



Universidad Nacional de Loja

Área de la Energía las Industrias y los
Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

TÍTULO:

**“VISIÓN ARTIFICIAL APLICADA PARA EL RECONOCIMIENTO
DEL LENGUAJE DE SEÑAS”**

Tesis previa a la obtención del
Título de Ingeniero en Sistemas.

AUTOR:

Luis Germán Sivisapa Aguilera

DIRECTOR:

Ing. Henry Patricio Paz Arias Mg. Sc.

LOJA – ECUADOR

2014

CERTIFICACIÓN DEL DIRECTOR

Ing. Henry Patricio Paz Arias Mg. Sc.

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

CERTIFICA:

Que el egresado Luis Germán Sivisapa Aguilera autor del presente trabajo de titulación cuyo tema versa sobre: **“VISIÓN ARTIFICIAL APLICADA PARA EL RECONOCIMIENTO DEL LENGUAJE DE SEÑAS”**, ha sido dirigido orientado y discutido bajo mi asesoramiento y reúne satisfactoriamente los requisitos exigidos en una investigación de este nivel por lo cual autorizo su presentación y sustentación.

Loja, octubre 29 de 2014



Ing. Henry Patricio Paz Arias Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

AUTORÍA

Yo, **Luis Germán Sivisapa Aguilera**, declaro ser autor del presente trabajo de tesis 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 tesis en el Repositorio Institucional - Biblioteca Virtual.

Autor: Luis Germán Sivisapa Aguilera

Firma:



Cédula: 1104746936

Fecha: 1 de diciembre del 2014

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

Yo, **Luis Germán Sivisapa Aguilera**, declaro ser autor de la tesis titulada: “**VISIÓN ARTIFICIAL APLICADA PARA EL RECONOCIMIENTO DEL LENGUAJE DE SEÑAS**”, como requisito para optar al grado de **Ingeniero en 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 de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, primer día del mes de diciembre del dos mil catorce.



Firma:

Autor: Luis Germán Sivisapa Aguilera

Cédula: 1104746936

Dirección: Loja (Zamora Huayco)

Correo Electrónico: lgsivisapaa@unl.edu.ec

Teléfono: 2139299

Celular: 0939169722

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Henry Patricio Paz Arias, Mg. Sc.

Tribunal de Grado: Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

Ing. Gabriela Viñan Rueda, Mg. Sc.

Ing. Franco Hernán Salcedo López, Mg. Sc.

AGRADECIMIENTO

Me complace de sobremanera a través de este trabajo ofrecer mis más sinceros agradecimientos a Dios por guiarme en el desarrollo de este sueño tan anhelado, por regalarme salud y bienestar, a mis padres ya que sus sabios consejos hacen posible que pueda terminar cada una de mis metas con éxito.

De manera muy especial, agradezco a la Universidad Nacional de Loja, a sus autoridades y a toda su planta Docente y Administrativa, que con su apoyo incondicional supieron guiarnos y compartir sus sabias enseñanzas.

También quisiera hacer extensiva mi gratitud a mis Directores del Trabajo de Titulación, Ing. Pablo Fernando Ordóñez Ordóñez Mg. Sc. y al Ing. Henry Patricio Paz Arias Mg. Sc., por su dedicada orientación, seguimiento y supervisión continua y a la Asociación de Sordos de Loja “Virgen del Cisne”, ya que me permitió que se pueda llevar a cabo este proyecto, a través de la dotación completa de información referente al tema.

Luis Germán Sivisapa Aguilera

DEDICATORIA

El presente trabajo de titulación está dedicado, a Dios por ser quien con su gran amor nos ayuda en el logro de nuestros sueños, y quien nos da la fortaleza para seguir, a mis abnegados padres en agradecimiento al don de la vida, y quienes con sus sabios consejos han hecho de mí una persona con valores, a mis hermanas y hermano quienes me apoyaron de manera moral y económica para poder culminar con éxito esta meta propuesta.

Luis Germán Sivisapa Aguilera

CESIÓN DE DERECHOS

Luis Germán Sivisapa Aguilera, autor principal del presente Trabajo de Titulación, autorizo a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales no Renovables, y por ende a la carrera de Ingeniería en Sistemas, hacer uso del mismo en lo que se estime sea conveniente.

a. Título

“Visión Artificial Aplicada para el Reconocimiento del Lenguaje de
Señas”.

b. Resumen

El desarrollo del presente Trabajo de Titulación tiene como finalidad realizar la implementación de una herramienta que facilite el aprendizaje del lenguaje de señas en tiempo real, a través del empleo de técnicas de visión artificial y procesamiento digital de imágenes.

La metodología de programación empleada es la metodología RAD (Rapid Application Development), ya que el sistema a realizar estaba planificado para un tiempo de desarrollo limitado, y se requería de la interacción de los usuarios con los desarrolladores.

Para el reconocimiento de la mano se realizó un algoritmo para seguir objetos segmentados a través del color, al igual que para la predicción se hace uso de los algoritmos de clasificación el KNN (K-Nearest Neighbors) y SVM (Support Vector Machine), ya que son empleados con frecuencia en el desarrollo de este tipo de sistemas, estos dos algoritmos fueron combinados para reducir el porcentaje de error y obtener un mayor porcentaje de aciertos durante la predicción.

En los resultados se detallan los procesos utilizados para el desarrollo del sistema de acuerdo a los requerimientos planteados inicialmente por los miembros de la asociación de Sordos de Loja “Virgen del Cisne”.

La importancia de esta investigación radica en el empleo de herramientas de software libre para el desarrollo del proyecto, de esta manera el sistema no requiere de componentes hardware o software con características potentes, lo que lo hace adecuado para una eficiente distribución.

Summary

The present research work aims to perform the implementation of a tool that facilitates the learning of sign language in real time, through the use of computer vision techniques and digital image processing.

The methodology of programming employed is the methodology RAD (Rapid Application Development), due to the system was planned for a limited development time, and it was required the user interaction with the developers.

For hand recognition algorithm was performed to pursue segmented objects through color, in the same way for prediction where the algorithm of classification are used the KNN (K-Nearest Neighbors) and SVM (Support Vector Machine) because they are frequently used in the development of this kind of systems, these two algorithms were combined to reduce the error rate and for getting a higher percentage of correct answers during the prediction.

In the results the processes used to develop the system according to the established requirements initially by members of the Association of Deaf people of Loja named "Virgen del Cisne" are detailed.

The importance of this research is in the employing of free software tools for the development of the project, so the system don't require hardware or software components with powerful features, making it suitable for an efficient distribution.

Certificado por la Dra. Roció Peñaranda (Ver Anexo 2. Certificación de Traducción.).

Índice de Contenidos

CERTIFICACIÓN DEL DIRECTOR	I
AUTORÍA	II
CARTA DE AUTORIZACIÓN DE TESIS	III
AGRADECIMIENTO	IV
DEDICATORIA.....	V
CESIÓN DE DERECHOS	VI
a. Título.....	VII
b. Resumen.....	VIII
Summary	IX
Índice de Contenidos.....	X
Índice de Figuras	XV
Índice de Tablas	XVII
c. Introducción.....	18
d. Revisión de Literatura	20
CAPÍTULO I. LAS DISCAPACIDADES	20
1.1 Qué es la discapacidad.....	20
1.1.1 Tipos de Discapacidad.....	20
1.2 Discapitados en el Ecuador.....	21
1.2.1 Asociación de Sordos de Loja “Virgen del Cisne”	24
CAPÍTULO II. LA INTELIGENCIA ARTIFICIAL.....	26
2.1 La Inteligencia	26
2.2 Inteligencia Artificial	26
2.3 Técnicas de la Inteligencia Artificial	26

2.3.1	Redes neuronales artificiales (Artificial Neural Networks).....	26
2.3.1.1	Ventajas de las Redes neuronales	27
2.3.2	Ingeniería del conocimiento (Knowledge Engineering).....	27
2.3.2.1	Ventajas de la implementación de la Ingeniería de Conocimiento:.....	28
2.3.3	Procesamiento del lenguaje natural (Natural Language Processing)	29
2.3.4	Minería de datos (Data Mining).....	29
CAPÍTULO III. VISIÓN ARTIFICIAL		31
3.1	Definición.....	31
3.2	La Imagen Digital	31
3.3	Componentes de un sistema de visión artificial.....	32
3.4	Fases en un sistema de visión artificial	33
3.5	Casos de éxito que hacen uso de la visión artificial.	34
3.5.1	Aplicaciones No industriales.....	34
3.5.1.1	Sistema de Reconocimiento Automático de Matrículas (ANPR)	35
3.5.1.1.1	Algoritmo para el reconocimiento de las placas vehiculares	35
3.5.1.2	Reconocimiento Óptico de Caracteres (OCR)	36
3.5.2	Aplicaciones Industriales	42
3.5.3	Aplicaciones para la Inspección y Control de Calidad	42
3.5.3.1	Algoritmo aplicado en el control de Calidad.....	42
3.6	Casos de fracaso que hacen uso de la visión artificial.	46
CAPÍTULO IV. ELEMENTOS PARA EL EMPLEO DE LA VISIÓN ARTIFICIAL.....		49
4.1	Componentes Software	49
4.1.1	Sistema Operativo.....	49
4.1.2	Herramientas de desarrollo	50

4.1.2.1	Lenguajes de programación	50
4.1.2.2	Librerías de desarrollo	51
4.1.2.2.1	JavaVis.....	52
4.1.2.2.2	OpenCV	54
4.1.2.2.3	Image Processing ToolBox.....	56
4.1.2.2.4	X-Vision.....	58
4.2	Componentes Hardware	59
4.3	Algoritmos para el reconocimiento de la mano	60
4.4	Algoritmos para la detección de la mano	61
4.5	Algoritmos para el reconocimiento del lenguaje de señas	62
CAPÍTULO V. METODOLOGÍAS DE DESARROLLO		70
5.1	Metodologías de desarrollo ágil.....	70
5.1.1	Programación Extrema (XP)	70
5.1.2	SCRUM	71
5.1.3	Rapid Application Development (RAD).....	72
5.1.4	Diferencias entre Metodologías Ágiles	73
e.	Materiales y Métodos.....	75
1.	Métodos	75
1.1	Metodología de Desarrollo.....	76
f.	Resultados	79
Fase 1: Determinación de la Factibilidad para el desarrollo del sistema informático para el reconocimiento del lenguaje de señas.		79
1.2	Análisis revisión literaria.....	79
1.3	Componentes hardware y software	80
1.3.1	Software.....	80

