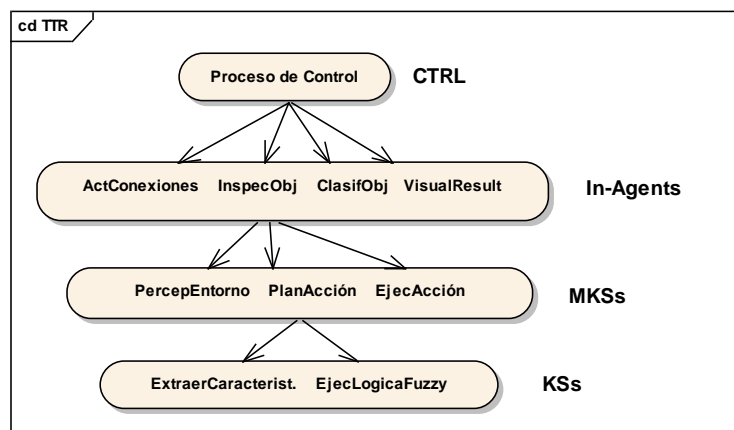


Figura 141: Tareas de alto nivel TAN

En donde, tiene una TTR (Tarea en tiempo real) por cada in-agente que se defina, de modo que en este caso se tienen cuatro tareas de tiempo real.

En resumen estas son las tareas a programar en el algoritmo desarrollado en MatLab.

Dónde:

- CTRL es un agente que está compuesto por una serie de métodos para el manejo de percepción y acción, un módulo de control que se responsabiliza por la ejecución de tiempo real de cada uno de los componentes, y un servidor inteligente que se encarga de producir respuesta para hacer el razonamiento.
- Un agente interno (IA – *In-Agent*) que es una entidad interna que posee el conocimiento necesario para solucionar un problema en particular, a distintos niveles de abstracción y el método que se aplica depende del tiempo que tenga disponible el sistema para razonar antes de proporcionar una respuesta al entorno. Este conocimiento puede incorporar técnicas de inteligencia artificial. Un in-agent se caracteriza por un comportamiento periódico, es una tarea específica y dispone de un plazo máximo de ejecución, *deadline*.
- Una fuente de conocimientos múltiple (MKS - *A Multiple Knowledge Source*) que implementa el concepto de los algoritmos *anytime* y *multiple methods*, dando diversas soluciones al mismo problema con diferente tiempo de cómputo y diferente nivel de calidad. Así, se permite acotar el tiempo de cómputo para el caso peor de las entidades especificadas sin necesidad de, a priori, podar las funcionalidades de las técnicas de IA utilizadas. Estas técnicas, por su propia naturaleza, tienen tiempos de cómputo no limitados o poco realistas, pudiendo utilizar técnicas de IA en la resolución de problemas de tiempo real críticos. Para poder garantizar su planificabilidad es necesario conocer los tiempos de respuesta en los peores casos de las entidades en ejecución que los constituyen.

El MKS agrupa varios niveles de fuentes de conocimiento.

- Una fuente de conocimientos (KS - *Knowledge Source*) representa el conocimiento (métodos, heurísticas, sistemas basados en reglas o cualquier otra técnica de IA) básico de resolución de una parte de un problema. El conocimiento puede ser representado en forma procedural, mediante un lenguaje basado en reglas

o mediante la utilización de otros formalismos de representación del conocimiento. El KS se puede caracterizar por el tipo de operación que realiza en el sistema, así:

- ✓ KS de percepción cuando realiza operaciones de entrada, entonces leerá los valores de variables observables a través de cámaras.
- ✓ KS de cognición cuando hace operaciones de razonamiento e inferencia, operará sobre valores observables, no observables e internos calculando valores de variables internas, de salida o no observables.
- ✓ KS de acción cuando sus operaciones son de salida. Actuará sobre el mundo mediante actuadores a partir de los valores de las variables de salida previamente calculadas.

2.2.2.7. Modelo del conocimiento

Este modelo muestra el conocimiento usado por el sistema para solucionar sus TAN incluyendo sus especificaciones de TR. A través del diagrama de conceptos se presenta el esquema del dominio (*ver Figura 142*).

El único agente que interactúa con el sistema es el usuario u operador que se encarga de manejarlo. De resto el conocimiento del sistema se ha obtenido de personas (los expertos o agentes que proporcionan el conocimiento del proceso a realizar por el sistema controlador y de su manejo), pero que para efectos del sistema final no tendrá ninguna relación.

2.2.2.7.1. Factores o variables del proceso de control del micro mundo

Los factores que se investigan de acuerdo a su posible análisis son:

- **Factores Controlables**

Las variables del proceso o variables de entrada que se pueden fijar en el nivel de operación, es decir dentro del grupo de variables que hacen parte del banco de pruebas, se encuentran:

- ✓ Posición de la botella.
- ✓ Posición de la cámara.
- ✓ Posición de la luz.
- ✓ Tipo de iluminación.

- ✓ Color del interior del banco de pruebas.

- **Factores Estudiados (Indirectos)**

Las variables que se investigan en el experimento, en cuanto a cómo influyen o afectan a la variable de respuesta presentes en las imágenes son:

- ✓ Posición de la botella dentro del banco de pruebas
- ✓ Tonalidad
- ✓ Brillo
- ✓ Nitidez
- ✓ Sombra
- ✓ Textura
- ✓ Contraste

- **Factores No Controlables**

Las variables que no se pueden controlar durante la operación normal del proceso en este caso son las siguientes:

- ✓ Fisuras en la botella.
- ✓ Posición igual de la etiqueta en todas las botellas.
- ✓ Desgaste de la botella.
 - Cambios en la tonalidad.
 - Pérdida de transparencia del vidrio.

Una vez identificados y clasificados los factores, se selecciona el diseño experimental, en donde el objetivo del experimento se ha utilizado como un criterio general de clasificación.

2.2.2.7.2. Diseño y Construcción del Banco de Pruebas

En el banco de pruebas se controlan las siguientes variables del sistema:

- **Posición y ángulo de la cámara**

Se establece la mejor posición de la cámara para la captura de las fotografías, ubicándose en la parte frontal del banco de pruebas. También se construye un soporte con el cual se asegura dicha posición y ángulos de la cámara. Para la captura de las

imágenes se utiliza una cámara de la *Figura 64*, y el banco de pruebas se lo puede apreciar en la *Figura 66* y *Figura 67*.

Los criterios que se tienen en cuenta en la elección de la posición más adecuada para la cámara son:

- ✓ Menor cantidad de brillos presentes en el fondo y en la boca de la botella.
- ✓ Reducción de la sombra en la parte frontal de la botella.
- ✓ Textura más uniforme observada en el fondo del banco de pruebas.
- ✓ Mayor cantidad de información obtenida del fondo de la botella en cuanto al ángulo de toma de la fotografía.

- **Posición y tipo de luz**

Se hace un análisis para detectar la mejor posición de la iluminación dentro del banco de pruebas; después del análisis correspondiente se centran las pruebas en la parte superior y posterior (trasera), utilizando 2 posiciones para las lámparas tipo reflector. Finalmente se diseñan y elaboran a mano los reflectores utilizando papel aluminio y papel cebolla. Se realizan pruebas con cuatro tipos diferentes de luz, escogiéndose en color blanco.

Los criterios que se tienen en cuenta en la elección de la posición y del tipo de luz apropiados para la toma de las imágenes son:

- ✓ Menor cantidad de brillos presentes tanto en el fondo del banco de pruebas, como en fondo y en la boca de la botella.
- ✓ Reducción del número de sombras presentes en la imagen.
- ✓ Reducción de la sombra en la parte frontal de la botella.
- ✓ Disminución del efecto de reflexión causado por el vidrio transparente de la botella.
- ✓ Menor contraste entre la base donde se coloca la botella y el fondo del banco de pruebas.
- ✓ Textura más uniforme observada en el fondo del banco de pruebas.
- ✓ Disminución de anomalías presentes en la imagen, tales como aparición de rayos de luz y diferentes colores en la botella, proyección del color de la etiqueta en el vidrio de la botella.

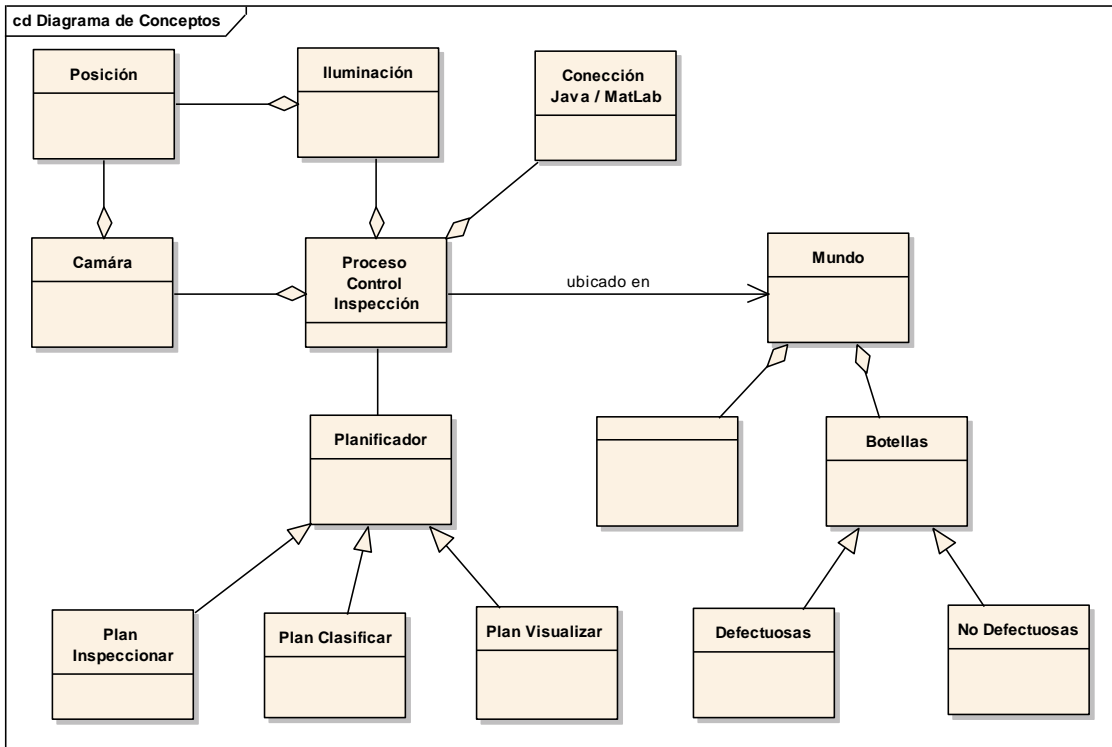


Figura 142: Diagrama de esquema de dominio para el sistema de inspección en el micro mundo

En cuanto al conocimiento de las tareas, el planificador se puede apreciar a través de una máquina de estados de la siguiente forma (ver Figura 143):

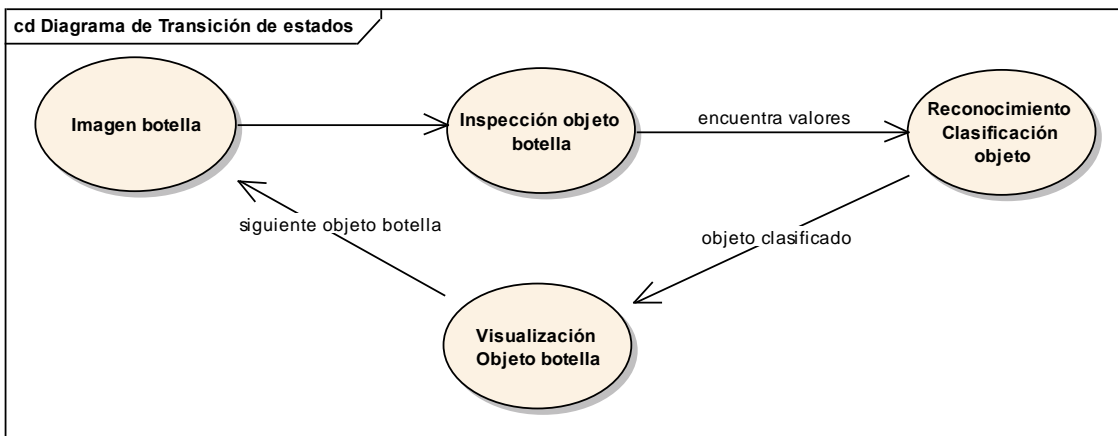


Figura 143: Diagrama de transición de estados del planificador

2.2.2.8. Modelo de diseño

2.2.2.8.1. Toma de muestras fotografías

Después de hacer diversos análisis teniendo en cuenta las variables del sistema y obtenido los datos necesarios extraídos se determina una unidad experimental de 830 imágenes, que representan 830 muestras posibles de botellas con defectos y sin ellos, que se presentan frecuentemente a la salida de la lavadora.

Para el desarrollo de este proyecto se empleó una cámara digital tipo Web-Cam con una resolución de 640 por 480 píxeles, lo suficiente para tener imágenes digitales con una calidad representativa de la escena original, baja carga computacional en el procesamiento con una velocidad de captura de 30 fps (frames por segundo). El objetivo de trabajar con una Web-Cam fue brindar versatilidad al sistema desarrollado, permitiendo que él mismo opere con computadores de escritorio, además de los bajos costos en la adquisición de este tipo de cámaras y la fácil disponibilidad de las mismas para los usuarios.

Para empezar se preparan todas las muestras de botellas de 375 cm³, partiendo de imágenes de igual tamaño y en formato JPEG¹⁷. A estas imágenes se les aplica las tareas de la sección anterior con la ayuda del software MatLab, y como fase inicial se desarrolla el algoritmo.

2.2.2.8.2. Desarrollo del algoritmo en tiempo diferido

Una vez estudiados y comprendidos los métodos y funciones internas de los Toolboxes de Image Processing, Wavelet y Statistics del MatLab se procede a realizar las funciones que realizan los cálculos de la aplicación.

En este momento se implementan las tareas mencionadas en los modelos anteriores, para ello, se desarrolla el algoritmo para inspección del objeto botella en tiempo diferido, una vez concluidas las respectivas pruebas de código, se implementan en tiempo real para su respectiva evaluación de desempeño.

Las funciones generales implementadas en código MatLab, se describen en la TABLA XIII.

¹⁷ JPEG (Del inglés Joint Photographic Experts Group): Formato de archivo de imagen más común utilizado por las cámaras fotográficas digitales y otros dispositivos de captura de imagen.

TABLA XIII: FUNCIONES DEL ALGORITMO IMPLEMENTADAS EN MATLAB

Función	Descripción General
algoritmoInspeccion	<p>RESULTADO = algoritmoInspeccion (rutaLocalizacion, nombreImagen): función que prepara la imagen, se extraen datos y características de la misma por medio de funciones desarrolladas, los cuales posteriormente son evaluados en la red neuro-difusa para su clasificación y obtener el resultado, además guarda la imagen analizada.</p> <ul style="list-style-type: none"> • RESULTADO: resultado de la inspección, es de tipo double • rutaLocalizacion: dirección o ruta de localización de la imagen de entrada • nombreImagen: nombre del fichero de imagen de entrada.
umbralOtsu	<p>[sigmaB2max, umbralOptimo] = umbralOtsu (ImagenFiltrada): función que calcula el umbral óptimo de la imagen.</p> <ul style="list-style-type: none"> • umbralOptimo: valor estadístico de la imagen. • sigmaB2max: valor estadístico de la imagen. • ImagenFiltrada: imagen filtrada de entrada.
std2	<p>desviacion = std2(imagenRecortada): función de MatLab que calcula la desviación estándar de los valores en imagenRecortada.</p> <ul style="list-style-type: none"> • imagenRecortada: imagen de entrada que puede ser numérica o lógica • desviacion: valor estadístico de la imagen
defectosEnRGB	<p>imagenConFallasRGB, area_ImagenConFallasRGB, Imagen_DiferenciaRGB = defectosEnRGB (imagenRecortadaRgb): función que extrae los planos R, G y B de la imagen, los compara y aprovecha su diferencia para intensificar la zona defectuosa, así obtiene el valor y una imagen en escala de gris.</p> <ul style="list-style-type: none"> • imagenRecortadaRgb: imagen de entrada de tipo RGB • imagenConFallasRGB: imagen de salida que muestra el defecto o zona defectuosa, es de tipo double logica • area_ImagenConFallasRGB: valor de salida del área defectuosa, es de tipo double. • Imagen_DiferenciaRGB: imagen de salida en escala de gris, representa la diferencia entre los planos RGB.
bordesImagenDiferenciaRGB	<p>imagenDiferenciaRGBFallas = bordesImagenDiferenciaRGB (Imagen_DiferenciaRGB): función que analiza los bordes de la imagen</p>

	<p>de entrada, los resalta o excluye dependiendo de los valores de cada píxel.</p> <ul style="list-style-type: none">• Imagen_DiferenciaRGB: imagen de entrada en donde resaltan los colores RGB en escala la gris.• imagenDiferenciaRGBFallas: imagen de salida que muestra la zona de defecto, es de tipo lógica.
extraerBorde1	<p>imagenFallaBorde1 = extraerBorde1 (imagenGris): función que prepara la imagen por medio de funciones de erosión, dilatación, reconstrucción morfológica, y complemento (inversión), posteriormente la analiza por medio de bordes y estos son resaltados o excluidos dependiendo de los valores de cada píxel.</p> <ul style="list-style-type: none">• imagenGris: imagen de entrada original en escala gris.• imagenFallaBorde1: imagen de salida que muestra la zona de defecto, es de tipo lógica.
extraerBorde2	<p>imagenFallaBorde2 = extraerBorde2 (imagenFiltrada, umbral): función que utiliza filtros en la imagen para posteriormente analizarla por medio de bordes, estos son resaltados o excluidos dependiendo de los valores de cada píxel.</p> <ul style="list-style-type: none">• imagenFiltrada: imagen de entrada en gris con reducción de ruido.• imagenFallaBorde2: imagen de salida que muestra la zona de defecto, es de tipo lógica.• umbral: valor estadístico de la imagen calculado anteriormente

- **Descripción de pasos del Algoritmo de toma de imágenes en tiempo diferido**

Las funciones descritas en la *TABLA XIII* y funciones internas de los toolboxes de Matlab, son llamadas como si fueran métodos de Java, con sus respectivos parámetros y retorno de variables.

La primera parte de la función principal llamada **algoritmoInspeccion**, carga y prepara la imagen para su posterior extracción de datos.

Cargar la Imagen

```
I=imread('C:/imagen.jpg');
```

La función **imread** lee la imagen de un archivo, indicando el camino del directorio C:/ en el cual se encuentra la imagen para que MATLAB pueda cargarla.

En caso del presente algoritmo, la ruta y nombre de la imagen son proporcionados desde la aplicación hecha en Java, las mismas puede ser cambiadas desde la aplicación a conveniencia.

```
function resultado =  
algoritmoInspección(rutaLocalizacion,nombreImagen)  
nombreYRuta=strcat(rutaLocalizacion,nombreImagen);  
I=imread(nombreYRuta);
```

La función `imshow(I)` muestra la imagen deseada en pantalla (ver *Figura 144*).

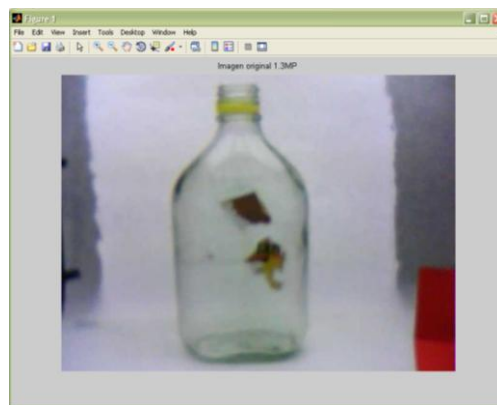


Figura 144: Imagen Original 640 x 480 (1,3M píxeles)

Recortar la Imagen

```
rect=[130 0 300 480];  
imagenRecortada = imcrop(I,rect);
```

La función **imcrop** permite recortar una imagen en un rectángulo especificado (ver *Figura 145*).



Figura 145: Imagen Recortada

Pasar a escala de grises

```
imagenGris=rgb2gray(imagenRecortada);
```

La función **rgb2gray** convierte la imagen de color verdadero RGB a una imagen en escala de grises (256 niveles de gris) (ver Figura 146).



Figura 146: Imagen en Gris

```
imagenFiltrada=wiener2(imagenGris,[5 5]);
```

La función **wiener2** es un filtro para reducir ruidos en la imagen (ver Figura 147).

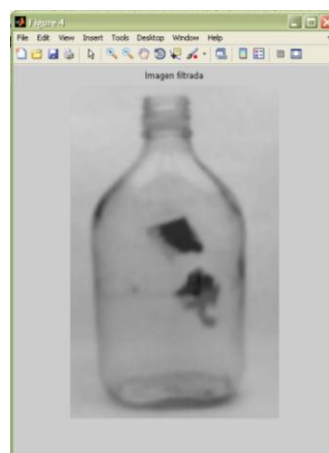


Figura 147: Imagen Filtrada

Análisis de Datos: Valor de Sigma, Umbral y desviación estándar

Dada a que una imagen $f(x,y)$ se representa en MatLab como una matriz, se pueden analizar sus datos estadísticamente.

```
function [sigmaB2max, umbralOptimo] = umbralOtsu(imagenFiltrada)
```

La función **umbralOtsu** está hecha en un archivo *.m* de MatLab que calcula el umbral óptimo y extrae el valor **sigma**, aquí se ha extraído una característica o valor de la imagen que será tomado en cuenta al momento del entrenamiento con lógica Fuzzy.

```
desviacion = std2(imagenRecortada);
```

La Función **std2** computariza la **desviación estándar** de una matriz de elementos, en este caso de los valores de *imagenRecortada*. Este valor también nos sirve para el análisis y entrenamiento.

Extracción de regiones y medida de áreas

Para extraer estas características se hace uso de las funciones secundarias: *defectosEnRGB*, *bordesImagenDiferenciaRGB*, *extraerBorde1* y *extraerBorde2* de la *TABLA XIII*.

Función *defectosEnRGB*

```
function [imagenConFallasRGB, area_ImagenConFallasRGB,  
Imagen_DiferenciaRGB] = defectosEnRGB(imagenRecortadaRgb)
```

Extrae los colores individualmente en plano RGB, Red=Rojo, Green=Verde y Blue=Azul, esto devuelve 3 imágenes en escala de gris (ver *Figura 148*) que se filtran con la función *wiener2*.

```
PlanoRojo = imagenRecortadaRgb(:, :, 1);  
PlanoVerde = imagenRecortadaRgb(:, :, 2);  
PlanoAzul = imagenRecortadaRgb(:, :, 3);  
PlanoRojoFiltrado=wiener2(PlanoRojo,[5 5]);  
PlanoVerdeFiltrado=wiener2(PlanoVerde,[5 5]);  
PlanoAzulFiltrado=wiener2(PlanoAzul,[5 5]);
```

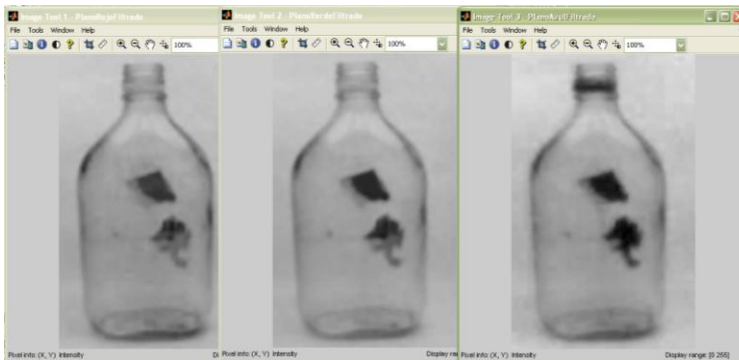


Figura 148: Imágenes en gris extraídas del plano R, G y B respectivamente.

Luego se convierten las imágenes en lógicas (1=blanco, 0=negro) (ver Figura 149), mediante la comparación de un umbral arbitrario (threshold) utilizando el operador relacional "<".

```

imagenLogica1=(PlanoRojoFiltrado < 55);
imagenLogica2=(PlanoVerdeFiltrado < 65);
imagenLogica3=(PlanoAzulFiltrado < 63);

```

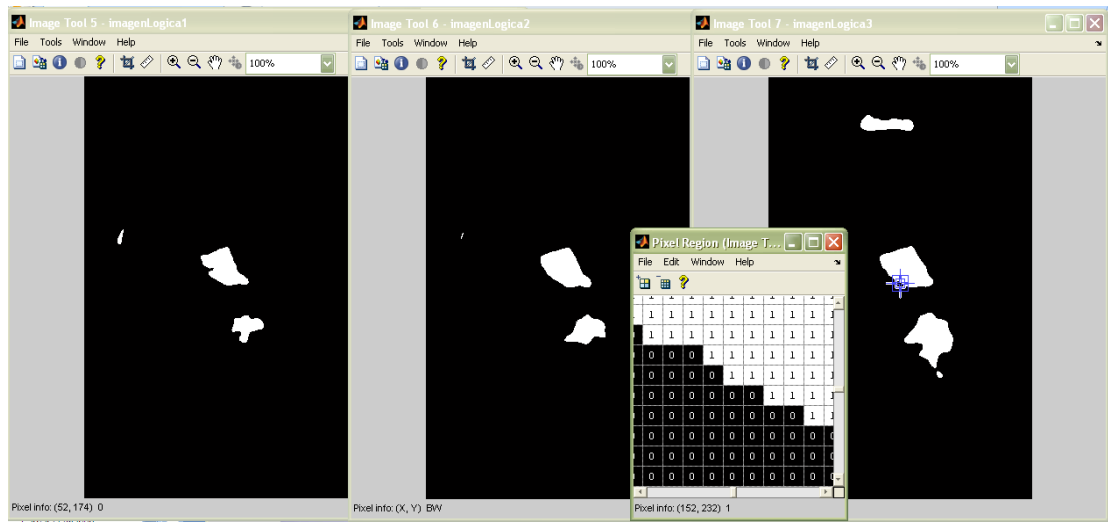


Figura 149: Imágenes Lógicas.

Se realiza una suma de las 3 imágenes lógicas, como si fueran matrices y se crea una imagen resultante (ver Figura 150).

```

Imagen_Resultante=imagenLogica1+imagenLogica2+imagenLogica3;

```

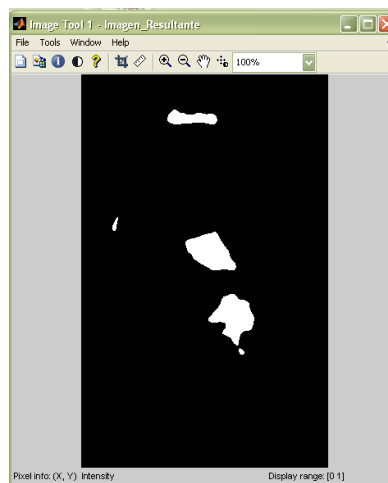


Figura 150: Imagen Resultante

Por otro lado aplico la función **imabsdiff** sobre las tres imágenes en gris anteriores de la *Figura 148*, esta función devuelve una imagen con la diferencia absoluta (valor) entre ellas (ver *Figura 151*).

```
Imagen_Diferencia = imabsdiff(PlanoRojoFiltrado,PlanoAzulFiltrado);
Imagen_DiferenciaRGB =imabsdiff(Imagen_Diferencia,PlanoVerdeFiltrado);
```



Figura 151: Imagen Diferencia de RGB.

La *Figura 151* se envía como variable de retorno de la función **defectosEnRGB**, siguiendo con la misma función se procede a cortar la parte inferior (base de la botella) de la imagen de la *Figura 145*, es decir solo se analiza la parte superior extrayendo cada uno de los planos R, G y B, y además se reduce el ruido.

```
nueva = uint8(zeros(size(imagenRecortadaRgb)));
nueva = imcomplement(nueva);
rect=[0 0 300 450];
imagenR = imcrop(imagenRecortadaRgb,rect);
nueva(1:450,1:300,:) = imagenR;
imagenRecortadaRgb = nueva;
```

```
PlanoRojo = imagenRecortadaRgb(:, :, 1);
PlanoVerde = imagenRecortadaRgb(:, :, 2);
PlanoAzul = imagenRecortadaRgb(:, :, 3);
```

```
PlanoRojoFiltrado=wiener2(PlanoRojo,[5 5]);
PlanoVerdeFiltrado=wiener2(PlanoVerde,[5 5]);
PlanoAzulFiltrado=wiener2(PlanoAzul,[5 5]);
```


Se realiza la diferencia de los 3 planos filtrados mediante la combinación de cada uno de los mismos y se comparan mediante un umbral arbitrario.

```
s1=imabsdiff(PlanoRojoFiltrado,PlanoVerdeFiltrado);  
ss1=s1<28;  
s2=imabsdiff(PlanoRojoFiltrado,PlanoAzulFiltrado);  
ss2=s2<40;  
s3=imabsdiff(PlanoVerdeFiltrado,PlanoAzulFiltrado);  
ss3=s3<39;  
s4=imabsdiff(PlanoVerdeFiltrado,PlanoRojoFiltrado);  
ss4=s4<30;  
s5=imabsdiff(PlanoAzulFiltrado,PlanoRojoFiltrado);  
ss5=s5<40;  
s6=imabsdiff(PlanoAzulFiltrado,PlanoVerdeFiltrado);  
ss6=s6<39;
```

Luego se efectúa la operación lógica AND entre las 6 imágenes combinadas resultantes y se aplica la función **imcomplement** (ver *Figura 152*).

```
imagenComplementoPlanos = imcomplement(ss1 & ss2 & ss3 & ss4 & ss5 &  
ss6 );
```

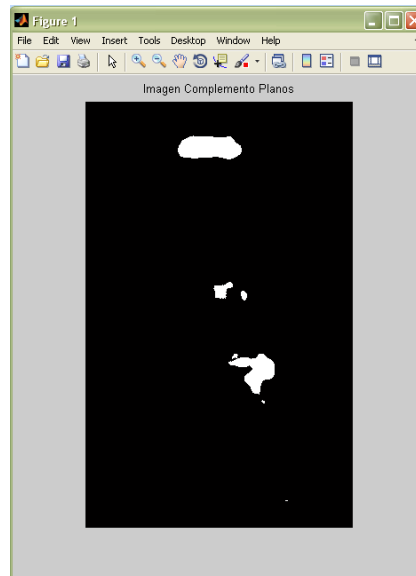


Figura 152: Imagen Complemento entre Planos

Por otro lado a la imagen de la *Figura 150* se le aplica una restricción para poder discriminar zonas no defectuosas, mediante la obtención de los valores promedio de los píxeles de la imagen en gris correspondientes y comparándolos con un valor analizado, esto se hace con **regionprops**, esta función extrae un conjunto de medidas por cada componente (objeto) conectado en la imagen binaria, se utiliza la propiedad de medida de contorno **PíxelIdxList** (ver *Figura 153*).

```

imagenGris = rgb2gray(imagenRecortadaRgbAux);
pixel = regionprops(Imagen_Resultante, 'all');

for k = 1:numel(pixel)
    idx = pixel(k).PixelIdxList;
    xx = mean(imagenGris(idx));
    if ((xx <= 48.0))
        Imagen_Resultante(pixel(k).PixelIdxList)=0;
    end;
end

```

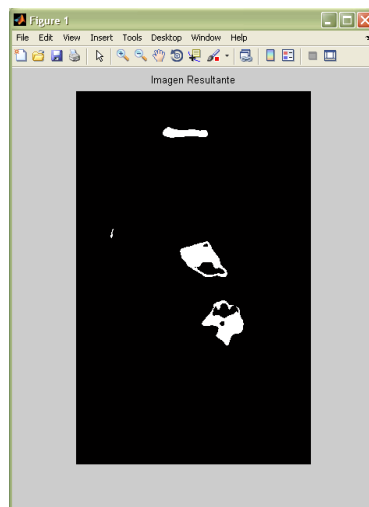


Figura 153: Imagen Resultante después de la restricción

Se suma las imágenes de la *Figura 153* y *Figura 152*, para obtener la imagen que representa las zonas con defectos desde una imagen en RGB (ver *Figura 154*).

```

imagenConFallasRGB=(logical(Imagen_Resultante))+imagenComplementoPlanos;

```

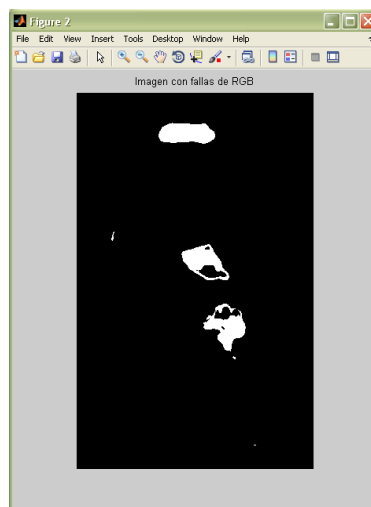


Figura 154: Imagen con Fallas de RGB

Finalmente en esta función, se obtiene el valor del área total de la *Figura 154*, está representada de color blanco en la imagen, se aplica la función **regionprops** con la propiedad de medida **Area** y se realiza una suma de ellas. En este paso se ha extraído otro valor para el análisis y entrenamiento.

Área: Calcula el área en píxeles cuadrados de la región.

```
area=regionprops(imagenConFallasRGB, 'Area');  
area_ImagenConFallasRGB=sum(cell2mat(struct2cell(area)));
```

Función **bordesImagenDiferenciaRGB**

```
function imagenDiferenciaRGBFallas = bordesImagenDiferenciaRGB  
(Imagen_DiferenciaRGB)  
I=wiener2(Imagen_DiferenciaRGB,[5 5]);
```

A la imagen de la *Figura 151*, se le aplica la reducción de ruido con **wiener2** y la detección de bordes por medio de la función **edge** de MatLab, con los operadores **canny** y **sobel**, esta función encuentra los bordes de una imagen en distintos niveles de intensidad y devuelve una matriz binaria compuesta de “1’s” los cuales indican los bordes y de “0’s” que indican la parte negra o hueca de la imagen (ver *Figura 155*).

```
PSF = fspecial('gaussian',4,10);  
Blurred = imfilter(I,PSF,'symmetric','conv');  
  
[junk threshold] = edge(Blurred, 'canny');  
fudgeFactor = 0.95;  
BW_s = edge(Blurred,'sobel', threshold * fudgeFactor);
```

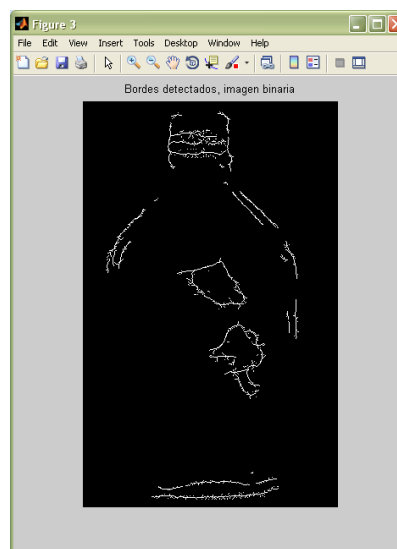


Figura 155: Bordes Detectados por operadores Canny y Sobel.

Luego se aplican algunas funciones de filtrado morfológico (dilatación, llenado de huecos, limpieza de bordes, creación estructuras en los elementos y eliminación de ruido, como se puede observar en la *Figura 156*.

```
se90 = strel('line',2,190);
se0 = strel('line',2.01,95);
BWsdil = imdilate(BWs, [se90 se0]);
BWdfill = imfill(BWsdil, 'holes');
BWnobord = imclearborder(BWdfill, 4);

seD = strel('disk',1);
BWfinal = imerode(BWnobord,seD);
BWfinal = imerode(BWfinal,seD);
```

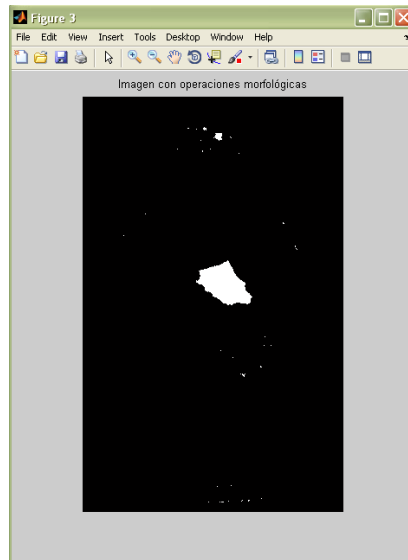


Figura 156: Imagen con operaciones morfológicas

Asimismo se aplica **regionprops**, esta función extrae un conjunto de medidas por cada componente (objeto) conectado en la imagen binaria, se utiliza la propiedad de medida de contorno **PixelIdxList**, la cual permite discriminar regiones de no interés, resultándonos la siguiente imagen (ver *Figura 157*).

```
pixel = regionprops(BWfinal, 'PixelIdxList');
for k = 1:numel(pixel)
    idx = pixel(k).PixelIdxList;
    xx = mean(I(idx));
    if ((xx >= 85.0))
        BWfinal(pixel(k).PixelIdxList) = 0;
    end;
end

imagenDiferenciaRGBFallas=BWfinal;
```

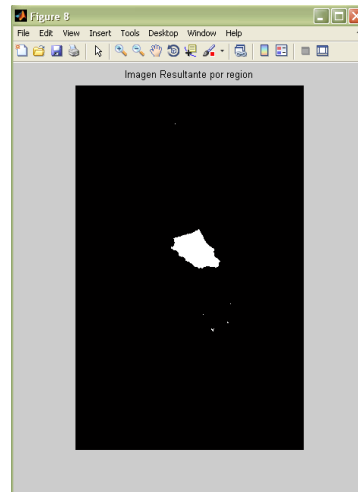


Figura 157: Imagen Diferencia RGB con defectos

A la imagen de la *Figura 157* se la envía como variable de retorno a la función principal **algoritmoInspeccion**.