1.3.2	Hardware	82
1.4	Lenguaje de Señas empleado.....	83
Fase 2: Diseño e Implementación del sistema.....		83
1.1	Planificación de requisitos	84
1.1.1	Requerimientos Funcionales:.....	84
1.1.2	Requerimientos No Funcionales:.....	85
1.1.3	Lista de Entidades a desarrollar	86
1.1.4	Diagramas de actividad.....	86
1.2	Diseño del usuario.	88
1.2.1	Modelo de datos.....	89
1.2.2	Flujos de ventanas principales del sistema	89
1.3	Construcción Rápida	93
1.3.1	Construcción del Reconocedor de la mano	93
1.3.2	Construcción del Reconocedor del Alfabeto de señas.....	98
1.3.3	Pruebas.....	102
1.3.4	Implantar la aplicación con la que el docente contará para realizar las actividades de aprendizaje.....	112
g.	Discusión.....	118
2.1.1	Talento Humano	119
2.1.2	Recursos Materiales.....	120
2.1.3	Recursos Técnicos.....	120
2.1.4	Servicios Básicos.....	121
h.	Conclusiones	122
i.	Recomendaciones	123
j.	Bibliografía	125
k.	Anexos.....	136
Anexo 1. Solicitud de permiso para el desarrollo del sistema en la Asociación de Sordos de Loja “Virgen del Cisne”.....		136

Anexo 2. Certificación de Traducción.	137
Anexo 3. Guía de Observación directa.	138
Anexo 4: Ley Orgánica de Discapacidades en el Ecuador.	139
Anexo 5. Encuesta Situación Actual de la Asociación de Sordos de Loja “Virgen del Cisne”.	142
Anexo 6. Encuesta para conocer el lenguaje de Señas que emplea en la enseñanza la Asociación de Sordos de Loja “Virgen del Cisne”.	144
Anexo 7. Lenguaje de Señas Ecuatoriano.	146
Anexo 8. Prototipos de pantallas iniciales.	149
Anexo 9. Instalación y Configuración de OpenCV.	152
Anexo 10. Estándar para Codificación en lenguaje C++.	156
Anexo 13. Encuesta funcionalidad del software.	164
Anexo 14. Licencias Aplicadas al sistema.	166

Índice de Figuras

Figura 1. Asociación de Sordos de Loja "Virgen del Cisne".....	24
Figura 2. Representación de la Imagen Digital.....	32
Figura 3. Fases en un sistema de Visión Artificial.....	33
Figura 4. Sistema para el control de Acceso.....	38
Figura 5. Control de Camiones.....	39
Figura 6. Lenguaje de Señas ASL.....	40
Figura 7. Control de tráfico.....	40
Figura 8. (a) Inspección Lateral, (b) Inspección de Fondo.....	44
Figura 9. Control de calidad en fresas.....	45
Figura 10. JavaVis.....	52
Figura 11. OpenCV.....	54
Figura 12. Algoritmos para la detección de la mano.....	60
Figura 13. Representación del algoritmo Cascade Haar.....	62
Figura 14. Algoritmo para la detección del abecedario de señas.....	63
Figura 15. Representación del modelo SVM.....	66
Figura 16. Reconocedor de Señas.....	67
Figura 17. Clasificador KNN con K diferentes valores.....	68
Figura 18. Diagrama de actividades en el proceso de aprendizaje del lenguaje de señas.....	87
Figura 19. Modelo del dominio.....	88
Figura 20. Pantalla principal para el entrenamiento.....	90
Figura 21. Pantalla segmentación del color.....	91
Figura 22. Pantalla de traducción del lenguaje.....	92
Figura 23. Captura de la mano para la clasificación.....	92
Figura 24. Algoritmo para reconocer la mano.....	94
Figura 25. Pruebas de cobertura (a) modelo, (b) controlador.....	106
Figura 26. Identificación de errores de ingreso.....	109
Figura 27. Prueba con Iluminación Adecuada.....	110
Figura 28. Prueba con Iluminación inadecuada.....	110
Figura 29. Prueba nocturna.....	111
Figura 30. Prueba con distancia mayor a 1m.....	112
Figura 31. Resultados pregunta 1.....	113
Figura 32. Resultados pregunta 2.....	114
Figura 33. Resultados pregunta 3.....	115
Figura 34. Resultados pregunta 4.....	115
Figura 35. Resultados pregunta 5.....	116
Figura 36. Abecedario del Lenguaje de Señas Ecuatoriana.....	147
Figura 37. Números del Lenguaje de Señas Ecuatoriana.....	148
Figura 38. Prototipo pantalla Principal.....	149
Figura 39. Prototipo pantalla Entrenamiento con Imágenes.....	150
Figura 40. Prototipo pantalla Entrenamiento con Webcam.....	150
Figura 41. Prototipo pantalla Detección de Piel.....	151

Figura 42. Prototipo pantalla Acerca de...	151
Figura 43. Configuración Opencv.conf	153
Figura 44. Agregar OpenCV al path.	154

Índice de Tablas

TABLA I. DATOS GENERALES POR PROVINCIA.....	21
TABLA II. ORGANIZACIONES QUE TRABAJAN EN MATERIA DE DISCAPACIDADES EN LA PROVINCIA DE LOJA.	22
TABLA III. VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS OPERATIVOS MÁS CONOCIDOS.....	49
TABLA IV. LENGUAJES DE PROGRAMACIÓN QUE SOPORTAN VISIÓN ARTIFICIAL.....	51
TABLA V. MÉTODOS Y ALGORITMOS DE JAVAVIS.....	53
TABLA VI. COMPONENTES Y FUNCIONES DE OPENCV.....	55
TABLA VII. VENTAJAS Y DESVENTAJAS DE IMAGE PROCESSING TOOLBOX.	57
TABLA VIII. COMPONENTES Y FUNCIONES DE IMAGE PROCESSING TOOLBOX....	58
TABLA IX. CARACTERÍSTICAS MÍNIMAS INSTALACIÓN DE S.O.	59
TABLA X. DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES.	73
TABLA XI. PAÍSES DONDE SE HA IMPLANTADO LOS SISTEMAS ANPR.	79
TABLA XII. CARACTERÍSTICAS DEL HARDWARE PARA EL DESARROLLO.....	82
TABLA XIII. REQUERIMIENTOS FUNCIONALES.....	84
TABLA XIV. REQUERIMIENTOS NO FUNCIONALES.	85
TABLA XV. LISTA DE ENTIDADES A DESARROLLAR.	86
TABLA XVI. VENTAJAS NO USAR UNA BASE DE DATOS.....	89
TABLA XVII. FUNCIONES EMPLEADAS EN LA CONSTRUCCIÓN DEL ALGORITMO DE RECONOCIMIENTO DE LA MANO.	95
TABLA XVIII. IMPLEMENTACIÓN DEL ALGORITMO PARA RECONOCER LA MANO..	96
TABLA XIX. IMPLEMENTACIÓN DEL CLASIFICADOR KNN.....	98
TABLA XX. IMPLEMENTACIÓN DEL CLASIFICADOR SVM.	99
TABLA XXI. PORCENTAJE DE PREDICCIÓN CLASIFICADOR SVM.	99
TABLA XXII. PORCENTAJE DE PREDICCIÓN CLASIFICADOR KNN.....	100
TABLA XXIII. ALGORITMO PARA RECONOCER EL ALFABETO DE SEÑAS.....	101
TABLA XXIV. PORCENTAJE DE PREDICCIÓN CLASIFICADOR SVM Y KNN.	101
TABLA XXV. IMPLEMENTACIÓN DE LAS PRUEBAS UNITARIAS.	104
TABLA XXVI. COBERTURA DEL PROYECTO.....	107
TABLA XXVII. NIVELES DE COBERTURA.	108
TABLA XXVIII. TALENTO HUMANO.	120
TABLA XXIX. RECURSOS MATERIALES.	120
TABLA XXX. RECURSOS TÉCNICOS.	120
TABLA XXXI. SERVICIOS BÁSICOS.	121
TABLA XXXII. PRESUPUESTO TOTAL.	121
TABLA XXXIII. GUÍA DE OBSERVACIÓN DIRECTA.....	138
TABLA XXXIV. CRONOGRAMA DEL PLAN DE INSTALACIÓN Y CAPACITACIÓN.....	162

c. Introducción

A nivel mundial el tema de la discapacidad es tratado en todos los gobiernos con su afán por afrontar hasta donde sea posible dicho problema, especialmente en los accesos a diversas áreas como educación, salud, transporte, etc. [1], por lo cual el uso de las herramientas tecnológicas provee de varias alternativas para su tratamiento. El empleo de la inteligencia artificial con sus variadas técnicas como, visión por ordenador, reconocimiento del lenguaje, aprendizaje automático, etc., han permitido la implementación de muchos sistemas, que son utilizados por personas cuyas capacidades no se encuentran dentro de lo “normal” [2].

En la asociación de sordos de Loja “Virgen del Cisne”, se capacita a personas con la discapacidad del oído, mediante la enseñanza del Lenguaje de Señas Ecuatoriano, enseñanza en la que existen problemas que afronta el tutor, ya que se debe explicar las posiciones de cada una de las señas a cada uno de los estudiantes, proceso que conlleva tiempo elevado y enseñanza lenta, los estudiantes por otro lado presentan dificultades para la comprensión del lenguaje, confusión al reconocer signos entre otras (Ver Anexo 5), que están ligadas a la enseñanza lenta.

Es por ello que el presente Trabajo de Titulación se enfoca en el desarrollo de un sistema para el aprendizaje del lenguaje de señas en el nivel básico, con el uso de la visión artificial en tiempo real que mejore la enseñanza-aprendizaje de las personas que conforman la asociación, todo el código implementado así como la documentación de la aplicación cuenta con licencias de software libre para que pueda servir como base en la construcción de sistemas de similares características.

La estructura del informe inicia con un resumen del Trabajo de Titulación en donde se detallan los aspectos más importantes del mismo, seguido de ello se realiza una breve introducción que permite que el lector conozca de antemano el contenido del informe, a continuación se presenta la revisión literaria la cual cuenta con cinco capítulos, el primero trata sobre la introducción a las discapacidades y los centros que brindan apoyo en todo el país, el segundo, tercero, cuarto y quinto capítulo muestran información acerca de la inteligencia artificial, la visión artificial como técnica, las metodologías ágiles de desarrollo más empleadas así como los diferentes algoritmos y clasificadores

que permiten la predicción del Lenguaje de Señas, seguidamente del marco teórico se describe los métodos, técnicas y metodología empleada en el desarrollo, luego de ello se presentan los resultados obtenidos en el proyecto, y finalmente el informe concluye con la discusión, los resultados obtenidos, las conclusiones y recomendaciones.

d. Revisión de Literatura

Capítulo I. Las Discapacidades

Las discapacidades son situaciones de las que nadie está exento, es por ello que tenemos que tener conocimiento acerca de esta temática, en el presente capítulo se da a conocer un concepto así como los tipos de discapacidades, el número de discapacitados por provincia en todo el Ecuador, y las empresas públicas y privadas que brindan capacitación a personas con este tipo de condición.

1.1 Qué es la discapacidad

La discapacidad es una deficiencia en un sujeto con la repercusión directa en su capacidad de realizar actividades en los términos considerados normales para cualquier sujeto de sus características (edad, género,...) [3]. En la actualidad la discapacidad se ve desde una perspectiva ecológica (es decir, desde la interacción persona-ambiente). El Instituto de Medicina de la OMS¹ (1991) sugiere que las limitaciones de una persona se convierten en discapacidad sólo como consecuencia de la interacción de la persona con un ambiente que no le proporciona el adecuado apoyo para reducir sus limitaciones funcionales [4].

1.1.1 Tipos de Discapacidad

A nivel mundial existen diversos tipos de discapacidad clasificados de acuerdo a los siguientes criterios [5]:

- **Física:** Se define como una imposibilidad en su capacidad motriz o corporal, esto significa que partes afectadas son los brazos y/o las piernas.
- **Psíquica:** representa una disminución en las habilidades adaptativas de un determinado individuo.
- **Sensorial:** este tipo de discapacidad se define como la disminución de la actividad de uno o varios sentidos como: la visión, oído, habla.

¹ Organización Mundial de la Salud.

- **Intelectual:** se refiere al bajo rendimiento en la capacidad intelectual y cognitiva. Algunas de estas discapacidades son Síndrome de Down, Retraso Mental.

1.2 Discapacitados en el Ecuador

Según los datos que maneja actualmente el Consejo Nacional de Igualdad de Discapacidades (CONADIS), en el Ecuador existe 2.49 % de discapacidad, todo este porcentaje se encuentra distribuido por provincias, en la TABLA I, se muestra el número de discapacitados por provincia recogidos tras el censo de mayo del 2013 [6,7]:

TABLA I. DATOS GENERALES POR PROVINCIA.

Provincia	Auditiva	Física	Intelectual	Lenguaje	Psicológico	Visual	Total
Azuay	2584	14681	4935	356	642	2864	26062
Bolívar	1000	2271	1253	156	161	851	5692
Cañar	825	2994	1544	184	280	695	6522
Carchi	1033	2294	904	75	245	549	5100
Chimborazo	2488	5214	2959	165	146	1192	12164
Cotopaxi	1361	4014	2071	283	193	1166	9088
El oro	1602	8145	5413	166	764	1856	17946
Esmeraldas	994	5944	3557	238	296	1633	12662
Galápagos	32	126	107	2	17	30	314
Guayas	8551	38929	20414	902	2866	9036	80698
Imbabura	2234	4238	1764	143	339	1040	9758
Loja	1553	5026	4198	154	620	1502	13053
Los ríos	1231	9882	3745	250	387	1729	17224
Manabí	3449	23495	5812	278	3239	5676	41949
Morona	336	1864	967	115	169	641	4092

Santiago							
Napo	492	1735	851	135	75	476	3764
Orellana	451	2166	761	115	186	925	4604
Pastaza	334	1078	584	28	79	324	2427
Pichincha	7951	23610	11122	712	2289	5981	51665
Santa Elena	969	4579	2077	84	162	809	8680
Santo domingo de los Tsáchilas	973	5069	1971	103	401	1120	9637
Sucumbíos	510	2287	1118	78	178	688	4859
Tungurahua	2055	4244	2502	172	321	930	10224
Zamora Chinchipe	397	1559	821	65	95	366	3303
Total	43405	175444	81450	4959	14150	42079	361487

El tipo de discapacidad que mayoritariamente prevalece es la Discapacidad Física e Intelectual, para la cual existen varias entidades públicas y privadas que han establecido programas para tratar de contrarrestar en gran medida dicho problema, en la provincia de Loja existen 17 fundaciones y asociaciones (ver TABLA II), que prestan atención a personas con este tipo de discapacidad [8]:

TABLA II. ORGANIZACIONES QUE TRABAJAN EN MATERIA DE DISCAPACIDADES EN LA PROVINCIA DE LOJA.

1	Asociación de Discapacitados Reina del Cisne.
2	Asociación de Discapacitados Virgen de la Nube.
3	Asociación de Invidentes de la Provincia de Loja.

4	Asociación de Padres Pro-niños y Jóvenes Excepcionales de Loja.
5	Asociación de Participación Social de Personas con Discapacidades Cantón Gonzanamá.
6	Asociación de Personas con Discapacidad Física del Cantón Loja.
7	Asociación de Pindaleños con Discapacidad.
8	Asociación de Sordos de Loja "Virgen del Cisne".
9	Asociación Levántate y Anda.
10	Centro de Educación Especial "Divino Niño Jesús".
11	Centro Educativo de Audición y Lenguaje.
12	Fundación de Acción Social.
13	Fundación para la Atención al Discapacitado.
14	Fundación Tiflológica Ecuatoriana Punto 7.
15	Instituto de Educación Especial Ciudad de Loja N° 2.
16	Instituto Especial Fiscal para Ciegos Byron Eguiguren "Club de Leones".
17	Patronato Provincial de Amparo Social de Loja.

De todas estas fundaciones y asociaciones únicamente la Asociación de Sordos de Loja "Virgen del Cisne", es la que presta ayuda a las personas en la capacitación del lenguaje de señas, existen en Loja 1553 personas con discapacidad auditiva y 154 personas con discapacidad del habla [6], aunque es un porcentaje pequeño en comparación con los otros tipos de discapacidad, se debe prestar igual atención a este problema ya que el objetivo de todas las normas establecidas es eliminar las diferencias que existen, y poder integrarlos en el desarrollo de actividades consideradas "normales", especialmente en la educación, trabajo y salud como valor esencial para el desarrollo. Ecuador se ha convertido en uno de los primeros países que cuenta con normas y disposiciones legales que amparan a las personas que presentan alguna discapacidad,

posee una constitución en la que se mencionan algunos artículos (Ver Anexo 4), en los que se pueden destacar la integración, atención así como la sanción a personas o instituciones que incumplan con este tipo de leyes [9].

1.2.1 Asociación de Sordos de Loja “Virgen del Cisne”



Figura 1. Asociación de Sordos de Loja "Virgen del Cisne".

Esta es una organización a nivel local y provincial fundada el 8 de marzo de 1994 en la que se contó con el asesoramiento de Federación Nacional de Personas Sordas del Ecuador (FENASEC) y personas de los Estados Unidos, esta asociación busca romper las barreras de comunicación entre las personas sordas, la familia y la comunidad oyente en todos sus ámbitos.

Misión

Difundir la Lengua de Señas Ecuatoriana a la comunidad de sordos de la región sur del país y trabajar conjuntamente por el reconocimiento y fortalecimiento de nuestra comunidad.

Visión

Promover a través de los socios de nuestra comunidad el efectivo uso y cumplimiento de sus derechos en procura de la igualdad en todas las estructuras sociales.

Objetivos

General

- Acoger a todas las personas sordas de la provincia y la región sur del país.

Específicos

- Promover la inclusión familiar laboral y social.
- Brindar apoyo a la familia con personas sordas.

Capítulo II. La Inteligencia Artificial.

Con la aparición de las primeras máquinas consideradas ordenadores, se hace posible la concepción del hombre por crear máquinas que sean capaces de realizar cosas que solo los seres vivos son capaces de hacer como pensar y tomar decisiones en base a diferentes circunstancias, con ello la inteligencia artificial comienza a cobrar vida, este capítulo trata sobre la definición así como la descripción de las diversas técnicas que actualmente la inteligencia artificial posee.

2.1 La Inteligencia

Es una facultad especial propia de ciertas clases de seres orgánicos que les otorga, junto con el pensamiento, la voluntad de obrar, la conciencia de la existencia, los medios para establecer relaciones con el mundo exterior así como atender sus necesidades [10]. Sin duda alguna una de las interrogantes que por mucho tiempo científicos tratan de dar respuesta sin embargo no han tenido ningún éxito.

2.2 Inteligencia Artificial

La inteligencia Artificial resulta de la combinación de la ciencia Informática con la Fisiología y la Filosofía [11], tiene como propósito la creación de máquinas que sean capaces de captar la información en el entorno, procesarla y poder tomar decisiones en base a ella. “La principal aplicación de esta ciencia es la construcción de máquinas para la automatización de tareas que requieran de un comportamiento inteligente”.
Claudia Herrera.