Función extraerBorde1

```
imagenFallaBorde1 = extraerBorde1(imagenGris)
```

A la *Figura 146* (Imagen en Gris) se le realiza una abertura y cerramiento por reconstrucción mediante algoritmos morfológicos (ver *Figura 158*).

```
se = strel('disk', 13);  
Ie = imerode(imagenGris, se);  
Iobr = imreconstruct(Ie, imagenGris);  
Iobrd = imdilate(Iobr, se);  
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));  
Iobrcbr = imcomplement(Iobrcbr);
```

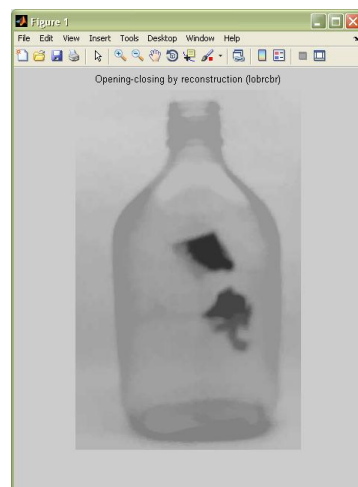


Figura 158: Abertura y cerramiento por reconstrucción

Del mismo modo a la *Figura 146* se le aplica la función **wiener2** para reducir el ruido, y así obtener un umbral mediante la función **edge** con el operador **canny**, dicho umbral se emplea a la *Figura 158* para aplicar otra vez la función **edge** con el operador **sobel**, este umbral indica la magnitud o sensibilidad para encontrar los bordes fuertes y débiles de la imagen, **edge** ignora todos los bordes que no son tan fuertes que el umbral (ver *Figura 159*).

```
I=wiener2 (imagenGris,[5 5]);

[junk threshold] = edge (I, 'canny');
fudgeFactor = 1;
BWs = edge (Iobrcbr,'sobel', threshold * fudgeFactor);
```

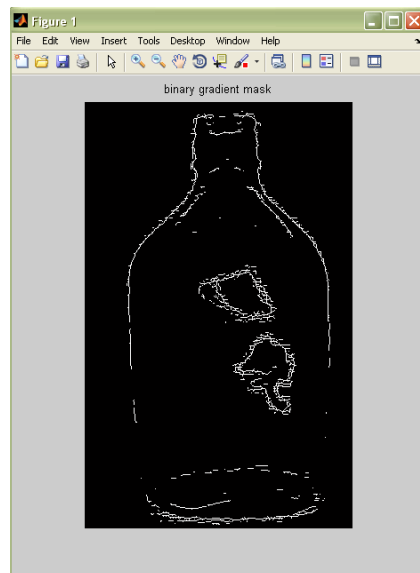


Figura 159: Imagen binaria en bordes.

Efectuó luego operaciones morfológicas como unión de líneas (bordes) continuas, relleno de huecos, limpieza de bordes y erosión, esto se hace con otros parámetros (valores) previamente analizados (ver *Figura 160*).

```
se90 = strel('line', 3, 90);
se0 = strel('line', 3, 180);
BWsdil = imdilate(BWs, [se90 se0]);

BWdfill = imfill(BWsdil, 'holes');

BWnobord = imclearborder(BWdfill, 4);

seD = strel('disk',2);
imagenFallaBorde1 = imerode(BWnobord,seD);
imagenFallaBorde1 = imerode(imagenFallaBorde1,seD);
```

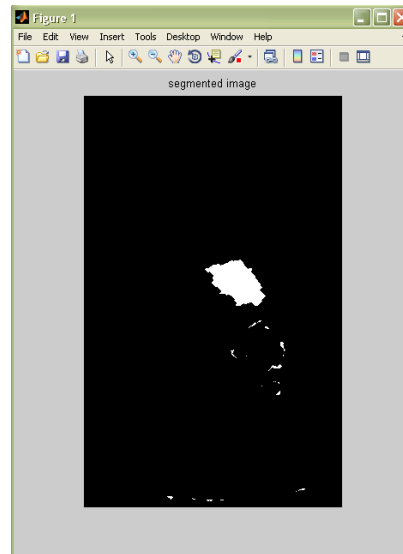


Figura 160: Imagen segmentada por morfología.

Asimismo se aplica la función **regionprops** con la propiedad **PixelIdxList**, para discriminar regiones o zonas de no interés (ver Figura 161).

```
pixel = regionprops(imagenFallaBorde1, 'PixelIdxList');  
for k = 1:numel(pixel)  
    idx = pixel(k).PixelIdxList;  
    xx = mean(I(idx));  
    if ((xx >= 112.0))  
        imagenFallaBorde1(pixel(k).PixelIdxList) = 0;  
    end;  
end
```

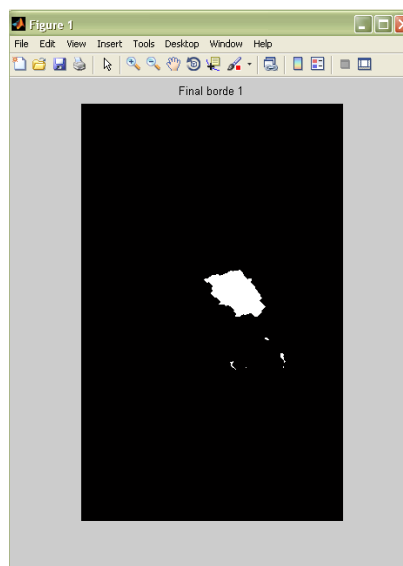


Figura 161: Imagen final con fallas de bordes 1

A esta imagen (Figura 161), se la envía como variable de retorno en esta función.

Función extraerBorde2

```
function imagenFallaBorde2 = extraerBorde2(imagenFiltrada)
```

En esta función se discrimina la parte superior de la imagen, la cual consiste en generar una nueva imagen a partir la *Figura 147*, y resulta la imagen de la *Figura 162*.

```
recovered = uint8(ones(size(I)));
recovered(:,:,1) = umbral-19;
rect=[0 100 301 380];
I = imcrop(I,rect);
recovered(100:480,1:301,:) = I;
I=recovered;
```



Figura 162: Imagen Recortada

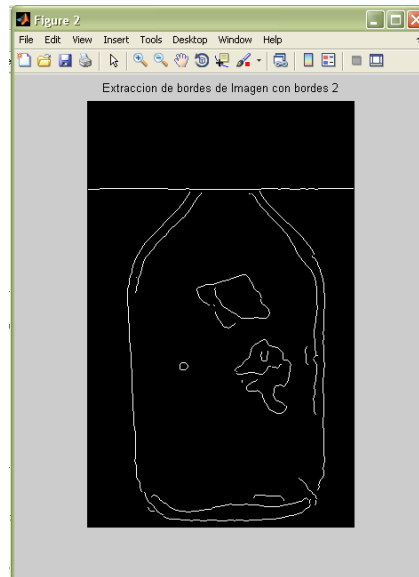


Figura 163: Imagen Segmentada por bordes 2

Seguidamente se emplea el filtro gaussiano **fspecial** a la imagen de la *Figura 162*, este elimina el ruido mejor manera y además emborrona menos los bordes. Al aplicar estos filtros conviene tener cuidado ya que puede perderse la información de los píxeles. También se aplica **edge** para extraer bordes (ver *Figura 163*).

```
PSF = fspecial('gaussian',3,10);
Blurred = imfilter(I,PSF,'symmetric','conv');
WEIGHT = edge(Blurred,'canny',0.2);
```

Consecutivamente se aplica dilatación o ampliación de los bordes encontrados anteriormente mediante la función **strel** con sus respectivos parámetros, estos fueron anteriormente analizados mediante prueba y error. Esto ayuda a extraer las zonas con defectos más significativos al cerrarse la estructura (ver *Figura 164*).


```
se90 = strel('line', 2.2,180);  
se0 = strel('line', 2.2, 90);  
BWsdil = imdilate(WEIGHT, [se90 se0]);
```

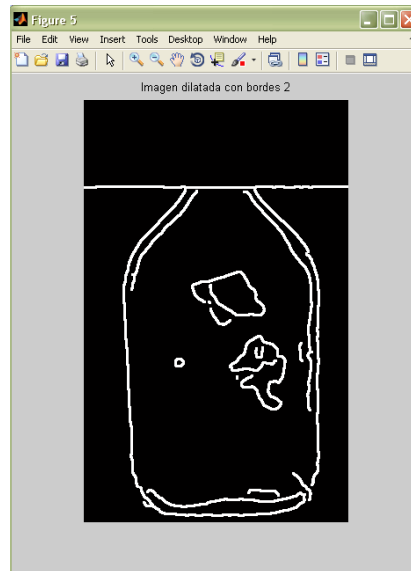


Figura 164: Imagen dilatada con bordes 2

Se llenan los huecos con **imfill**, es decir la estructura que fue cerrada o conectada, y se limpian los bordes extras mediante la consideración de parámetros en la función **imclearborder** (ver Figura 165).

```
BWdfill = imfill(BWsdil, 'holes');  
BWnobord = imclearborder(BWdfill, 4);
```

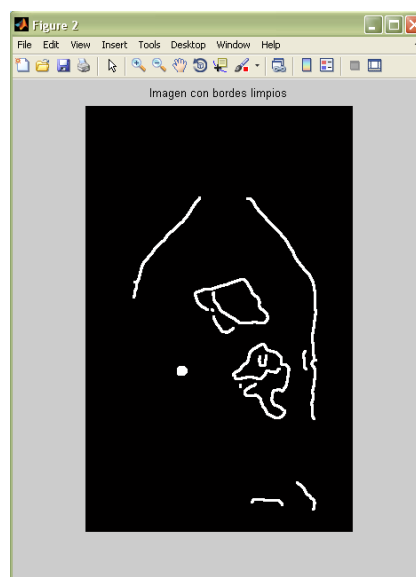


Figura 165: Imagen rellenada

La función **strel** permite crear una estructura morfológica de una determinada forma y anchura de **N** píxeles, y la función **imerode** permite que la operación morfológica erosione a la imagen almacenada en **seD**, para limpiar los bordes no deseados de la imagen, como sigue de la siguiente manera (ver *Figura 166*).

```
seD = strel('disk', 1);
imagenFallaBorde2 = imerode(BWnobord, seD);
imagenFallaBorde2 = imerode(imagenFallaBorde2, seD);
seD = strel('disk', 2);
imagenFallaBorde2 = imerode(imagenFallaBorde2, seD);
```

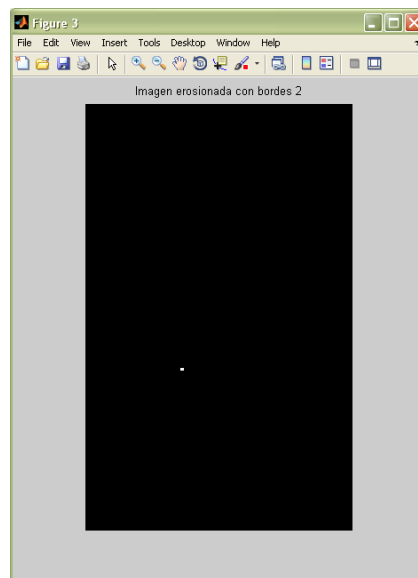


Figura 166: Imagen erosionada con bordes 2

Por consiguiente se realiza un último análisis de la imagen anterior, para así poder discriminar alguna zona de no interés persistente, este consiste en extraer valores de los píxeles de la imagen binaria de la *Figura 166* con la función **regionprops** y su propiedad de medida de forma '**PixellIdxList**', la cual extrae los índices lineales de los píxeles en la región o regiones (blanco = 1).

En la imagen en gris de la *Figura 147* se localizan los índices de la *Figura 166* y se computariza cada región por medio de su promedio **mean** y varianza **var**, estas medidas se comparan por medio de restricciones examinadas en las zonas de las imágenes; es decir los valores de los píxeles en gris menores representan zonas más oscuras y los mayores a las más claras, entonces se asigna el valor 0 (negro) al píxel de la *Figura 166* si cumple la restricción con el valor umbral constante sondeado.

Luego se obtiene la *Figura 167* que representa la imagen final de la función **extraerBorde2** y se la envía como variable de retorno de esta función.

```
pixel = regionprops(imagenFallaBorde2, 'PixelIdxList');  
  
for k = 1:numel(pixel)  
    idx = pixel(k).PixelIdxList;  
    promedio = mean(I(idx));  
    varianza = var(double(I(idx)));  
  
    if ((promedio >= 146.0))  
        imagenFallaBorde2(pixel(k).PixelIdxList) = 0;  
    end;  
  
    if ((varianza >= 292.0))  
        imagenFallaBorde2(pixel(k).PixelIdxList) = 0;  
    end;  
end
```

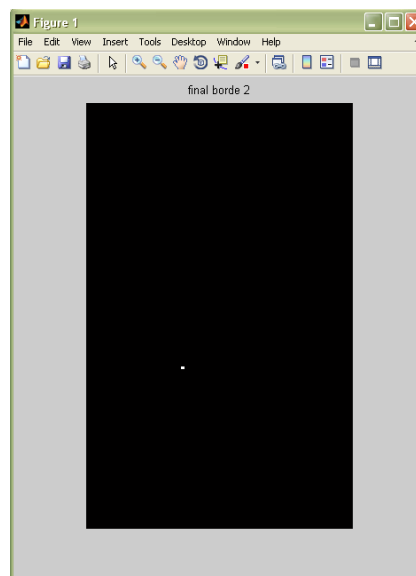


Figura 167: Imagen final con fallas de bordes 2

Regresando a la función principal **algoritmoInspeccion** del algoritmo se utiliza **imadd**, función que suma las imágenes de retorno de las funciones secundarias **bordesImagenDiferenciaRGB**, **extraerBorde1** y **extraerBorde2** (ver *Figura 168*).

```
imagenDiferenciaRGBDefecto = bordesImagenDiferenciaRGB  
(Imagen_DiferenciaRGB);  
  
imagenFallaBorde1 = extraerBorde1(imagenGris);  
imagenFallaBorde2 = extraerBorde2(imagenFiltrada, umbral);
```

```

imagenesFallasBordes=imadd(imagenDiferenciaRGBDefecto,imagenFallaBorde
1);
imagenesFallasBordes=imadd(logical(imagenesFallasBordes),imagenFallaBo
rde2);

```

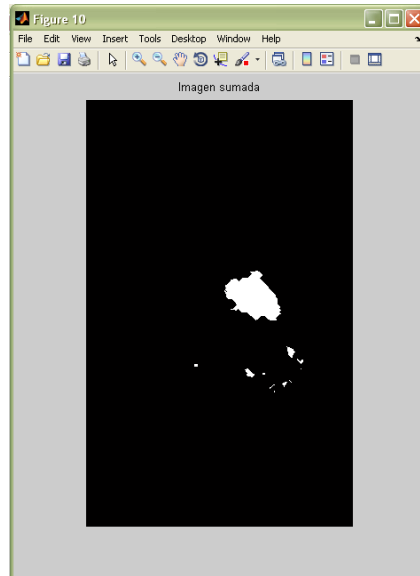


Figura 168: Imagen sumada por bordes

A la imagen anterior se le aplica **regionprops** con la propiedad **Area**, esta(s) área(s) corresponde(n) al actual número de píxeles blancos en la región y son puestos en un arreglo, para poder sumarlos con la función **sum** y así obtener un área total.

```

areas=regionprops(imagenesFallasBordes,'Area');
areaImagenConBordes=sum(cell2mat(struct2cell(areas)));

```

El valor de `areaImagenConBordes` es la última característica, para empezar a estudiar y analizar estadísticamente las variables extraídas, se van a comparar los resultados, encontrar diferencias y rangos numéricos de cada una de las imágenes de muestra escogidas para su posterior entrenamiento en la red neuro-difusa y de esta forma permitir un reconocimiento exitoso.

De la misma forma se aplica **imadd** para sumar la *Figura 154* y la *Figura 168*, resultando la imagen final binaria (ver *Figura 169*).

```

imagenesFallasFINAL =
imadd(logical(imagenConFallasRGB),logical(imagenesFallasBordes));

```

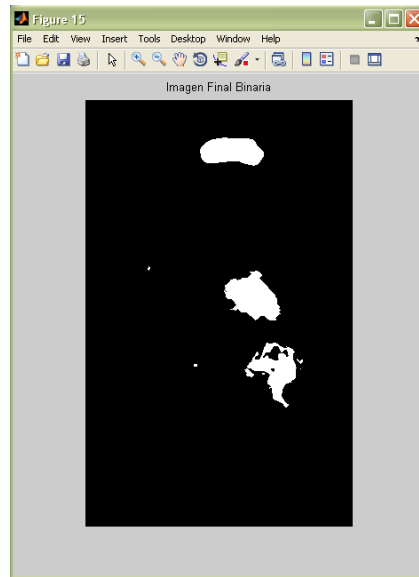


Figura 169: Imagen final binaria.

Un método alternativo para la visualización del objeto segmentado es colocando un contorno alrededor de la imagen original en RGB de la botella (ver *Figura 145*), es decir combinando la imagen original con los resultados obtenidos en la segmentación y extracción de características. El contorno es creado por la función `bwperim` (ver *Figura 170*).

```
ImagenBinariaFinal = bwperim(imagenesFallasFINAL, 6);  
imagenRecortada (ImagenBinariaFinal) = 255;
```

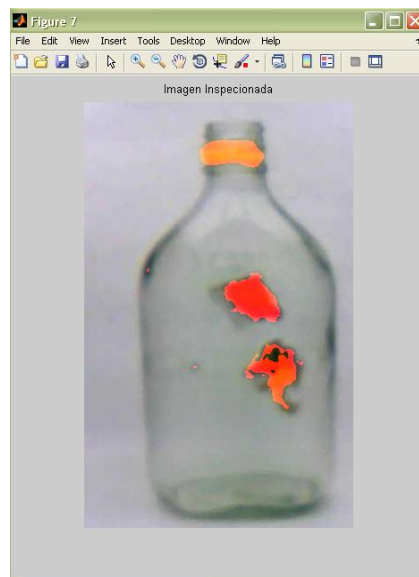


Figura 170: Imagen Original Inspeccionada

Por consiguiente se guarda la imagen en un archivo JPG, con su respectivo directorio para ser visualizada en la interfaz gráfica.

```
reducida = imresize(imagenRecortada,[320 NaN]);
rutaActual = pwd; %Identica el directorio actual
rutanueva=strcat(rutaActual, '/imagenAnalizada');
fNombre = fullfile(rutanueva,nombreImagen);
imwrite(reducida,fNombre);
```

Los cuatro datos extraídos: Desviación estándar, Sigma, Área de falla en RGB y Área de falla en Bordes, sirven para el entrenamiento del sistema de inspección y reconocimiento de botellas, basado en Lógica Fuzzy, estos datos pasan a ser los 4 elementos de cada uno de los 38 vectores que corresponden a las 38 imágenes de muestra de las botellas, En la *TABLA XIV* se pueden apreciar algunos de los datos extraídos con el algoritmo en MatLab.

TABLA XIV: DATOS EXTRAÍDOS CON EL ALGORITMO

Nº	Descripción de la Botella	Datos Extraídos					
		Desviación Estándar	Sigma	AreaRGB		AreaBordes	
				pixeles ²	cm ²	pixeles ²	cm ²
1	Limpia1	26.23566871869	3.83583764632e+02	0	0	22	0,015
2	Limpia2	25.901730530251	3.71260780439e+02	0	0	9	0,006
3	Limpia3	26.883302884235	4.03487980915e+02	0	0	3	0,002
4	Limpia4	26.115436227761	3.85416847939e+02	0	0	16	0,011
5	Limpia5	25.946697141422	3.85245044492e+02	40	0,028	0	0
6	Limpia6	26.072934038888	3.93162854417e+02	31	0,021	4	0,002
7	Limpia7	25.871312772694	3.92497831280e+02	9	0,006	24	0,016
8	Limpia8	25.706771253025	3.64638243697e+02	8	0,0056	0	0
9	Limpia9	26.647450083056	4.04906284588e+02	14	0,0098	0	0
10	Limpia10	26.648889117017	4.15534415022e+02	0	0	0	0
11	Pico malo1 (anillo)	28.235716299977	4.34049881149e+02	1202	0,841	385	0,269
12	Pico malo2 (anillo)	28.257282563866	4.28390481379e+02	1208	0,845	715	0,500
13	Pico malo3 (anillo y plástico)	27.294417863031	4.21819334963e+02	1010	0,707	456	0,31
14	Pico malo4 (anillo y plástico)	27.19393502116	4.18081137004e+02	946	0,662	2	0,001
15	Pico malo5 (anillo y colilla)	30.848135838451	5.10332615677e+02	1975	1,382	249	0,174



16	Pico malo6 (anillo y colilla)	31.211005382206	5.01582411931e+02	2939	2,057	59	0,041
17	Pico malo7 (sorbete)	30.642935115397	4.6816567575e+02	2001	1,400	1956	1,369
18	Pico malo8 (sorbete)	30.467420738326	4.8565203519e+02	1581	1,106	1923	1,346
19	Mala 1 (etiqueta 10%)	32.737000604212	5.72492389274e+02	303	0,212	2906	2,034
20	Mala 2 (etiqueta 10%)	31.849447246168	5.34683867172e+02	10	0,007	2624	1,836
21	Mala 3 (etiqueta 25%, anillo y colilla)	33.027459337425	5.52035627382e+02	2490	1,743	2536	1,775
22	Mala 4 (etiqueta 25%, anillo y colilla)	31.897678406944	5.50293545934e+02	1811	1,267	5593	3,915
23	Mala 5 (etiqueta 25%, plástico y colilla)	32.228882102317	5.45561038683e+02	2285	1,600	3205	2,244
24	Mala 6 (etiqueta 25%, plástico y colilla)	31.536188350758	5.42387676554e+02	1496	1,047	2970	2,079
25	Mala 7 (etiqueta 40% y colilla)	35.731748123946	6.14517877777e+02	3390	2,373	5100	3,570
26	Mala 8 (etiqueta 40% y colilla)	32.236046188563	5.45314482221e+02	221	0,155	5098	3,569
27	Mala 9 (anillo y etiqueta 50%)	41.060766727676	8.03472485822e+02	8021	5,615	6860	4,802
28	Mala 10 (anillo y etiqueta 50%)	33.940052794234	6.27172173997e+02	1043	0,730	8441	5,909
29	Mala 11 (anillo y etiqueta 60%)	37.441696256151	6.36903059709e+02	5757	4,030	5726	4,008
30	Mala 12 (anillo y etiqueta 60%)	35.137166913774	6.41986685456e+02	1308	0,916	6691	4,684
31	Mala 13 (anillo y etiqueta 90%)	50.898406103789	1.557794142e+03	15773	11,042	13560	9,492
32	Mala 14 (anillo y etiqueta 90%)	43.605480257527	1.2514364385e+03	1111	0,778	23767	16,638

33	Mala 15 (etiqueta 95%)	52.643252055500	1.653053043e+03	21104	14,773	28092	19,665
34	Mala 16 (etiqueta 95%)	43.998716911732	1.19084975253e+03	79	0,055	95	0,067
35	Mala 17 (botella completa)	70.547970737929	2.9777780062e+03	63644	44,552	9377	6,564
36	Mala 18 (botella completa)	68.177067557606	2.7502251329e+03	70742	49,521	2600	1,820
37	No existe botella 1	16.967621992593	1.92761379398e+02	0	0	0	0
38	No existe botella 2	17.149359825784	1.92514519680e+02	0	0	0	0

El área máxima considerada para que una botella no defectuosa (buena), pueda clasificarse correctamente es de 58 píxeles², para una mejor comprensión, ésta área se puede estimar en centímetros (cm) de cada lado del área, es decir en cm².

Para calcular el área de píxeles de la región en centímetros (cm²), se calcula la raíz cuadrada del área especificada para aproximar el valor al lado horizontal (coordenada X) y al lado vertical (coordenada Y), se recuerda que estos valores se encuentran en una matriz y el área se calcula en unidades cuadráticas, por ejemplo.

AreaRGB en píxeles² = 58

$$\sqrt{58} = 7,615773106 \text{ píxeles}$$

- 7,615773106 píxeles (X)
- 7,615773106 píxeles (Y)

Se utiliza la equivalencia 1 píxel (x) = 0.026458 cm y se calcula¹⁸.

- 7,615773106 píxeles (X) * 0.026458 cm = **0,20149812 cm** de lado horizontal (coordenada X), y así mismo para;
- 7,615773106 píxeles (Y) * 0.026458 cm = **0,20149812 cm** de lado vertical (coordenada Y).

Los píxeles generalmente tienen la forma de cuadrados o rectángulos, es decir, se tiene 0,20149812 cm de cada lado, por lo que se admite, que con este valor no se

¹⁸ Equivalencia tomada de la Sección Revisión de Literatura - Capítulo III - apartado 3.2.1.2 y 3.2.1.3

puede detectar una botella como defectuosa en el área de 58 píxeles² anteriormente mencionada. Por último se calcula el área:

$$\text{Área} = \text{Lado (Base)} * \text{Lado (Altura)}$$

$$\text{Área} = 0,20149812 \text{ cm} * 0,20149812 \text{ cm} = \mathbf{0,041 \text{ cm}^2}$$

Por consiguiente, para calcular el área de una botella completamente defectuosa, se escoge el valor máximo de 75976 píxeles², procediendo a efectuar el mismo cálculo anterior.

$$\text{AreaRGB en píxeles}^2 = 75976$$

$$\sqrt{75976} = 275,637443 \text{ Píxeles}$$

- 275,637443 píxeles (X)
- 275,637443 píxeles (Y)

Se utiliza la equivalencia 1 píxel (x) = 0.026458 cm y se calcula.

- 275,637443 píxeles (X) * 0.026458 cm = **7,29281547 cm** de lado horizontal (coordenada X), y así mismo para;
- 275,637443 píxeles (Y) * 0.026458 cm = **7,29281547 cm** de lado vertical (coordenada Y).

En consecuencia, se tiene 7,29281547 cm de cada lado en el área de 75976 píxeles² anteriormente mencionada, es decir tenemos 7,29281547 cm de defecto que se podría identificar como una etiqueta completa en la botella, Por último calculamos el área:

$$\text{Área} = \text{Lado (Base)} * \text{Lado (Altura)}$$

$$\text{Área} = 7,29281547 \text{ cm} * 7,29281547 \text{ cm} = \mathbf{53,185 \text{ cm}^2}$$

Por otro lado, los archivos .m del algoritmo implementado para el sistema de inspección y clasificación, se encuentra disponible como código fuente en la memoria de almacenamiento de este proyecto. A continuación se describen algunas funciones internas de MatLab utilizadas en el código (ver TABLA XV).

TABLA XV: FUNCIONES UTILIZADAS EN EL ALGORITMO DE INSPECCIÓN

Nombre	Descripción	Entrada	Salida
Strel	SE = strel (<i>shape</i> , parameters) Crea un elemento estructurante morfológico, del tipo especificado por shape (forma de línea, disco, octógono, etc.)	Tipo de elemento estructurante (shape) y parámetros necesarios para generarlo (parameters).	Elemento estructurante SE
Imopen	IM2 = imopen(IM,SE) Ejecuta un cierre morfológico sobre una imagen IM en escala de gris o binaria.	Imagen IM y elemento estructurante SE con el que se realiza la apertura.	Imagen IM con la apertura, puede ser cualquier clase numérico o lógico y cualquier dimensión
regionprops	Mide un conjunto de propiedades para cada región etiquetada en la matriz L	Matriz etiquetada (L) y lista de propiedades	Array con las medidas realizadas en cada región
Imdilate	Dilata o amplía la imagen de entrada utilizando el elemento estructurante SE	Imagen binaria o en escala de grises, elemento estructurante SE	Imagen dilatada
Imerode	Realiza la erosión de la imagen de entrada con el elemento estructurante SE.	Imagen binaria o en escala de grises, elemento estructurante SE	Imagen erosionada.
Imfill	BW2 = imfill(BW,'holes') Llena los agujeros en la imagen binaria BW. Un agujero es un conjunto de píxeles de fondo que no pueden ser alcanzadas por el llenado en el fondo del borde de la imagen.	La imagen de entrada BW puede ser numérico o lógico, y debe ser real y no dispersa. Puede tener cualquier dimensión.	La imagen de salida tiene la misma clase que la imagen de entrada.
imclearborder	IM2 = imclearborder (IM,conn) suprime las estructuras que son más delgadas que sus alrededores y que están conectadas al borde de la imagen. (En otras palabras, utilizar esta función para borrar el borde de la imagen.)	- IM puede ser una imagen en escala de grises o imagen binaria. - con especifica la conectividad deseada.	La imagen de salida, IM2, está en escala de grises o binario, respectivamente.
Imabsdiff	Z = imabsdiff (X, Y) Diferencia absoluta de dos imágenes es decir resta cada elemento de matriz Y de el elemento correspondiente de la matriz X y devuelve la diferencia absoluta en el elemento correspondiente de la matriz de salida Z.	X e Y son verdaderas matrices numéricas, no dispersa con la misma clase y tamaño.	Z tiene la misma clase y tamaño que X e Y.
imreconstruct	IM = imreconstruct (marcador, máscara) realiza la reconstrucción morfológica del	Marcador y la máscara pueden ser dos imágenes de	IM imagen devuelta es una imagen de



	<p>marcador de imagen bajo la máscara de la imagen.</p> <p>Marcador debe ser el mismo tamaño que la máscara, y sus elementos deben ser menor que o igual a los elementos correspondientes de la máscara.</p>	<p>intensidad o dos imágenes binarias con el mismo tamaño.</p>	<p>intensidad o binarios, respectivamente.</p>
imcomplement	<p>$M2 = \text{imcomplement}(IM)$ calcula el complemento de la imagen IM.</p> <p>En el complemento de una imagen binaria, los ceros se convierten en unos y otros se convierten en ceros, el negro y el blanco se invierten. En el complemento de una imagen RGB o intensidad, cada valor de píxel se resta del valor máximo de píxeles con el apoyo de la clase (o 1.0 para las imágenes de doble precisión) y la diferencia se utiliza como el valor de píxel en la imagen de salida. En la imagen de salida, las zonas oscuras se vuelven zonas más claras y la luz se oscurecen.</p>	<p>IM puede ser una imagen binaria, escala de grises o una imagen RGB.</p>	<p>IM2 tiene la misma clase y tamaño que el IM.</p>
fspecial	<p>$h = \text{fspecial}('gaussian', \text{hsize}, \text{sigma})$ crea un filtro h de dos dimensiones del tipo especificado. fspecial devuelve h como un núcleo de correlación, que es la forma apropiada para utilizar con imfilter.</p> <p>En este tipo devuelve un filtro de paso bajo gaussiano rotación simétrica de tamaño hsize con una desviación estándar sigma (positivo).</p>	<p>- 'gaussian': Filtro de paso bajo gaussiano - hsize puede ser un vector que especifica el número de filas y columnas en h, o puede ser un escalar, en cuyo caso h es una matriz cuadrada. - sigma desviación estándar (positivo).</p>	<p>h es de clase doble.</p>
imfilter	<p>$B = \text{imfilter}(A, h, 'symmetric', 'conv')$ filtra la matriz multidimensional A con el filtro multidimensional h.</p>	<p>- La matriz A puede ser lógico o una matriz numérica no dispersa de cualquier clase y dimensión. - symmetric: Valores de la matriz de entrada fuera de los límites de la matriz se calculan por el espejo de reflexión de la matriz a través de la frontera de matriz. - 'conv': imfilter realiza un filtrado multidimensional utilizando convolución.</p>	<p>El resultado B tiene el mismo tamaño y clase como A.</p>
mean	<p>$M = \text{mean}(A)$ devuelve los valores medios de los elementos a lo largo de las diferentes dimensiones de una matriz.</p>	<p>A: Imagen</p>	<p>M: Promedio</p>
var	<p>$V = \text{var}(X)$ devuelve la varianza de X para los vectores. Para matrices, var (X) es un vector fila que contiene la varianza de cada columna de X.</p>	<p>X: Imagen doblé en matriz</p>	<p>V: varianza, es el cuadrado de la desviación estándar</p>
bwperim	<p>$BW2 = \text{bwperim}(BW1, \text{conn})$</p>	<p>. BW1: Imagen que</p>	<p>Imagen binaria</p>

	<p>devuelve una imagen binaria que contiene solamente los píxeles del perímetro de los objetos en la imagen de entrada BW1.</p> <p>Un píxel es parte del perímetro si es distinto de cero y que está conectado a al menos un píxel de valor cero. La conectividad por defecto es 4 para dos dimensiones, 6 de tres dimensiones.</p>	<p>puede ser lógica o numérico . conn: Conectividad deseada.</p>	<p>que contiene solamente los píxeles del perímetro</p>
Imwrite	<p>imwrite(Imagen,filename,fmt) Graba la imagen en el archivo especificado por el nombre del archivo en el formato especificado por fmt.</p>	<p>Imagen: puede ser un M-por-N (escala de grises) o M-por-N-por-3 matriz (imagen color verdadero) Filename: es una cadena que especifica el nombre del archivo de salida.</p>	<p>Archivo de imagen guardado o grabado</p>

2.2.3. Identificación de los conocimientos tácticos

2.2.3.1. Análisis de conocimientos tácticos

Para el sistema de control Fuzzy de 4 entradas y 1 salida, las reglas pueden ser representadas por la matriz de la *TABLA XVI*, en donde se observa las etiquetas de las variables de entrada como: Desviación Estándar, Sigma, Área de fallas en RGB y Área de fallas en Bordes; y la etiqueta de la variable de salida denomina Defecto.



TABLA XVI: REGLAS POSIBLES PARA EL SISTEMA DE CONTROL FUZZY

N° Regla	Antecedente 1				Antecedente 2				Antecedente 3		Antecedente 4		Consecuente	
	Desviación Estándar				Sigma				AreaRGB		AreaBordes		Defecto	
	baja	media	mediaalta	alta	bajo	medio	medioalto	alto	poca	mucha	poca	mucha	No Existe (buena)	Existe (mala)
Dominio	[16.5 - 17.3]	[24.41- 28.6]	[27 - 32.01]	[31.33 - 77.92]	[190 - 197]	[370-467]	[421-520]	[530 - 3010]	[0-58]	[58-73e+003]	[0-58]	[58-73e+003]	1	0
1	X				X				X		X			X
2	X				X				X			X		X
3	X				X					X				X
4	X				X					X		X		X
5	X					X			X		X			X
6	X					X			X			X		X
7	X					X				X	X			X
8	X					X				X		X		X
9	X						X		X		X			X
10	X						X		X			X		X
11	X						X			X	X			X
12	X						X			X		X		X
13	X							X	X		X			X
14	X							X	X			X		X
15	X							X		X	X			X
16	X							X		X		X		X
17		X			X				X		X			X
18		X			X				X			X		X
19		X			X					X	X			X
20		X			X					X		X		X

RESULTADOS

21		X				X			X		X		X	
22		X				X			X			X		X
23		X				X				X	X			X
24		X				X				X		X		X
25		X					X		X		X			X
26		X					X		X			X		X
27		X					X			X	X			X
28		X					X			X		X		X
29		X						X	X			X		X
30		X						X	X			X		X
31		X						X		X	X			X
32		X						X		X		X		X
33			X		X				X		X			X
34			X		X				X			X		X
35			X		X					X	X			X
36			X		X					X		X		X
37			X			X			X		X			X
38			X			X			X			X		X
39			X			X				X	X			X
40			X			X				X		X		X
41			X				X		X			X		X
42			X				X		X			X		X
43			X				X			X	X			X
44			X				X			X		X		X
45			X					X	X			X		X
46			X					X	X			X		X
47			X					X		X	X			X
48			X					X		X		X		X



RESULTADOS

49			X	X				X		X			X
50			X	X				X			X		X
51			X	X					X	X			X
52			X	X					X		X		X
53			X		X			X		X			X
54			X		X			X			X		X
55			X		X				X	X			X
56			X		X				X		X		X
57			X			X		X		X			X
58			X			X		X			X		X
59			X			X			X	X			X
60			X			X			X		X		X
61			X				X	X		X			X
62			X				X	X			X		X
63			X				X		X	X			X
64			X				X		X		X		X

De acuerdo con la tabla anterior, para el sistema en total existen 64 reglas posibles, entonces se puede deducir con su debida sintaxis 4 de ellas, en la *TABLA XVII*.