2.3 Técnicas de la Inteligencia Artificial

Existen varias técnicas de la inteligencia artificial que permiten la implementación computacional, de acuerdo a los requerimientos que surgen para un determinado problema, entre las más conocidas tenemos:

2.3.1 Redes neuronales artificiales (Artificial Neural Networks)

Se constituyeron inicialmente como una simulación abstracta de los sistemas nerviosos biológicos, formados por un conjunto de unidades llamadas neuronas o nodos conectados unos con otros. Las conexiones de estos nodos se asemejan a las dendritas y axones de los sistemas nerviosos biológicos [12], las neuronas se modelan mediante unidades de proceso. Cada unidad de proceso se compone de una red de conexiones de entrada, una función de red (de propagación), encargada de computar la entrada

total combinada de todas las conexiones, un núcleo central de proceso, encargado de aplicar la función de activación, y la salida, por donde se transmite el valor de activación a otras unidades.

2.3.1.1 Ventajas de las Redes neuronales

El empleo de las redes neuronales presentan algunas ventajas, a continuación se describe cada una de ellas:

- **Son sistemas distribuidos no lineales:** una neurona es un elemento no lineal por esta razón puede tomar cualquier valor cuando está en conjunto con otras neuronas.
- **Son sistemas tolerantes a fallos:** una red neuronal al ser un sistema distribuido permite el fallo de algunos elementos individuales [13].
- **Adaptabilidad:** esta es una capacidad especial ya que las redes neuronales pueden adaptarse a entornos inicialmente no conocidos.
- **Establecer relaciones no lineales entre datos:** puede relacionar diferentes tipos de datos y compararlos con la probabilidad de ocurrencia.

2.3.2 Ingeniería del conocimiento (Knowledge Engineering)

Es parte de la inteligencia artificial, su estudio se basa en el diseño y desarrollo de Sistemas Expertos o Sistemas Basados en el Conocimiento (SBC) [14], el trabajo que tiene que desempeñar un Ingeniero de Conocimiento es obtener todo el conocimiento generado por expertos humanos para que luego sea codificado e integrado a un sistema.

Una de las tareas más difíciles es la adquisición de la información para reducir la complejidad, para realizar este proceso se han de seguir una secuencia de pasos:

1. **Identificación del problema/s:** se debe realizar el análisis correspondiente para determinar si el problema es conveniente resolverlo mediante la técnica de los Sistemas de Base de Conocimiento.

2. **Conceptualización:** en este paso se debe entender las características del problema, así como los detalles que conllevan para la resolución por parte del experto.
3. **Formalización:** se debe considerar diferentes esquemas de razonamiento para poder modelizar la información recogida.
4. **Implementación:** se deberá tomar la decisión sobre el modo de representar el conocimiento.
5. **Prueba:** Probar el correcto funcionamiento del sistema y en caso de descubrir errores regresar a los pasos anteriores para su corrección.

2.3.2.1 Ventajas de la implementación de la Ingeniería de Conocimiento

La construcción de sistemas con este tipo de técnica presentan las siguientes ventajas:

- **Respuestas no subjetivas:** Los sistemas ofrecen respuestas sólidas ya que no poseen ninguna clase de emociones.
- **Explicación del razonamiento:** El sistema experto puede explicar clara y detalladamente el razonamiento que conduce a una conclusión [15].
- **Respuesta rápida:** A diferencia de los humanos estos sistemas procesan la información y emiten las respuestas de manera muy rápida.
- **Coste reducido.**
- **Permanencia:** La experiencia es permanente, a diferencia de lo que ocurre con los expertos humanos.

2.3.3 Procesamiento del lenguaje natural (Natural Language Processing)

Esta es una disciplina de la Inteligencia Artificial que se ocupa de la investigación de mecanismos computacionales para la comunicación entre personas y máquinas mediante el uso de un lenguaje común (Lenguajes Naturales).

Además permite a las personas comunicar el conocimiento que poseemos del mundo, mediante el empleo de alguna de las siguientes interrogantes [16]:

- **Informar:** no es necesario volver sobre caminos ya visitados. Emitimos y recibimos información constantemente a través del lenguaje.
- **Preguntar:** el lenguaje nos permite recabar información del resto del mundo.
- **Compartir:** sobre todo las experiencias para evitar pérdidas de tiempo volviendo a realizar lo que ya se ha hecho.

Realizando estudios acerca del lenguaje podremos entendernos y entender el mundo que nos rodea, pero no se puede realizar todo esto de forma rápida ya que existen algunos inconvenientes como:

- **Descripciones Incompletas:** esto hace imposible la interpretación rápida de la comunicación es por ello que se debe extraer parte del contexto del lenguaje.
- **Ambigüedad de Significado:** una misma palabra puede tener varios significados en una misma conversación.
- **No completitud:** no se podrá tener un lenguaje completo, ya que cada vez se agregan nuevas palabras expresiones y significados.

2.3.4 Minería de datos (Data Mining)

La minería de datos es una exploración sobre un gran volumen de datos de manera que con esa valiosa información se pueda predecir, pronosticar hechos sobre la toma de decisiones. Lo que en verdad hace el data mining es reunir las ventajas de varias áreas como la Estadística, la Inteligencia Artificial, la Computación Gráfica, las Bases de Datos

y el Procesamiento Masivo, principalmente usando como materia prima las bases de datos ya que contienen toda la información de una determinada actividad [17].

Algunas de las ventajas de la minería de datos son [18]:

- **Los modelos son fáciles de entender:** en la que personas sin una importante experiencia en estadísticas pueden interpretar el modelo y compararlo con sus propias ideas.
- **Enormes bases de datos pueden ser analizadas:** estas bases de datos pueden ser enormes tanto en largo como en ancho.
- **La minería de datos descubre información que no se esperaba obtener:** varios estudios, han descubierto que combinaciones particulares de factores entregan efectos inesperados que entregan valor a la compañía.
- **Los modelos son confiables:** El modelo es probado y comprobado usando técnicas estadísticas antes de ser usado.
- **Los modelos se construyen de manera rápida:** el modelado se torna mucho más fácil puesto que muchos algoritmos son probados y sólo el mejor modelo es entregado al usuario.

Capítulo III. Visión Artificial

La visión artificial ha permitido que las industrias mejoren su producción, junto con el empleo de equipos electrónicos se hace posible realizar las tareas de los seres humanos con mayor rapidez y precisión, en el presente capítulo se describe la definición de la visión artificial así como las fases que se deben cumplir para el desarrollo de un sistema y finalmente los casos de éxito y fracaso que han permitido la construcción de nuevos y mejores sistemas.

3.1 Definición

Según Marr, “Visión es un proceso que a partir de imágenes del mundo exterior produce una descripción útil para el observador y no tiene información irrelevante” [19].

La Visión Artificial es una técnica de la Inteligencia Artificial, que tiene por objeto tratar de simular todo el proceso de visión de los seres humanos a través de una aplicación, mediante la captura, análisis, procesamiento y reconocimiento de imágenes captadas a través de un dispositivo.

3.2 La Imagen Digital

Una imagen digital es una celda que se encuentra compuesta por píxeles, en el que cada uno de estos píxeles ocupa un lugar en la memoria de la computadora y se encuentra representado por un número, este número compone el color y brillo de la imagen [20]. De la misma manera a la imagen digital se la puede considerar como una matriz con un número finito de filas y columnas Figura 2, la intersección de filas y columnas dan lugar a cada uno de los píxeles, estos poseen un valor entre 0 y 255 que representa el nivel de color [21].



Figura 2. Representación de la Imagen Digital².

3.3 Componentes de un sistema de visión artificial

- **Fuente de iluminación:** es la característica esencial de la visión por computador, ya que de ella depende poder captar una escena, aplicando la técnica adecuada se puede lograr exactitud en los resultados para los que el sistema es desarrollado, a la iluminación se la puede manipular de forma artificial mediante el empleo de lámparas Led, Laser, Fibra Óptica, Fluorescente, etc., [22]. La importancia de la iluminación radica en los cambios bruscos de la luz en los lugares donde el sistema será puesto en marcha.

Es así que el sistema de iluminación debe cumplir ciertas características que influye directamente sobre las prestaciones del sistema de visión. Con condiciones de iluminación inestable o variable, ocasiona un pre-procesamiento muy costoso y complejo de la imagen, puesto que la imagen no es más que la representación de la información de la luz [23].

- **Cámara:** es el dispositivo que hace posible captar escenas. Además de la tecnología de fabricación, un aspecto importante del sensor de una cámara digital es su factor de forma. Los dos tipos básicos son lineal y matricial. Las cámaras lineales obtienen mucha menos información de la escena pero poseen una tasa de transferencia mucho más elevada que las cámaras matriciales

² Imagen Obtenida de OpenCV Tutorials.

- **Computador:** este es el dispositivo en el cual se procesarán las imágenes que son captadas a través de una cámara.

3.4 Fases en un sistema de visión artificial

Durante el proceso que un sistema realiza para la solución de un determinado problema mediante el uso de la visión artificial, se siguen cuatro fases esenciales que le permiten optimizar el tiempo y los recursos, a continuación se describen cada una de las fases que implementan este proceso [24]:

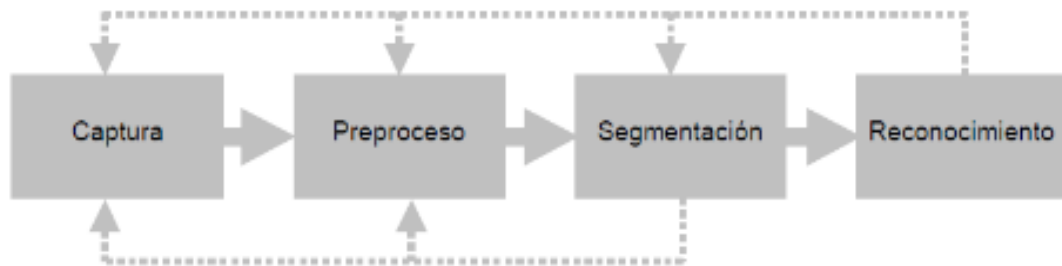


Figura 3. Fases en un sistema de Visión Artificial³.

1. **Captura:** En esta fase se recopila la información (las imágenes).
2. **Procesamiento:** Con la ayuda de otros algoritmos se delimitan las regiones de interés y se eliminan partes no deseadas de las imágenes, de manera que facilite las etapas posteriores.
3. **Segmentación:** se aíslan los elementos que sean de nuestro interés del resto de la imagen.
4. **Reconocimiento:** Con la ayuda de un análisis previo se comparan las características obtenidas con las generadas en ese momento y se establecen las diferencias entre objetos.

³ Imagen obtenida de: Visión por Computador para UAS.

Estas fases se siguen de manera secuencial en un sistema habitual de visión artificial Figura 3, más puede suceder que no sigan esa secuencia siempre, sino que se realimenten hacia atrás en el ciclo. Así, si la segmentación falla, se puede volver al procesado o a la captura para intentar arreglar el fallo [25].

3.5 Casos de éxito que hacen uso de la visión artificial.

Actualmente la Visión por Computador o Visión Artificial ha tomado gran importancia ya que permite una mayor productividad en actividades que son repetitivas, estas técnicas son empleadas con mayor frecuencia en procesos de inspección y ensamblaje donde los humanos a pesar de hacerlo bien emplean un tiempo considerable retrasando así ciertos procesos.

Todos los sistemas existentes que hacen uso de la visión artificial pueden ser clasificados en:

- Aplicaciones Industriales (por ejemplo, control de calidad).
- Aplicaciones no industriales (por ejemplo, control del tráfico).

Las ventajas que se obtienen al momento de emplear un sistema en cualquiera de los campos antes mencionados se detallan a continuación [26]:

- Automatizar tareas repetitivas de inspección realizadas por operadores.
- Realizar controles de calidad de productos que no era posible verificar por métodos tradicionales.
- Realizar inspecciones de objetos sin contacto físico.
- Realizar la inspección del 100% de la producción (calidad total) a gran velocidad.
- Reducir el tiempo de ciclo en procesos automatizados.
- Realizar inspecciones en procesos donde existe diversidad de piezas con cambios frecuentes de producción.

3.5.1 Aplicaciones No industriales

Las aplicaciones no industriales que han sido implementadas con mayor frecuencia en distintos países a nivel mundial son las que a continuación se nombran:

3.5.1.1 Sistema de Reconocimiento Automático de Matrículas (ANPR)

El Primer Sistema de Reconocimiento Automático de Matrículas o Automatic Number Plate Recognition por sus siglas en inglés (ANPR), fue inventado en 1976 en la rama de desarrollo de la policía de Reino Unido, los primeros prototipos estuvieron en funcionamiento en 1979, luego de recibir varios contratos se logró producir varios sistemas para la industria [27]. En 1981 gracias a este tipo de sistema se pudo realizar el primer arresto por robo de vehículos, estos tipos de sistemas son aplicaciones de visión por computador que trabajan con un conjunto de dispositivos hardware y software para la captura de imágenes en tiempo real y el reconocimiento óptico y la extracción de caracteres (OCR) [26,27], son muchas las aplicaciones que se pueden realizar con este tipo de sistemas, desde el registro de entrada y salida en parqueaderos las 24 horas del día, permitir accesos a vehículos autorizados y proveer información del flujo vehicular [30].

El sistema de Visión Artificial se ejecutará sobre un PC convencional y estará enlazado con el resto de las aplicaciones o bases de datos, los procesos básicos que realizan estos tipos de sistemas es capturar imágenes de las placas vehiculares para luego con la ayuda de técnicas especializadas proceder a la binarización, segmentación y adelgazamiento de las mismas para luego mediante un algoritmo de OCR reconocer los caracteres que contiene dicha placa vehicular, una de las grandes dificultades que es agregada a este tipo de sistemas es que trabaja en un entorno no controlado, la luz, los reflejos, las condiciones climáticas y el horario en que se encuentre laborando hacen que varíe la iluminación [29], sin embargo una vez que esta información está procesada, será convertida en pequeños paquetes de datos, que se pueden transmitir fácilmente (de ser necesario) a otros dispositivos remotos para su almacenamiento y utilización.

3.5.1.1.1 Algoritmo para el reconocimiento de las placas vehiculares

Para la detección de las placas vehiculares se sigue un algoritmo en el que se procesa las imágenes, con el propósito de encontrar una región de interés, que contiene la información buscada, en la mayoría de los sistemas ANPR se siguen los pasos descritos a continuación [28,29]:

- **Adquisición de las imágenes.**

En este paso se obtienen las imágenes captadas desde un dispositivo.

- **Pre-procesado de las imágenes.**

Consiste en aplicar diferentes métodos de la visión artificial con el propósito de obtener mayor calidad en las imágenes o video, en esta etapa se realizan la binarización, detección de bordes, selección de contornos, etc.

- **Búsqueda de Candidatos.**

La entrada a esta etapa son las imágenes o video captado, las salidas serán dependientes del número de candidatos encontrados en el video que cumplan con las características mínimas como área y color de una placa, existe un filtro el cual permite la clasificación de candidatos con áreas inferiores a mil píxeles con el fin de eliminar el ruido.

- **Verificación y Segmentación.**

Las salidas de la fase anterior son tomadas como entradas para la selección de una región de interés (ROI) donde se la aislara de toda la imagen, dependiendo del ángulo de captura se podrán observar un número de placas.

- **Reconocimiento de caracteres.**

Finalmente una vez obtenido toda la región de interés que contiene la placa, mediante la aplicación de OCR se evaluarán cada uno de los caracteres por separado y mediante el conjunto de redes neuronales se obtendrán cada uno de los caracteres contenidos en la imagen de la placa.

3.5.1.2 Reconocimiento Óptico de Caracteres (OCR)

El OCR es una aplicación que mediante el software de visión artificial lee texto de una imagen, este proceso identifica automáticamente símbolos o caracteres de determinado alfabeto para luego almacenarlos en forma de datos que puedan ser manipulados a través de un editor de texto [32], el software inspeccionará una determinada imagen pixel a pixel, buscando formas que coincidan con los rasgos de los caracteres disponibles en las fuentes del programa de no conseguirlo tratará de identificar los caracteres a través del análisis de sus características, de tal manera que el reconocimiento de los mismos no se limite exclusivamente a un determinado número de

fuentes [33], se rechazarán los caracteres no válidos, aquellos que no cumplan con determinadas normas como: altura mínima y máxima, área mínima y máxima.

Aunque actualmente el OCR puede llegar a mantener la estructura de los documentos originales en el archivo de salida, e incluso reconocer caracteres contenidos en documentos manuscritos, diagramas, partituras, etc., no hay que olvidar que su nivel de efectividad sigue siendo limitado, lo que hace necesaria una posterior revisión y corrección manual del texto generado. En el mercado existe una amplia oferta de software y librerías OCR, cuyo nivel de reconocimiento está ligado a la precisión y efectividad.

Existen múltiples programas realizados aplicando las técnicas de visión artificial y el OCR, entre los más conocidos tenemos [27-33]:

- **Control de Accesos:** usando la matrícula a modo de “llave” o “mando” para acceder a un estacionamiento [35].



a. Ubicación de los equipos de captura.



b. Captura de imágenes de los objetos.

Figura 4. Sistema para el control de Acceso⁴.

La Figura 4, muestra una de las implementaciones del control de acceso al cuartel general de la policía municipal de Valladolid, el proyecto consiste en la instalación de dos cámaras de video que registran las entradas y salidas de vehículos de cualquier tipo (coches patrulla, motos, grúas, etc.). El objetivo es el registro automático evitando que un agente esté dedicado a esta función casi en exclusiva durante los tres turnos diarios, el dato de matrícula válido es procesado y almacenado en una base de datos que recoge los datos de entrada y salida de un vehículo con referencia de fecha y hora de cada evento. El sistema permite conocer en cada momento el parque de vehículos que se encuentran en el estacionamiento o fuera del mismo.

- **Control de camiones:** situando un lector de matrículas junto a la báscula que mide la carga del camión [35].

⁴ Imagen obtenida de: Técnicas de Gestión de Tráfico.



Figura 5. Control de Camiones⁵.

La Financiera Maderera S.A. (FINSA) en Santiago de Compostela (España), ha implementado un sistema, tanto en la entrada como la salida de la planta, se han instalado cámaras de reconocimiento de matrículas para registrar las matrículas delantera y trasera de los vehículos pesados, con la particularidad de que estos tienen distinta matrícula en la cabina y en el remolque por lo que el sistema debe desechar cualquier otra matrícula que no sea una trasera de remolque. Además, el sistema clasifica los vehículos en función de su nacionalidad y carga el dato automáticamente en el albarán de entrada y salida, junto con el dato de peso del vehículo.

- **Inventariado de vehículos:** además de capturar la imagen de la matrícula se podría adquirir imágenes adicionales del vehículo, para poder determinar el estado del mismo en el instante de ingreso al aparcamiento, en caso de sufrir algún siniestro.
- **Reconocimiento de Lenguaje de Signos ASL:** con el diseño de este proyecto se pretende identificar uno a uno cada símbolo del alfabeto de signos para personas sordas, este proyecto sirve como el inicio de un largo camino para facilitar la comunicación entre oyentes y personas sordas.

⁵ Imagen obtenida de: Técnicas de Gestión de Tráfico.



Figura 6. Lenguaje de Señas ASL⁶.