TABLA XVII: REGLAS LINGÜÍSTICAS

No Regla	Términos lingüísticos
REGLA 1	IF la Desviación Estándar es baja AND el sigma es bajo AND el Área en RGB es poca AND Área en bordes es poca, THEN Defecto Existe
REGLA 17	IF la Desviación Estándar es media AND el sigma es bajo AND el Área en RGB es poca AND Área en bordes es poca, THEN Defecto Existe
REGLA 21	IF la Desviación Estándar es media AND el sigma es medio AND el Área en RGB es poca AND Área en bordes es poca, THEN Defecto no Existe
REGLA 28	IF la Desviación Estándar es media AND el sigma es medioalto AND el Área en RGB es mucha AND Área en bordes es mucha, THEN Defecto Existe

Hasta aquí se ha evaluado cada una de las reglas, el próximo paso es determinar la salida Fuzzy comparando las fuerzas de todas las reglas que especifican el mismo consecuente, es decir la misma acción de salida (*ver TABLA XVIII*).

- El defecto existe con un grado de verdad de 0
- El defecto no existe con un grado de verdad de 1

TABLA XVIII: REGLAS LINGÜÍSTICAS CON SALIDA FUZZY

No Regla	Términos lingüísticos	Salida
REGLA 1	IF la Desviación Estándar es baja AND el sigma es bajo AND el Área en RGB es poca AND Área en bordes es poca, THEN Defecto Existe	0
REGLA 17	IF la Desviación Estándar es media AND el sigma es bajo AND el Área en RGB es poca AND Área en bordes es poca, THEN Defecto Existe	0
REGLA 21	IF la Desviación Estándar es media AND el sigma es medio AND el Área en RGB es poca AND Área en bordes es poca, THEN Defecto no Existe	1
REGLA 28	IF la Desviación Estándar es media AND el sigma es medioalto AND el Área en RGB es mucha AND Área en bordes es mucha, THEN Defecto Existe	0

2.2.3.2. Comprobación de los conocimientos tácticos

Se ha validado y revisado todas las reglas, además se ha comprobado que el accionar experto y el desarrollador coinciden con las representaciones de los conocimientos tácticos. En tal sentido, esto permitió ajustar las reglas de manera idónea.

Etapa 2.3 Formalización de los conocimientos y definición de la arquitectura

2.3.1. Motor de inferencia y base de conocimientos

- El modelo fuzzy o difuso utilizado es de la forma conocida como Takagi Sugeno
- Reglas de inferencia de tipo IF/THEN
- Método de optimización **hybrid**.

TABLA XIX: BASE DE CONOCIMIENTOS

Variables	Valores	Descripción
Desviación Estándar	Baja, media, mediaalta, alta	Valor estadístico de una imagen, es una medida de dispersión, que indica cuánto pueden alejarse los valores respecto al promedio (media), si los píxeles son claros la desviación estándar es baja, en cambio si los píxeles son oscuros la desviación es alta.
Sigma	Bajo, medio, medioalto, alto	Valor estadístico que indica el grado de variación en un conjunto de medidas. Si el valor de sigma es alto, se puede decir que existe mucha variabilidad en la imagen, pero si el valor de sigma es pequeño o bajo, con toda seguridad se puede decir que la imagen tiene poca variabilidad y por consiguiente es más uniforme.
AreaRGB	Poca, mucha	Valor numérico que denota el área defectuosa en píxeles de la imagen de la botella, extraída por la detección en colores RGB
AreaBordes	Poca, mucha	Valor numérico que denota el área defectuosa en píxeles de la imagen de la botella, extraída por la detección de bordes.

Los parámetros del sistema de inferencia difuso obtenidos son los siguientes:

TABLA XX: PARÁMETROS DEL SISTEMA DE INFERENCIA DIFUSO

Nombre	'inspeccionRedNeuroDifusa'		
Tipo sistema de inferencia	Sugeno		
Variables de entrada	Variable	Rango	Tipo de FP
	Desviación	[16.75 - 70.83]	trapmf
	Sigma	[192.51 - 3002.30]	trapmf
	AreaRGB	[0 - 70742]	trapmf
AreaBordes	[0 - 29092]	trapmf	
Variables de salida	Defecto	[0 1]	Constante
Nº FP variables de entrada	[4 4 2 2]		
Nº FP variables de salida	64		
Número de Reglas	64		
Método Y (andMethod)	'min' (mínimo)		
Método O (orMethod)	'probor' (Máximo)		
Método Defusificación defuzzMethod	'wtaver'		
Método de Implicancia impMethod	'prod'		
Método de agregación(aggMethod)	'sum'		

2.3.2. Codificación de Reglas y desarrollo del controlador difuso

2.3.2.1. Preparación de Datos con la ayuda del software MatLab

A los 38 vectores que se utilizan para el entrenamiento del modelo Anfis (*modelo difuso de este proyecto*), se los conoce como patrones (entradas/salidas) del modelo Anfis, los vectores de entrada son los valores extraídos del algoritmo anteriormente especificados, en cambio, a los vectores de salidas se les asignó un valor según su defecto (*ver TABLA XVIII*), partiendo de una botella ideal no defectuosa (buena) con valor 1 y la defectuosa (mala) con 0, razón por la cual se los conoce como rango del vector salida.

En la *TABLA XXI* se muestra los 38 vectores encontrados según el estado de la botella.

TABLA XXI: VECTORES PARA EL ENTRENAMIENTO

Nº	Descripción	Vector de Entrada [Desviación est., sigma, areaRGB, areaBordes]	Vector Salida
1	Limpia1	[26.235668718692285, 3.835837646328803e+02, 0, 22]	1
2	Limpia2	[25.901730530251110, 3.712607804393007e+02, 0, 9]	1
3	Limpia3	[26.883302884235402, 4.034879809151485e+02, 0, 3]	1
4	Limpia4	[26.115436227761847, 3.854168479394065e+02, 0, 16]	1
5	Limpia5	[25.946697141422693, 3.85245044492068e+02, 40, 0],	1
6	Limpia6	[26.072934038888786, 3.93162854417522e+02, 31, 4]	1
7	Limpia7	[25.871312772694490, 3.92497831280245e+02, 9, 24]	1
8	Limpia8	[25.706771253025580, 3.646382436976625e+02, 8, 0]	1
9	Limpia9	[26.647450083056746, 4.049062845885590e+02, 14, 0]	1
10	Limpia10	[26.648889117017678, 4.155344150227687e+02, 0, 0]	1
11	Pico malo1 (anillo)	[28.235716299977630, 4.34049881149e+02, 1202, 385]	0
12	Pico malo2 (anillo)	[28.257282563866560, 4.28390481379e+02, 1208, 715]	0
13	Pico malo3 (anillo y plástico)	[27.294417863031192, 4.21819334963e+02, 1010, 456]	0
14	Pico malo4 (anillo y plástico)	[27.193935021168148, 4.18081137004541e+02, 946, 2]	0
15	Pico malo5 (anillo y colilla)	[30.848135838451450, 5.10332615677e+02, 1975, 249]	0
16	Pico malo6 (anillo y colilla)	[31.211005382206828, 5.015824119318e+02, 2939, 59]	0
17	Pico malo7 (sorbete)	[30.642935115397425, 4.6816567575e+02, 2001, 1956]	0



18	Pico malo8 (sorbete)	[30.467420738326600, 4.8565203519e+02, 1581, 1923]	0
19	Mala 1 (etiqueta 10%)	[32.737000604212260, 5.724923892748e+02, 303, 2906]	0
20	Mala 2 (etiqueta 10%)	[31.849447246168160, 5.3468386717227e+02, 10, 2624]	0
21	Mala 3 (etiqueta 25%, anillo y colilla)	[33.027459337425460, 5.52035627382e+02, 2490, 2536]	0
22	Mala 4 (etiqueta 25%, anillo y colilla)	[31.897678406944564, 5.50293545934e+02, 1811, 5593]	0
23	Mala 5 (etiqueta 25%, plástico y colilla)	[32.228882102317850, 5.45561038683e+02, 2285, 3205]	0
24	Mala 6 (etiqueta 25%, plástico y colilla)	[31.536188350758420, 5.42387676554e+02, 1496, 2970]	0
25	Mala 7 (etiqueta 40% y colilla)	[35.731748123946230, 6.14517877777e+02, 3390, 5100]	0
26	Mala 8 (etiqueta 40% y colilla)	[32.236046188563100, 5.453144822217e+02, 221, 5098]	0
27	Mala 9 (anillo y etiqueta 50%)	[41.060766727676380, 8.03472485822e+02, 8021, 6860]	0
28	Mala 10 (anillo y etiqueta 50%)	[33.940052794234376, 6.27172173997e+02, 1043, 8441]	0
29	Mala 11 (anillo y etiqueta 60%)	[37.441696256151710, 6.36903059709e+02, 5757, 5726]	0
30	Mala 12 (anillo y etiqueta 60%)	[35.137166913774735, 6.41986685456e+02, 1308, 6691]	0
31	Mala 13 (anillo y etiqueta 90%)	[50.898406103789200, 1.557794142e+03, 15773, 13560]	0
32	Mala 14 (anillo y etiqueta 90%)	[43.605480257527500, 1.2514364385e+03, 1111, 23767]	0
33	Mala 15 (etiqueta 95%)	[52.643252055500476, 1.653053043e+03, 21104, 28092]	0
34	Mala 16 (etiqueta 95%)	[43.998716911732610, 1.190849752536732e+03, 79, 95]	0
35	Mala 17 (botella completa)	[70.547970737929330, 2.9777780062e+03, 63644, 9377]	0
36	Mala 18 (botella completa)	[68.177067557606250, 2.7502251329e+03, 70742, 2600]	0
37	No existe botella 1	[16.967621992593040, 1.927613793983196e+02, 0, 0]	0
38	No existe botella 2	[17.149359825784163, 1.925145196803449e+02, 0, 0]	0

Para la validación del modelo Fuzzy, se toman 38 entradas con sus respectivas salidas Anfis, estos nuevos vectores son diferentes a los que se utilizaron para el entrenamiento y que se las puede apreciar en la *TABLA XXII*.

Estas 38 entradas se tomaron al azar de otros datos extraídos, el objetivo es de validar el modelo Anfis. El modelo Anfis compara estos nuevos vectores con los 38 ya entrenados, los clasifica y tiene la capacidad de reconocerlos.

TABLA XXII: VECTORES PARA LA VALIDACIÓN.

Nº	Descripción	Vector de Entrada [Desviación est., sigma, areaRGB, areaBordes]	Vector Salida
1	Limpia1	[26.403773433671660, 3.94285217049310e+02, 17, 27]	1
2	Limpia2	[25.744796370836927, 3.73729714260096e+02, 0, 8]	1
3	Limpia3	[26.883302884235402, 4.09720927494677e+02, 0, 6]	1
4	Limpia4	[25.869245425236677, 3.90669396136534e+02, 0, 0]	1
5	Limpia5	[26.485133360291540, 4.01781843007476e+02, 58, 33]	1
6	Limpia6	[25.895712048522880, 3.95906127814924e+02, 29, 4]	1
7	Limpia7	[26.147819365838604, 3.97318374413720e+02, 19, 42]	1
8	Limpia8	[24.814523785508147, 3.63857033852423e+02, 2, 0]	1
9	Limpia9	[26.732947500929928, 4.15611022088665e+02, 18, 20]	1
10	Limpia10	[26.514324543009185, 4.11781891968897e+02, 0, 0]	1
11	Pico malo1 (anillo)	[28.282853787630916, 4.404504323330e+02, 1228, 391]	0
12	Pico malo2 (anillo)	[28.078368079308820, 4.387601246077e+02, 1212, 380]	0
13	Pico malo3 (anillo y plástico)	[27.342663601404666, 4.279144738442e+02, 1033, 456]	0
14	Pico malo4 (anillo y plástico)	[27.097190160638530, 4.20662524269355e+02, 921, 2]	0
15	Pico malo5 (anillo y colilla)	[31.103185326087146, 5.17798264750e+02, 2196, 4880]	0
16	Pico malo6 (anillo y colilla)	[31.211005382206828, 5.0959331717002e+02, 2939, 80]	0
17	Pico malo7 (sorbete)	[30.768936238163295, 4.77784125409e+02, 2002, 2086]	0
18	Pico malo8 (sorbete)	[30.467420738326600, 4.92951953640e+02, 1581, 1923]	0
19	Mala 1 (etiqueta 10%)	[32.773447829375584, 5.810615418370e+02, 303, 2946]	0
20	Mala 2 (etiqueta 10%)	[31.510919293995734, 5.40325247835353e+02, 1, 2607]	0
21	Mala 3 (etiqueta 25%, anillo y colilla)	[33.143918104411500, 5.59398273165e+02, 2625, 4800]	0
22	Mala 4 (etiqueta 25%, anillo y colilla)	[31.595927146261012, 5.55556408766e+02, 1502, 5370]	0
23	Mala 5 (etiqueta 25%, plástico y colilla)	[32.813095919416430, 5.70590918502e+02, 2807, 5604]	0

24	Mala 6 (etiqueta 25%, plástico y colilla)	[31.626267664483034, 5.51706456453e+02, 1422, 2238]	0
25	Mala 7 (etiqueta 40% y colilla)	[36.049623292559760, 6.27845279653e+02, 3482, 5100]	0
26	Mala 8 (etiqueta 40% y colilla)	[32.054984383104640, 5.508980724681e+02, 221, 3975]	0
27	Mala 9 (anillo y etiqueta 50%)	[41.280514297124800, 8.24796662124e+02, 8028, 9107]	0
28	Mala 10 (anillo y etiqueta 50%)	[33.830855689760840, 6.369514433317e+02, 975, 8184]	0
29	Mala 11 (anillo y etiqueta 60%)	[37.596718116159600, 6.54007939607e+02, 5995, 6922]	0
30	Mala 12 (anillo y etiqueta 60%)	[35.079604808823170, 6.53658111903e+02, 1244, 6577]	0
31	Mala 13 (anillo y etiqueta 90%)	[51.104207492104810, 1.582026421e+03, 15956, 29092]	0
32	Mala 14 (anillo y etiqueta 90%)	[43.505617648361180, 1.2669979708e+03, 1033, 23637]	0
33	Mala 15 (etiqueta 95%)	[52.748812261168640, 1.676664450e+03, 21727, 28304]	0
34	Mala 16 (etiqueta 95%)	[43.915508810028830, 1.20967194151913e+03, 20, 95]	0
35	Mala 17 (botella completa)	[70.833265446401230, 3.0023062029e+03, 64776, 9598]	0
36	Mala 18 (botella completa)	[68.030837200295960, 2.7743604809e+03, 69978, 2577]	0
37	No existe botella 1	[16.774467336145830, 1.96336324354101e+02, 0, 0]	0
38	No existe botella 2	[16.759920803878640, 1.93481371666796e+02, 0, 0]	0

2.3.2.2. Construcción del sistema con Fuzzy Logic Toolbox: Aprendizaje e inferencia través de ANFIS

La idea básica detrás de estas técnicas de adaptación neuro-aprendizaje es muy sencilla.

Estas técnicas proporcionan un método para el procedimiento de modelado difuso para aprender información sobre un conjunto de datos, con el fin de calcular los parámetros de la función de pertenencia que mejor permita al sistema asociado de inferencia difusa, realizar un seguimiento de la entrada / salida de datos dadas. Este método de aprendizaje funciona de manera similar a la de las redes neuronales. La función de Fuzzy Logic Toolbox que realiza esta función de pertenencia de ajuste de parámetros se llama ANFIS. ANFIS se puede acceder ya sea desde la línea de comandos, o a través de la interfaz gráfica de usuario ANFIS Editor.

En el programa MatLab R2009a, en el Command Window escribir el comando **anfisedit** y luego oprimir *enter* como se muestra en la *Figura 171*.

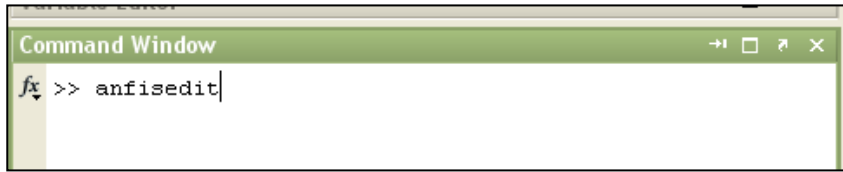


Figura 171: Command Window de MatLab

Seguidamente aparecerá la ventana del Anfis Editor (*ver Figura 172*).

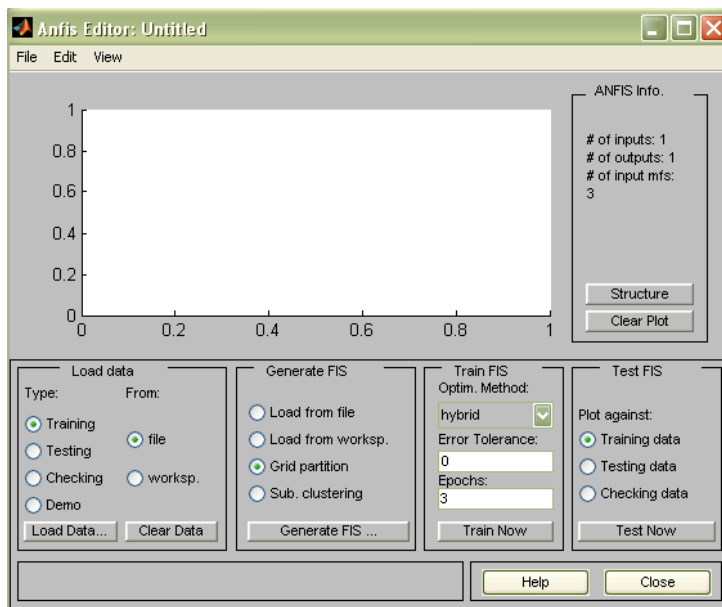


Figura 172: Ventana de Anfis Editor

- **Cargando datos de entrenamiento y validación del sistema Neurofuzzy**

El entrenamiento del sistema Neurofuzzy se realiza con los 38 vectores de la *TABLA XXI*, para que el sistema aprenda a clasificar a las entradas, esto lo hace a través del reconocimiento de patrones o clasificación, para esto se le brinda información digitalizada de cada botella, este archivo *.m* desarrollado en MatLab se encuentra en Anexo 5.

Para acceder a los datos se ejecuta el archivo *.m* abierto desde MatLab, estos datos se muestran como variables en el Workspace (espacio de trabajo) (*ver Figura 173*).

Name	Value	Min	Max
chk_data	<38x5 double>	0	69978
trn_data1	<38x5 double>	0	70742
trn_data2	<38x5 double>	0	72762
trn_data3	<38x5 double>	0	72762

Figura 173: Datos cargados en el Workspace de MatLab

En la ventana de la *Figura 172*, para cargar el conjunto de datos de entrenamiento ubicarse en el panel *Load data* y seleccionar **Training, worksp.** luego click en el botón *Load Data....*, una pequeña ventana aparece y permite escribir el nombre de la variable de entrada **trn_data1** que se encuentra en el *workspace*, como se muestra a continuación (ver *Figura 174*).

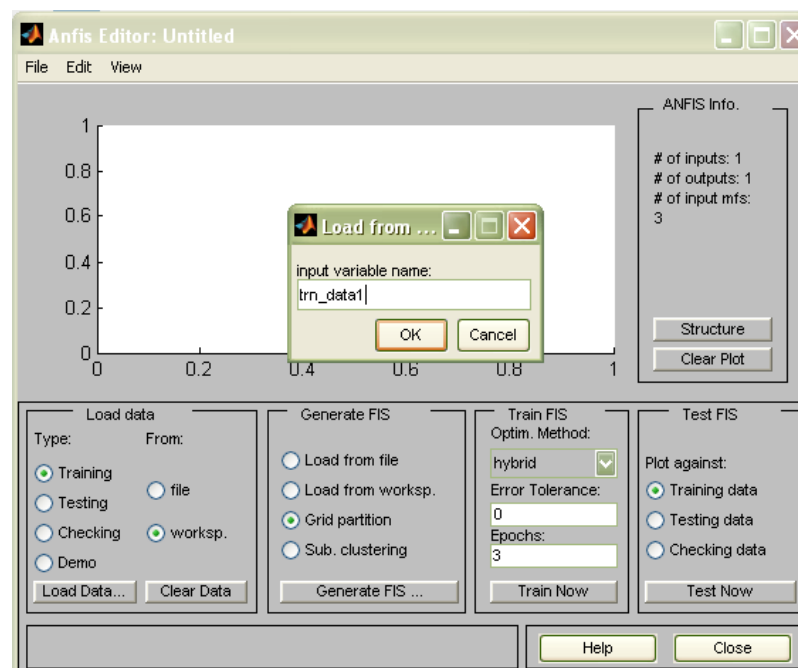


Figura 174: Cargar datos de entrenamiento en Anfis

Los datos de entrenamiento aparecen en la gráfica como un conjunto de círculos, en la *Figura 175* se puede ilustrar esta representación.

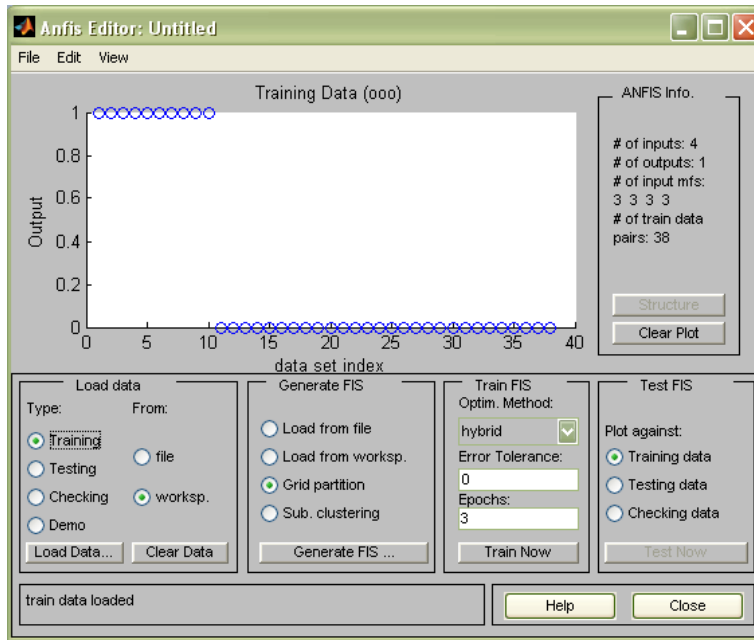


Figura 175: Datos de entrenamiento cargados (conjunto de círculos).

Para cargar los datos validación en el panel *Load data* de la Figura 175 seleccionar **Checking, worksp.**, y así cargar la variable **chk_data** desde el *workspace*. Estos datos aparecen en la gráfica como (+)'s superpuestos a los datos de entrenamiento. Los datos de *chk_data* también se encuentran en el archivo *.m* que se encuentra en el Anexo 6.

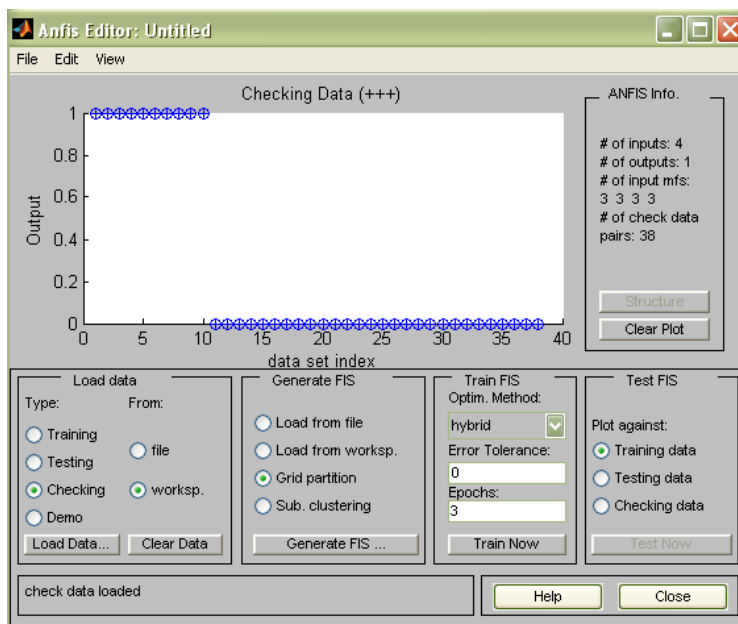


Figura 176: Datos de validación cargados (Conjunto de +'s)

Este conjunto de datos se utilizará para entrenar el sistema difuso mediante el ajuste de los parámetros de la función de pertenencia como el mejor modelo de estos datos. El siguiente paso consiste en especificar un sistema inicial de inferencia difuso (FIS) para ANFIS a entrenar.

- **Inicializar y generar el FIS (Sistema de Inferencia Difuso)**

En esta parte se puede escoger la forma de inicializar los parámetros del FIS, una es utilizando los parámetros de preferencia propia, ósea a nuestra base de conocimientos, y si no se tiene ninguna preferencia de cómo parametrizar las funciones de pertenencia iniciales, ANFIS puede hacerlo automáticamente.

Como ya se analizaron los vectores de entrada y los rangos establecidos, se utilizaron en cierto grado las dos formas de generar el FIS:

En la ventana de la *Figura 176* en el panel *Generate FIS*, se escoge **Grid partition**, y click en botón **Generate FIS**, se muestra un menú desde el que puede elegir el número de funciones de pertenencia MFS, y el tipo de entrada y salida de las mismas (ver *Figura 177*). Tener en cuenta que sólo hay dos opciones para la función de pertenencia de salida: constante y lineal. Esta limitación de las opciones de salida de la función de pertenencia se debe a que ANFIS sólo funciona en los sistemas de tipo Sugeno. Finalmente click en **OK** de la *Figura 177*.

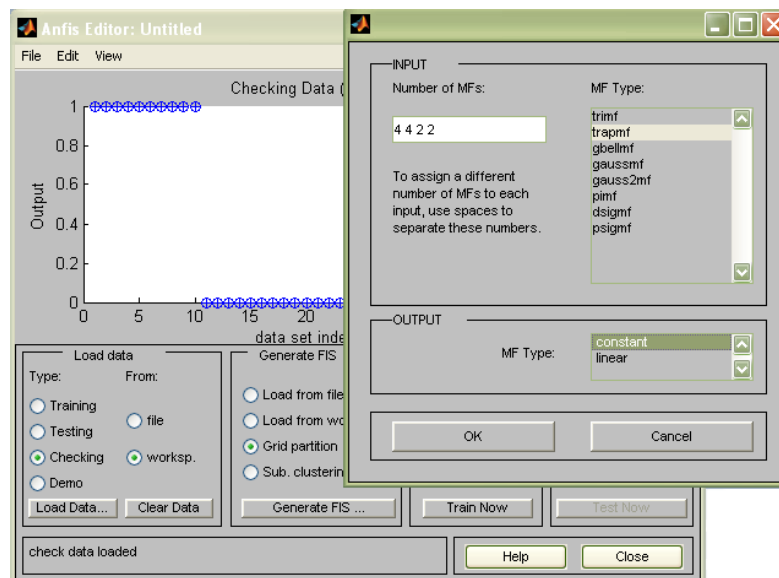


Figura 177: Ingreso y tipo de funciones de pertenencia.

El paso anterior genera automáticamente el diagrama de tipo Sugeno, las variables de entrada y salida con su respectivo nombre, rango, tipo de curva y parámetros, las gráficas de las funciones de pertenencia y la combinación de reglas.

Haciendo click en el menú **Edit, FIS Properties** de la *Figura 176*, se modifica las propiedades de acuerdo a la *TABLA XX*.

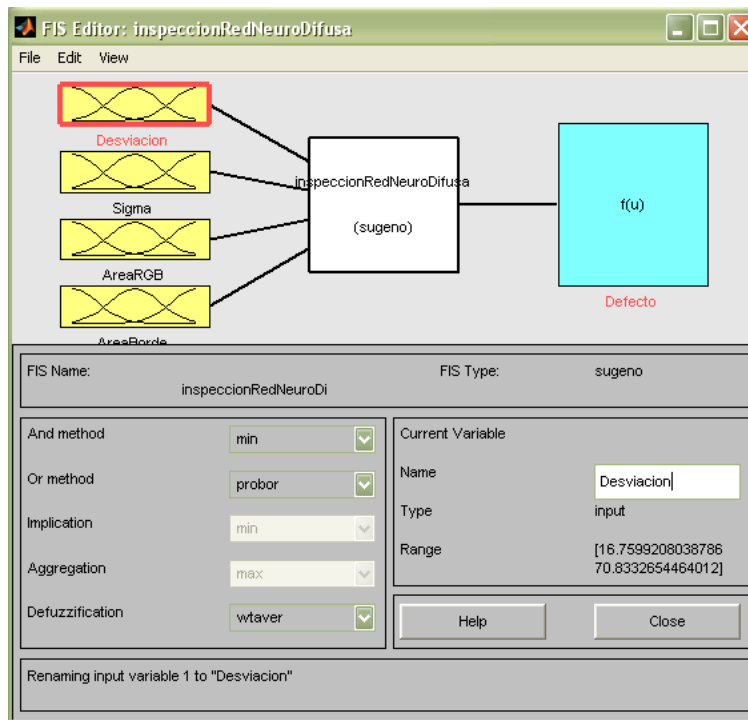


Figura 178: Definición de las variables en la ventana FIS Editor

La ventana de edición del FIS (ver *Figura 178*) muestra las variables de entrada y salida así como el método de inferencia que se elige, el cual corresponde al método Sugeno. En la misma ventana en la parte izquierda se puede modificar los métodos de los operadores lógicos *AND* y *OR*, los métodos de implicación, de agregación y de defuzzificación.

En la parte derecha, se puede cambiar el nombre de la variable que se encuentre seleccionada, por ejemplo, modificar el nombre “input1” por “Desviacion”.

El segundo paso es definir los conjuntos difusos de las variables de entrada y salida.

La *Figura 179* muestra la ventana donde se definen los conjuntos difusos de las variables.

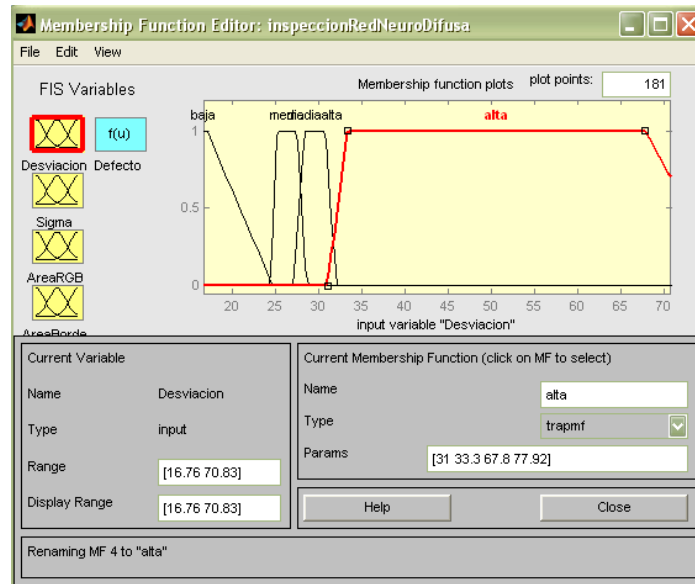


Figura 179: Editor de Funciones de Pertenencia, Membership Editor

En la parte izquierda, se puede modificar el rango de la función de pertenencia, en el cual estará delimitada. En la parte derecha, es posible modificar el nombre de la función de pertenencia (etiquetas lingüísticas), los parámetros y también su forma, la cual puede ser triangular, trapezoidal, gaussiana, etc.

Por último es necesario definir las reglas que harán funcionar al sistema, para poder modificar las reglas del modelo se debe acceder al *Rule Editor*, haciendo doble click sobre el modelo (ver Figura 180).

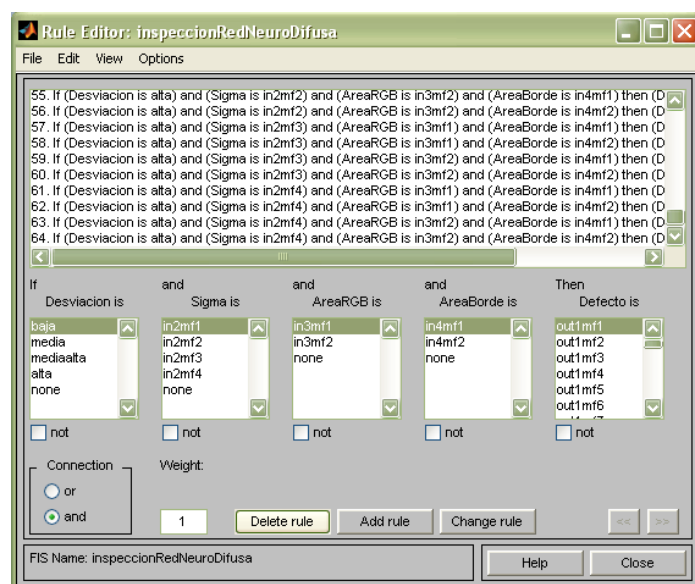


Figura 180: Ventana Editor de Reglas, Rule Editor

En la ventana de edición de las reglas se presentan las variables con los conjuntos difusos que se definieron anteriormente. Ayudándose de la *TABLA XVI* de inferencia de la base de conocimientos, se selecciona un conjunto de cada variable junto a las condiciones que los relacionan.

Una vez finalizada la configuración del controlador es posible observar las reglas que se generaron en el editor de reglas y como se activan cuando se ingresan valores de entrada. En la *Figura 181* se muestra la ventana de las reglas eligiendo **View, Rules** de la *Figura 180*.

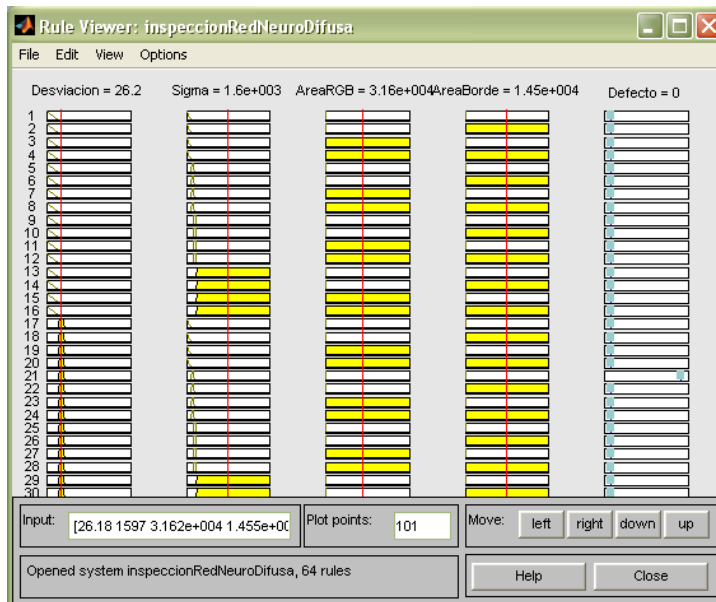


Figura 181: Ventana Rule Viewer

En la ventana se observan las variables de entrada así como los conjuntos que las conforman en color amarillo. Las líneas rojas representan los valores de entrada que se ingresan al controlador y se pueden observar los grados de membrecía a los que corresponden dentro del conjunto difuso. La variable de salida con sus conjuntos difusos en color azul muestran las reglas que se cumplen y en la parte inferior de la columna se puede apreciar la línea generado por la inferencia. El valor de salida que produce la defusificación se puede observar en la parte superior de la columna.

Para poder implementar el controlador es necesario guardar el trabajo realizado, con el menú **File, export > to file...** Luego, importar el archivo al *workspace*, para que MatLab lo pueda reconocer y pueda ser implementado.

Para regresar al entrenamiento Anfis, clic en **Edit, Anfis** de la *Figura 181*. Ahora bien, para cargar el FIS; en la ventana de la *Figura 182* en el panel **Generate FIS**, clic en **Load from worksp.** o **Load from file**, esto depende donde se haya exportado y luego click en el botón **Load...**

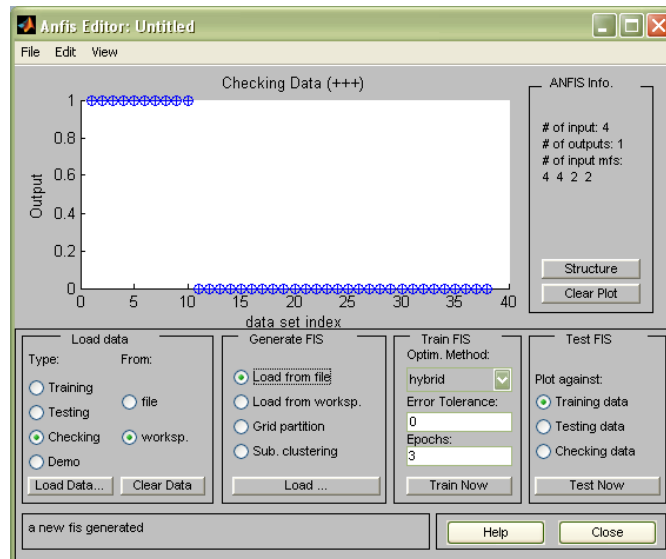


Figura 182: Cargar el FIS

Para visualizar el modelo de la estructura FIS, en el panel **ANFIS Info.** clic en el botón **Structure**, Las ramas de este gráfico nodal están codificadas por colores, para indicar si se utilizan o no en las reglas los *and*, *not* u *or*. Haciendo clic en los nodos indica información acerca de la estructura (ver *Figura 183*).