Este tipo de sistema deberá poder interpretar los Símbolos del Alfabeto Americano (ASL), Figura 6, y dar su significado en tiempo real. El programa interpreta los símbolos utilizando una cámara de video integrada en el sistema mediante la cual se realizarán capturas periódicas de los signos que haga el usuario e imprimirá estos de forma consecutiva a través de la pantalla [34,35].

- **Mejora del tráfico en un cruce regulado por semáforos, mediante un sistema basado en Visión Artificial**



Figura 7. Control de tráfico⁷.

⁶ Imagen obtenida de Reconocimiento automático de lenguaje de signos: Lenguaje ASL.

⁷ Imagen obtenida de "Mejora del Tráfico en un cruce regulado por semáforos mediante un sistema basado en Visión Artificial".

El concepto principal de este proyecto es la creación de un sistema inteligente para mejorar el tráfico, de aquí que este proyecto también se conozca como TEA (Traffic Enhancement Application). Instalando cámaras IP y antenas Bluetooth en la parte superior de estos semáforos se puede conocer el estado del tráfico en un determinado instante, para regular el tráfico de la manera más adecuada y equitativa. Los beneficios que pueden resultar de la aplicación de este proyecto son numerosos y una vez que se simule o se ponga en funcionamiento una prueba real a gran escala se podrán cuantificar con precisión.

El proceso que se realiza para la toma de decisiones se lo hace a través de los tiempos que el semáforo tarda en cambiar los colores así [38]:

1. Rojo:

- Capturan una imagen de la vía que regulan para un determinado instante, usando la cámara IP instalada.
- Buscan si algún equipo de emergencia se aproxima por la vía que regulan, usando la antena Bluetooth instalada.
- Envían la imagen de la vía que regulan y los datos de los equipos de emergencia que se aproximan (si existiesen) a un sistema, denominado centro de control de intersección.

2. Amarillo:

- Procesa las imágenes de todos los semáforos para un determinado instante.
- Procesa la información sobre equipos de emergencia que se aproximan a la intersección (si existiesen).
- A través de su sistema de toma de decisiones y de los datos que maneja, manda la decisión óptima a los semáforos para que estos la ejecuten y regulen el tráfico acorde a la decisión tomada.

3. Verde:

- Ejecutan la decisión tomada por el centro de control de intersección.

3.5.2 Aplicaciones Industriales

El campo de la industria es el más beneficiado con la construcción de aplicaciones que facilitan las labores repetitivas que inicialmente eran realizadas por el hombre, a continuación se describen las aplicaciones más relevantes y los campos donde se las aplica:

3.5.3 Aplicaciones para la Inspección y Control de Calidad

La inspección y control de calidad se refiere a la comprobación visual o mecánica de qué objetos cumplan con determinados criterios, una de las finalidades de los controles de calidad consiste en detener la producción de algún producto si el sistema de producción comienza a generar productos que no cumplen con las normas estándares generales, la inspección de un objeto manufacturado puede tomar muchas formas, que podría involucrar las siguientes tareas [39]:

- a) Verificar la presencia de cada característica esperada.
- b) Verificar las dimensiones de esas características (por ejemplo radio y longitud de un cilindro).
- c) Verificar las interrelaciones entre características (por ejemplo distancias entre centros de gravedad y ángulos entre planos).

3.5.3.1 Algoritmo aplicado en el control de Calidad

Los métodos empleados en el algoritmo para el control de calidad van desde los aspectos más sencillos como la captación de las imágenes hasta culminar con el reconocimiento y la interpretación de las formas de los objetos para los cuales se requiere la inspección, a continuación serán descritas cada una de las etapas que se recorren antes de poder emplear la inspección y el control de calidad en diferentes campos de la vida real [38-40]:

- **Etapas de Captura**

El primer paso del proceso para el sistema de visión por computador es la adquisición y la digitalización de las imágenes, el objetivo es realzar las técnicas fotográficas, para las características visibles del objeto a procesar.

- **Etapas de Pre-procesamiento**

En la etapa de captura las imágenes no son aún directamente utilizables razón por la cual se hace necesario, el pre-procesamiento, que permite la corrección de problemas en la iluminación y contraste además permite la aplicación de transformaciones que acentúan las características que deben ser extraídas facilitando así la aplicación de operaciones en etapas posteriores.

Algunas de las operaciones que en la etapa de pre-procesamiento son:

- **Conversión de color o niveles de Gris:** mejora en la calidad de la imagen.
- **Transformadas geométricas:** este tipo de método no modifica la información únicamente el aspecto visual como rotar, trasladar, zoom, etc.
- **Transformación de Histogramas:** permite la modificación de la imagen ya que facilita la segmentación de los objetos.
- **Filtrado secuencial y Frecuencial:** generan importantes cambios en la imagen original.

- **Etapas de Segmentación**

Con la ayuda de esta etapa es fácil la separación de dos o más regiones de una imagen a partir del histograma, de la misma manera el análisis de los bordes permiten la extracción de características puntuales en cuanto a la orientación forma y dimensiones. Una vez obtenidos los bordes de la imagen es necesario seguir segmentando la imagen con el propósito de aplicar algoritmos de medición, inspección y control.

- **Etapas de Extracción y Selección de Características**

Una vez que se tiene una imagen lo más definida posible es necesario extraer un conjunto de características o también denominado Vector de Características, este tipo de vector representa información cuantitativa que son básicas en la definición de una clase, estas características pueden ser geométricas y topológicas.

1. **Geométricas:** representan algún tipo de propiedad geométrica como: área, perímetro, etc.
 2. **Topológicas:** representan propiedades que tienen que ver con la estructura del objeto como: número de hoyos, etc.
- **Etapa de Reconocimiento e Interpretación de Formas**

Es la última etapa del algoritmo en el que se asignan nombres de acuerdo a la información que muestran sus descriptores o patrones, para el logro eficiente del reconocimiento se emplean técnicas de reconocimiento geométrico de formas como el aprendizaje supervisado, algoritmos de clasificación no supervisado, clustering, redes neuronales, etc.

Algunos de los proyectos que han sido implementados en empresas a nivel mundial son:

- **La detección de anomalías en envases de vidrio**



Figura 8. (a) Inspección Lateral, (b) Inspección de Fondo.

Uno de los problemas a la hora de envasar un producto alimenticio en frascos de vidrio es la presencia en la línea de llenado de algunos envases con defectos del vidrio o cuerpos extraños. Esto obliga a la incorporación de un puesto de inspección que detecte las anomalías del envase, para la realización de este puesto de inspección se propone un sistema de inspección automática utilizando técnicas de visión artificial.

Para llevar a cabo la inspección se lo puede dividir al problema en dos tipos [43]:

- **Inspección lateral:** Figura 8 (a), en donde se inspecciona las paredes laterales del frasco en busca de defectos como, vidrios pegados, rajaduras, defectos del tipo “hilos de teléfono”, etc.
- **Inspección de fondo:** Figura 8 (b), en donde el objeto de interés es el fondo del frasco y se buscan defectos de los denominados “púas de macho” y vidrios pegados o sueltos.

En todos los casos los cuerpos extraños que no sean de vidrio resultan fácilmente detectables.

- **Selección y control de calidad en frutas**



Figura 9. Control de calidad en fresas⁸.

La calidad de la fresa está directamente influenciada por los diferentes defectos que pueda presentar el producto, muchos de estos productos presentan alteraciones como manchas en distintos colores, dependiendo de los defectos presentados, las fresas pueden ser desechadas por no estar aptas para el consumo humano.

⁸ Imagen obtenida de “Sistema de Visión Artificial para la inspección selección y control de calidad de fresas”.

Los diferentes defectos presentados en la piel de la fresa son diferenciados principalmente de acuerdo a su color [39-42]:

- **Detección de defectos claros:** este defecto es ocasionado por un hongo tipo moho al tratarse como hongo es clasificada como una enfermedad por ende la fresa no es apta para el consumo en fresco.
- **Detección de defectos oscuros:** en la Figura 8, se observa este tipo de defecto, se puede deber a magulladuras, maduración excesiva o picaduras de insectos, siempre que la suma de los defectos no ocupe más del 20% de la fruta puede ser aceptada, si la suma de los defectos supera el 20% la fruta será clasificada como no apta para el consumo.

3.6 Casos de fracaso que hacen uso de la visión artificial.

Los sistemas que hacen uso de la visión artificial han sido desarrollados de acuerdo a los objetivos planteados inicialmente sin embargo como es una técnica que actualmente se encuentra en pleno auge, existen diferentes métodos que pueden ser utilizados para dar solución a un determinado problema, la elección de cada uno de ellos depende exclusivamente del análisis que haga la persona encargada.

Es una ventaja que no exista información en la que el empleo de la visión artificial resulte inadecuado, sin embargo habrán procesos en los que la solución con recursos humanos sea la más adecuada, no obstante, la solución con recursos humanos es menos estructurada que la solución artificial, el desarrollo de los algoritmos implican el empleo de métodos de prueba y error, como consecuencia de ello los artículos revisados tienen un punto importante denominado trabajos futuros, a través de los cuales los desarrolladores ponen a consideración, lo que se puede mejorar de acuerdo a la experiencia ya adquirida a los posibles defectos o problemas que posean los proyectos propuestos.

De esta manera se puede evidenciar algunas mejoras en los siguientes proyectos:

1. Sistema de Visión Artificial para el control de calidad en piezas cromadas

Durante el desarrollo del sistema fue posible observar aspectos que pueden complementar y mejorar el sistema de visión [42]:

- Complementar el sistema con una base de datos que contenga plantillas que cumplan con determinadas características, lo que generará la posibilidad que el sistema pueda reconocer otros modelos o accesorios ya que actualmente cuenta con un solo patrón de plantillas.
- Debido a la variabilidad de la luz ambiental, se recomienda diseñar e implementar un sistema de luz artificial, basado en fluorescentes utilizando la técnica de iluminación difusa, de manera que sea posible una iluminación uniforme y se reduzca el índice de reflexión. Con este sistema de iluminación se pretende que el sistema se adapte fácilmente a cualquier entorno de trabajo sin la necesidad de recalibrado.
- Con la finalidad de mejorar las estrategias de inspección y que se puedan procesar métodos más robustos es recomendable desarrollar un sistema de visión basado en hardware, este tipo de hardware permite realizar de manera más óptima el reconocimiento de patrones mediante clasificadores estadísticos y redes neuronales.