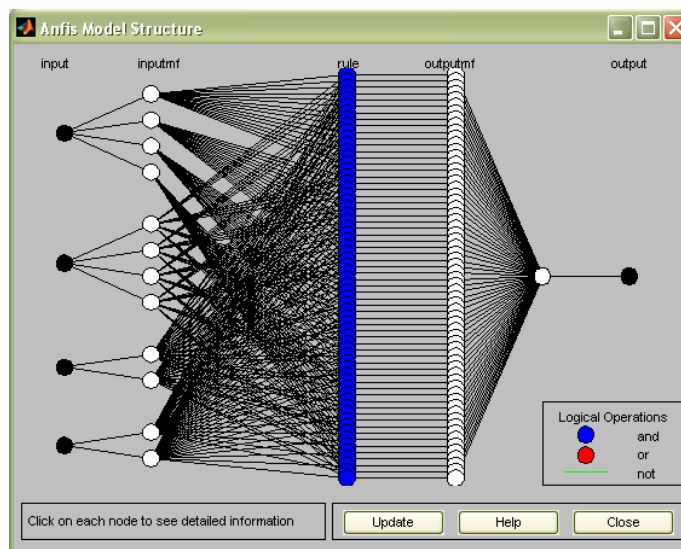


Figura 183: Estructura FIS

2.3.2.2.1. Entrenando Anfis

Los dos opciones de parámetros ANFIS de optimización de métodos disponibles para la formación del FIS son **hybrid** (por defecto, es una mezcla de mínimos cuadrados y retropropagación) y **backpropa** (backpropagation). La tolerancia de error se utiliza para crear un criterio de formación parada, que está relacionada con el tamaño de error. La formación se detendrá después de que el error en los datos de entrenamiento permanezca dentro de la tolerancia. Es mejor dejar a 0 si no se sabe cómo se va a comportar su error en el entrenamiento.

Para empezar a entrenar, en el panel **Train FIS** de la *Figura 184* realizar lo siguiente:

- Selección el método de optimización **hybrid**.
- Establecer el número de épocas de entrenamiento a 40,
- Seleccione **Train now**.

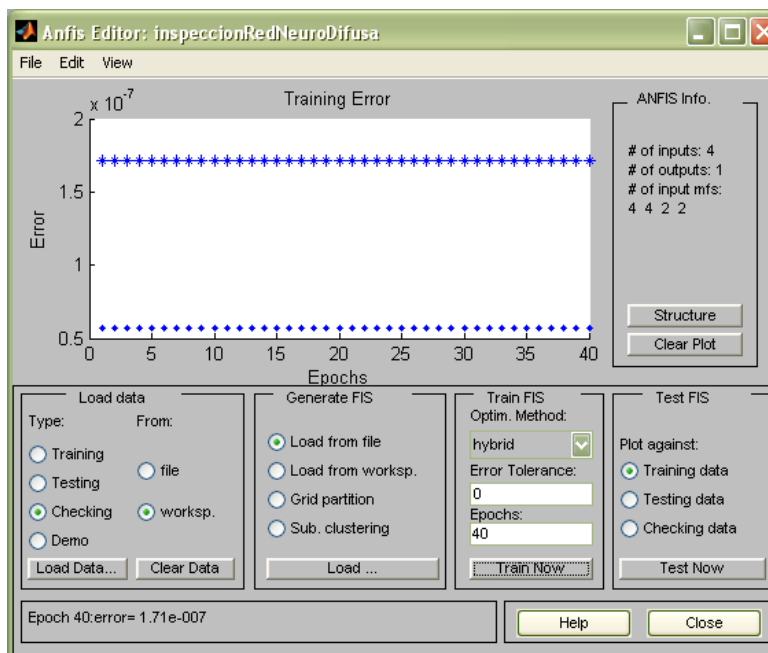


Figura 184: Entrenando Anfis

Comportamiento del error

En la *Figura 184*, la gráfica de la parte superior indica el error de entrenamiento y en la inferior, el error de verificación.

En este momento ANFIS elige los parámetros del modelo relacionados con la comprobación de errores mínima. Es aquí donde son útiles los datos de validación ANFIS, los cuales ya fueron cargados anteriormente. Nótese que el error es mínimo $1.71e-007$ (0.000000171), si de lo contrario fuese grande esto indicaría que este conjunto de funciones de pertenencia escogidas no sería la mejor opción para el modelado de los datos de entrenamiento. Y además que el conjunto de datos de validación no son sumamente diferentes a los datos de entrenamiento, esto radica en que se debe extraer o realizar un buen conjunto de datos que refleje a una buena información.

Prueba de los datos con el FIS entrenado

Para probar o comprobar el FIS con los datos de validación, hacer clic en **Checking data** en el panel **Test FIS** de la ventana *Figura 184*, y hacer clic en **Test Now**. Ahora, se puede comprobar los datos de validación contra la salida del FIS y se muestra bastante bien.

La siguiente gráfica indica que hay una pequeña discrepancia ($5.6999e-008$) entre la salida de la comprobación de datos y la salida del FIS.

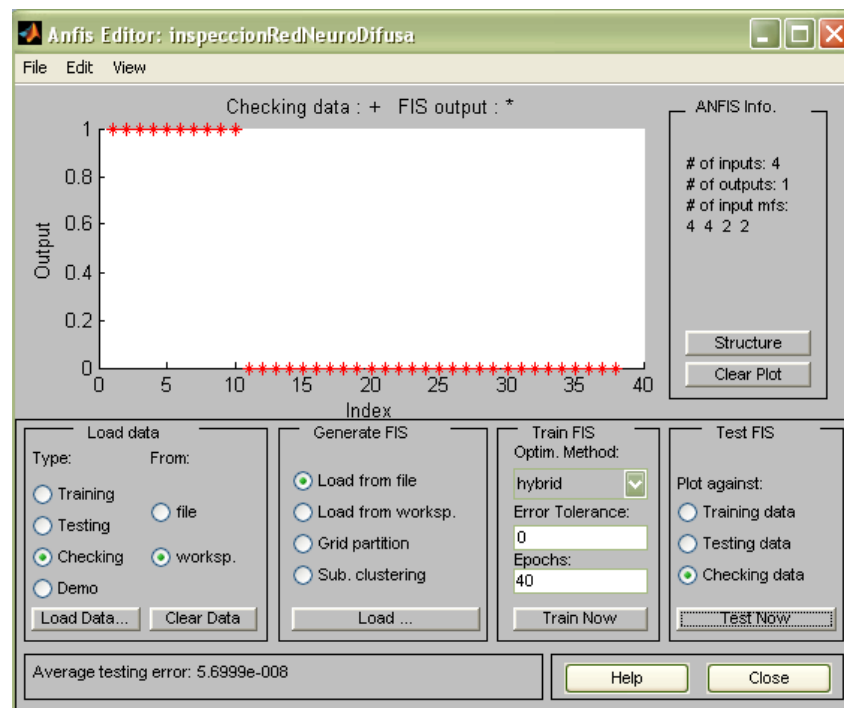


Figura 185: Prueba de datos de validación

Con esto termina la construcción del modelo ANFIS y se procede a realizar pruebas con otras botellas.

Para poder evaluar este controlador FIS e incorporarlo al algoritmo se lo exporta a un archivo *.fis*, tal como se muestra en la *Figura 186*. Clic en el menú **File**, luego **Export> To File**, y elegimos el nombre y el directorio donde guardarlo.

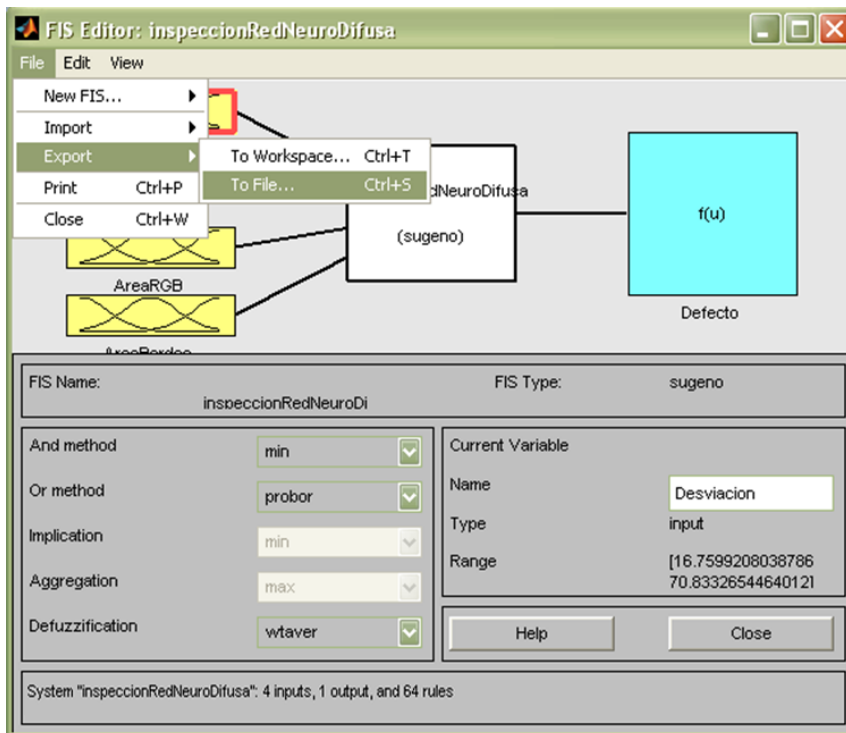


Figura 186: Exportar el Controlador FIS

2.3.2.2.2. Evaluación del FIS

La función **evalfis** de MatLab, permite evaluar las entradas del vector de datos que corresponden a las características extraídas de la imagen de las botellas. Esta evaluación se incluye al final en el algoritmo de MatLab para calcular el resultado de clasificación en tiempo real.

```
datos_extraidos=[desviacion sigmaB2max area_ImagenConFallasRGB
areaImagenConBordes];
red=readfis('inspeccionRedNeuroDifusa.fis');
RESULTADO = evalfis(datos_extraidos,red);
```


Este resultado se obtiene desde MatLab por medio de la conexión que se hizo con JAVA, esta conexión se especifica en la FASE 3, La interfaz gráfica en Java se encargara de visualizar de forma gráfica y sonora este resultado.

Etapa 2.4 Selección de la Herramienta e implementación

El sistema de control basado en conocimiento está desarrollado en dos partes, por una lado el desarrollo del sistema global y por otra parte a la del proceso de control en sí, a cual está destinado.

Para el desarrollo del sistema global se utiliza el lenguaje de programación JAVA en la plataforma NetBeans y se comenzó a la implementación del prototipo como lo establece la metodología IDEAL, lo cual implica que para haber logrado el prototipo, se desarrollan una serie de prototipos hasta lograr la satisfacción de los requisitos determinados en el análisis.

En el presente proyecto se empleó una gran cantidad de tiempo en la implementación de la interfaz de usuario ya que se debía mantener una interfaz amigable que mantenga el proceso y se puedan apreciar los resultados de forma visual y sonora.

El proceso de control de inspección se desarrolló en MatLab con ayuda de sus Toolboxes, para conseguir este objetivo, se estudió la herramienta Image Processing Toolbox para el procesamiento digital de imágenes dando paso al desarrollo del algoritmo; la programación y configuración del controlador se realizó utilizando Fuzzy Logic Toolbox.

Etapa 2.5: Validación y evaluación del prototipo

El desarrollo de un sistema basado en conocimiento no se considera que está acabado una vez que funciona este, sino que se continúa desarrollando y actualizando tanto el conocimiento del sistema como los métodos de procesamiento, quedando reflejados los progresos o modificaciones en el campo, área o sistema.

4. FASE 3: EJECUCIÓN DE LA CONSTRUCCIÓN DEL SISTEMA INTEGRADO

Etapa 3.1: Requisitos y diseño de la integración con otros sistemas

En el presente proyecto el Sistema de control está integrado e interactúa con los lenguajes de programación en cual fue programado y configurado.

Utilizando equipos de cómputo a nivel mono-usuario, el sistema de control de inspección cuenta con un equipo básico de cómputo consistente en:

TABLA XXIII: CARACTERÍSTICAS MÍNIMAS DE HARDWARE

CARACTERÍSTICAS MÍNIMAS DE HARDWARE	
Procesador	Pentium(R) Dual Core 2.10 GHz
Memoria Ram	4 Gb
Almacenamiento	Disco de 120 GB
Periféricos	Lectora de DVD/CD, puerto USB, Cámara Web, teclado y mouse.

Para el correcto funcionamiento del sistema se puede contar con los siguientes programas instalados:

TABLA XXIV: SOFTWARE INSTALADO

SOFTWARE INSTALADO	
Sistema Operativo	Windows XP
Programas	<ul style="list-style-type: none"> - MatLab R2009a <ul style="list-style-type: none"> ✓ MATLAB 7.8 ✓ Fuzzy Logic Toolbox 2.2.9 ✓ Image Processing Toolbox 6.3 ✓ Parallel Computing Toolbox 4.1 ✓ Signal Processing Toolbox 6.11 ✓ Statistics Toolbox 7.1 ✓ Wavelet Toolbox 4.4 - Java <ul style="list-style-type: none"> ✓ jdk1.6.0_03 ✓ jre1.6.0_03 - MySql 5.0, SQLyog 4.0 - Java Media Framework (JMF) 2.1.1e
Librería de enlace dinámico	JMatLink.dll
Driver	Controlador de la Cámara Web

Todos los instaladores y archivos ejecutables de la aplicación del proyecto están adjuntados en el CD de la memoria de almacenamiento.

3.1.1. Instalación de JAVA (jdk1.6.0_03 y jre1.6.0_03)

Si el computador no tiene instalado Java se lo procederá a instalar de la siguiente manera.

- Ejecute el archivo **jdk-6u3-windows-i586-p.exe** (ver Figura 187) y presione **Accept** para aceptar los términos de la licencia, como se muestra en la Figura 188.

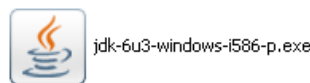


Figura 187: Archivo de instalación de Java 6

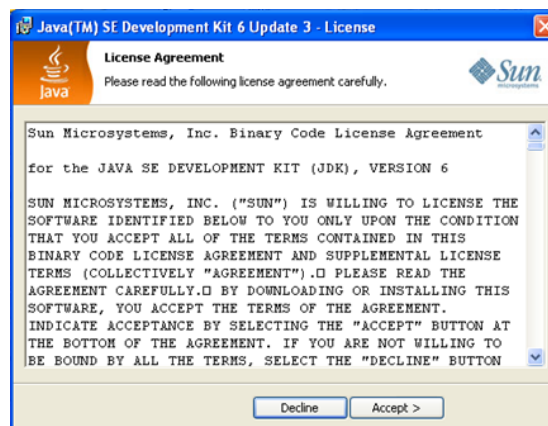


Figura 188: Licencia de Java SE Development Kit 6

- Se pueden personalizar las opciones que desea instalar y el directorio donde se van a copiar los archivos, luego presione **Next** para continuar (ver Figura 189).

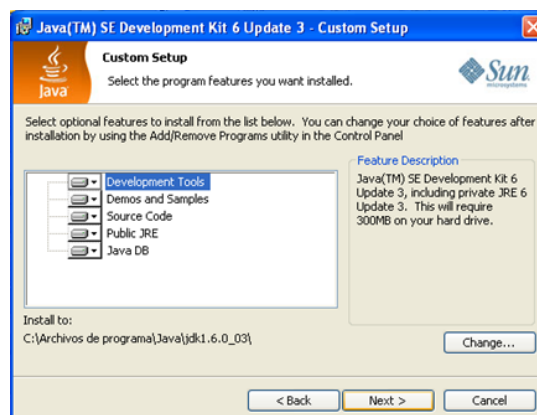


Figura 189: Configuración personalizada del JDK de Java

- Espere mientras avanza el proceso de instalación (ver Figura 190).

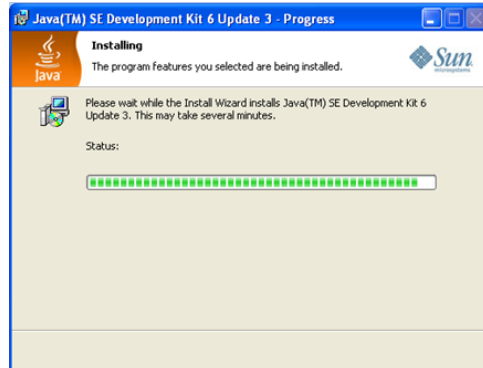


Figura 190: Proceso de Instalación de JDK de Java

- A continuación se visualizan otras opciones en las que se puede personalizar la instalación de JRE¹⁹ y el directorio donde se van a copiar los archivos, luego presione **Next** para continuar (ver Figura 191).

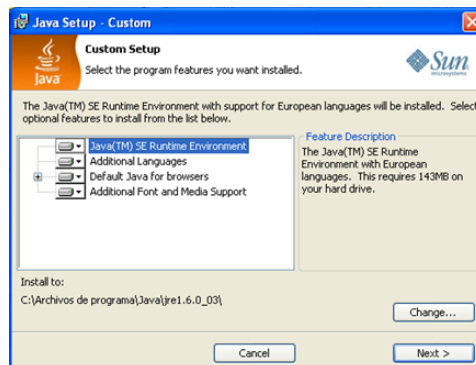


Figura 191: Configuración personalizada del JRE de Java

- Espere mientras avanza el proceso de instalación (ver Figura 192).



Figura 192: Proceso de Instalación de JRE de Java

¹⁹ Conjunto de utilidades que permite la ejecución de programas Java

- Cuando termina el proceso de instalación aparece una pantalla con el mensaje **Wizard Completed**; si desea ver el archivo **readme** con indicaciones y notas de la versión, seleccione el cuadro de chequeo **Show the readme file**; y presione **Finish** para terminar.



Figura 193: Instalación completa de Java

- Para verificar si la instalación se realizó correctamente, abra una consola del sistema (**Inicio > Programas > Accesorios > Símbolo del sistema**) y ejecute el comando **java -version**, debe aparecer una consola similar a la que se muestra en la **Figura 194** a continuación.

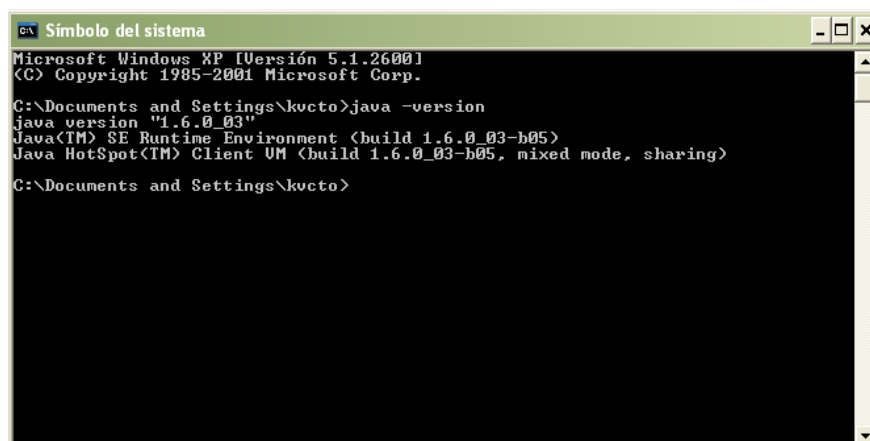


Figura 194: Consola del sistema

3.1.2. Instalación de Java Media Framework (JMF)

- Ejecute el archivo haciendo doble clic sobre el *jmf-2_1_1e-windows-i586.exe* (ver *Figura 195*) y se procede a preparar el proceso de instalación como lo indica la *Figura 196*.

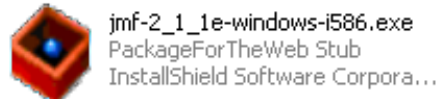


Figura 195: Archivo de Instalación de Java Media Framework

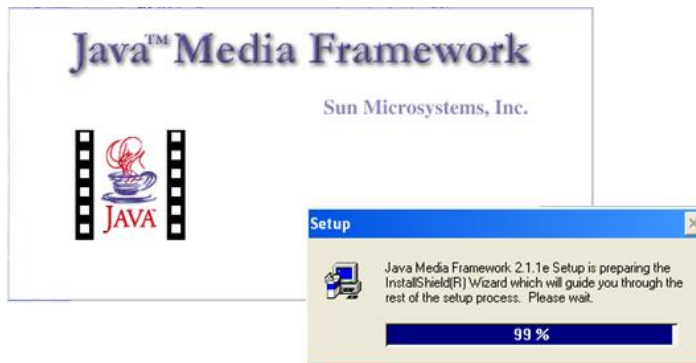


Figura 196: Preparando proceso de Instalación de JMF

- Haga clic en **Yes** (Si) después de haber leído el contrato de licencia (ver *Figura 197*).

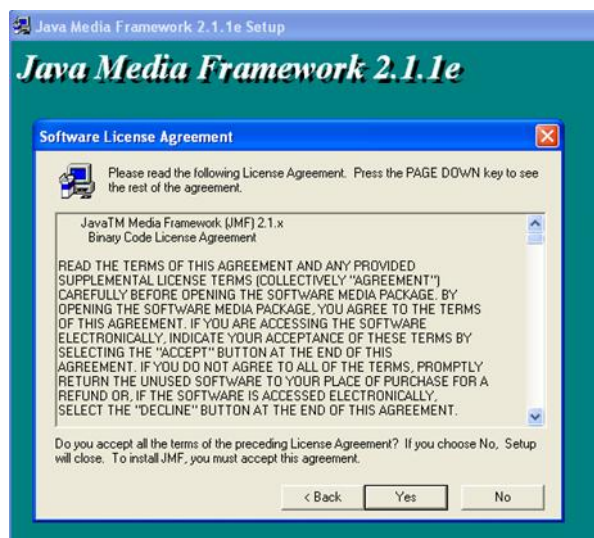


Figura 197: Contrato de Licencia JMF

- Haga clic en **Next** (Siguiete) para instalar el Java Media Framework en el directorio predeterminado (ver Figura 198).

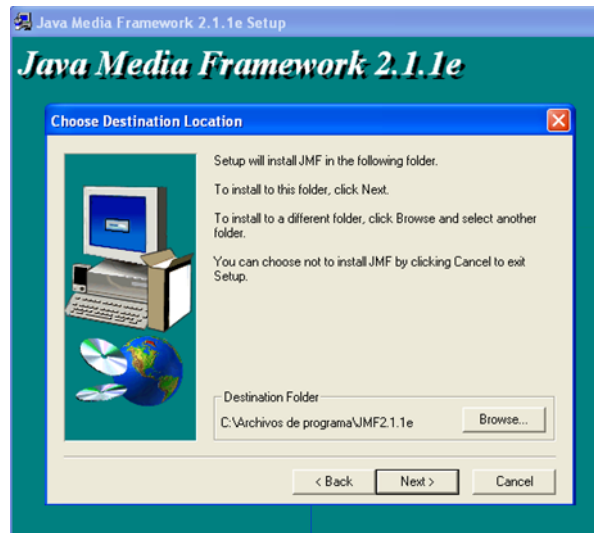


Figura 198: Destino de Instalación

- Haga clic en **Next** (Siguiete) para instalar Java Media Framework con las opciones por defecto (ver Figura 199).

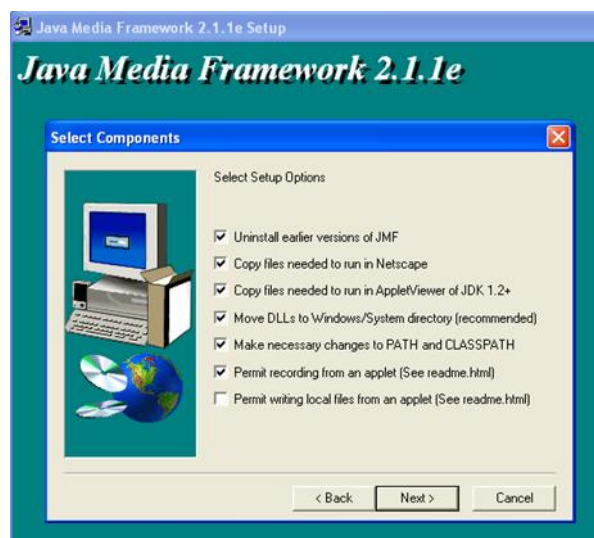


Figura 199: Selección de opciones de Instalación de JMF

- El sistema nos muestra los archivos de acceso directo de JMF en el equipo (ver Figura 200).

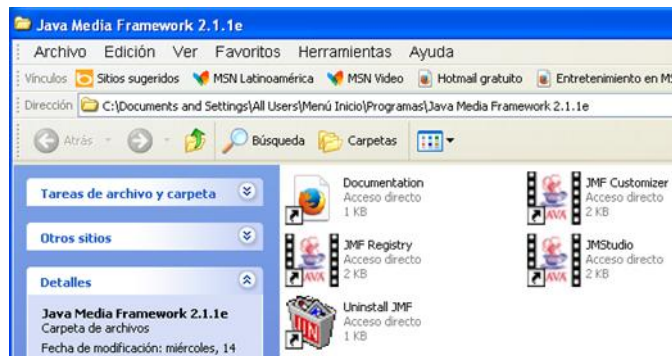


Figura 200: Archivos Instalados

- Finalice la instalación reiniciando el equipo, como se aprecia en la *Figura 201*.

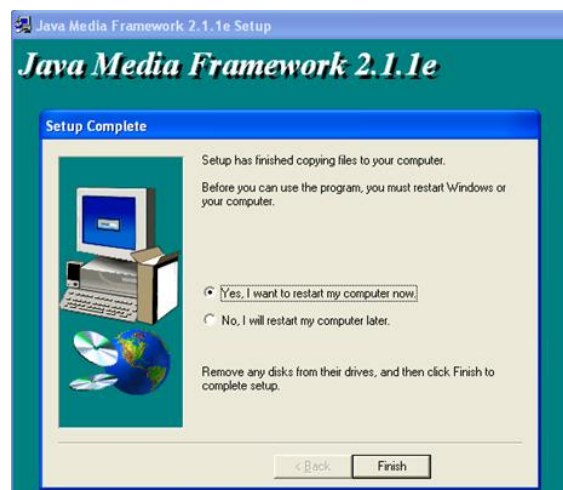


Figura 201: Instalación Finalizada

3.1.2.1. Configuración de JMF

Para la configuración de JMF se debe tener instalado el driver de la cámara a utilizar en la aplicación y esta debe estar debidamente conectada para verificar su funcionamiento.

- Haga clic en el icono **JMStudio** (ver *Figura 202*) creado en el escritorio después de la instalación y se verá una ventana como la *Figura 203*.



Figura 202: Ícono de JMF instalado



Figura 203: Ventana de Java Media Studio

- Luego hacer clic en menú **File > Preferences** como lo muestra la *Figura 204*.

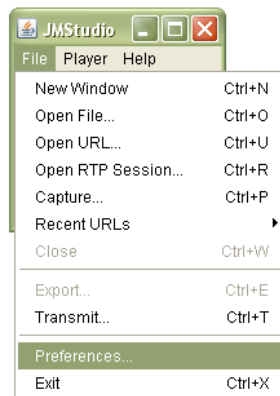


Figura 204: Opción de Preferencia

- Se muestra el Editor del Registro JMF y elegir la pestaña **Capture Devices** del menú como lo indica la *Figura 205*.

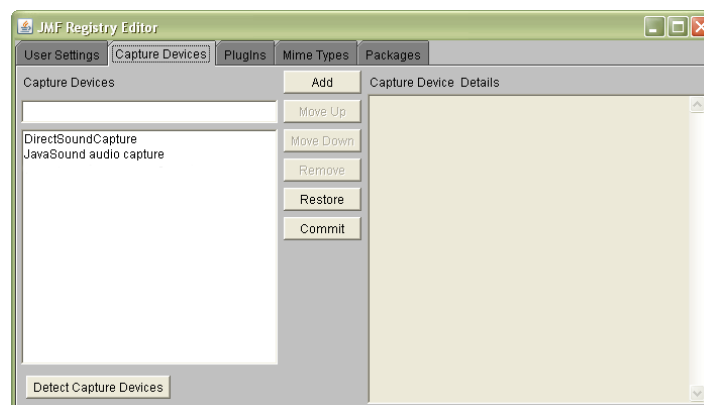


Figura 205: Editor de Registro de JMF

- Haga clic en el botón **Detect Capture Devices** para iniciar el para buscar los dispositivos conectados al ordenador y espere un momento (ver *Figura 206*).



Figura 206: Buscando Dispositivos de Captura

- Si su cámara web está conectada a su PC y la personalización de JMF se ha realizado correctamente, el Editor de registro debe mostrar los dispositivos de **vwf** que se enumeran a continuación (ver Figura 207).

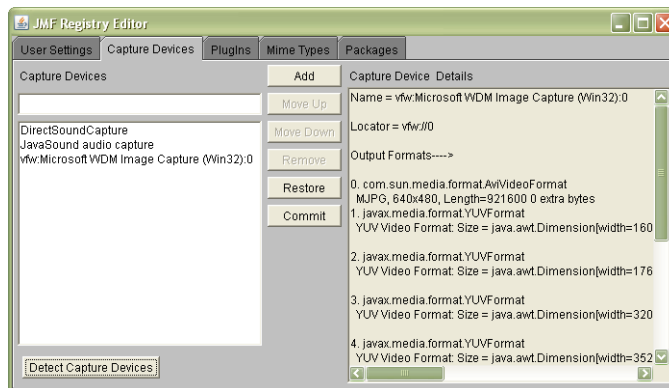


Figura 207: Dispositivos conectados

- Si hay más dispositivos diferentes al **vwf**, se los puede remover haciendo clic en la opción **Remove** (ver Figura 208), luego cierre la ventana de Editor de Registro.

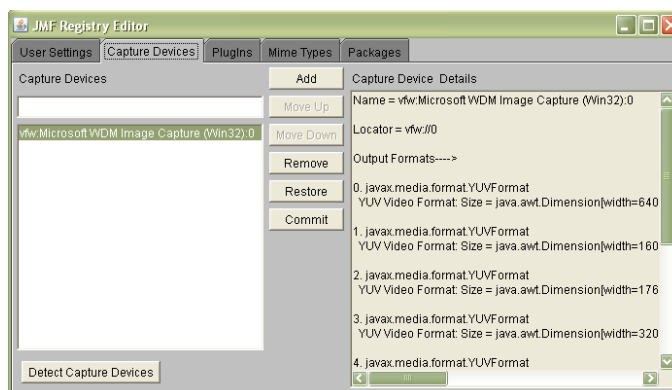


Figura 208: Dispositivos eliminados

- Una vez seleccionado el dispositivo en la *Figura 203*, hacer clic en el menú **File > Capture** como se indica en la *Figura 209*.

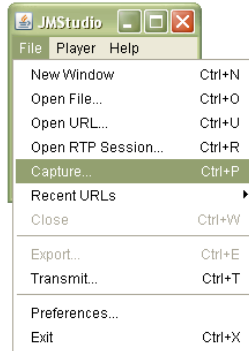


Figura 209: Opción Captura

- Se presentará la ventana para seleccionar el dispositivo de captura y sus opciones de configuración. Seleccione las opciones como indica la *Figura 210*.

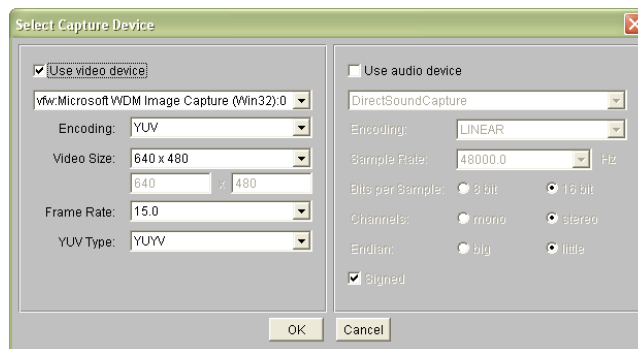


Figura 210: Seleccionar el Dispositivo de Captura

- En la *Figura 210* anterior clic en la opción **OK**, se visualiza la imagen en video de la cámara seleccionada, como lo muestra la *Figura 211*.



Figura 211: Imagen de video de la cámara seleccionada

- Para salir de la configuración del dispositivo, clic en el menú **File > Exit** como lo indica la *Figura 212*.

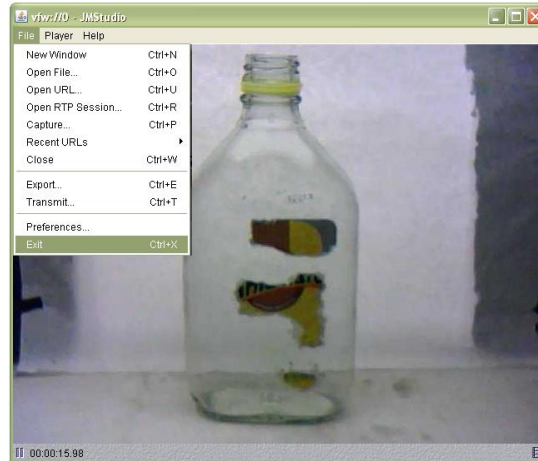


Figura 212: Salir de la configuración de JMF

3.1.3. Instalación de MATLAB R2009A

Inserte el DVD en la unidad de su computador, el programa de instalación normalmente se inicia automáticamente.

- Espere un momento mientras Microsoft Visual C++ 2005 Redistributable Package (x86) instala componentes en caso de que el equipo que no tenga Visual C++ 2005 instalado (*ver Figura 213*).

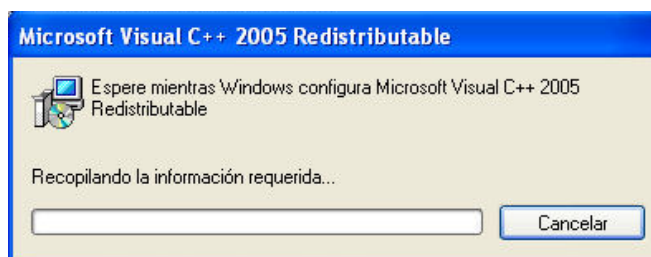


Figura 213: Configuración de Microsoft Visual C++ 2005 Redistributable

- En este paso elegir la opción **Install manually without using the Internet**, es decir se va instalar manualmente sin una conexión a Internet. Y luego hacer clic en **Next** (Siguiente) para continuar con la instalación (*ver Figura 214*).

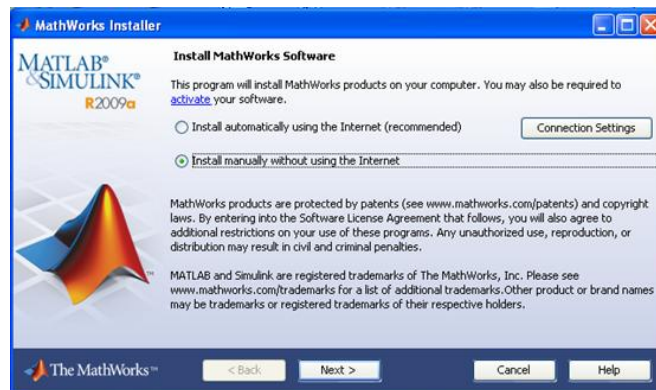


Figura 214: Opciones de Instalación de MatLab

- **Acuerdo de licencia de software**

Revise el acuerdo de licencia de software y, si está de acuerdo con los términos, seleccione **Yes** (Sí) y haga clic en **Next** (Siguiete) (ver Figura 215).

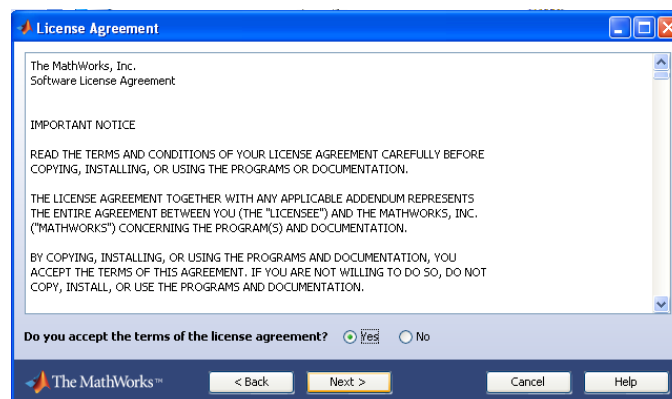


Figura 215: Acuerdo de Licencia

- **Introduzca clave de instalación**

Si usted no tiene una conexión a Internet, y decide instalar manualmente, el programa de instalación muestra el cuadro de diálogo Clave de instalación de archivos. Una clave de instalación del archivo identifica los productos que puede instalar.