2. Reconocimiento automático de lenguaje de signos: Lenguaje ASL

Los puntos importantes de este proyecto son la segmentación por umbral y la extracción de características que nos permite realizar la clasificación de signos [36]:

- Se puede mejorar la extracción de características realizando diferentes fases de segmentación para cada captura de imagen, teniendo en cuenta las variaciones de tono y luminosidad de la piel de la mano.
- Se puede reducir también el tiempo de cálculo escogiendo características de la imagen que requieran un menor tamaño del vector previo a la fase de clasificación, ya que cuanto mayor es la longitud del vector, mayor es el tiempo de cálculo en la fase de clasificación.
- Este programa en concreto está sujeto a una serie de restricciones de uso, como la utilización de fondos e iluminaciones adecuados.

3. Mejora del tráfico en un cruce regulado por semáforos

Las siguientes directrices recogen una serie de mejoras que se pueden realizar a este proyecto [38]:

- Mejorar el tiempo de respuesta del proceso de tratamiento digital de imágenes, puesto que varía en función del número de elementos que detecta en la imagen y que deben ser clasificados o no como vehículos.
- Detección de vehículos especiales: no ha sido posible realizar una detección automática de vehículos de emergencia basada exclusivamente en el tratamiento de imágenes, una posibilidad más sofisticada sería habilitar la posibilidad de recibir mensajes de prioridad de paso. Estos mensajes podrían estar emitidos por los sistemas de control de flotas de los equipos de emergencia, ya que estos conocen la posición y trayectoria inmediata de estos vehículos.
- Mejorar el algoritmo de estimación de la velocidad de vehículos que se aproximan a la intersección, aplicando cálculo trigonométrico se puede obtener una estimación precisa de la velocidad de los vehículos en cada momento, que puede resultar interesante como dato del estado del tráfico.

Capítulo IV. Elementos para el empleo de la Visión Artificial.

Los sistemas de visión artificial son sistemas que requieren de elementos computacionales y de captura para realizar su función, estos sistemas necesitan de hardware que permita la adquisición de las imágenes, así como de un software, que nos permita programar las tareas específicas que va a realizar este tipo de sistema, y las librerías que ayudan durante el procesamiento de las imágenes, a través de este capítulo se dará a conocer todas las herramientas que ayudarán durante el proceso de construcción del sistema, los sistemas operativos, los lenguajes de programación, las librerías y el hardware que es compatible con el uso de la visión artificial.

4.1 Componentes Software

El empleo de este tipo de componentes sirve como la base para la ejecución de sistemas que hagan uso de la visión por ordenador.

4.1.1 Sistema Operativo

El desarrollo de un sistema de visión artificial se lo puede realizar sobre cualquier plataforma o Sistema Operativo (S.O), el requisito principal que debe poseer este, es tener soporte para el uso de las librerías de visión artificial, actualmente todos los S.O disponibles en el mercado poseen estas características, sin embargo se debe tener en cuenta características que permitan el uso adecuado del mismo, en la siguiente tabla se presentan algunas ventajas y desventajas de los S.O más conocidos [45–47].

TABLA III. VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS OPERATIVOS MÁS CONOCIDOS.

Sistema Operativo	Ventajas	Desventajas
Windows	<ul style="list-style-type: none">• Interfaces amigables al usuario.• Múltiples versiones.	<ul style="list-style-type: none">• Falta de estabilidad en muchas versiones.• No gratuito y poco flexible.• Requiere de equipos potentes y con alta memoria.

Linux	<ul style="list-style-type: none"> • Sistemas estables y seguros. • Software libre. • Flexible. • Gran seguridad. • No cuenta con una sola empresa que lo respalde. 	<ul style="list-style-type: none"> • No fácil de usar por usuarios corrientes. • Documentación y terminología muy técnica. • Incompatibilidad con programas de Windows.
Mac	<ul style="list-style-type: none"> • Basado en Linux. • Mejor Interfaz gráfica del mercado. • Utilización sencilla. • Gran seguridad. 	<ul style="list-style-type: none"> • Hardware Exclusivo. • Dificultad al encontrar software. • Costoso.

4.1.2 Herramientas de desarrollo

Este tipo de herramientas permiten la implementación de aplicaciones de manera lógica ya que brindan los componentes necesarios para su desarrollo.

4.1.2.1 Lenguajes de programación

Existen múltiples lenguajes de programación en el mercado que nos sirven durante la construcción de sistemas o programas, muchos de ellos con licencias libres y comerciales sin embargo antes de la elección de un lenguaje de programación se debería hacer un análisis de las características o necesidades que se requerirán en el proceso de construcción, así como el debido soporte que brinde con las librerías de visión artificial.

Algunos de los lenguajes de programación que poseen la capacidad para integrar las librerías de visión artificial son descritos en la siguiente tabla [48–54]:

TABLA IV. LENGUAJES DE PROGRAMACIÓN QUE SOPORTAN VISIÓN ARTIFICIAL.

Lenguaje de programación	Ventajas	Desventajas
Java	<ul style="list-style-type: none"> • Multiplataforma. • Es de distribución Libre. • Portable. • Lenguaje Simple. • JRE para cualquier plataforma. • Orientado a objetos. 	<ul style="list-style-type: none"> • La velocidad.
Matlab	<ul style="list-style-type: none"> • Amplio soporte. • Alta precisión. 	<ul style="list-style-type: none"> • Comercial. • Problemas eventuales de velocidad.
C++	<ul style="list-style-type: none"> • Rapidez. • Orientado a Objetos. • Compilado. 	<ul style="list-style-type: none"> • No portable. • Mayor control = Mayor dificultad.
.Net	<ul style="list-style-type: none"> • Posibilidades de utilización de amplia gama de lenguajes. 	<ul style="list-style-type: none"> • No disponible en otras plataformas.
Python	<ul style="list-style-type: none"> • Multiplataforma. • Comunidad. • Multiparadigma. • Código Abierto. 	<ul style="list-style-type: none"> • Servidores no poseen Python. • Configuración difícil.

4.1.2.2 Librerías de desarrollo

Las librerías que con mayor frecuencia se las emplea en la construcción de sistemas de visión artificial, por criterios como:

- Documentación existente.
- Facilidad en el uso.
- Cantidad de funciones disponibles en su API.
- Cantidad de aplicaciones desarrolladas.

Son JavaVis, OpenCV, Image Processing Toolbox, X-vision cada una de estas librerías son descritas a continuación:

4.1.2.2.1 JavaVis

La librería de visión artificial JavaVis, Figura 10, fue desarrollada en el Departamento de Ciencia de Computación e Inteligencia Artificial de la Universidad de Alicante, empieza a desarrollarse como un trabajo de la asignatura Sistemas Informáticos de Ingeniería en Informática [55]. En esta librería se integran clases y métodos para dar soporte a los algoritmos que se deseen implementar, con el uso de clases de soporte básicas para: definir el tipo de datos de imagen, manejar lectura y escritura de ficheros, usar una interfaz gráfica que nos permita ver el resultado de aplicar determinados algoritmos. En base a estas clases básicas, se han desarrollado distintos métodos “clásicos” y útiles de visión computacional, de esta forma, disponemos de un conjunto de métodos de los que podemos hacer uso para desarrollar métodos más complejos [25].

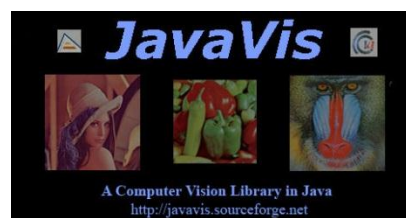


Figura 10. JavaVis⁹.

Esta librería se encuentra disponible desde su página oficial¹⁰, se trata de una librería de uso libre y está disponible para los sistemas operativos Windows y Linux, La librería se basa en clases básicas, que sirven para poder desarrollar métodos más complejos. Las principales ventajas que presenta son [56]:

- Adaptada para plataformas portátiles.
- Posibilidad de ampliación fácilmente.
- Una interfaz gráfica.

⁹ Imagen obtenida de JavaVis.

¹⁰ Página oficial JavaVis <http://javavis.sourceforge.net/index.html>

- Posee un formato de imagen especial que permite almacenar secuencias de imágenes y datos de tipo geométrico.

En la TABLA V, se muestra las funciones básicas que posee la librería:

TABLA V. MÉTODOS Y ALGORITMOS DE JAVAVIS.

Algoritmos	Descripción
Formato de Imagen	Se dispone la posibilidad de manejar secuencia de imágenes distinguidas entre frames.
Datos Geométricos	Permite operar con datos (puntos, segmentos, polígonos y aristas).
Formato de Fichero	JavaVis dispone de un formato de fichero propio, además permite la lectura de jpeg, gif, etc.
Funciones	<p>Una función en java es una clase que define un algoritmo.</p> <ul style="list-style-type: none"> • Transform: permite conversión de formatos de una imagen: FColorToGray, FGrayToGray, FGrayToColor, FRGBToHSB, frgb-ToHSI, FRGBToYCbCr, FBinarize. • Smoothness: reduce el ruido de las imágenes: FSmoothAverage, FSmoothMedian, FSmoothGaussian. • Manipulation: manipula una imagen, girar, escalar: FOpenWindow, FSkew, FFlip, FFuzzyKmeans, FK -medias, FMirror, FScale, FRotate. • Geometry: aplica funciones geométricas a las imágenes: FRandomPoint, FRandomSegment, FInterSegment, FGeoToGray, FHoughCirc, FHough -Line. • Edges: trabaja con los bordes: FCanny , Flink , FNitzberg , FSusan • Math: permite implementar operadores morfológicos: FErode, FDilate. • ImageBD: permiten las búsquedas en las bases de datos: FCalcHistoColor, FCalcHistoBD, FSearchImage, FSOM.