Si usted tiene la clave, seleccione la opción **I have the File Installation Key for my license** (tengo la clave de instalación de archivos para mi licencia), introduzca la clave de instalación de archivos y haga clic en **Next** (Siguiete) (ver Figura 216).

Esta clave se encuentra en el DVD de MATLAB en la carpeta llamada Clave y Activación.



Figura 216: Clave de Instalación

- **Elegir el tipo de instalación**

En el cuadro de diálogo Tipo de instalación, especifique si desea realizar una instalación **Typical** (Típica) o **Custom** (Personalizada) y haga clic en **Next** (Siguiente).

Elija típica si no es necesario especificar los productos que desea instalar y no necesita acceder a las opciones de instalación.

En este caso se Selecciona **Custom** (Personalizada) ya que es necesario especificar los productos a instalar (ver Figura 217).



Figura 217: Tipo de Instalación

- **Especificar la carpeta de instalación**

Especifique el nombre de la carpeta en la que desea instalar los productos de The MathWorks. Puede aceptar la carpeta de instalación por defecto o especificar el nombre de una carpeta de instalación diferente. Si no existe la carpeta, el instalador la crea (ver Figura 218).

Si comete un error al escribir un nombre de carpeta y quiere empezar de nuevo con el nombre de la carpeta por defecto, haga clic en **Restore Default Folder** (Restaurar carpeta predeterminada). Para continuar con la instalación, haga clic en **Next** (Siguiente).

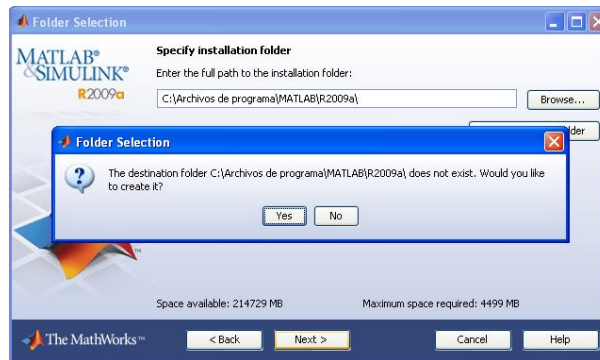


Figura 218: Carpeta de Instalación

- **Especifique los productos a instalar**

Si está realizando una instalación personalizada, puede especificar que los productos que desea instalar en el cuadro de diálogo de selección de productos (ver Figura 219). Este cuadro de diálogo muestra todos los productos asociados a la licencia que ha seleccionado o con la clave de activación que ha especificado. En el cuadro de diálogo, todos los productos están preseleccionados para su instalación. Si no desea instalar un producto en particular, desactive la casilla de verificación junto a su nombre.

Para este caso se debe seleccionar los siguientes productos:

- ✓ MATLAB 7.8
- ✓ Fuzzy Logic Toolbox 2.2.9
- ✓ Image Processing Toolbox 6.3
- ✓ Parallel Computing Toolbox 4.1
- ✓ Signal Processing Toolbox 6.11
- ✓ Statistics Toolbox 7.1
- ✓ Wavelet Toolbox 4.4

Después de seleccionar los productos que desea instalar, haga clic en **Next** (Siguiente), para continuar con la instalación. Después de hacer clic en **Next**, es posible que reciba un mensaje que indica que uno o varios de los productos que ha seleccionado dependen de otros productos.

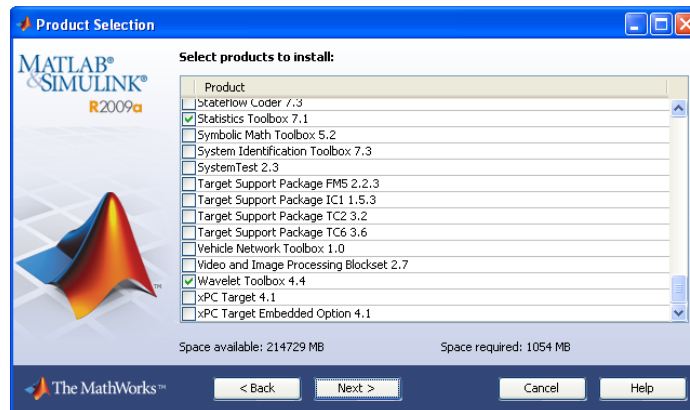


Figura 219: Selección de Producto

- **Especificar opciones de instalación**

Para instalaciones personalizadas, puede especificar varias opciones de instalación, incluyendo (ver Figura 220).

- ✓ Configuración de permisos de todos los archivos instalados sólo de lectura.
- ✓ La determinación de accesos directos para el software MATLAB en el menú Inicio y en el escritorio.
- ✓ Especificación de archivos del sistema operativo, asociados con MATLAB, basados en su extensión de archivo. Por ejemplo, si asocia los archivos con la extensión `.m` con MATLAB, el sistema operativo identifica el tipo de estos archivos como MATLAB M-file.

Después de seleccionar las opciones de instalación, haga clic en **Next** (Siguiente) para continuar con la instalación.

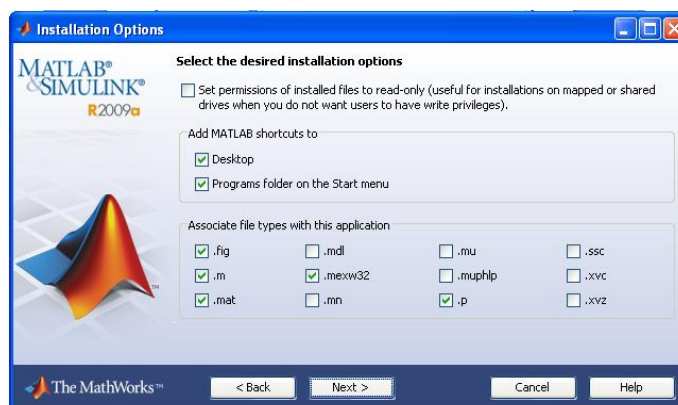


Figura 220: Opciones de Instalación

- **Confirmar sus elecciones**

Antes de que comience a copiar los archivos en el disco duro, el instalador muestra un resumen de las opciones de instalación. Para cambiar la configuración, haga clic en **Back** (Atrás). Para continuar con la instalación, haga clic en **Install** (Instalar) (ver *Figura 221*).

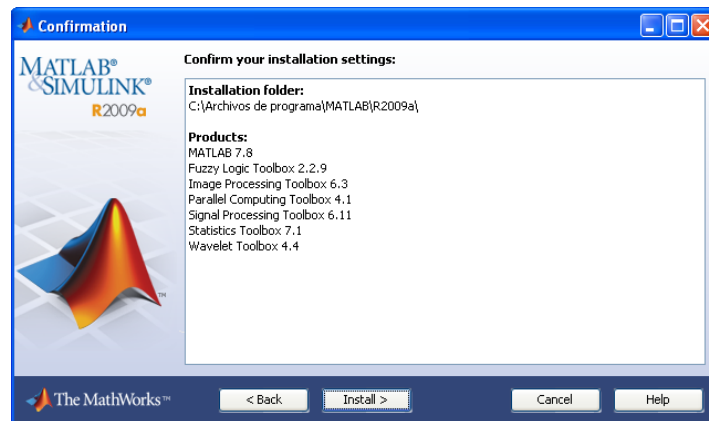


Figura 221: Confirmación de Elecciones

- El programa de instalación muestra un cuadro de diálogo de estado para mostrar el progreso de la instalación (ver *Figura 222*).

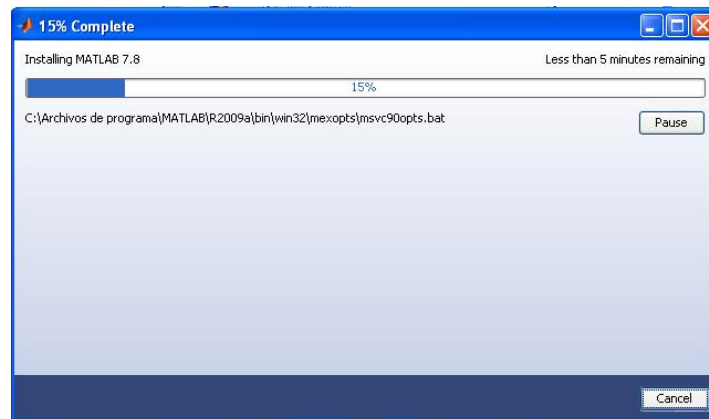


Figura 222: Proceso de Instalación

- **Finalización de la instalación**

Cuando la instalación se realiza correctamente, el programa de instalación muestra el cuadro de diálogo **Installation is complete** (Instalación completa). En este cuadro de diálogo, puede elegir activar el software que acaba de instalar. Usted no puede utilizar

el software que ha instalado hasta que lo active. The MathWorks recomienda activar inmediatamente después de la instalación.

Haga clic en **Next** (Siguiete) para continuar con la activación.

Si decide salir del instalador sin realizar la activación, desactive la opción **Activate MATLAB** (Activar MATLAB) y haga clic en **Finish** (Finalizar) (ver Figura 223).

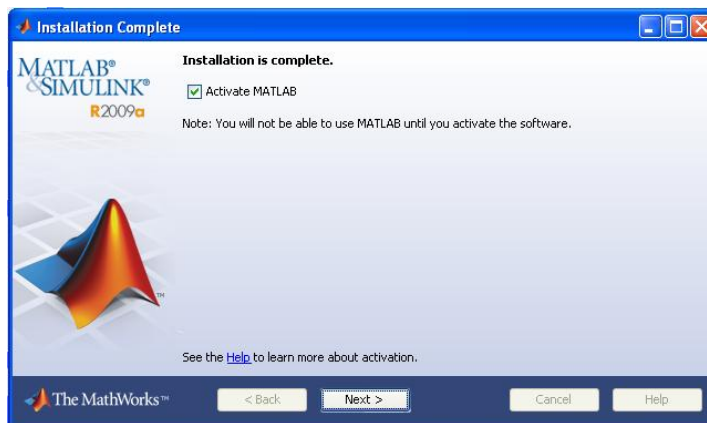


Figura 223: Instalación completa

- **Activación sin conexión a Internet**

Para activar el software sin internet, seleccione **Activate manually without the Internet** (Activar manualmente sin la opción de Internet) y haga clic en **Next** (Siguiete) (ver Figura 224).



Figura 224: Activación de Matlab

- **Especificar ruta de archivo de licencia**

Para activar sin conexión a Internet, debe tener un archivo de licencia. El Archivo de Licencia identifica los productos que se pueden ejecutar. Haga clic en la opción **Browse...** (Buscar) (ver Figura 225).



Figura 225: Activación sin Internet

- **Especifique la ruta del archivo de Licencia**

Especifique la ubicación de su archivo de licencia y clic en **Select**. El archivo viene en el DVD de instalación de MATLAB en la carpeta *License_Activation* llamado *lic_standalone.dat* (ver Figura 226).

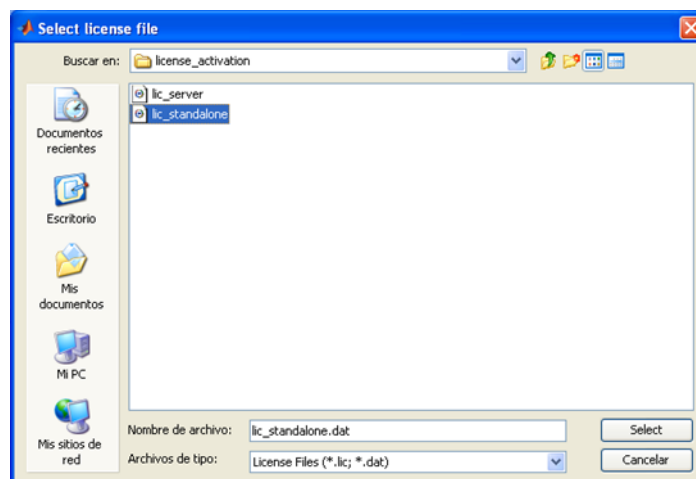


Figura 226: Selección del archivo de licencia

Una vez especificada la ruta completa del archivo de licencia en el cuadro de texto haga clic en **Next** (Siguiente) (ver Figura 227).



Figura 227: Archivo Seleccionado

- **Activación finalizada**

Después de la activación de su instalación, puede ejecutar el software de MathWorks. En este caso no ejecutamos MATLAB, y se desactiva la opción **Start MATLAB** (Iniciar MATLAB) y haga clic en **Finish** (Finalizar) para salir del proceso de activación (ver Figura 228).

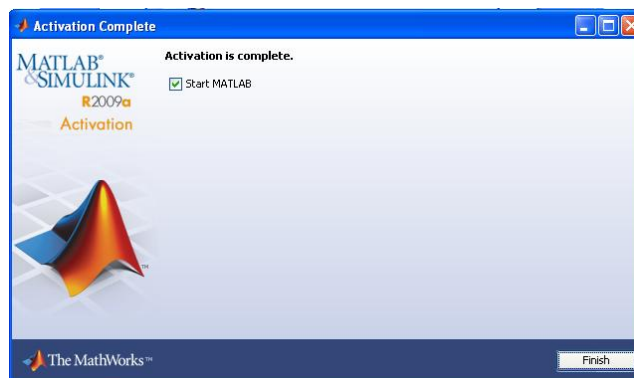


Figura 228: Activación completa de MatLab

3.1.4. Instalación y Configuración de Gestor de Base de Datos: My SQL 5.0

Abrir el archivo llamado **mysql-5.0.24a-win32.zip** (ver Figura 229), en donde se encuentra el instalador llamado *Setup.exe* y ejecutarlo.



Figura 229: Archivo de instalación de MySQL

- En la ventana (ver Figura 230) dar clic en **Next**.



Figura 230: Asistente de configuración de MySQL 5.0

- Marcar la opción **Typical** como lo indica la Figura 231.



Figura 231: Tipo de Instalación

- La ventana siguiente (Figura 232) se muestra la configuración a instalar y dar clic en **Install**.

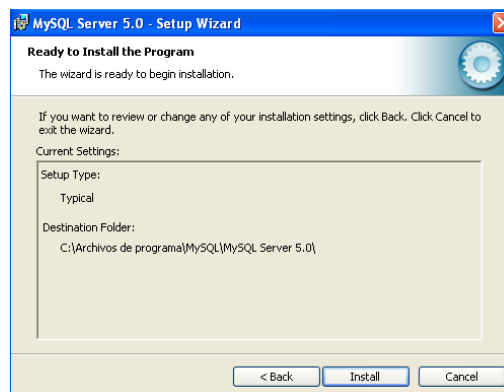


Figura 232: Configuración actual a instalar

- La *Figura 233* indica la instalación en proceso.

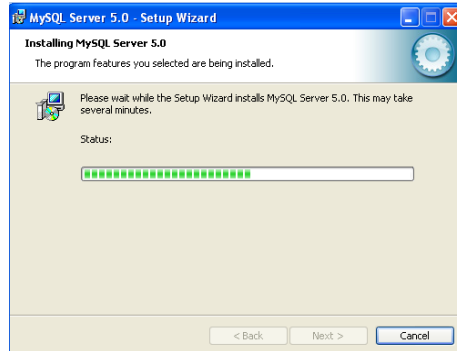


Figura 233: Instalación en proceso

- Antes de la finalización de la instalación aparece una ventana para registrarse en MySQL.com, pero en este caso se dará clic en **Skip Sign-Up** y luego en **Next** (ver *Figura 234*).

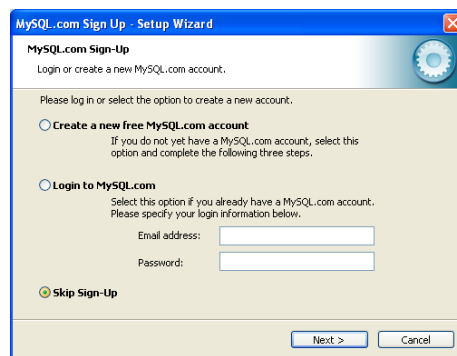


Figura 234: Registro de Cuenta de MySQL

- Tras la instalación aparecerá el asistente de configuración de MySQL Server, marcar la opción **Configure the MySQL Server now** y pulsar clic sobre el botón **Finish**, para finalizar la primera parte de la instalación (ver *Figura 235*).



Figura 235: Instalación completa

- Si todo lo anterior fue correcto debe presentarse en la pantalla un cuadro de dialogo similar a la figura. Dar clic en **Next** (ver Figura 236).

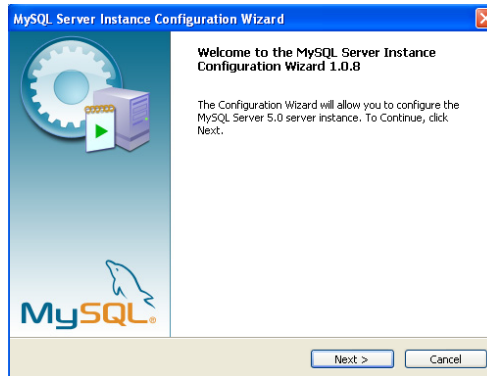


Figura 236: Configuración de la instancia del servidor de MySQL

- En la Figura 237 se escoge la opción **estándar configuration** para una configuración de uso general para el servidor.

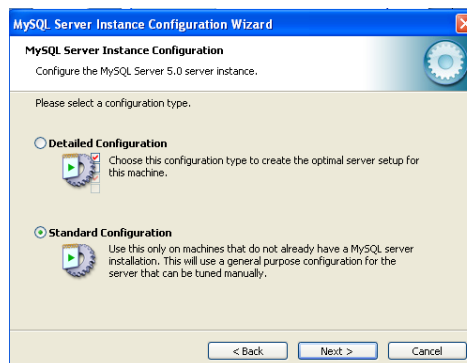


Figura 237: Tipo de configuración de la instancia de servidor

- En la Figura 238, la opción recomendada es **Install As Windows Service** y no arrancar automáticamente con el arranque del sistema y clic en **Next**.



Figura 238: Opciones de servicio en Windows.

- Se ingresa la contraseña para el administrador (*root*) como lo indica la *Figura 239*.



Figura 239: Opciones de seguridad

- Pulsar clic en el botón **Execute**, para que el asistente de configuración de MySQL Server 5.0 aplique todas las opciones indicadas anteriormente (ver *Figura 240*).

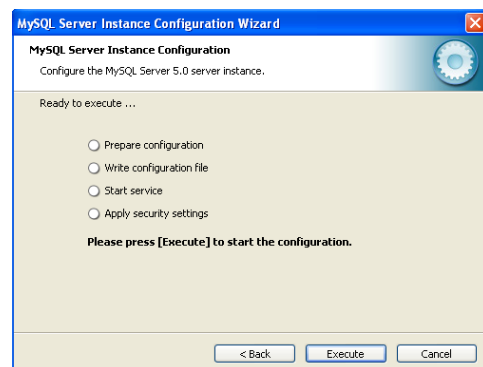


Figura 240: Ejecutar las opciones de Instancia

- Si no hay problemas mostrará esta ventana indicando que el proceso de instalación y configuración de MySQL Server ha terminado y se ha instalado e iniciado el Servicio que ejecutará MySQL (ver *Figura 241*).

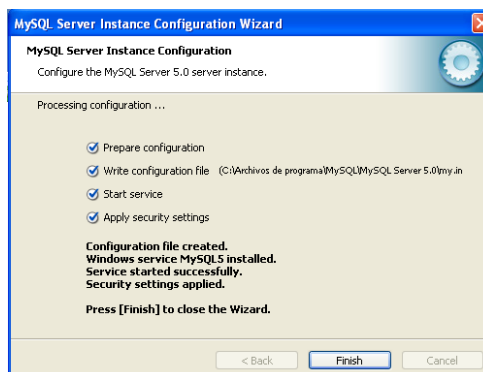


Figura 241: Configuración del servicio de MySQL Finalizada

3.1.5. Instalación de SQLyog 4.0

Dar doble clic al archivo instalador **SQLyog40.exe** (ver Figura 242).



Figura 242: Archivo de Instalación de SQLyog

- En la ventana (ver Figura 243) que aparece dar clic en el boton **Next**.



Figura 243: Asistente de configuración de SQLyog 4.0

- En la Figura 244 se acepta los términos de licencia y dar clic en **Next**.

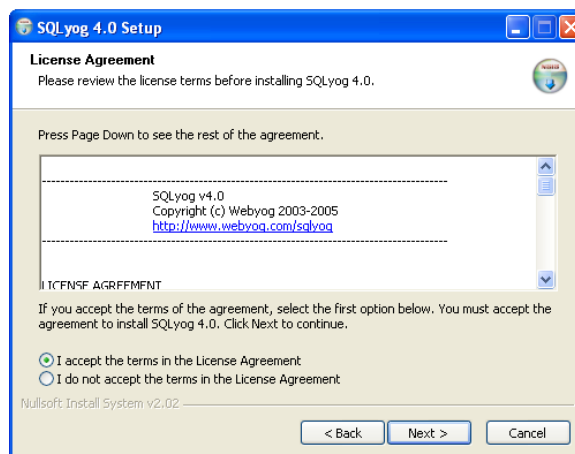


Figura 244: Términos de Licencia de SQLyog 4.0

- Dejar la ruta por defecto donde se va a instalar el programa C:\Archivos de Programa\SQLyog, y dar clic en **Install**, (ver Figura 245).

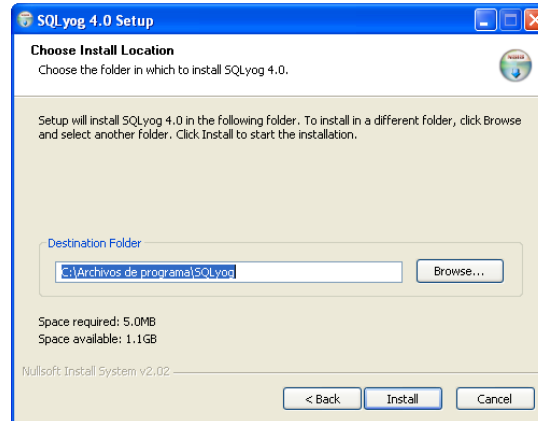


Figura 245: Directorio de Instalación

- El programa empezara a instalarse como se muestra en la *Figura 246*.

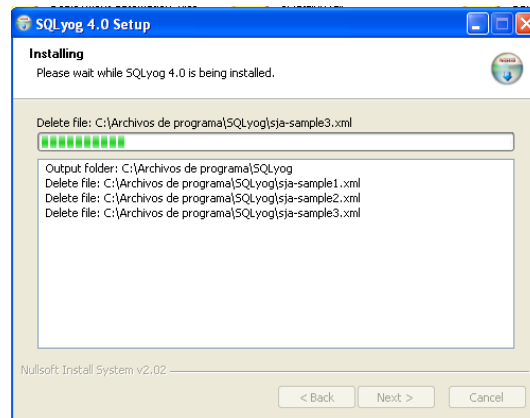


Figura 246: Proceso de Instalación

- Finalmente se activa la casilla **Run SQLyog 4.0** para correr el programa y configurarlo y dar clic en el boton **Finish** (ver *Figura 247*).

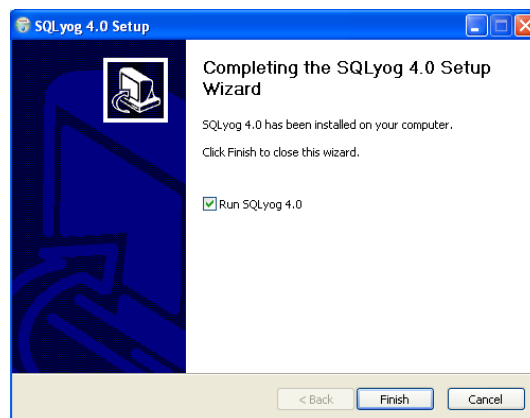


Figura 247: Instalación completa de SQLyog 4.0

- Se visualiza el entorno del programa SQLyog y dar clic en **OK** (ver Figura 248).

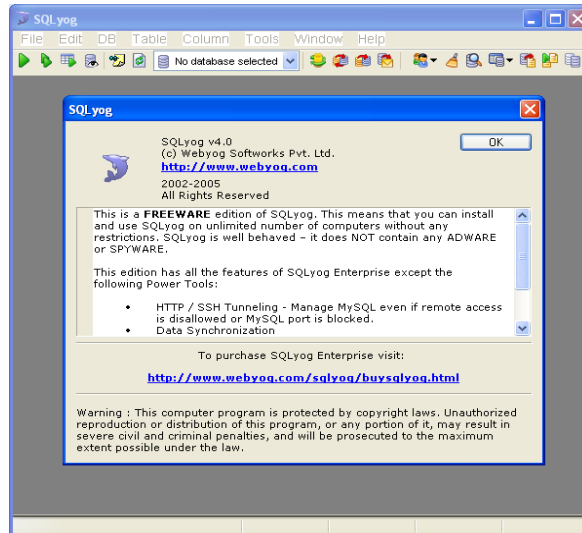


Figura 248: Entorno de SQLyog

3.1.5.1. Configurar SQLYOG para conectar a MYSQL

- En la ventana de la Figura 249, dar clic en la opción **New** para realizar una nueva conexión con MySQL.

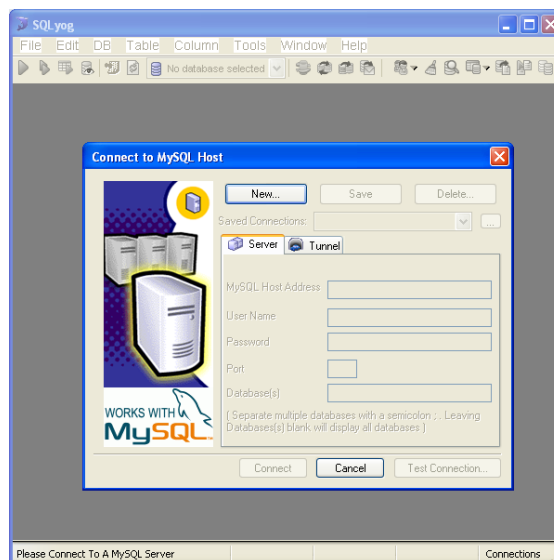


Figura 249: Nueva Conexión

- En la ventana de la Figura 250, ingresar el nombre de la nueva conexión y clic en **OK**.

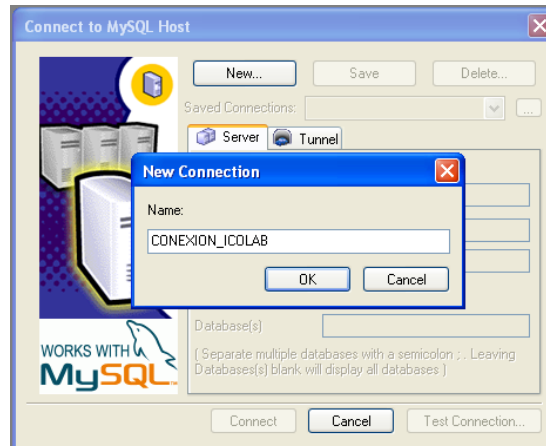


Figura 250: Nombre de nueva conexión

- En la ventana de conexión de la *Figura 251*, se ingresa la contraseña de **root**, la cual es **root**. Luego damos clic en el botón **Connect**.

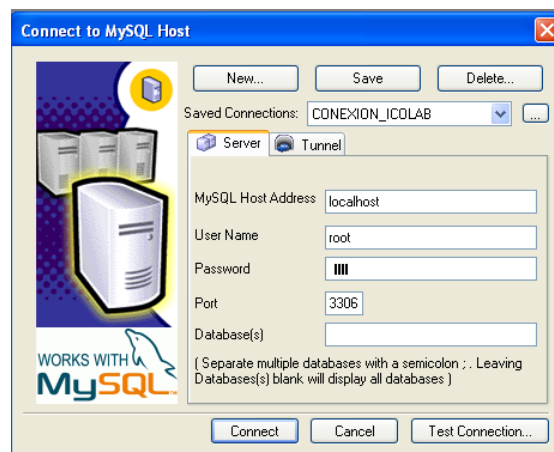


Figura 251: Conexión con Host de MySQL

- Se guarda los detalles de la conexión dando clic en **Si** (ver *Figura 234*).

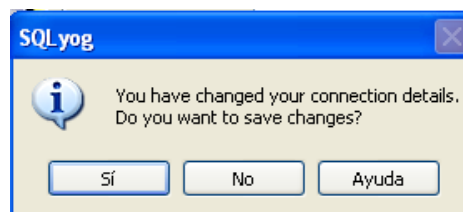


Figura 252: Guardar detalles de conexión

- En la interfaz gráfica de SQLyog escribimos la sentencia **create database icolab_bd**, en el editor Query, como lo indica la *Figura 253*.

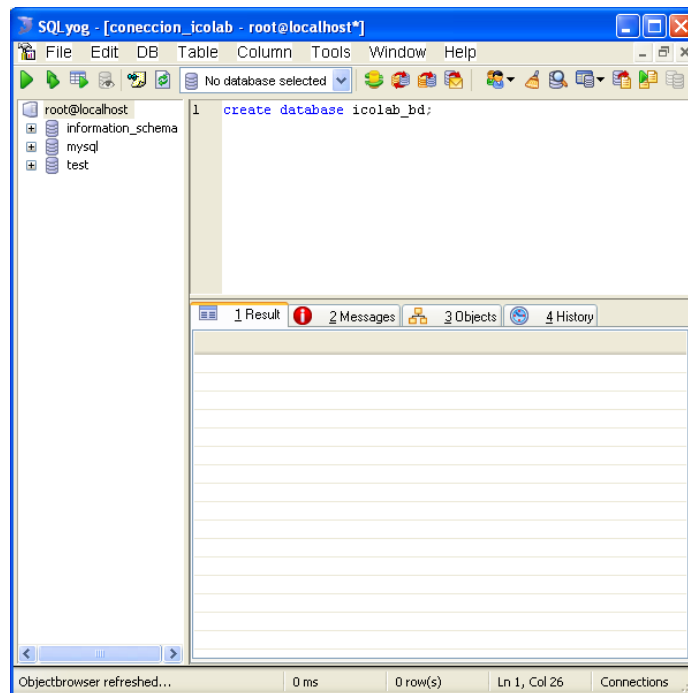


Figura 253: Creación de base de datos

- Para ejecutar esta sentencia damos clic en el botón **Execute Current Query** (ver Figura 254).

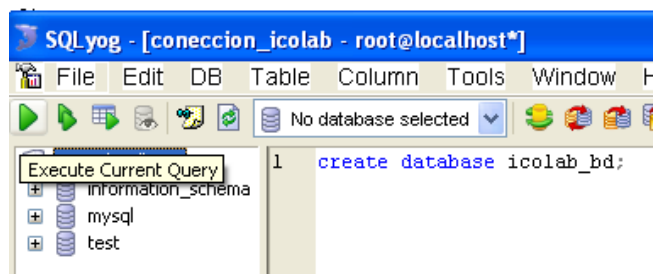


Figura 254: Ejecutar sentencia

- Verificamos que la base de datos ha sido creada como lo indica la Figura 255.

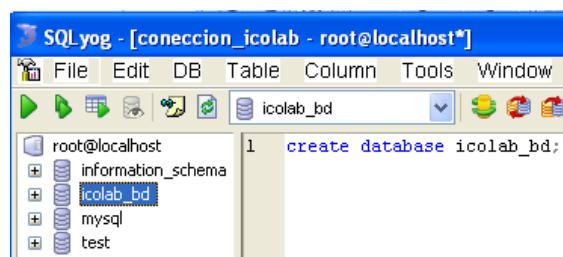


Figura 255: Base de Datos Creada

- Finalmente cerrar la interfaz gráfica de dando clic **File> Exit> OK** (ver Figura 256).

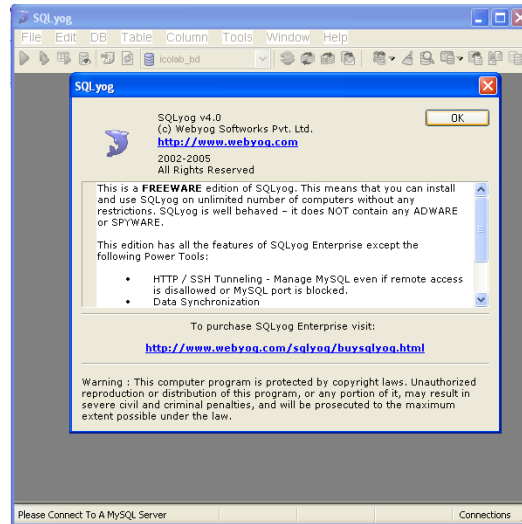


Figura 256: Salir de SQLyog

3.1.6. Conexión de MATLAB con la Interfaz Gráfica del Sistema realizada en JAVA

La interfaz gráfica de Java debe comunicarse con las funciones programadas en MatLab, para ello es necesaria la utilización de la librería *JMatLink* (Figura 257), que es la encargada de unir estos dos programas. Una vez que Java toma las imágenes, mediante unas líneas de código se llama a la función de MatLab que realiza la inspección de las imágenes.



JMatLink.dll

Figura 257: Archivo de comunicación con MATLAB

El código de unión es el siguiente:

```
JMatLink engine = new JMatLink();
engine.engOpen();

engine.engOutputBuffer();
engine.engEvalString(funcion);

String letra = engine.engGetOutputBuffer();
engine.engClose();
```

Figura 258: Código de conexión JMatLink

Para completar la instalación se debe copiar el archivo *JMatLink.dll* en los directorios *c:\windows* y *c:\windows\system32*, como se muestra en la *Figura 259*. Este archivo se adjunta en disco de la memoria de almacenamiento del proyecto.

Luego añadir a la variable de entorno PATH, la siguiente línea:

“C:\Archivos de programa\MATLAB\R2009a\bin”

Asegurarse de reemplazar la ruta adecuada de la instalación de MATLAB.

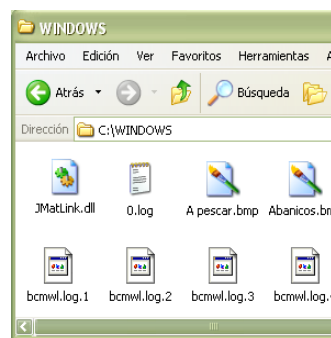


Figura 259: Instalación del archivo de librería JMatLink

3.1.7. Instalación de ICOLab v1.0

Ejecute el archivo *Instalador_ICOLab_v.01.exe* (ver *Figura 260*), seguidamente se mostrará el asistente de instalación y presione **Siguiente>** (ver *Figura 261*).



Figura 260: Archivo de Instalación de ICOLab v1.0



Figura 261: Asistente de Instalación ICOLab v1.0

- Seleccione el directorio de instalación y clic en **Siguiente**> (ver Figura 262).

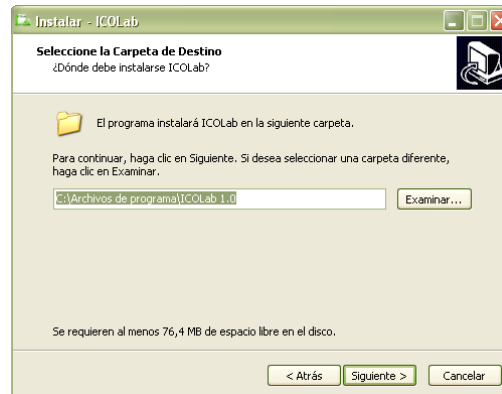


Figura 262: Carpeta de Instalación de ICOLab v1.0

- Seleccione la carpeta donde colocar los accesos directos de ICOLab v1.0, en este caso se deja por defecto y dar clic en **Siguiente**> (ver Figura 263).

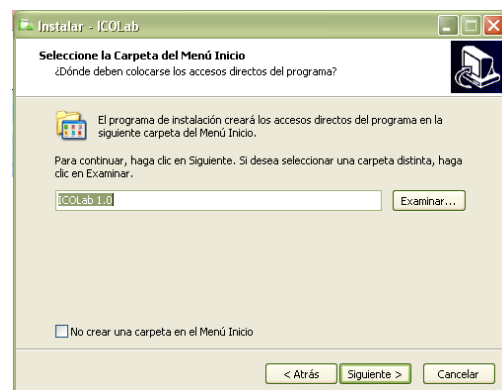


Figura 263: Carpeta de Accesos Directos

- Para crear el ícono de acceso directo, marcamos la opción *Crear un ícono en el escritorio* como lo indica la Figura 264.

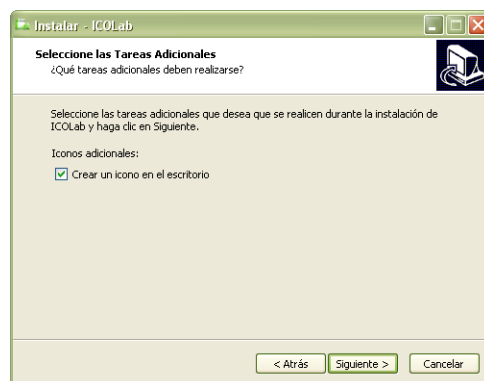


Figura 264: Crear ícono

- En la *Figura 265* se muestra la configuración para instalar el programa, dar clic en **Instalar** para ejecutarla.

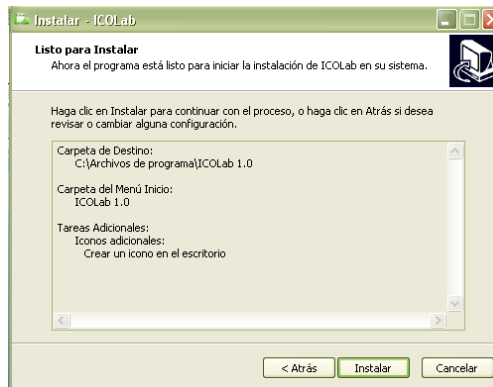


Figura 265: Configuraciones realizadas a instalar

- En la *Figura 266* se muestra el proceso de instalación.

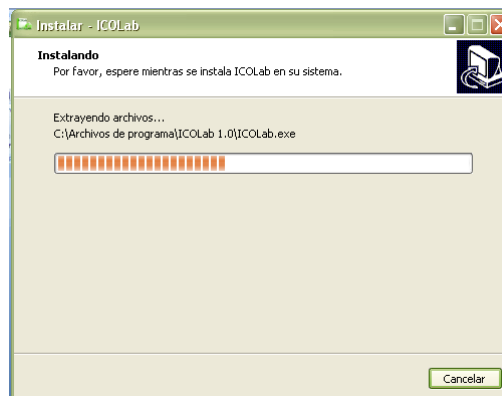


Figura 266: Proceso de Instalación ICOLab v1.0

- Finalmente se muestra la *Figura 267*, la cual indica que la instalación ha sido completada.

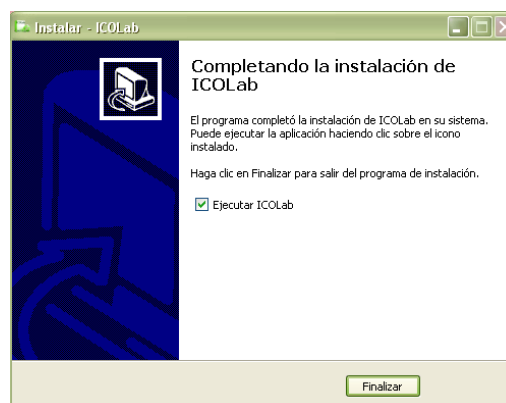


Figura 267: Instalación completa de ICOLab v1.0

Al ejecutar el programa se abrirá la ventana principal con la opción de Acceso al Inicio.

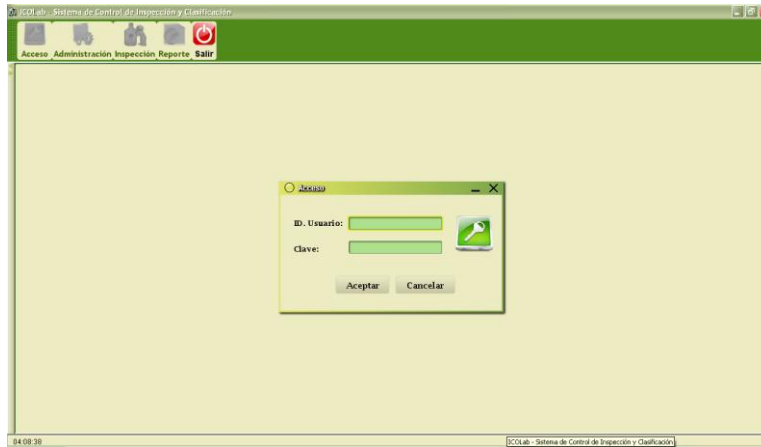


Figura 268: Ventana Principal de ICOLab 1.0

Para el buen funcionamiento de *ICOLab v1.0* se realizó el Manual del Usuario que viene adjunto a este proyecto.

3.1.8. Verificar la variables de entorno PATH

Una vez instalado y configurado todos los programas que son de ayuda para el correcto funcionamiento del sistema ICOLab, se verifican las variables de entorno **PATH**, **PATH** es la variable del sistema que utiliza el sistema operativo para buscar los ejecutables necesarios. Para verificar o cambiar las variables de entorno:

- Haga clic con el botón secundario en **Mi PC** y, a continuación, haga clic en **Propiedades**.
- Haga clic en la pestaña **Opciones avanzadas**.
- Haga clic en **Variables de entorno**.
- Buscar la variable **Path**
- Marque la variable y, haga clic en **Modificar** para verificar cambiar o cambiar su nombre o su valor.

Los valores que deben existir son los siguientes:

- **C:\Archivos de programa\Java\jdk1.6.0_03\jre\bin;**
- **C:\Archivos de programa\MATLAB\R2009a\bin;**
- **C:\Archivos de programa\MATLAB\R2009a\bin\win32;**
- **C:\Archivos de programa\JMF2.1.1e\lib\jmf.jar**

Etapa 3.2: Implementación y evaluación del Sistema Integrado

El Sistema control de inspección se integró con todo el hardware y la programación: como librerías e interfaces cumpliendo con los requerimientos exigidos.

Tras la finalización del sistema, se efectuaron las pruebas correspondientes con el objeto de verificar si el sistema se adapta o no a los requerimientos del administrador y de los usuarios (estudiantes) que harán uso del sistema, además permite descubrir errores existentes y realizar las correcciones respectivas.

Las pruebas se efectuaron durante todo el proceso del Trabajo de Titulación en la Universidad Nacional de Loja, Área de Energía, las Industrias y los Recursos Naturales no Renovables, específicamente en el Laboratorio de Automatización y Control Industrial, utilizando para ello computadores con características de la *TABLA XXIII* y *TABLA XXIV*.

3.2.1. Prueba de Usabilidad (Funcionalidad, Diseño y Presentación)

Las pruebas fueron aplicadas en 2 niveles de personal: **Administradores** (Docentes investigadores que utilizan el Laboratorio de Automatización y Control); y **Usuarios** (Estudiantes de las carreras del Área de Energía, las Industrias y los Recursos Naturales No Renovables), todos estos posibles usuarios del sistema.

El esquema de pruebas se detalla en la siguiente tabla:

TABLA XXV: ESQUEMA DE PRUEBAS

ESQUEMA DE PRUEBAS	
Fecha	10 de Marzo del 2014
Alcance	Procesamiento digital de imágenes (PDI) apoyado en un entrenamiento Neuro-difuso, para la inspección de objetos defectuosos en el Laboratorio de Automatización y Control.
Items a Probar	Interacción del sistema y el usuario (Interfaz) Administrar Personal Creación de nuevo personal Modificar Personal Eliminar Personal Administrar Usuario Creación de nuevo Usuario Modificar Usuario Inhabilitar Usuario Operar Inspección

	Visualizar el resultado de la inspección Tiempo de espera del resultado Conexión de Cámara Conexión de MatLab Cambiar directorio de archivos Crear directorio de archivos Abrir directorio de archivos Consultar reporte estadístico de resultados Imprimir reporte estadístico de resultados
Estrategia	Manipulación del sistema por parte de los usuarios, obtención de información necesaria de pruebas a partir de encuestas.
Recursos	Computador Laboratorio de Automatización y Control
Calendario	Del 12 al 19 de Marzo 2014
Responsable	Víctor Espinoza Ochoa

Población de Estudio:

TABLA XXVI: POBLACIÓN DOCENTES

POBLACIÓN DE DOCENTES					
Carrera	Sistemas	Electromecánica	Electrónica y Telecomunicación.	Geología Ambiental y Ordenamiento Territorial	TOTAL
	1	1	1	1	4

TABLA XXVII: POBLACIÓN ESTUDIANTES AEIRNNR

POBLACIÓN DE ESTUDIANTES AEIRNNR							
Carrera	Sistemas		Electromecánica		Electrónica y Telecomunicación.	Geología Ambiental y Ordenamiento Territorial	TOTAL
Paralelo	A	B	A	B	A	A	
6to				17			17
10mo	22	23	21	-	26	18	110
							127

Fuente: Secretaría General del AEIRNNR, 2014

Tamaño de la Muestra:

Se determinó el tamaño de la muestra para la población de Estudiantes, utilizando la siguiente fórmula que tiene en cuenta el tamaño de la población, el nivel de confianza y el margen del error.

Para calcular el tamaño de la muestra suele utilizarse la siguiente fórmula [68]:

$$n = \frac{N\sigma^2Z^2}{N - 1 e^2 + \sigma^2Z^2}$$

Dónde:

n = el tamaño de la muestra.

N = tamaño de la población.

σ = Desviación estándar de la población que, generalmente cuando no se tiene su valor, suele utilizarse un valor constante de 0,5.

Z = Valor obtenido mediante niveles de confianza. Es un valor constante que, si no se tiene su valor, se lo toma en relación al 95% de confianza equivale a 1,96 (como más usual) o en relación al 99% de confianza equivale 2,58, valor que queda a criterio del investigador.

e = Límite aceptable de error muestral que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador [68].

Reemplazando valores en la fórmula se obtiene:

$$n = \frac{N\sigma^2Z^2}{N - 1 e^2 + \sigma^2Z^2} = \frac{127 \cdot 0,5^2 \cdot 1,96^2}{127 - 1 \cdot 0,09^2 + 0,5^2 \cdot 1,96^2} = 61,56 \cong 62$$

Después de calcular el tamaño de la muestra (n=62), se utilizó el tipo de muestreo probabilístico estratificado proporcional. “Estratificado” en medida que la población está dividida en sub-grupos de acuerdo a la carrera de procedencia. Y “proporcional”, ya que el tamaño de cada estrato ha sido establecido teniendo en cuenta la población de estudiantes de cada carrera, ciclo y paralelo.

El cálculo de factor de la proporción.

$$f = \frac{n}{N}$$

$$f = \frac{61,56}{127}$$

$$f = 0,484$$

Este factor se multiplica por cada población del paralelo de las carreras mencionadas en la TABLA XXVII y nos resulta la muestra siguiente:

TABLA XXVIII: MUESTRA ESTRATIFICADA PROPORCIONAL DE LOS ESTUDIANTES

MUESTRA DE ESTUDIANTES AEIRNNR							
Carrera	Sistemas		Electromecánica		Electrónica y Telecomunicación.	Geología Ambiental y Ordenamiento Territorial	TOTAL
Paralelo	A	B	A	B	A	A	
6to				8			8
10mo	11	11	10	-	13	9	54
							62

Por otro lado la muestra de Docentes es la misma, ya que la población es sumamente pequeña.

TABLA XXIX: MUESTRA DE DOCENTES

MUESTRA DE DOCENTES					
Carrera	Sistemas	Electromecánica	Electrónica y Telecomunicación.	Geología Ambiental y Ordenamiento Territorial	TOTAL
No.	1	1	1	1	4

Por consiguiente tenemos 66 encuestas aplicadas para la validación del sistema ICOLab: 4 a nivel de Administrador (Docentes) y 62 a nivel de Usuario (Estudiantes).

3.2.2. Funcionalidad del sistema ICOLab: Resultados obtenidos en tiempo real

Se analizaron 100 imágenes a las cuales se aplicaron al sistema ICOLab y por ende al algoritmo y al sistema FIS, se verificó que 5 botellas fueron detectadas incorrectamente.

Dentro de los envases analizados existieron algunos sin defectos que fueron detectados como defectuosos, esto se debe a la tonalidad del vidrio respecto a la a iluminación en el borde de la botella y se presente un tono más oscuro en esa área, esto provoca que el algoritmo de procesamiento digital de imágenes lo detecte, extraiga valores significativos y resulten como Falsos Positivos en las pruebas de estudio.

TABLA XXX: RESULTADOS DE LA PRUEBA DE ESTUDIO

Resultados de la Prueba de Estudio	Envases de botellas		TOTAL
	Con defectos(Malas)	Sin defectos (Buenas)	
Positivos	50 VP	5 FP	55
Negativos	0 FN	45 VN	45
TOTAL	50	50	100

VP: Verdadero Positivo, FP: Falso Positivo, FN: Falso Negativo, VN: Verdadero Negativo.

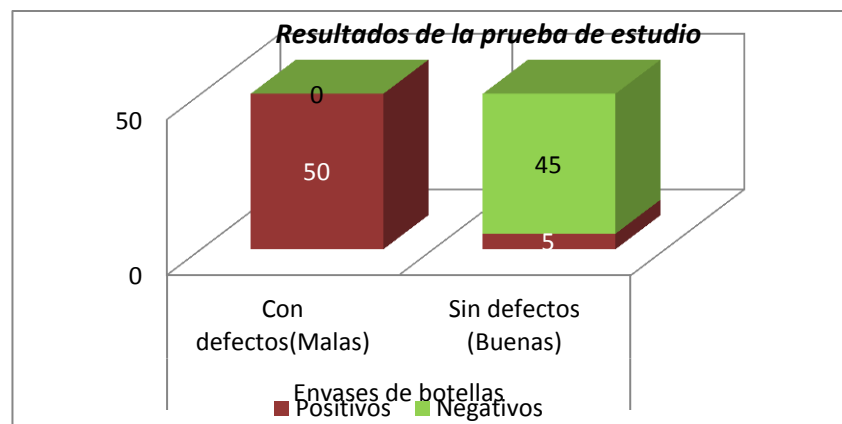


Figura 269: Gráfica Estadística de los resultados de prueba de estudio

Parámetros de desempeño

Sensibilidad: Capacidad o habilidad de detectar muestras negativas (con defecto) cuando realmente son negativas, es decir el porcentaje de botellas que efectivamente pueden ser clasificadas como defectuosas.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

$$\text{Sensibilidad} = \frac{50}{50 + 0} = 1 * 100\% \approx 100\%$$

Especificidad: Capacidad o habilidad de detectar muestras positivas (sin defecto) cuando realmente son positivas, es decir el porcentaje de botellas sin defectos y que están correctamente clasificadas.

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

$$\text{Especificidad} = \frac{45}{45 + 5} = 0.9 * 100\% \approx 90\%$$

En general, se considera que la prueba diagnóstica tiene una validez aceptable si su sensibilidad y su especificidad son iguales o superiores a 0,80^{2,12,13}.

Tasa de Falsos Positivos: Probabilidad de que una muestra con valor verdadero positivo dé un valor negativo, es decir, en una botella sin defectos (buena) se identifique un defecto.

$$\text{Tasa FP} = \frac{FP}{FP + VN}$$

$$\text{Tasa FP} = \frac{5}{5 + 45} = 0.1 * 100\% \approx 10\%$$

Tasa de Falsos Negativos: Probabilidad de que una muestra con valor verdadero negativo dé un valor positivo, es decir, en una botella con defectos (mala) no se identifique un defecto.

$$\text{Tasa FN} = \frac{FN}{FN + VP} \quad \text{Tasa FN} = \frac{0}{0 + 50} = 0 * 100\% \approx 0\%$$

Valor global o eficiencia: Es la proporción total de botellas clasificadas correctamente.

$$VG = VP + VN$$

$$VG = 50 + 45 = 95 \text{ botellas}$$

$$\text{Total botellas} = \frac{VP + VN}{VP + FP + VN + FN}$$

$$\text{Total botellas} = \frac{50 + 45}{50 + 5 + 45 + 0} = 0.95 * 100\% \approx 95\%$$

Etapa 3.3: Aceptación del sistema por el usuario

Una vez realizadas las pruebas se procedió a tabular los datos de las encuestas realizadas tanto al administrador como a los estudiantes.

ANÁLISIS DE PRUEBA: ADMINISTRADOR

Encuesta dirigida a 4 Docentes del Área de Energía, las Industrias y los Recursos Naturales No Renovables de la Universidad Nacional de Loja.

1. ACCESIBILIDAD

Cuadro 1. Pregunta N°1, ¿Se presentó algún problema al ingresar al sistema con su nombre de usuario y contraseña?

Respuesta	Nº Encuestados	%
Si	0	0
No	4	100
Total	4	100

Interpretación de los resultados:

Al mencionar, si se presentó algún problema al ingresar al sistema con su nombre de usuario y contraseña, 4 encuestados que representan el 100%, manifestaron que no tuvieron problemas de ingreso con usuario y contraseña al sistema. Lo cual se concluye que el acceso al sistema es seguro y fácil de entender, para gestionar de modo eficiente la información.

2. FUNCIONALIDAD

Cuadro 2. Pregunta N°2, Marque con una X, indicando si tuvo algún problema al realizar alguna de las siguientes actividades:

Respuesta	Nº Encuestados	%
a) La información se presenta de manera clara y entendible	0	0
b) Los mensajes de información y error son adecuados y entendibles.	0	0
c) El empleo de directorios de archivos fue conveniente	0	0
d) Los resultados de la inspección son aceptables y confiables.	0	0
e) El tiempo y la forma en que se presenta el resultado de la clasificación al ejecutar una inspección son adecuados.	1	25
f) El sistema permite visualizar en tiempo real la ejecución de la inspección.	0	0
g) El sistema ejecuta correctamente la administración de Personal	0	0
h) El sistema ejecuta correctamente la administración de Usuarios.	0	0
i) El sistema ejecuta correctamente la consulta de Reportes Estadísticos.	0	0
j) Ninguna	3	75
Total	4	100

Interpretación de los resultados:

De acuerdo a los resultados obtenidos de la Pregunta N°2, se concluye que la ejecución de las actividades son concretas y específicas dando facilidad de entendimiento al usuario para un mejor desempeño con el sistema.

Cuadro 3. Pregunta N°3, Considera que el sistema le permitió realizar estas actividades de manera:

Respuesta	Nº Encuestados	%
Rápida, eficiente y confiable	3	75
Regular	1	25
Lenta, deficiente, Nada Confiable	0	0
Total	4	100

Interpretación de los resultados:

De acuerdo a los resultados obtenidos en la Pregunta N°3, indica que 3 encuestados que representan el 75% respondieron que las actividades se realizan de manera rápida, eficiente y confiable, mientras que el 25% manifestó que el sistema permite realizar las actividades de forma regular. Con esto se concluye que las actividades se comportan correctamente, de manera ágil y segura, proporcionando al usuario un sistema robusto y entendible en todo desempeño.

Cuadro 4. Pregunta N°4, ¿Cómo evalúa la funcionalidad de la integración de las herramientas: MatLab en el procesamiento digital de imágenes y Java en la realización de la interfaz gráfica amigable del sistema ICOLab?

Respuesta	Nº Encuestados	%
Muy eficaz	3	75
Eficaz	1	25
Ineficaz	0	0
Muy ineficaz	0	0
Total	4	100

Interpretación de los resultados:

En lo relacionado a la funcionalidad de la integración de las herramientas: MatLab y Java, la mayoría representando un 75% respondieron que el sistema es muy eficaz, mientras que el 25% manifestó que el sistema es eficaz. Por lo que se concluye que la comunicación entre estas herramientas es muy conveniente y viable, para enviar y recibir datos de operaciones realizadas con matrices que contienen mucha información y así brindar un sistema con mejor rendimiento.

3. DISEÑO Y PRESENTACIÓN

Cuadro 5. Pregunta N°5, Califique la interfaz gráfica de la aplicación ICOLab.

Respuesta	Nº Encuestados	%
Muy amigable y fácil de manejar	4	100
Amigable	0	0
Regular	0	0
Confusa	0	0
Total	4	100

Interpretación de los resultados:

Respecto a la interfaz gráfica de la aplicación ICOLab, el 100% de los encuestados, manifestaron que la interfaz gráfica es muy amigable y fácil de manejar. Lo que se concluye que las ventanas son agradables, manejables y claras, para la navegabilidad e interacción del usuario dentro del sistema.

Cuadro 6. Pregunta N°6, ¿Cómo considera la visualización de la imagen (objeto botella) a través de la cámara en tiempo real?

Respuesta	Nº Encuestados	%
Excelente	4	100
Buena	0	0
Regular	0	0
Mala	0	0
Total	4	100

Interpretación de los resultados:

En lo relacionado a la visualización de la imagen, el 100% de los encuestados, respondieron que consideran excelente la visualización de la imagen (objeto botella) a través de la cámara en tiempo real proporcionada por el sistema. Lo que se concluye que la reproducción del escenario del proceso de inspección es ideal para simular su comportamiento en el sistema.

ANÁLISIS DE PRUEBA: ESTUDIANTES

Encuesta dirigida a los estudiantes de acuerdo a la TABLA XXVIII: MUESTRA ESTRATIFICADA PROPORCIONAL DE LOS ESTUDIANTES

Cuadro 7. Pregunta N°1, ¿Está de acuerdo en que el sistema de control ICOLab como herramienta de apoyo y prácticas, ayuda en el proceso de enseñanza-aprendizaje?

Respuesta	Nº Encuestados	%
Muy de acuerdo	34	55
De acuerdo	28	45
En desacuerdo	0	0
Muy en desacuerdo	0	0
Total	62	100

Interpretación de los resultados:

Al mencionar, si el sistema de control ICOLab como herramienta de apoyo y prácticas, ayuda en el proceso de enseñanza-aprendizaje, 34 encuestados que representan un 55% manifestaron que están muy de acuerdo, otras 28 personas que representan un 45% respondieron que están de acuerdo. Lo que se concluye que el sistema aporta con nuevos conocimientos y técnicas de exploración y construcción de procesos de control, para brindar mayor información en creación de proyectos de dicho campo.

Cuadro 8. Pregunta N°2, Cómo evalúa la funcionalidad de la integración de las herramientas: MatLab en el procesamiento digital de imágenes y Java en la realización de la interfaz gráfica amigable del sistema ICOLab?

Respuesta	Nº Encuestados	%
Muy eficaz	37	60
Eficaz	25	40
Ineficaz	0	0
Muy ineficaz	0	0
Total	62	100

Interpretación de los resultados:

En lo relacionado a la funcionalidad de la integración de las herramientas: MatLab y Java, 37 encuestados que representan el 60% respondieron que el sistema es muy eficaz, 25 encuestados que representan el 40% manifestaron que el sistema es eficaz. Esto indica que la comunicación entre estas herramientas es muy conveniente y viable, para enviar y recibir datos de operaciones realizadas con matrices que contienen mucha información y así brindar un sistema con mejor rendimiento.

Cuadro 9. Pregunta N°3, En la interfaz gráfica principal, al seleccionar el módulo de 'Inspección', esta le parece:

Respuesta	Nº Encuestados	%
Intuitiva	13	21
Entendible	49	79
Confusa	0	0
Incoherente	0	0
Total	62	100

Interpretación de los resultados:

Respecto a la opinión que le merece la interfaz gráfica principal, al seleccionar el módulo de “Inspección” los encuestados respondieron: 13 personas que representan el 21% manifestaron que es Intuitiva, mientras que 49 personas que representan el 79% respondieron que es entendible, por lo que se deduce que el módulo se presenta de manera clara, legible para su interpretación y usabilidad.

Cuadro 10. Pregunta N°4, ¿Cómo considera la visualización de la imagen (objeto botella) a través de la cámara en tiempo real?

Respuesta	Nº Encuestados	%
Excelente	24	39
Buena	38	61
Regular	0	0
Mala	0	0
Total	62	100

Interpretación de los resultados:

En lo relacionado a la visualización de la imagen (objeto botella) a través de la cámara en tiempo real, 24 encuestados que representan el 39% manifestaron que es excelente, mientras que 38 personas que representan el 61% consideran que es buena. De estos resultados se puede concluir que la reproducción del escenario del proceso de inspección es ideal para simular su comportamiento en el sistema.

Cuadro 11. Pregunta N°5, La rapidez de respuesta del equipo con respecto a la funcionalidad de la interfaz gráfica del sistema es:

Respuesta	Nº Encuestados	%
Eficiente	54	87
Poco eficiente	7	11
Lenta	1	2
Muy lenta	0	0
Total	62	100

Interpretación de los resultados:

Respecto a la rapidez de respuesta del equipo con relación a la funcionalidad de la interfaz gráfica del sistema, los encuestados manifestaron: 54 personas que

representan el 87% consideran que es eficiente, otros 7 encuestados que representan el 11% respondieron que es poco eficiente y solamente una persona que representan el 2% manifestó que es lenta. Esto indica que las herramientas y recursos utilizados para la implementación del sistema se comportan de manera aceptable en cuanto al tiempo de respuesta, dando lugar a un funcionamiento óptimo del sistema.

Cuadro 12. Pregunta N°6, Califique el tiempo y la forma en que se presenta el resultado de clasificación del objeto botella al ejecutar una inspección

Respuesta	Nº Encuestados	%
Excelente	70	85
Buena	12	15
Regular	0	0
Mala	0	0
Total	62	100

Interpretación de los resultados:

Con relación al tiempo y la forma en que se presenta el resultado de clasificación del objeto botella al ejecutar una inspección, 37 encuestados que representan el 60% manifestaron que es excelente, mientras que 25 personas que representan el 40% respondieron que es buena, lo que deja apreciar que el resultado se genera en un tiempo de respuesta ideal y aceptable, de la misma manera en cuanto a la presentación visual y sonora de dicho resultado se concluye que es comprensible e idóneo, permitiendo una mejor orientación al usuario frente al buen rendimiento del sistema.

Cuadro 13. Pregunta N°7, ¿Cómo considera la configuración y personalización de todo el sistema ICOLab para el Laboratorio de Automatización y Control?

Respuesta	Nº Encuestados	%
Supera mis expectativas	8	12
Cumple con mis expectativas	53	85
No supera mis expectativas	1	2
Total	62	100

Interpretación de los resultados:

Referente a la configuración y personalización de todo el sistema ICOLab para el Laboratorio de Automatización y Control la mayoría de encuestados en número de 53 que representan el 85% consideran que cumple con sus expectativas, mientras que 8 personas que representan el 12% manifestaron que supera sus expectativas, y un encuestado respondió que no supera sus expectativas; lo que da entender que el sistema está diseñado e implementado de manera aplicable y útil para así motivar y facilitar su uso a las personas que emplean el laboratorio, brindando una mejor experiencia en este tipo de sistemas.

Cuadro 14. Pregunta N°8, Según su criterio, ¿Cómo calificaría a los resultados del Sistema de Control de Inspección de Objetos (botellas) defectuosos en la inspección y clasificación?

Respuesta	Nº Encuestados	%
Resultados aceptables	38	61
Resultados confiables	24	38
Resultados no aceptables	0	0
Resultados incoherentes	0	0
Otros	0	0
Total	62	100

Interpretación de los resultados:

En relación a, cómo calificaría a los resultados del Sistema de Control de Inspección de Objetos (botellas) defectuosos en la inspección y clasificación, los encuestados según su criterio respondieron: 38 personas que representan el 61% manifestaron que los resultados son aceptables, mientras que 24 encuestados que representan el 38% dijeron que los resultados son confiables. Lo que se concluye que dichos resultados facilitan una inferencia efectiva y segura, permitiendo así una excelente eficacia en el proceso de inspección y por ende al éxito del sistema.

G. DISCUSIÓN

1. DESARROLLO DE LA PROPUESTA ALTERNATIVA

Para llevar a cabo la realización del presente Trabajo de Titulación, se evaluaron cada uno de los objetivos específicos planteados, indicando la forma en que se fueron cumpliendo. Los cuales se detallan a continuación:

- ✓ **Objetivo Específico 1:** Analizar la información para determinar las variables que se van a controlar en el sistema.

En la etapa del Modelo de la organización (*ver Sección Revisión de Literatura - Capítulo IV apartado 4.3 Metodología CommonKADS*), se cumplió con el objetivo planteado, donde se logró la búsqueda, organización y valorización de la información recolectada permitiendo alcanzar los conocimientos necesarios, conjuntamente con las técnicas de investigación para llevar a cabo el proceso del Trabajo de Titulación, para esto se aplicó consultas bibliográficas en la web, entrevista a los expertos en control automático (*Ver Anexo 3*), expertos operadores de la planta embotelladora seleccionada (*Ver Anexo 2*) y observaciones de campo (*Ver Anexo 1*).

Las variables encontradas dentro de la dinámica del sistema a controlar, tienen una estructura SISO, es decir, una entrada y una salida, en donde la entrada es la imagen capturada de la botella mientras que la salida es la definición del estado de dicha botella. Esto permite tener una visión completa del sistema y de esta manera poder aplicar las teorías de control inteligentes.

- ✓ **Objetivo Específico 2:** Identificar y estudiar las técnicas de detección e inspección de objetos para realizar pruebas con imágenes.

El cumplimiento de este objetivo se dio a través de la consulta al experto en procesamiento digital de imágenes (*Ver Anexo 4*), suplementando con referencias bibliografías [46] [27] y proyectos de visión artificial documentados [15] [17] [25] [30], lo cual proporcionó una pauta sobre las técnicas aplicables en visión artificial y por ende en procesamiento digital de imágenes.

Mediante el análisis experimental profundo de toda la información correspondiente a las características y propiedades de las imágenes, se determinaron las técnicas óptimas para dicho procesamiento, entre las cuales se mencionan:

- ✓ Técnicas de segmentación basada en: detección de bordes (operadores canny, sobel y sobel), umbrales, morfología matemática (operaciones de apertura, cierre, dilatación y erosión) las cuales permiten arrojar medidas geométricas como el área y el perímetro de dicha segmentación.
- ✓ Técnicas del espacio del color para clasificación de tonalidades en RGB.
- ✓ Técnicas basadas en estadísticas, como la desviación estándar, valor sigma y promedios permitiendo un análisis numérico.

Estas técnicas de procesamiento realzan y modifican las imágenes para mejorar su apariencia y destacar aspectos de la información contenida en la misma, satisfaciendo el objetivo planteado, al momento de permitir una buena identificación de los objetos evaluados y por ende permitiendo su adecuada clasificación.

- ✓ **Objetivo Específico 3:** Aplicar nuevas tecnologías como los Sistemas Neuro-Difusos y herramientas de diseño como Toolbox de Procesamiento de Imágenes del software MATLAB R2009a.

El software MATLAB R2009a nos ofrece unas potentes herramientas de desarrollo en diferentes áreas del control automático y para el cumplimiento de este objetivo se utilizaron dos de sus herramientas: Image Processing Toolbox (Toolbox de procesamiento de imágenes) y Fuzzy Logic Toolbox (Toolbox de Logica Fuzzy o Difusa).

El toolbox de procesamiento de imágenes se desarrolló en la Fase 2: Desarrollo de Prototipos de la etapa 2.2.2.8 Modelo de diseño (*ver Sección Revisión de Literatura – Capítulo IV: METODOLOGÍAS PARA LA CONSTRUCCIÓN DE SISTEMAS BASADOS EN CONOCIMIENTO*), permitiendo aplicar las técnicas de tratamiento de imágenes mencionadas en el Objetivo Especifico 2, debido a que contiene las funciones adecuadas para el procesamiento eficiente y rápido de las mismas. Se parte del estudio de las funciones y posibilidades disponibles en la herramienta, definiendo las variables de entrada y salida que emplea cada función, extrayendo las características más significativas de acuerdo al patrón de las imágenes, comparando y

asociando resultados comunes y diferentes, para luego analizarlos y definir un conjunto de datos o vector que contenga la información relevante de dicha imagen analizada.

En la Fase 2: Desarrollo de Prototipos de la Etapa 2.3 Formalización de los conocimientos y definición de la arquitectura (*ver Sección Revisión de Literatura – Capítulo IV: METODOLOGÍAS PARA LA CONSTRUCCIÓN DE SISTEMAS BASADOS EN CONOCIMIENTO*), se utilizó el Toolbox de Lógica Fuzzy que es una colección de funciones incorporada en MATLAB que proporciona herramientas para crear y editar sistemas controladores de inferencia difusos (FIS) mediante herramientas gráficas, en este paso al momento de construir el FIS se seleccionó un conjunto de datos o vectores, extraídos anteriormente con las técnicas de procesamiento digital de imágenes, para luego convertirlos en datos de entrada / salida y de esta manera entrenar y validar nuestro controlador FIS, con estos datos se logra crear el modelo de estructura de la red tipo Sugeno con sus respectivas reglas difusas. El comportamiento del error al momento de entrenar FIS con los datos ingresados, resulto un error mínimo, esto demuestra la buena utilización de técnicas y procedimientos al momento de extraer el conjunto de datos o vectores que refleje una buena información.

- ✓ **Objetivo Específico 4:** Diseñar un algoritmo para la detección e inspección de objetos como botellas de vidrio transparente utilizando procesamiento digital de imágenes.

De acuerdo al objetivo específico 3, se diseña el algoritmo en scripts *.m* (*ver Sección Revisión de Literatura – apartado 5.1.1. Como hacer un programa en MATLAB y Capítulo V: HERRAMIENTAS DE PROGRAMACIÓN Y DESARROLLO*) codificando funciones propias y de los Toolboxes anteriormente mencionados de MATLAB. Con ello el proceso implementado consta de varias etapas: adquisición de imágenes de las botellas, reprocesamiento a las imágenes capturadas, selección de regiones defectuosas y extracción del conjunto de datos o vector para evaluar el controlador FIS, el cual nos devuelve el resultado de la inspección y clasificación.

Los resultados de este diseño permiten definir que el sistema de inspección clasifica de forma efectiva y robusta dos estados de botellas (defectuosa o no defectuosa).

Este algoritmo se diseñó con el objetivo de generar un conjunto de herramientas y demostraciones que reforzarán las explicaciones teóricas impartidas en las clases de las carreras afines, dichas herramientas se podrán utilizar como material de laboratorio que permitan a los alumnos profundizar en la materia de control automático e inteligencia artificial, o como complemento de proyectos futuros.

- ✓ **Objetivo Específico 5:** Realizar las pruebas necesarias para comprobar el buen funcionamiento del Sistema de Control.

En la Fase 3: Ejecución de la Construcción del Sistema Integrado de la Etapa 3.1: Requisitos y diseño de la integración con otros sistemas, el Sistema de control de inspección se integró con el hardware y toda la programación: algoritmo, interfaces y librerías, cumpliendo con los requerimientos exigidos.

Tras la finalización del sistema, se efectuaron las pruebas correspondientes de acuerdo a la *Etapa 3.2: Implementación y evaluación del Sistema Integrado*, con el objeto de verificar si el sistema se adapta o no a los requerimientos del administrador y de los usuarios (estudiantes) que harán uso del sistema.

La validación, funcionamiento y pruebas se llevaron a cabo mediante entrevistas (*Ver Anexos 10 y 11*) en el Laboratorio de Automatización y Control del Área de Energía, Las Industrias y los Recursos Naturales No Renovables de la Universidad Nacional de Loja, obteniendo resultados satisfactorios (*Etapa 3.3: Aceptación del sistema por el usuario*), tanto para el docente como para los estudiantes.

Teniendo en cuenta las prácticas, experiencias y resultados obtenidos, se amplía el conocimiento en estudiantes, docentes e investigadores, incentivando el desarrollo de nuevas alternativas que mejoren las condiciones de vida y trabajo de la región.

2. VALORACIÓN TÉCNICA – ECONÓMICA – AMBIENTAL

Luego de la construcción e implementación del sistema es indispensable realizar la valoración técnica, económica y ambiental del mismo, permitiendo así determinar que se desarrolló de manera satisfactoria y factible con todos los recursos necesarios, por tan razón se ha evidenciado que la aplicación cumple con los objetivos planteados al inicio del Trabajo de Titulación, contribuyendo al desarrollo de aplicaciones enfocadas a la teoría de control de procesos. Por lo tanto se establece a continuación un costo relativo y los beneficios que proporciona.

2.1. Valoración Técnica Económica

Los costos principales del proyecto son establecidos por los gastos de trabajo y desarrollo del sistema. La cuenta de gastos de trabajo para el tiempo gastado en el proyecto es por el ingeniero del conocimiento (autor del trabajo de titulación). La opción de recursos técnicos y tecnológicos para desarrollo del sistema experto está basada en la naturaleza del problema y las facilidades del computador y/o recursos de desarrollador.

Dentro de las herramientas empleadas desde el punto de vista tecnológico, es fundamental el uso del software matemático MATLAB, permitiendo utilizar su potente procesamiento de datos con matrices para cumplir con las exigencias de rapidez y eficiencia en el funcionamiento del sistema, al mismo tiempo proporciona modelos que simulan el comportamiento humano mediante la construcción y configuración del controlador difuso para el sistema de inspección. Por otro lado, la interfaz gráfica del sistema terminado es amigable y agradable al usuario, permitiendo una interacción de forma ágil y entendible, para cumplir con las características mencionadas, dicha interfaz se configuró en el IDE NetBeans con el lenguaje JAVA, debido a su familiaridad con el desarrollador del sistema.

De la igual importancia, se requiere una base de datos en MySql para administrar y gestionar los datos de ingreso/salida y los generados por el sistema.

Dentro de los equipos de implementación, se utiliza un banco de pruebas debidamente diseñado y construido para controlar diferentes factores que intervienen proceso de inspección y clasificación, integrando además una cámara para tener la percepción del ambiente. El banco de pruebas es acoplado a una banda transportadora activada por

un motor, situada en el Laboratorio de Automatización y Control Industrial del AEIRNNR, teniendo disponibilidad de uso por parte de las autoridades pertinentes. Por consiguiente esto permite representar el proceso del mundo real y efectuar el proceso de inspección y clasificación en tiempo real con resultados de calidad.

Desde el ámbito económico para el desarrollo e implantación se contó con los suficientes recursos humanos y económicos que permitieron el éxito del Trabajo de Titulación cumpliendo con sus objetivos. A razón, se utiliza software de libre distribución en su mayoría, permitiendo acceso de información a estas herramientas, no obstante se realizó un pago por la licencia de la herramienta MatLab y sus Toolboxes utilizados.

Debido a la adquisición de recursos materiales para la construcción del banco de pruebas, de gastos de trabajo del desarrollador y de la licencia pagada, los costos se incrementaron considerablemente, pero sin embargo los resultados y beneficios en generación de nuevos conocimientos ameritan su retribución. Cabe recalcar que no existió mayor inconveniente ya que estos recursos han sido adquiridos con inversión del desarrollador del proyecto de Trabajo de Titulación. A continuación se detalla los recursos utilizados para la elaboración del proyecto.

RECURSOS HUMANOS:

TABLA XXXI: RECURSOS HUMANOS

Descripción	Cantidad	Nº de Horas	Valor Unitario	Valor Total
<ul style="list-style-type: none"> Investigador del proyecto de Trabajo de Titulación ✓ Víctor Espinoza Ochoa	1	1920	\$ 1.50	\$ 2880.00
<ul style="list-style-type: none"> Director de Trabajo de Titulación 	1	--	--	--
<ul style="list-style-type: none"> Asesoramiento de Profesionales Vinculados con el Tema ✓ Experto en control automático	1	--	--	--
<ul style="list-style-type: none"> Experto en procesamiento digital de imágenes ✓ Experto operador de la Planta Embotelladora	1	--	--	--
			SUBTOTAL:	\$ 2880.00

RECURSOS TÉCNICOS Y TECNOLÓGICOS:

TABLA XXXII: RECURSOS TÉCNICOS Y TECNOLÓGICOS

Descripción	Cantidad	Valor Unitario	Valor Total
HARDWARE			
• Computador HP Pentium Dual Core Inside.	1	\$ 1200,00	\$ 1200,00
• Impresora Lexmark z600	1	\$ 60,00	\$ 60,00
• Cámara WebCam	1	\$ 20,00	\$ 20,00
• Memoria flash 4 GB	1	\$ 10,00	\$ 10,00
SOFTWARE			
• Plataforma Java	1	--	--
• Java Media Framework JMF2.1.1e	1	--	--
• NetBeans 6.01	1	--	--
• MySQL 5.0	1	--	--
• SQLyog v4.0	1	--	--
• MatLab R2009a	1	\$ 189,00	\$ 189,00
• Enterprise Architect 4.10.736	1	--	--
• Microsoft Project 2003	1	\$ 289,99	\$ 289,99
COMUNICACIONES			
• Internet	1000 horas	\$ 0,80	\$ 800,00
EQUIPOS DE IMPLEMENTACIÓN			
• Banda transportadora de Laboratorio de Automatización y Control	1	--	--
SUBTOTAL:			\$ 2568,99

RECURSOS MATERIALES:

TABLA XXXIII: RECURSOS MATERIALES

Descripción	Cantidad	Valor Unitario	Valor Total
• Resma de Hojas A4	5	\$ 5,00	\$ 25,00
• Cartucho	2	\$ 20,00	\$ 40,00
• Empastados	4	\$ 10,00	\$ 40,00
• Encuadernado (Manuales)	4	\$ 25,00	\$ 100,00
• Caja de Clips	1	\$ 0,50	\$ 0,50
• Capetas (Perfiles)	4	\$ 0,35	\$ 1,40
• Perforadora	1	\$ 2,00	\$ 2,00

• Grapadora	1	\$ 3,00	\$ 3,00
• Borrador	2	\$ 0,25	\$ 0,50
• Lápiz	4	\$ 0,25	\$ 1,00
• CDs (Diseño e impresión)	4	\$ 5,00	\$ 15,00
• Anillados	6	\$ 1,00	\$ 6,00
• Fotocopias	500	\$ 0,02	\$ 10,00
• Plancha de Folio Blanco 3mm	1	\$ 9,80	\$ 9,80
• Focos Ahorradores 130	2	\$ 2,00	4,00
• Tornillos y bisagras	--	\$ 2,00	2,00
• Espuma Flex	5	\$ 3,00	15,00
• Tubo de Silicona	1	\$ 3,80	\$ 3,80
		SUBTOTAL:	\$ 279.00

TABLA XXXIV: RESUMEN DE RECURSOS

RECURSOS	SUBTOTALES
Recursos Humanos	\$ 2880,00
Recursos Técnicos y Tecnológicos	\$ 2568,99
Recursos Materiales	\$ 279,00
Subtotal	\$ 5727,99
Imprevistos 10 %	\$ 572.79
TOTAL:	\$ 6300.78

2.2. Valoración Ambiental

Ya que este proyecto se orienta al control automático en el campo industrial, y sobre todo al manejo de envases de vidrio encontramos que, en la embotelladora estudiada (*Ver Anexo 1*), tiene entre sus políticas ambientales la reutilización y reciclado de botellas de vidrio, para de esta manera tener la oportunidad de disminuir costos, aumentando su competitividad, sostenibilidad y mejorando su imagen frente a la administración, la sociedad en general y el ambiente.

Con este sistema de control se coadyuva a estas políticas ambientales en el uso de objetos de vidrio (botellas). Además con todo esto se establece que el presente sistema tendrá un mínimo impacto ambiental, debido a que la tecnología y recursos utilizados en el mismo no afectarán en mayor medida a nuestro entorno y por ende al medio ambiente.

H. CONCLUSIONES

- La aplicación de las metodologías *IDEAL* y *CommonKADS-RT*, facilitan la construcción del conocimiento en su ambiente o dominio ya establecido, proporcionando una documentación entendible y ordenada, por ende reducir tiempo y costo en el desarrollo del sistema.
- El uso del software matemático *MATLAB R2009a* y sus *Toolboxes* permitió desarrollar el procesamiento de datos, de forma rápida y potente para el sistema, efectuando una conexión eficiente entre *MATLAB* y la interfaz gráfica realizada en *JAVA* mediante la librería *JMatLink.dll* facilitando una comunicación dinámica para las tareas de control del sistema.
- De acuerdo a los resultados obtenidos, la buena iluminación en las imágenes eliminó el ruido, permitiendo trabajar en condiciones óptimas.
- Los resultados del Trabajo de Titulación, permiten determinar, que un sistema de inspección y clasificación basado en técnicas de procesamiento de imágenes y red neuro-difusa inspecciona y clasifica de forma efectiva y robusta dos estados de botellas (defectuosa o no defectuosa) entre 2 y 2,4 segundos.
- El Sistema de control de inspección y clasificación fue validado correctamente por los estudiantes y docentes del AEIRNNR, despertando interés y aceptación en este tipo de teorías inteligentes aplicadas en el mismo. De acuerdo a los resultados obtenidos y tomando en cuenta las pruebas, se determinó que la eficiencia global de la inspección es del 95% con validez aceptable, permitiendo dar soluciones no solo en el ámbito académico, si no a las empresas industriales y a la sociedad en general en el área de control de procesos.

I. RECOMENDACIONES

- Para el desarrollo de sistemas de control, se recomienda el empleo de las metodologías *IDEAL* y *CommonKADS-RT*, debido a que estos sistemas son basados en conocimiento en tiempo real, de modo que estas metodologías ayuden a la extracción de sus características inteligentes.
- Aprovechar la agilidad que *MATLAB* proporciona en diversos campos del control de sistemas, teniendo en cuenta que posee una entendible programación orientada a objetos, a su posibilidad de poder comunicarse con otros lenguajes de programación.
- El algoritmo desarrollado en este proyecto permite su modificación para lograr una inspección y/o reconocimiento en diversos campos del control, seguridad y demás aplicaciones similares de visión artificial.
- Para proyectos futuros sería positivo implementar la activación un brazo electro-neumático, utilizando la variable o resultado de salida del sistema; ya que en este proyecto dicho resultado solo se puede apreciar de manera visual y sonora en la aplicación.
- El controlador Neuro-difuso construido presenta gran robustez al momento de controlar las variables del proceso de inspección, sin embargo como trabajo futuro sería factible incorporar a los consecuentes de salida, valores que indiquen alguna duda al momento de inspeccionar un objeto extraño no debidamente entrenado en la red.

J. BIBLIOGRAFÍA

- [1] Javier Aracil and Francisco Gordillo, *Dinámica de Sistemas*, Primera ed. Madrid, España: Alianza Editorial, 1997.
- [2] Peter Senge, *La Quinta Disciplina en la práctica (The Fifth Discipline: The Art and Practice of the Learning Organization)*, Primera ed. Barcelona, España: Ediciones Granica, 1998.
- [3] José Cotero Ochoa. (2000) Capítulo 2: Control clásico y Control Inteligente. [Online].
<http://www.mde.iteso.mx/titulacion/estudio%20de%20caso/ejemplos/marco%20teorico%20en%20progreso.pdf>
- [4] Paul Erick Méndez Monroy, "Desarrollo de un Observador y Controlador Neuro Difuso para un Sistema Interconectado," Facultad de Ingeniería, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, Ciudad de México, Tesis Profesional 2007.
- [5] B. Martín Del Río and A. S. Molina, *Redes Neuronales y Sistemas Neurodifusos*. México: Alfaomega, 2002.
- [6] José Carlos Chahuara, "Control neuro-difuso aplicado a una Grúa Torre," Facultad de Ingeniería Electrónica, Universidad Nacional Mayor de San Marcos - UNMSM, Lima, Tesis 2005.
- [7] Araceli Grande Meza, "Observadores Difusos y Control Adaptable Difuso Basado en Observadores," DEPARTAMENTO DE CONTROL AUTOMÁTICO, INSTITUTO POLITÉCNICO NACIONAL, México, D.F., Tesis 2003.
- [8] Ailén Sabadí Hernández, "La Lógica Difusa. Características y Aplicaciones," Universidad de las Ciencias Informáticas, La Habana - Cuba, Trabajo de Información 2005.

- [9] Hugo A. Banda Gamboa, "INTELIGENCIA ARTIFICIAL - PRINCIPIOS Y APLICACIONES," Departamento de Informática y Ciencias de Computación, Escuela Politécnica Nacional, Quito, Obra de Investigación 2012.
- [10] Díaz Jhon and Muñoz Freddy., "Comparación Entre Un Controlador Difuso Y Un Controlador PID Para Un Motor Universal," *ENGI Revista Electrónica De La Facultad De Ingeniería - UDEC*, vol. 1, no. 2, p. 2, Julio 2013.
- [11] Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, Segunda ed. Padstow, Cornwall, England: John Wiley & Sons, Ltd, 2004.
- [12] Sujit Nath Pant and Keith E. Holbert. (2004, Agosto) Fuzzy Logic in Decision Making and Signal Processing. [Online]. <http://enpub.fulton.asu.edu/powerzone/fuzzylogic/chapter%206/frame6.htm>
- [13] Perla María Saldivar Castillo, "Control de un Brazo Mecánico mediante Técnicas de Computación Suave," Centro de Investigación y Desarrollo de Tecnología Digital, INSTITUTO POLITÉCNICO NACIONAL, Tijuana - México, Tesis de Master 2007.
- [14] J-S. Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665-685, Mayo/Junio 1993.
- [15] Ana Domènech Martorell, "Reconocimiento Gestual a través de Webcam," Escuela Técnica Superior Ingeniería - ICAI, Universidad Pontificia Comillas, Madrid, Proyecto Fin de Carrera 2009.
- [16] Steven Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, Segunda ed. San Diego, California: Technical, 1999.
- [17] Carmen Virginia Gámez Jiménez, "Diseño y Desarrollo de un Sistema de Reconocimiento de Caras," Escuela Politécnica Superior, Universidad Carlos III de Madrid, Madrid, Proyecto Fin de Carrera 2009.
- [18] Jesús Vega Uribe and Marlon Reyes Figueroa, "Transformaciones lineales y no

- lineales para espacios de color en procesamiento digital de imágenes," *Revista internacional de métodos numéricos para cálculo y diseño en ingeniería*, vol. 22, no. 3, p. 232, Diciembre 2006, <http://upcommons.upc.edu/revistes/bitstream/2099/4751/1/RR223A.pdf>.
- [19] Genaro Gutiérrez Becerril. (2008, Junio) Detección de Fallas Estructurales en Envases de Vidrio no Cilíndricos y Grabados Utilizando Morfología Matemática y Wavelet. [Online]. http://itzamna.bnct.ipn.mx:8080/dspace/bitstream/123456789/5911/1/1388_Centro%20de%20Investigaci%C3%B3n%20en%20ComputaciOn%20%28CIC%29tesis_Febrero_2010_173707036.pdf
- [20] Instituto de Tecnologías Educativas, Ministerio de Educación de España. GIMP, aplicaciones didácticas - La imagen digital. [Online]. http://www.ite.educacion.es/formacion/materiales/86/cd/pdf/m2_caracteristicas_de_la_imagen_digital.pdf
- [21] Bienvido Trillo and Muñoz Pedro, "Iniciacion a GIMP," Centro del Profesorado de Orcera, Granada, Curso 2006.
- [22] Foto-Mosaik-Edda. (2000, Junio) Definición, tamaño de imagen, DPI y PPI. [Online]. http://www.fmedda.com/es/article/dpi_ppi
- [23] MiMoriarty. (2013, Marzo) Las unidades de medida del diseño gráfico. [Online]. <http://mimoriarty.wordpress.com/2013/03/26/las-unidades-de-medida-del-diseno-grafico/>
- [24] Carlos Alfonso Roa Bravo, "Métodos de proyección multivariada de imágenes aplicados al control estadístico de procesos," Venezuela, 2004.
- [25] Elías García Santillán, "Detección y clasificación de objetos dentro de un salón de clases empleando técnicas de procesamiento digital de imágenes," Universidad Autónoma Metropolitana, México D.F., Tesis de Maestría ISBN: 978-960-6766-85-5, 2008.

- [26] Silvia Satorres Martínez, "Detección de bordes en una imagen," Departamento de Ingeniería electrónica, Telecomunicación y Automática., Universidad de Jaén - Área de Ingeniería de Sistemas y Automática, Curso Práctica 3ª 2005/2006.
- [27] Roberto Rodríguez and Juan Sossa, *Procesamiento y Análisis Digital de IMÁGENES.*: Alfaomega Ra-Ma, 2012.
- [28] Diego González Aguilera, "Procesamiento de Imágenes," Departamento de Ingeniería Cartográfica y del Terreno - Escuela Politécnica Superior de Ávila, Universidad de Salamanca, Salamanca - España, Cursos Enseñanzas-Técnicas 2008.
- [29] Oscar Fernandez Asuncion. (2009, Abril) DETECCION DE BORDES: Algoritmo de Canny. [Online]. <http://oefa.blogspot.com/2009/04/deteccion-de-bordes-algoritmo-de-canny.html>
- [30] Dina Patricia Jarrín Zambrano. (2009, Agosto) Universidad Politécnica Nacional - Facultad de Ingeniería Eléctrica y Electrónica. [Online]. <http://dSPACE.epn.edu.ec/bitstream/15000/8792/3/T11363%20CAP%20III.pdf>
- [31] José Domingo Carrillo Verdun, "Metodología para el Desarrollo De Sistemas Expertos," UNIVERSIDAD POLITÉCNICA DE MADRID, Madrid, España, Tesis Doctoral 5, 1987.
- [32] Miguel Jiménez and Pérez Claudia, "Detección, Diagnóstico y Tratamiento de Tuberculosis Pulmonar en el Centro de Salud Túpac Amaru, asistido por un Sistema Experto.," ASAMBLEA NACIONAL DE RECTORES, Perú, Tesis 2007.
- [33] Francisco Marcelo Rizzi, "Sistema Experto Asistente de Requerimientos," Universidad Politécnica de Madrid / Instituto Tecnológico de Buenos Aires, Buenos Aires - Argentina, Tesis de Magister 2001.
- [34] A Gómez, N Juristo, C Montes, and J. Pazos, "Ingeniería del Conocimiento," 1997.
- [35] Pedro Salcedo Lagos, "CommonKADS y el Lenguaje de Modelado Unificado -



- UML," Facultad de Educación, Universidad de Concepción, Concepción - Chile, Proyecto de Investigación 2002.
- [36] Mónica Henao Cálad, "CommonKADS una buena herramienta para la Gerencia del Conocimiento," *Revista Universidad EAFIT*, vol. 36., no. 118, p. 11, Abril-Mayo - Junio 2000.
- [37] Jorge Roberto Jova Rodríguez, "Metodología CommonKADS en el desarrollo de sistemas expertos," *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 10, no. 5, pp. 7-8, Octubre 2012.
- [38] Mónica Henao Cálad, "CommonKADS-RT: Una Metodología para el Desarrollo de Sistemas Basados en el Conocimiento de Tiempo Real ," Universidad Politécnica de Valencia, Valencia, España, Tesis Doctoral 2001.
- [39] Julián Gutiérrez Santos, "Integración de librerías ITK en MATLAB mediante objetos MEX para la segmentación del hígado procedente de imágenes de CT en entornos multiplataforma," Departamento De Electrónica, Automática e Informática Industrial, Universidad Politécnica de Madrid, Madrid - España, Trabajo de Titulación GVA-ELAI-UPM@PFC0092-2009, 2009. [Online]. http://www.elai.upm.es/webantigua/spain/Investiga/GCII/personal/jgutierrez/julian_data/PFC_Julian.zip
- [40] MathWorks ®, Inc. (2012, Enero) Descripción general. [Online]. <http://www.mathworks.es/products/matlab/index.html>
- [41] MathWorks ®, Inc. (2004) MATLAB. [Online]. <http://www.mathworks.es/products/matlab/description1.html>
- [42] Alexeis Companioni, *La programación en Matlab*. La Habana-Cuba, 2000.
- [43] Sixto Romero and Francisco Moreno. (2008, Marzo) El Asistente Matemático MatLab. [Online]. http://www.uhu.es/sixto.romero/MATLAB_MASTER.pdf
- [44] Rafael Palacios, "Curso rápido de Matlab," in *Palacios, Rafael.*, 2004, p. 41.

- [45] MathWorks ®, Inc. (2011) MATLAB Toolboxes. [Online].
<http://www.mathworks.es/help/>
- [46] Eduardo Laorden Fiter, "Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesamiento digital de imágenes de MATLAB," Escuela Universitaria de Ingeniería Técnica de Telecomunicación, Universidad Politécnica de Madrid, Madrid, Proyecto de Fin de Carrera 2012.
- [47] Jang J.-S. Roger and Ned Gulley. (1997, Abril) MATLAB: Fuzzy Logic Toolbox. [Online].
http://www.mty.itesm.mx/dtie/centros/csi/materias/ia4002-1/docs/Fuzzy_Toolbox.pdf
- [48] The MathWorks, Inc., "Fuzzy Logic Toolbox For Use with MATLAB," Natick, Massachusetts, Guía de Usuario Julio 2002. [Online].
<http://www.mathworks.com/products/fuzzy-logic/>
- [49] Inc Mathworks. (2012) Constraints of anfis. [Online].
<http://www.mathworks.es/help/toolbox/fuzzy/fp715dup12.html#FP328A>
- [50] David L. Martínez. (2010, Enero) Comparación de Frameworks de Desarrollo de Páginas Web con AJAX. [Online].
<http://www.sicuma.uma.es/es/formacion/doctoradoargentina/independientes/argentina08/LaRed-Eugenia/index.html>
- [51] Oracle Corporation. (2013) Bienvenido a NetBeans y www.netbeans.org. [Online].
https://netbeans.org/index_es.html
- [52] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha, *The Java™ Language Specification*, Tercera ed. Santa Clara, California: Sun Microsystems, Inc., 2005, [Consulta:05-11-2011], <http://docs.oracle.com/javase/specs/jls/se5.0/jls3.pdf>.
- [53] Joaquin Gracia. (2003) UML: Diagramas UML. [Online].
<http://www.ingenierosoftware.com/analisisydiseno/uml.php>
- [54] Sparx Systems Pty Ltd. (2007) Enterprise Architect - Herramienta de diseño UML. [Online]. <http://www.sparxsystems.com.ar/products/ea.html>



- [55] Stefan Müller. (2008, Junio) JMatLink: MATLAB Java Classes. [Online]. <http://www.held-mueller.de/JMatLink/>
- [56] Sun Microsystems, Inc. (2003, Mayo) Documentation. [Online]. <http://www.oracle.com/technetwork/java/javase/documentation/index-135173.html>
- [57] David Gilbert. (2003, Julio) Reference Guide. [Online]. <http://reporting.pentaho.com>
- [58] BI-Spain.com. (2006, Enero) Pentaho compra JFreeReport para completar su oferta de reporting en código abierto. [Online]. <http://www.bi-spain.com/articulo/19665/open-source-software-libre/pentaho-compra-jfreereport-para-completar-su-oferta-de-reporting-en-codigo-abierto>
- [59] David Gilbert. (2011) JFreeChart. [Online]. <http://www.jfree.org/jfreechart/>
- [60] JBoss Community. (2012, Enero) Hibernate. [Online]. <http://hibernate.bluemars.net/>
- [61] Gavin King, Christian Bauer, Max Rydahl Andersen, Emmanuel Bernard, and Steve Ebersole. (2012, Febrero) Hibernate - Tutorial in PDF. [Online]. http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/pdf/hibernate_reference.pdf
- [62] Publicación Tschernig ®. (2010, Diciembre) Tecnología al Instante. [Online]. http://www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=1061
- [63] Electrónica Básica, Copyright © 2013. (2013, Enero) Electronica-Basica.com. [Online]. <http://www.electronica-basica.com/web-cam.html>
- [64] Genius, KYE Systems Corp. (2012, Mayo) Genius Productos. [Online]. <http://www.geniusnet.com/wSite/ct?xltem=19463&ctNode=1303>
- [65] Javier Gómez Ferri and Arantxa Muñoz Grau, "Técnicas cualitativas de investigación social," Departamento de Sociología y Antropología social, Universidad de Valencia, Valencia - España, Taller 1, 2011.

- [66] Frank Morales. (2012, Octubre) Cooperacion en Red Euro Americana para el Desarrollo Sostenible (Creadess). [Online]. <http://www.creadess.org/index.php/informate/de-interes/temas-de-interes/17300-conozca-3-tipos-de-investigacion-descriptiva-exploratoria-y-explicativa>
- [67] Atenea Alonso and Lorena y otros. García, "MÉTODOS DE INVESTIGACIÓN DE ENFOQUE EXPERIMENTAL," Universidad Autónoma de Madrid, Madrid - España, Ensayo 5, 2010.
- [68] Mario Orlando Suárez Ijujes and Fausto Amilcar Tapia Zambrano, *Interaprendizaje de Estadística Básica*, Primera Edición ed. Ibarra, Ecuador: Universidad Técnica del Norte, 2012, <http://repositorio.utn.edu.ec/handle/123456789/2341>.



κ. ANEXOS

Anexo 1: Observación sistemática en la embotelladora

Los principales métodos utilizados para la recopilación del sistema de control fueron la observación.

VALLTO AGROINDUSTRIA CIA. LTDA.

Fábrica: Hacienda “La Palmira”



Figura 270: VALLTO AGROINDUSTRIA CIA. LTDA.



Figura 271: Bandeja de Reciclado



Figura 272: Proceso de Lavado / Secado de Botellas

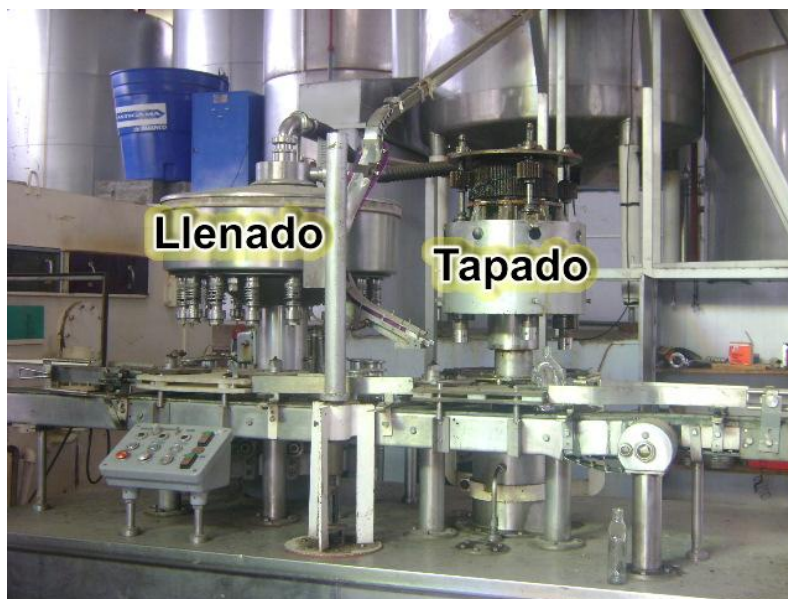


Figura 273: Proceso Llenado y Tapado de las Botellas



Figura 274: Producto Final



Figura 275: Empacado Final del Producto

Anexo 2: Entrevista aplicada al Experto Operador de la planta Embotelladora

Entrevista Realizada al Experto Operador de la Planta Embotelladora

"Vallto Agroindustria Cía. Ltda."

Sr. César Vallejo Rojas

1. ¿Qué productos se elabora en la planta embotelladora?

Aguardiente de caña, llamado "Cañita"

2. ¿En cuántas presentaciones viene este producto?

En 375 c.c y 750 c.c pero en la actualidad solo producimos el de 375 c.c

3. ¿De qué depende su producción?

Cantidad y calidad de producto

4. ¿Cuál es el proceso de envasado para llegar al producto final?

Reciclaje, lavado y secado, llenado, sellado y tapado, Etiquetado, embalaje y almacenamiento

5. ¿De qué forma se da este proceso de envasado?

- Manual
 Mecánico
 Automático

6. ¿En el proceso de envasado del producto, se necesita alguna inspección?

Si, al momento de ingresar las botellas recicladas verificamos que no estén figurados con picos rotos o en el resto de la botella.

Otra inspección que se realiza es a la salida de la botella de laadora/secadora para ver si quedaron algunos residuos o defectos dentro o fuera de la botella.

Y última es al momento de embalaje, en el cual se realiza si están bien selladas y tapadas.



7. ¿De qué forma se realizan inspecciones mencionadas?

De forma manual

8. ¿Cuáles son los tipos de defectos o residuos que quedan a la salida de la lavadora/secadora al momento de inspeccionar?

Anillos de las tapas, plástico de seguridad de las tapas, colillas de cigarrillo, etiquetas e impurezas de suciedad

9. ¿Le serviría un sistema automático de control de inspección y clasificación en este proceso?

Sí, por que nos ahorraría los de estar revisando cada botella de manera visual y así reducir tiempo en la producción

10. ¿Qué requisitos usted agregaría para el sistema de control de inspección?

Que sea rápido

Que el resultado de la inspección sea monitoreado por medio de una pantalla visual y con sonido

Que se puedan revisar las botellas que han sido inspeccionadas

Que pudiera sacar reportes para ver cuántas botellas han sido defectuosas y cuáles no

GRACIAS POR SU COLABORACIÓN

Anexo 3: Entrevista aplicada al Experto en Control Automático

Entrevista Realizada al Experto en Control Automático

Julio Cuenca

Ing. Electromecánico

Máster en Informática Industrial y Automatización

Docente de la Carrera de Ingeniería Electromecánica de la UNL

1. ¿Cuáles son los factores para controlar un sistema ?

Primeramente que exista la necesidad de implementar el control
Para inhibir o impedir conductas indeseables del sistema
Incrementar Calidad y Cantidad
Disminuir tiempo y costo
Ayudar a personas en factores de riesgo, etc.

2. ¿Cuáles serían los pasos para controlar un sistema ?

Identificar el tipo de sistema, si son: si es mecánico, eléctrico, hidráulico
híbrido, térmico, artificial es decir una imitación de tipo biológico, etc.
Luego determinar el comportamiento del sistema, ósea en que
implica: si es lento o rápido, si es complejo, analizar si
es de una variable o multivariable, Si no sabe se puede averiguar
con algún especialista en el campo.

3. ¿Es posible controlar todos los sistemas en procesos industriales?

Si su dinámica es compleja o no lineal, no se puede controlar
completamente con controladores convencionales, entonces
estos sistemas de control requieren un conocimiento muy
preciso del proceso sobre el que se aplican.

4. ¿Cuáles son estos controladores convencionales?

Controladores PDI (Proporcional-Integral-Directivo) PI,
para diseñar estos controladores se utilizan ecuaciones
y modelos matemáticos los cuales representan los
valores de los parámetros del sistema.

5. ¿Cómo se podrían controlar sistemas complejos como la inspección de objetos en procesos industriales?

Con controladores de algoritmos inteligentes como:
redes neuronales, lógica difusa, algoritmos genéticos,
control productivo y control neuro-difuso

6. ¿De qué manera funcionan estos controladores inteligentes?

Utilizan el conocimiento experto en vez de ecuaciones
diferenciales para describir un sistema, este conocimiento
puede expresarse de manera natural, empleando las variables

7. ¿Ha aplicado estos controladores inteligentes en sistemas no lineales?

Si, en trabajos como: Péndulo Invertido y control de
Tonques no lineales.

8. ¿Qué resultados han conseguido en los trabajos anteriores?

Resultados eficientes y confiables y una de las
ventajas fue la reducción de tiempo en cuanto a su implementación
y sintonización debido a su naturaleza

9. ¿Qué desventaja tienen los controladores inteligentes?

Si un problema tiene solución mediante un modelo
matemático, existe la posibilidad de obtener peores resultados
usando lógica Difusa.

En las redes neuronales se aprecia un mayor tiempo de
aprendizaje para obtener mejores resultados en la salida.

En lógica difusa es la dificultad de interpretación de valores
difusos, es decir mala interpretación de las variables y sus valores.

Anexo 4: Entrevista aplicada al Experto en Procesamiento Digital de Imágenes

Entrevista Realizada al Experto en Procesamiento Digital de Imágenes

Pedro Pablo Moreno Zúñiga

Ing. de Informática – Universidad Pontificia de Salamanca

Máster en Seguridad Informática

Proyectos en Visión Artificial

1. ¿ Que es el Procesamiento Digital de Imágenes (PDI)?

Este procesamiento es una rama de la visión artificial y tiene como
objetivo mejorar el aspecto de las imágenes y hacer más evidentes ciertas
detalles que se deseará hacer notar en ellas para la búsqueda de información.
Antes de esto las imágenes deben estar digitalizadas fotográficamente,
mediante cámaras o electrónicamente mediante monitores de televisión.

2. ¿ En que campos o aplicaciones es útil el PDI?

Con el desarrollo de la tecnología en hardware y software en un sinnúmero
de aplicaciones como en: Seguridad, Grabado, lectura de documentos, Cartografía,
Crecimiento de bacterias, Modelación pedagógica, Imágenes médicas, Navegación
autónoma, Reconocimiento de objetos, Astronomía, Seguridad fotográfica, Mapas
climatológicos, Impresión de buletines digitales, transferencias y comparación
de documentos, en su mayoría muchos campos.

3. ¿ Podría mencionar algunas técnicas para el PDI?

* Correcciones de color (Ajuste de brillo y Contraste)
* Segmentación por espacios del color RGB, morfología matemática
(dilatación, Erosión), histograma, umbral, detección de bordes, esta segmentación
se trata de la descripción de una región que se sobresale al fondo de la imagen.
* Análisis estadístico de los datos que cubren la imagen, como promedio,
Sumatoria, Varianza estándar
La morfología matemática implica transformaciones del segmento
como la dilatación Erosión

4. ¿Cuáles son los aspectos a estudiar en una imagen?

Brillo, Tonalidad y Contraste

5. ¿Cuáles serían las etapas para llevar a cabo un PDI?

Generalmente hay 3 o 4 etapas que se puede aplicar
Pre-procesado: Esto consiste en eliminar ruido de la imagen, resaltar áreas de que nos interesan, convertirlas a escala de grises o como los convergen, en si todo esto nos ayuda a resaltar detalles.
Procesado: Se lo realiza con las técnicas de segmentación anteriormente mencionadas en las que se detectan cambios abruptos en tonalidad.
Extracción de características: Del paso anterior se pueden seleccionar los segmentos de modo que se extraiga información por ejemplo: Tamaño, forma es decir características geométricas, etc. Además se pueden extraer valores estadísticos del conjunto de datos de la imagen.
Por último se pueden implementar un reconocimiento mediante algoritmos para identificar patrones de las imágenes.

6. ¿Que programas (software) trabajan y aplican técnicas para realizar PDI?

Matlab, LabView, Matlab Imaging Library, Scion Image, Calico Software, Eidas Imagine, Aptelon Imaging.

7. ¿Para usted cuál es el software más recomendable y eficiente?

Bueno para algunas aplicaciones he utilizado Matlab, ya que los algoritmos desarrollados pueden transferirse a automatismos programables o PLC's para ejecutar algún control en tiempo real dependiendo de la aplicación desarrollada. Además de que sus funciones vienen documentadas con la ayuda del programa y se nos facilita su implantación y desarrollo.

8. ¿Sería factible implementar entonces el PDI en procesos como en la inspección de objetos mediante cámaras en el campo industrial?

Si sería factible, pero primero se determinaría las condiciones de trabajo de la aplicación, las variables que intervienen, los requerimientos y requisitos, además del hardware y software se utilizarse

9. ¿Me recomendaría alguna bibliografía sobre PDI?

- Procesamiento Digital de imágenes con Matlab y Simulink de los autores Erik Cuevas Daniel Zaldívar y Marco Pérez Cisneros.
- Procesamiento y Análisis Digital de imágenes de los autores Roberto Rodríguez y Juan Sosa.

Anexo 5: Conjunto de Datos para el entrenamiento del FIS

Datos Entrenamiento 1

```

%Cargar los datos de entrenamiento 1
Limpia1= [26.235668718692285 3.835837646328803e+02 0 22, 1];
Limpia11= [25.901730530251110 3.712607804393007e+02 0 9, 1];
Limpia2= [26.883302884235402 4.034879809151485e+02 0 3, 1];
Limpia22= [26.115436227761847 3.854168479394065e+02 0 16, 1];
Limpia3= [25.946697141422693 3.852450444920681e+02 40 0, 1];
Limpia33= [26.072934038888786 3.931628544175227e+02 31 4, 1];
Limpia4= [25.871312772694490 3.924978312802452e+02 9 24, 1];
Limpia44= [25.706771253025580 3.646382436976625e+02 8 0, 1];
Limpia5= [26.647450083056746 4.049062845885590e+02 14 0, 1];
Limpia55= [26.648889117017678 4.155344150227687e+02 0 0, 1];

%pico malo
pm1 = [28.235716299977630 4.340498811497389e+02 1202 385, 0];
pm11 = [28.257282563866560 4.283904813794064e+02 1208 715, 0];
pm2= [27.294417863031192 4.218193349637940e+02 1010 456, 0];
pm22 = [27.193935021168148 4.180811370045416e+02 946 2, 0];
pm3 = [30.848135838451450 5.103326156771842e+02 1975 249, 0];
pm33 = [31.211005382206828 5.015824119318161e+02 2939 59, 0];
pm4 = [30.642935115397425 4.681656757582682e+02 2001 1956, 0];
pm44 = [30.467420738326600 4.856520351956007e+02 1581 1923, 0];

%malas etiquetas y otras impurezas
mala1= [32.737000604212260 5.724923892748437e+02 303 2906, 0];
mala11=[31.849447246168160 5.346838671722708e+02 10 2624, 0];
mala2 = [33.027459337425460 5.520356273828855e+02 2490 2536, 0];
mala22=[31.897678406944564 5.502935459348334e+02 1811 5593, 0];
mala3 = [32.228882102317850 5.455610386831531e+02 2285 3205, 0];
mala33=[31.536188350758420 5.423876765548753e+02 1496 2970, 0];
mala4 = [35.731748123946230 6.145178777771853e+02 3390 5100, 0];
mala44=[32.236046188563100 5.453144822217015e+02 221 5098, 0];
mala5 = [41.060766727676380 8.034724858227376e+02 8021 6860, 0];
mala55=[33.940052794234376 6.271721739975451e+02 1043 8441, 0];
mala6= [37.441696256151710 6.369030597094160e+02 5757 5726, 0];
mala66=[35.137166913774735 6.419866854568311e+02 1308 6691, 0];
mala7= [50.898406103789200 1.557794142347661e+03 15773 13560, 0];
mala77=[43.605480257527500 1.251436438503570e+03 1111 23767, 0];
mala8 = [52.643252055500476 1.653053043742872e+03 21104 28092, 0];
mala88=[43.998716911732610 1.190849752536732e+03 79 95, 0];
mala9 = [70.547970737929330 2.977778006238261e+03 63644 9377, 0];
mala99=[68.177067557606250 2.750225132978256e+03 70742 2600, 0];

vacio1 = [16.967621992593040 1.927613793983196e+02 0 0, 0];
vacio11 = [17.149359825784163 1.925145196803449e+02 0 0, 0];

% %vector Datos de entrenamiento 1
trn_data = [Limpia1; Limpia11; Limpia2; Limpia22; Limpia3; Limpia33;
Limpia4; Limpia44; Limpia5; Limpia55;
pm1; pm11; pm2; pm22; pm3; pm33; pm4; pm44;
mala1; mala11; mala2; mala22; mala3; mala33; mala4; mala44;
mala5; mala55;

```



```
mala6; mala66; mala7; mala77; mala8; mala88; mala9; mala99;  
vacio1; vacio11];
```

Datos Entrenamiento 2

%Cargar los datos de entrenamiento 2

```
Limpia1= [26.014122547014352 387.19 0 0, 1];  
Limpia11= [26.116146510655017 386.50 0 52, 1];  
Limpia2= [26.765390256628372 403.50 0 0, 1];  
Limpia22= [26.136572486399032 394.82 0 34, 1];  
Limpia3= [25.877123469735096 386.66 13 0, 1];  
Limpia33= [26.194617884735802 401.52 52 57, 1];  
Limpia4= [25.695798682484153 395.65 6 24, 1];  
Limpia44= [25.869442279010390 373.12 43 58, 1];  
Limpia5= [26.428855514488420 406.51 0 0, 1];  
Limpia55= [27.512224299238050 420.22 58 58, 1];
```

%pico malo

```
pm1 = [28.078368079308820 432.426698709604000 1110 379, 0];  
pm11 = [28.397798051102473 437.668591064038000 1263 715, 0];  
pm2= [27.261611765019275 425.559705274035000 924 39, 0];  
pm22= [28.051526476769180 446.120821687076000 986 329, 0];  
pm3= [30.848135838451450 515.953118018976000 1885 40, 0];  
pm33 = [31.211005382206828 509.593317170025000 2939 80, 0];  
pm4= [30.553892371237716 474.416662303086000 1957 1940, 0];  
pm44= [30.939803047395547 496.465793995160000 1838 7883, 0];
```

%malas etiquetas y otras impurezas

```
mala1= [32.659768471546950 578.104752166874000 198 2820, 0];  
mala11= [31.849447246168160 542.723877371140000 18 2693, 0];  
mala2 = [32.704107460854720 553.871209197702000 2229 2455, 0];  
mala22 = [31.984481584406020 559.427577112549000 1811 6609, 0];  
mala3 = [32.132047926492454 549.966270030003000 2285 2742, 0];  
mala33 = [31.881464870291550 563.454096807871000 1694 5672, 0];  
mala4 = [35.614970336671550 622.349151076859000 3275 4885, 0];  
mala44 = [32.286337788980106 555.649339913609000 259 5098, 0];  
mala5 = [40.902918875479195 821.139690070249000 7673 6720, 0];  
mala55 = [35.377827113543475 681.948626731847000 1069 8569, 0];  
mala6 = [37.332731975063870 650.283228259260000 5757 3448, 0];  
mala66 = [35.348915363386325 655.455206093008000 1386 7276, 0];  
mala7 = [50.693295640410940 1575.107151367410000 15192 13136, 0];  
mala77 = [43.708537188833980 1271.503944803990000 1111 25043, 0];  
mala8 = [52.374468842022510 1666.565365649350000 21104 11739, 0];  
mala88 = [44.204066081577320 1220.413495315530000 116 125, 0];  
mala9 = [70.424703390787500 2994.048977205210000 63644 9341, 0];  
mala99 = [68.469431777955250 2782.959907143160000 72762 3214, 0];
```

```
vacio1 = [16.648187862763663 192.858116016473000 0 0, 0];  
vacio11 = [17.370457065788940 196.037881537456000 0 56, 0];
```

%vector Datos de entrenamiento 2

```
trn_data2 = [Limpia1; Limpia11; Limpia2; Limpia22; Limpia3; Limpia33;  
Limpia4; Limpia44; Limpia5; Limpia55;  
pm1; pm11; pm2; pm22; pm3; pm33; pm4; pm44;
```

```

malal; mala11; mala2; mala22; mala3; mala33; mala4; mala44;
mala5; mala55;
mala6; mala66; mala7; mala77; mala8; mala88; mala9; mala99;
vacio1; vacio11];

```

Datos Entrenamiento 3

%Cargar los datos de entrenamiento 3

```

Limpia1= [26.506773433671660 420.49 17 20, 1];
Limpia11= [26.815246510655017 406.20 0 50, 1];
Limpia2= [26.194617884735802 403.70 0 0, 1];
Limpia22= [25.136572486399032 394.82 0 4, 1];
Limpia3= [25.877123469735096 386.36 17 0, 1];
Limpia33= [26.765390256628372 401.82 0 7, 1];
Limpia4= [26.605798682484153 395.35 0 2, 1];
Limpia44= [26.669442279010390 393.62 44 0, 1];
Limpia5= [25.338855514488420 403.21 0 2, 1];
Limpia55= [26.212224299238050 350.28 5 8, 1];

```

%pico malo

```

pm1 = [28.582853787630916 430.450432333023000 1200 391, 0];
pm11 = [28.478368079308820 435.760124607745000 1803 283, 0];
pm2= [27.042663601404666 428.914473844249000 1533 356, 0];
pm22= [27.597190160638530 423.662524269355000 621 33, 0];
pm3= [29.103185326087146 507.798264750520000 1196 3840, 0];
pm33 = [39.848135838451450 505.953118018976000 3885 100, 0];
pm4= [31.768936238163295 477.784125409938000 2406 986, 0];
pm44= [30.757420738326600 472.951953640462000 1343 1023, 0];

```

%malas etiquetas y otras impurezas

```

mala1= [31.773447829375584 541.061541837037000 103 2746, 0];
mala11= [32.510919293995734 580.325247835353000 100 2907, 0];
mala2 = [31.143918104411500 559.398273165265000 1625 5800, 0];
mala22 = [33.595927146261012 545.456408766752000 2502 4370, 0];
mala3 = [31.813095919416430 550.590918502761000 1807 2604, 0];
mala33 = [33.626267664483034 571.706456453297000 2422 5238, 0];
mala4 = [32.049623292559760 557.845279653134000 2482 310, 0];
mala44 = [36.054984383104640 610.898072468104000 321 595, 0];
mala5 = [33.280514297124800 634.796662124775000 9028 8107, 0];
mala55 = [41.830855689760840 826.951443331767000 875 184, 0];
mala6 = [35.596718116159600 653.007939607314000 1995 6522, 0];
mala66 = [37.079604808823170 652.658111903464000 5244 6977, 0];
mala7 = [43.104207492104810 1282.026421253170000 19563 23092, 0];
mala77 = [51.505617648361180 1566.997970822710000 1453 29637, 0];
mala8 = [53.748812261168640 1676.664450405020000 21727 28304, 0];
mala88 = [42.915508810028830 1209.671941519130000 1000 9532, 0];
mala9 = [69.833265446401230 2002.306202957460000 69776 2598, 1];
mala99 = [70.030837200295960 3774.360480976820000 64978 9577, 1];

```

```

vacio1 = [16.754467336145830 195.336324354101000 0 0, 1];
vacio11 = [16.779920803878640 194.481371666796000 0 0, 1];

```



```
%vector Datos de entrenamiento 3
trn_data3 = [Limpia1; Limpia11; Limpia2; Limpia22; Limpia3; Limpia33;
Limpia4; Limpia44; Limpia5; Limpia55;
             pm1; pm11; pm2; pm22; pm3; pm33; pm4; pm44;
             mala1; mala11; mala2; mala22; mala3; mala33; mala4; mala44;
             mala5; mala55;
             mala6; mala66; mala7; mala77; mala8; mala88; mala9; mala99;
             vacio1; vacio11];
```


Anexo 6: Conjunto de Datos para la validación del FIS

Datos Validación 1

```
%Cargar los datos de validación 1
```

```
Limpia1= [26.403773433671660 394.285217049310000 17 27, 1];
Limpia11= [25.744796370836927 373.729714260096000 0 8, 1];
Limpia2= [26.883302884235402 409.720927494677000 0 6, 1];
Limpia22= [25.869245425236677 390.669396136534000 0 0, 1];
Limpia3= [26.485133360291540 401.781843007476000 58 33, 1];
Limpia33= [25.895712048522880 395.906127814924000 29 4, 1];
Limpia4= [26.147819365838604 397.318374413720000 19 42, 1];
Limpia44= [24.814523785508147 363.857033852423000 2 0, 1];
Limpia5= [26.732947500929928 415.611022088665000 18 20, 1];
Limpia55= [26.514324543009185 411.781891968897000 0 0, 1];
```

```
%pico malo
```

```
pm1 = [28.282853787630916 440.450432333023000 1228 391, 0];
pm11 = [28.078368079308820 438.760124607745000 1212 380, 0];
pm2= [27.342663601404666 427.914473844249000 1033 456, 0];
pm22= [27.097190160638530 420.662524269355000 921 2, 0];
pm3= [31.103185326087146 517.798264750520000 2196 4880, 0];
pm33 = [31.211005382206828 509.593317170025000 2939 80, 0];
pm4= [30.768936238163295 477.784125409938000 2002 2086, 0];
pm44= [30.467420738326600 492.951953640462000 1581 1923, 0];
```

```
%malas etiquetas y otras impurezas
```

```
malal= [32.773447829375584 581.061541837037000 303 2946, 0];
mala11= [31.510919293995734 540.325247835353000 1 2607, 0];
mala2 = [33.143918104411500 559.398273165265000 2625 4800, 0];
mala22 = [31.595927146261012 555.556408766752000 1502 5370, 0];
mala3 = [32.813095919416430 570.590918502761000 2807 5604, 0];
mala33 = [31.626267664483034 551.706456453297000 1422 2238, 0];
mala4 = [36.049623292559760 627.845279653134000 3482 5100, 0];
mala44 = [32.054984383104640 550.898072468104000 221 3975, 0];
mala5 = [41.280514297124800 824.796662124775000 8028 9107, 0];
mala55 = [33.830855689760840 636.951443331767000 975 8184, 0];
mala6 = [37.596718116159600 654.007939607314000 5995 6922, 0];
mala66 = [35.079604808823170 653.658111903464000 1244 6577, 0];
mala7 = [51.104207492104810 1582.026421253170000 15956 29092, 0];
mala77 = [43.505617648361180 1266.997970822710000 1033 23637, 0];
mala8 = [52.748812261168640 1676.664450405020000 21727 28304, 0];
mala88 = [43.915508810028830 1209.671941519130000 20 95, 0];
mala9 = [70.833265446401230 3002.306202957460000 64776 9598, 0];
mala99 = [68.030837200295960 2774.360480976820000 69978 2577, 0];
```

```
vacio1 = [16.774467336145830 196.336324354101000 0 0, 0];
```

```
vacio11 = [16.759920803878640 193.481371666796000 0 0, 0];
```

```
% %Vector Datos de validacion
```

```
chk_data = [Limpia1; Limpia11; Limpia2; Limpia22; Limpia3; Limpia33;
Limpia4; Limpia44; Limpia5; Limpia55;
pm1; pm11; pm2; pm22; pm3; pm33; pm4; pm44;
malal; mala11; mala2; mala22; mala3; mala33; mala4; mala44;
mala5; mala55;
```



```
mala6; mala66; mala7; mala77; mala8; mala88; mala9; mala99;  
vacio1; vacio11];
```

Datos Validación 2

```
%Cargar los datos de validación 2  
Limpia1= [26.014122547014352 3.806553641724079e+02 0 1, 1];  
Limpia11= [25.787781095899852 3.675284205846882e+02 0 52, 1];  
Limpia2= [25.869245425236677 3.848705474597394e+02 0 34, 1];  
Limpia22= [25.996199248959737 3.872076115116821e+02 0 28, 1];  
Limpia3= [26.082747747181443 3.951847415278160e+02 29 25, 1];  
Limpia33= [26.137115060776573 3.904472319142315e+02 38 19, 1];  
Limpia4= [25.101985839085610 3.672371424188737e+02 27 58, 1];  
Limpia44= [25.706771253025580 3.646382436976625e+02 8 0, 1];  
Limpia5= [27.402626552509147 4.335663280257684e+02 24 4, 1];  
Limpia55= [26.648889117017678 4.155344150227687e+02 0 0, 1];  
  
%pico malo  
pm1 = [27.193935021168148 4.180811370045416e+02 946 2, 0];  
pm11 = [28.257282563866560 4.283904813794064e+02 1208 715, 0];  
pm2= [31.211005382206828 5.015824119318161e+02 2939 80, 0];  
pm22= [30.642935115397425 4.681656757582682e+02 2001 1956, 0];  
pm3= [30.467420738326600 4.856520351956007e+02 1581 1923, 0];  
pm33 = [27.294417863031192 4.218193349637940e+02 1010 456, 0];  
pm4= [28.235716299977630 4.340498811497389e+02 1202 385, 0];  
pm44= [30.467420738326600 5.265022319318161e+02 1500 1003, 0];  
  
%malas etiquetas y otras impurezas  
mala1= [32.737000604212260 5.724923892748437e+02 303 2906, 0];  
mala11= [31.849447246168160 5.346838671722708e+02 10 2624, 0];  
mala2 = [33.027459337425460 5.520356273828855e+02 2490 2536, 0];  
mala22 = [31.897678406944564 5.502935459348334e+02 1811 5593, 0];  
mala3 = [32.228882102317850 5.455610386831531e+02 2285 3205, 0];  
mala33 = [31.536188350758420 5.423876765548753e+02 1496 2970, 0];  
mala4 = [35.731748123946230 6.145178777771853e+02 3390 5100, 0];  
mala44 = [32.236046188563100 5.453144822217015e+02 221 5098, 0];  
mala5 = [41.060766727676380 8.034724858227376e+02 8021 6860, 0];  
mala55 = [33.940052794234376 6.271721739975451e+02 1043 8441, 0];  
mala6 = [37.441696256151710 6.369030597094160e+02 5757 5726, 0];  
mala66 = [35.137166913774735 6.419866854568311e+02 1308 6691, 0];  
mala7 = [50.898406103789200 1.557794142347661e+03 15773 13560, 0];  
mala77 = [43.626684867625116 1.249598847200897e+03 1071 25043, 0];  
mala8 = [43.605480257527500 1.251436438503570e+03 1111 23767, 0];  
mala88 = [52.643252055500476 1.653053043742872e+03 21104 28092, 0];  
mala9 = [68.177067557606250 2.750225132978256e+03 70742 2600, 0];  
mala99 = [68.030837200295960 2774.360480976820000 69978 2577, 0];  
  
vacio1 = [16.967621992593040 1.927613793983196e+02 0 0, 0];  
vacio11 = [17.149359825784163 1.925145196803449e+02 0 0, 0];  
  
% %Vector Datos de validación 2  
chk_data2 = [Limpia1; Limpia11; Limpia2; Limpia22; Limpia3; Limpia33;  
Limpia4; Limpia44; Limpia5; Limpia55;
```

pm1; pm11; pm2; pm22; pm3; pm33; pm4; pm44;
mala1; mala11; mala2; mala22; mala3; mala33; mala4; mala44;
mala5; mala55;
mala6; mala66; mala7; mala77; mala8; mala88; mala9; mala99;
vacio1; vacio11];

Anexo 7: Tablas de la Base de Datos del Sistema

Nombre de la base de datos: **icolab_bd**

✓ Tabla Personal

Table Name: Schema: **icolab_bd**

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_persona	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
cedula	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nombres	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
apellidos	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
direccion	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
telefono	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
genero	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
estado_civil	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
empleado	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
usuario	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

✓ Tabla Usuario

Table Name: Schema: **icolab_bd**

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_usuario	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
nombre	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
clave	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nivel_acceso	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
usuario	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
persona	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

✓ Tabla objetobotella

Table Name: Schema: **icolab_bd**

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_objeto	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
fecha	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nombre_objeto	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
defecto	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
NUsuario	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
usuario	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

✓ Tabla reporteresultados

Table Name: Schema: **icolab_bd**

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_resultado	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
estado	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
total	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
porcentaje	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
fecha	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
usuario	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Anexo 8: Solicitud para realizar pruebas de validación y encuestas



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

Loja, 11 de Marzo del 2014.

Mg. Sc.

Hernán Leonardo Torres Carrión.

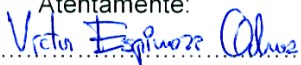
COORDINADOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS

Presente.-

Yo, Víctor Moisés Espinoza Ochoa, egresado de la carrera de ingeniería en sistemas con cédula de ciudadanía 0704520279, por medio de la presente solicito a usted muy comedidamente lo siguiente: se me permita realizar las pruebas de validación y encuestas del mi proyecto de tesis denominado **“PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) APOYADO EN UN ENTRENAMIENTO NEURO-DIFUSO, PARA LA INSPECCIÓN DE OBJETOS DEFECTUOSOS”**, a los estudiantes del Décimo Módulo de la Carrera a su coordinación, dichas encuestas se realizarían el día 18 de marzo a las 10H00 en el Laboratorio de Automatización y Control del Área de Energía, las Industrias y los Recursos Naturales No Renovables.

Por la atención que se digne a la presente le antelo mi más sincero agradecimiento.

Atentamente:



Víctor Moisés Espinoza Ochoa

CI: 070452027-9



Anexo 9: Autorización para realizar pruebas de validación y encuestas



UNIVERSIDAD
NACIONAL
DE LOJA

OFICIO-CIS-UNL



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

Of. N° 236 CIS-AEIRNNR-UNL
Loja, 12 de marzo de 2014

Señor Ingeniero
Luis Antonio Chamba Eras
DOCENTE DE LA CARRERA DE INGENIERIA EN SISTEMAS.
Ciudad.

De mi consideración:

En atención a la petición realizada por el **Sr. Victor Moisés Espinoza Ochoa** el día 11 de marzo del 2014, en donde me solicita que se le permita realizar las pruebas de validación y encuestas para su proyecto "**PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) APOYADO EN UN ENTRENAMIENTO NEURO-DIFUSO, PARA LA INSPECCIÓN DE OBJETOS DEFECTUOSOS**", a realizarse el día 18 de marzo del año en curso, a los estudiantes del Módulo X; por lo que me permito indicar que esta Coordinación autoriza el pedido del mencionado señor.

Sin otro particular me suscribo de usted.

Atentamente,

Ing. Hernán Leonardo Torres Carrión M.Sc.
COORDINADOR CIS.

C.c. Archivo
Elisa Orellana

Anexo 10: Modelo de encuestas aplicadas a los administradores (Docentes)



UNIVERSIDAD NACIONAL DE LOJA
ÁREA DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

Buenos días/tardes, con la finalidad de verificar la funcionalidad del sistema de control ICOlab versión 1.0, la cual emplea el procesamiento digital de imágenes apoyado en un entrenamiento Neuro-difuso para la inspección y clasificación de objetos defectuosos (botellas de vidrio transparente), herramienta que interviene como apoyo en prácticas del Laboratorio de Automatización y Control de la institución, se solicita de manera comedida se otorgue unos minutos de su tiempo y responda las siguientes preguntas.

Perfil del Administrador

Nombre: Ing. JULIO CUELLER TINITANA
Carrera: ELECTROELECTRÓNICA Unidad: CONTROL

1. ACCESIBILIDAD

1.1 ¿Se presentó algún problema al ingresar al sistema con su nombre de usuario y contraseña?

SI

NO

2. FUNCIONALIDAD

2.1 Marque con una X, indicando si tuvo algún problema al realizar alguna de las siguientes actividades:

- | | |
|---|-------------------------------------|
| a) La información se presenta de manera clara y entendible | <input checked="" type="checkbox"/> |
| b) Los mensajes de información y error son adecuados y entendibles. | <input checked="" type="checkbox"/> |
| c) El empleo de directorios de archivos fue conveniente | <input checked="" type="checkbox"/> |
| d) Los resultados de la inspección son aceptables y confiables. | <input checked="" type="checkbox"/> |
| e) El tiempo y la forma en que se presenta el resultado de la clasificación al ejecutar una inspección son adecuados. | <input checked="" type="checkbox"/> |
| f) El sistema permite visualizar en tiempo real la ejecución de la inspección. | <input checked="" type="checkbox"/> |
| g) El sistema ejecuta correctamente la administración de Personal | <input checked="" type="checkbox"/> |
| h) El sistema ejecuta correctamente la administración de Usuarios. | <input checked="" type="checkbox"/> |
| i) El sistema ejecuta correctamente la consulta de Reportes Estadísticos. | <input checked="" type="checkbox"/> |



2.2 Considera que el sistema le permitió realizar estas actividades de manera:

- Rápida, eficiente y confiable
- Regular
- Lenta, deficiente, Nada Confiable

2.3 ¿Cómo evalúa la funcionalidad de la integración de las herramientas: MatLab en el procesamiento digital de imágenes y Java en la realización de la interfaz gráfica amigable del sistema ICOLab?

- Muy eficaz
- Eficaz
- Ineficaz
- Muy ineficaz

3. DISEÑO Y PRESENTACIÓN

3.1 Califique la interfaz gráfica de la aplicación ICOLab.

- Muy amigable y fácil de manejar
- Amigable
- Regular
- Confusa

3.2 ¿Cómo considera la visualización de la imagen (objeto botella) a través de la cámara en tiempo real?

- Excelente
- Buena
- Regular
- Mala

Firma: _____

GRACIAS POR SU COLABORACIÓN

Anexo 11: Modelo de encuestas aplicadas a los usuarios del sistema (Estudiantes)



UNIVERSIDAD NACIONAL DE LOJA ÁREA DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

Buenos días/tardes, se está realizando una encuesta para verificar la funcionalidad de la herramienta tecnológica de control ICOLab versión 1.0, la cual emplea el procesamiento digital de imágenes apoyado en un entrenamiento Neuro-difuso para la inspección y clasificación de objetos defectuosos (botellas de vidrio transparente), herramienta que interviene como apoyo en las prácticas del Laboratorio de Automatización y Control de la institución, se solicita de manera comedida se otorgue unos minutos de su tiempo y responda las siguientes preguntas.

Perfil del Usuario

Nombre: Darío Broeze
Carrera: Ingeniería en Sistemas Módulo: 10^{mo} Paralelo: A

- ¿Está de acuerdo en que el sistema de control ICOLab como herramienta de apoyo y prácticas, ayuda en el proceso de enseñanza-aprendizaje?
 - Muy de acuerdo
 - De acuerdo
 - En desacuerdo
 - Muy en desacuerdo
- ¿Cómo evalúa la funcionalidad de la integración de las herramientas: MatLab en el procesamiento digital de imágenes y Java en la realización de la interfaz gráfica amigable del sistema ICOLab?
 - Muy eficaz
 - Eficaz
 - Ineficaz
 - Muy ineficaz
- En la interfaz gráfica principal, al seleccionar el módulo de 'Inspección', esta le parece:
 - Intuitiva
 - Entendible
 - Confusa
 - Incoherente



4. ¿Cómo considera la visualización de la imagen (objeto botella) a través de la cámara en tiempo real?
- Excelente
 - Buena
 - Regular
 - Mala
5. La rapidez de respuesta del equipo con respecto a la funcionalidad de la interfaz gráfica del sistema es:
- Eficiente
 - Poco eficiente
 - Lenta
 - Muy lenta
6. Califique el tiempo y la forma en que se presenta el resultado de clasificación del objeto botella al ejecutar una inspección.
- Excelente
 - Buena
 - Regular
 - Mala
7. ¿Cómo considera la configuración y personalización de todo el sistema ICOLab para el Laboratorio de Automatización y Control?
- Supera mis expectativas
 - Cumple con mis expectativas
 - No supera mis expectativas
8. Según su criterio, ¿Cómo calificaría a los resultados del Sistema de Control de Inspección de Objetos (botellas) defectuosos en la inspección y clasificación?
- Resultados aceptables
 - Resultados confiables
 - Resultados no aceptables
 - Resultados incoherentes
 - Otros _____

Firma: _____

GRACIAS POR SU COLABORACIÓN



UNIVERSIDAD NACIONAL DE LOJA
ÁREA DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO
RENOVABLES

Buenos días/tardes, se está realizando una encuesta para verificar la funcionalidad de la herramienta tecnológica de control ICOLab versión 1.0, la cual emplea el procesamiento digital de imágenes apoyado en un entrenamiento Neuro-difuso para la inspección y clasificación de objetos defectuosos (botellas de vidrio transparente), herramienta que interviene como apoyo en las prácticas del Laboratorio de Automatización y Control de la institución, se solicita de manera comedida se otorgue unos minutos de su tiempo y responda las siguientes preguntas.

Perfil del Usuario

Nombre: Guillermo Palacios
Carrera: Electromecánica Módulo: 6^{to} Paralelo: B

1. ¿Está de acuerdo en que el sistema de control ICOLab como herramienta de apoyo y prácticas, ayuda en el proceso de enseñanza-aprendizaje?
 - Muy de acuerdo
 - De acuerdo
 - En desacuerdo
 - Muy en desacuerdo

2. ¿Cómo evalúa la funcionalidad de la integración de las herramientas: MatLab en el procesamiento digital de imágenes y Java en la realización de la interfaz gráfica amigable del sistema ICOLab?
 - Muy eficaz
 - Eficaz
 - Ineficaz
 - Muy ineficaz

3. En la interfaz gráfica principal, al seleccionar el módulo de 'Inspección', esta le parece:
 - Intuitiva
 - Entendible
 - Confusa
 - Incoherente



4. ¿Cómo considera la visualización de la imagen (objeto botella) a través de la cámara en tiempo real?
- Excelente
 - Buena
 - Regular
 - Mala
5. La rapidez de respuesta del equipo con respecto a la funcionalidad de la interfaz gráfica del sistema es:
- Eficiente
 - Poco eficiente
 - Lenta
 - Muy lenta
6. Califique el tiempo y la forma en que se presenta el resultado de clasificación del objeto botella al ejecutar una inspección.
- Excelente
 - Buena
 - Regular
 - Mala
7. ¿Cómo considera la configuración y personalización de todo el sistema ICOLab para el Laboratorio de Automatización y Control?
- Supera mis expectativas
 - Cumple con mis expectativas
 - No supera mis expectativas
8. Según su criterio, ¿Cómo calificaría a los resultados del Sistema de Control de Inspección de Objetos (botellas) defectuosos en la inspección y clasificación?
- Resultados aceptables
 - Resultados confiables
 - Resultados no aceptables
 - Resultados incoherentes
 - Otros _____

Firma: Guller...

GRACIAS POR SU COLABORACIÓN

Anexo 12: Certificación de Traducción del Resumen



Lic. Silvana Jaramillo Aguilar
PROFESORA DEL INSTITUTO
"FINE-TUNED ENGLISH"

CERTIFICA:

Que el documento aquí compuesto es fiel traducción del idioma español al idioma inglés del resumen para el artículo científico de la tesis titulada **"PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) APOYADO EN UN ENTRENAMIENTO NEURO-DIFUSO, PARA LA INSPECCIÓN DE OBJETOS DEFECTUOSOS"**, del señor VÍCTOR MOISÉS ESPINOZA OCHOA, egresado de la carrera de Ingeniería en Sistemas.

Lo certifica en honor a la verdad y autoriza al interesado hacer uso del presente en lo que a sus intereses convenga.

Loja, 31 de marzo de 2014



Lic. Silvana Jaramillo Aguilar
PROFESORA DE F.T.E.



Anexo 13: Cronograma de Actividades

CRO NO GRAMA DE ACTIVIDADES ICOLAB						
Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesora	Nombres de los recursos
1	1. Modelo de la organización (M. CommonKADS)	11 días	vie 04/02/11	vie 18/02/11		Victor Espinoza
2	Condiciones iniciales	5 días	vie 04/02/11	jue 10/02/11		Victor Espinoza
3	Especificaciones del proyecto	6 días	vie 11/02/11	vie 18/02/11	2	Victor Espinoza
4	2. FASE 1: IDENTIFICACIÓN DE LA TAREA (M. IDEAL)	70 días	lun 07/03/11	vie 10/06/11		Victor Espinoza
5	Etapa 1.1: Plan de requisitos y adquisición de conocimientos	24 días	lun 07/03/11	jue 07/04/11		Victor Espinoza
6	Proceso de Adquisición de Conocimientos	24 días	lun 07/03/11	jue 07/04/11		Victor Espinoza
7	Primeras reuniones y evaluación de viabilidad	5 días	lun 07/03/11	vie 11/03/11		Victor Espinoza, Expertos
8	Extracción de conocimientos	5 días	lun 14/03/11	vie 18/03/11	7	Victor Espinoza, Expertos
9	Educación de conocimientos con expertos	4 días	mar 29/03/11	vie 01/04/11	8	Victor Espinoza, Expertos
10	Educación de conocimientos	4 días	lun 04/04/11	jue 07/04/11	9	Victor Espinoza, Expertos
11	Etapa 1.2: Evaluación y selección de la tarea	10 días	vie 08/04/11	jue 21/04/11	10	Victor Espinoza
12	Test de Madurez	10 días	vie 08/04/11	jue 21/04/11		Victor Espinoza
13	Etapa 1.3: Definición de las características del sistema	36 días	vie 22/04/11	vie 10/06/11	12	Victor Espinoza
14	Requerimientos Funcionales	7 días	vie 22/04/11	lun 02/05/11		Victor Espinoza
15	Requerimientos No Funcionales	7 días	mar 17/05/11	mié 25/05/11		Victor Espinoza
16	Casos de prueba o ensayo	3 días	jue 26/05/11	lun 30/05/11		Victor Espinoza, Expertos
17	Recursos necesarios	36 días	vie 22/04/11	vie 10/06/11		Victor Espinoza
18	Recursos Humanos	4 días	mar 31/05/11	vie 03/06/11		Victor Espinoza
19	Recursos Técnicos y Tecnológicos	5 días	lun 06/06/11	vie 10/06/11	18	Victor Espinoza
20	Recursos Económicos	5 días	vie 22/04/11	jue 28/04/11		Victor Espinoza
21	Análisis de costo / beneficio	18 días	jue 12/05/11	lun 06/06/11		Victor Espinoza
22	Costo del Proyecto	6 días	jue 12/05/11	jue 19/05/11		Victor Espinoza
23	Costos tangibles	3 días	jue 12/05/11	lun 16/05/11		Victor Espinoza
24	Costos intangibles	3 días	mar 17/05/11	jue 19/05/11		Victor Espinoza
25	Beneficios del Proyecto	9 días	vie 20/05/11	mié 01/06/11	23	Victor Espinoza
26	Beneficios tangibles	4 días	vie 20/05/11	mié 25/05/11		Victor Espinoza

CRO NO GRAMA DE ACTIVIDADES ICO LAB

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesora	Nombres de los recursos
27	Beneficios intangibles	5 días	jue 26/05/11	mié 01/06/11	26	Victor Espinoza
28	Calendario o Cronograma de actividades	3 días	jue 02/06/11	lun 06/06/11	27	Victor Espinoza
29	3. FASE 2: DESARROLLO DE LOS PROTOTIPOS	452 días	mar 07/06/11	mié 27/02/13		Victor Espinoza
30	Etapa 2.1: Concepción de la solución	21 días	mar 07/06/11	mar 05/07/11		Victor Espinoza
31	Modelado de Caso de Uso	7 días	mar 07/06/11	mié 15/06/11		Victor Espinoza
32	Diseño Arquitectónico de Sistema	12 días	lun 20/06/11	mar 05/07/11		Victor Espinoza
33	Modelo del Dominio	6 días	lun 20/06/11	lun 27/06/11		Victor Espinoza
34	Modelo de la Base de Datos	6 días	mar 28/06/11	mar 05/07/11	33	Victor Espinoza
35	Etapa 2.2: Adquisición y Conceptualización de los conocimientos	320 días	jue 07/07/11	mié 26/09/12	31	Victor Espinoza
36	Diccionario de Términos	3 días	jue 07/07/11	lun 11/07/11		Victor Espinoza
37	Identificación de las relaciones de la aplicación	275 días	lun 18/07/11	vie 03/08/12		Victor Espinoza
38	Modelado Dinámico de la Aplicación	10 días	lun 18/07/11	vie 29/07/11		Victor Espinoza
39	Modelo de tareas de alto nivel (TAN) (M. CommonKADS)	10 días	lun 01/08/11	vie 12/08/11	38	Victor Espinoza
40	Modelo de tareas en tiempo real (M. CommonKADS)	10 días	lun 15/08/11	vie 26/08/11		Victor Espinoza
41	Modelo del conocimiento (M. CommonKADS)	80 días	lun 29/08/11	vie 16/12/11	40	Victor Espinoza
42	Factores o variables del proceso de control	30 días	lun 29/08/11	vie 07/10/11		Victor Espinoza
43	Diseño y Construcción del Banco de Pruebas	50 días	lun 10/10/11	vie 16/12/11	42	Victor Espinoza
44	Modelo de diseño (M. CommonKADS)	145 días	lun 16/01/12	vie 03/08/12		Victor Espinoza
45	Toma de muestras fotografías	50 días	lun 16/01/12	vie 23/03/12		Victor Espinoza
46	Desarrollo del algoritmo en tiempo diferido	100 días	lun 19/03/12	vie 03/08/12		Victor Espinoza
47	Identificación de los conocimientos tácticos	38 días	lun 06/08/12	mié 26/09/12		Victor Espinoza
48	Análisis de conocimientos tácticos	16 días	lun 06/08/12	lun 27/08/12		Victor Espinoza
49	Comprobación de los conocimientos tácticos	20 días	jue 30/08/12	mié 26/09/12		Victor Espinoza
50	Etapa 2.3 Formalización de los conocimientos y definición de la arquitectura	74 días	mié 19/09/12	lun 31/12/12		Victor Espinoza
51	Motor de inferencia y base de conocimientos	18 días	mié 19/09/12	vie 12/10/12		Victor Espinoza
52	Codificación de Reglas y desarrollo del controlador difuso	53 días	jue 18/10/12	lun 31/12/12		Victor Espinoza

CRONOGRAMA DE ACTIVIDADES ICO LAB						
Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesora	Nombres de los recursos
53	Preparación de Datos con la ayuda del software Matlab	5 días	jue 18/10/12	mié 24/10/12		Victor Espinoza
54	Construcción del sistema con Fuzzy Logic Toolbox	48 días	jue 25/10/12	lun 31/12/12		Victor Espinoza
55	Entrenando Red Difusa	33 días	jue 25/10/12	lun 10/12/12		Victor Espinoza
56	Evaluación del Sistema de Inferencia Difuso	15 días	mar 11/12/12	lun 31/12/12	55	Victor Espinoza
57	Etapa 2.4 Selección de la Herramienta e implementación	32 días	mar 01/01/13	mié 13/02/13		Victor Espinoza
58	Selección de Recursos	2 días	mar 01/01/13	mié 02/01/13		Victor Espinoza
59	Elección del Lenguajes de Programación	3 días	jue 03/01/13	lun 07/01/13	58	Victor Espinoza
60	Codificación de Interfaces Gráficas	8 días	mar 08/01/13	jue 17/01/13	59	Victor Espinoza
61	Codificación de la Base de Datos	9 días	vie 18/01/13	mié 30/01/13	60	Victor Espinoza
62	Codificación de las Clases y programas	10 días	jue 31/01/13	mié 13/02/13	61	Victor Espinoza
63	Etapa 2.5: Validación y evaluación del prototipo	10 días	jue 14/02/13	mié 27/02/13		Victor Espinoza
64	Compilación de los programas	10 días	jue 14/02/13	mié 27/02/13		Victor Espinoza
65	4. FASE 3: EJECUCIÓN DE LA CONSTRUCCIÓN DEL SISTEMA INTEGRADO	18 días	jue 28/02/13	lun 25/03/13	64	Victor Espinoza
66	Etapa 3.1: Requisitos y diseño de la integración con otros sistemas	4 días	jue 28/02/13	mar 05/03/13		Victor Espinoza
67	Etapa 3.2: Implementación y evaluación del Sistema Integrado	10 días	mié 06/03/13	mar 19/03/13	66	Victor Espinoza
68	Etapa 3.3: Aceptación del sistema por el usuario	4 días	mié 20/03/13	lun 25/03/13	67	Victor Espinoza
69	5. FASE 4: CONSEGUIR UN MANTENIMIENTO PERFECTIVO	10 días	mar 26/03/13	lun 08/04/13	68	Victor Espinoza
70	6. FASE 5: LOGRAR UNA ADECUADA TRANSFERENCIA TECNOLÓGICA	40 días	mar 09/04/13	lun 03/06/13		Victor Espinoza
71	Etapa 5.1: Organizar la transferencia tecnológica	10 días	mar 09/04/13	lun 22/04/13		Victor Espinoza
72	Etapa 5.2: Completar la documentación del SBC de control construido	30 días	mar 23/04/13	lun 03/06/13	71	Victor Espinoza

Cronograma de Actividades

Anexo 14: Licencia Creative Commons del Proyecto



Procesamiento Digital de Imágenes (PDI) apoyado en un entrenamiento Neuro-Difuso, para la inspección de objetos defectuosos by Víctor Moisés Espinoza Ochoa is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License](https://creativecommons.org/licenses/by-nc-sa/4.0/).



Anexo 15: Anteproyecto de Tesis