	<ul style="list-style-type: none"> • Other: FHistogram, Skeleton, FOp, FSegmentHSB, FBlobs, FManhattan, FCapture, FAnnealing, FPca, FPcaRecon, FFlow.
Ejecución de Funciones	<p>La ejecución de funciones se la puede realizar por tres métodos:</p> <ol style="list-style-type: none"> 1. Se puede ejecutar desde consola del sistema operativo. 2. Se pueden ejecutar desde la interfaz gráfica. 3. Se puede llamar una función desde otra, para conseguir funciones más complejas.

4.1.2.2.2 OpenCV

OpenCV (Open Source Computer Vision), Figura 11, es una librería de funciones de programación dirigida principalmente a la visión por ordenador en tiempo real, desarrollada por Intel en 1999 y ahora con el apoyo de Willow Garage. La biblioteca fue originalmente escrita en C, lo que permite que OpenCV sea portátil para algunas plataformas específicas, Windows, Linux, MacOS e incluso móviles como Android o IOS, al ser una biblioteca multiplataforma una de sus principales ventajas es que puede ejecutarse en ordenadores sin grandes prestaciones, pudiendo realizarse aplicaciones de visión bastante sofisticadas y a buena velocidad [56].



Figura 11. OpenCV¹¹.

OpenCV está disponible desde su página oficial¹², desde la versión 2.0 OpenCV incluye tanto su interfaz de C tradicional, así como una nueva interfaz C++. Esta nueva interfaz busca reducir el número de líneas de código necesarias para codificar la funcionalidad de la visión, así como de reducir los errores de programación comunes, tales como

¹¹ Imagen obtenida de OpenCV.

¹² Página oficial <http://opencv.org>.

pérdidas de memoria (a través de la asignación automática de datos y cancelación de asignación), que pueden surgir al utilizar OpenCV, aproximadamente posee 300 funciones escritas en el lenguaje C.

Incluye cinco componentes diferentes para la visión por computador en tiempo real [25-58]:

TABLA VI. COMPONENTES Y FUNCIONES DE OPENCV.

Algoritmos	Descripción
core	Define las estructuras de datos básicas, incluyendo las funciones básicas utilizadas por todos los demás módulos: Vec, CvPoint, CvSize, CvRect, CvMat, IplImage, CreateImage, GetImageROI, circle, ellipse, line, rectangle.
imgproc	Contiene funciones de procesamiento de imágenes como transformaciones, deformaciones, perspectiva genérica, conversión de espacio de color, histogramas, etc.: Gaussian, Threshold, cvtColor, findContours, drawContours, convexHull, convexityDefects, Canny, HoughCircles, HoughLines.
video	Módulo de análisis de vídeo que incluye la estimación de movimiento, la sustracción de fondo, y los algoritmos de seguimiento de objetos: calcOpticalFlow, CamShift, BackgroundSubtractor.
calib3d	Contiene algoritmos de geometría como, calibración de cámara, algoritmos de correspondencia estéreo y elementos de la reconstrucción 3D: calibrateCamera.
objdetect	Módulo para la detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, tazas, las personas, vehículos, etcétera): CascadeClassifier, CvObjectDetection.
	Interfaz fácil de usar para la captura de vídeo, imagen y codecs de

highgui	vídeo: createTrackbar, imshow, namedWindow, destroyWindow, imread, VideoCapture, VideoWriter, addText, createButton.
gpu	Algoritmos para aceleración de diferentes módulos OpenCV: ImageProcessing, Object Detection, ImageFiltering.

4.1.2.2.3 Image Processing ToolBox

MATLAB dispone en la actualidad de un amplio abanico de programas de apoyo especializados, denominados Toolboxes, 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 los “toolboxes” de adquisición, procesamiento de imágenes y el Matlab Guide que nos permitirá realizar nuestro programa en un ambiente visual [59].

Image Processing Toolbox es una librería que implementa a Matlab, proporciona un conjunto completo de algoritmos estándar de referencia, funciones y aplicaciones para el procesamiento, el análisis y la visualización de imágenes, así como para el desarrollo de algoritmos. Puede realizar mejoras de imágenes, enfoque de imágenes borrosas, detección de funciones, reducción de ruidos, segmentación de imágenes, transformaciones geométricas y registro de imágenes. Muchas de las funciones de la toolbox son multiproceso, para aprovechar los ordenadores con varios núcleos o procesadores.

Es una herramienta muy potente, posee una licencia comercial, proporciona un conjunto de algoritmos funciones y aplicaciones para la adquisición, análisis y procesamiento de imágenes algunas de las más importantes se listan a continuación [60]:

- Análisis de imágenes y estadística
- Diseño de filtros y recuperación de imágenes.
- Mejora de imágenes.
- Operaciones morfológicas y geométricas.

- Definición de mapas de colores y modificación gráfica.
- Operaciones con Región de Intereses.

Lo que permite que esta sea una de las herramientas adecuadas para el análisis y procesamiento de las imágenes, Matlab utiliza estructuras de datos básicos como los arrays, este tipo de variables hacen adecuada la representación de imágenes, MATLAB guarda la mayoría de las imágenes como arrays bidimensionales (matrices), en los que cada elemento de la matriz corresponde a un único píxel en la imagen mostrada. Así, por ejemplo, una imagen compuesta por 200 filas y 300 columnas de puntos de color diferente sería guardada en MATLAB como una matriz de 200x300. Algunas imágenes, como las RGB2, requieren un array tridimensional, donde el primer plano de la tercera dimensión representa las intensidades de color rojo, la segunda el verde y la tercera azul. Esta convención hace que trabajar con imágenes en MATLAB sea similar a trabajar con cualquier otro tipo de matrices, y hace que toda la potencia que MATLAB ofrece al manejo de matrices sea accesible para aplicaciones de procesamiento de imágenes [61].

A continuación se detallan las características más relevantes de esta herramienta:

TABLA VII. VENTAJAS Y DESVENTAJAS DE IMAGE PROCESSING TOOLBOX.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Facilidad de programación. • Gran potencia de cálculo en operaciones con matrices. • Gran facilidad para obtener resultados gráficos. • Ideal para desarrollo teórico de algoritmos y demostraciones en enseñanza. 	<ul style="list-style-type: none"> • Requiere de una licencia.

En la TABLA VIII, se muestra algunas funciones con las cuales se pueden combinar para formar sistemas que contengan grandes prestaciones:

TABLA VIII. COMPONENTES Y FUNCIONES DE IMAGE PROCESSING TOOLBOX.

Algoritmos	Descripción
Import, Export, and Conversion	imread, imwrite, imfinfo, ind2gray, mat2gray
Display and Exploration	imshow, immovie, imtool, imageinfo, impixelinfo, immagbox
Geometric Transformation, Spatial Referencing, and Image Registration	imrotate, imresize, imref2d, imref3d,
Image Enhancement	roipoly, roipoly, roifill, imadd, imdilate, imerode
Image Analysis	houghlines, imfindcircles, graythresh, rgb2ntsc, xyz2uint16

4.1.2.2.4 X-Vision

Se trata de un entorno de trabajo modular y portable para el tracking visual. X-Vision ha sido diseñado para ser un entorno de programación de visión artificial en tiempo real, dando un alto rendimiento en estaciones de trabajo estándar provistos de una digitalizadora sencilla. Consiste en una pequeña serie de primitivas para realizar el seguimiento en imágenes, y un marco de trabajo para combinar las primitivas de tracking en sistemas para seguimientos más complejos [62].

La versión de X-Vision está escrita en C++, es de código abierto y se encuentra disponible desde su página oficial¹³, el código se compila con éxito bajo gcc/g++ en una variedad de plataformas, para utilizar el sistema, se necesita una fuente de entrada de vídeo, sin embargo es posible ejecutar a partir de secuencias de imágenes almacenados como archivos, el sistema consta de un conjunto básico de clases que

¹³ Página oficial <http://www.cs.jhu.edu/CIPS/xvision/>

definen una interfaz estándar para características rastreables, interfaces para pantallas y dispositivos de video, y algunas herramientas de software estándar.

4.2 Componentes Hardware

Los componentes hardware más importante en un sistema de visión artificial son:

- **Cámara**

Este elemento tiene como propósito la captura de las imágenes en tiempo real, no existe un tipo de cámara específico que se debe poseer para el desarrollo de un sistema que haga uso de la visión artificial, ya que cualquier dispositivo que permita la captura de video es adecuado, actualmente la mayoría de librerías de visión artificial no requiere tener una cámara instalada ya que constan con la característica de poder realizar el análisis y procesamiento en imágenes almacenadas en un medio físico como un disco duro, usb's, etc.

- **Computador**

A través de este dispositivo podremos analizar, procesar e interpretar las imágenes, además obtener un control total del sistema a desarrollar, de igual manera no se requiere un dispositivo con altas prestaciones, sin embargo se recomienda que se tome en cuenta los requerimientos para que el sistema operativo pueda funcionar adecuadamente, a continuación se detallan algunas características básicas que se debería tener para instalar el sistema operativo y la librería de visión artificial [63–66]:

TABLA IX. CARACTERÍSTICAS MÍNIMAS INSTALACIÓN DE S.O.

Sistema Operativo	Características	Características
Linux	Memoria RAM	1 Gb
	Disco	10 Gb
	Distro	-
Windows	Memoria RAM	2 Gb
	Disco	40 Gb
	Distro	XP o superior

Los datos mostrados en la TABLA IX, hacen referencia a las características que requiere cualquier distro de Linux para ser instalada y que funcione adecuadamente, mientras que las características descritas para Windows únicamente se pueden aplicar desde Windows XP en adelante.

4.3 Algoritmos para el reconocimiento de la mano

Dos de los algoritmos que con mayor frecuencia se los emplea en la detección de las manos son descritos a continuación [67]:

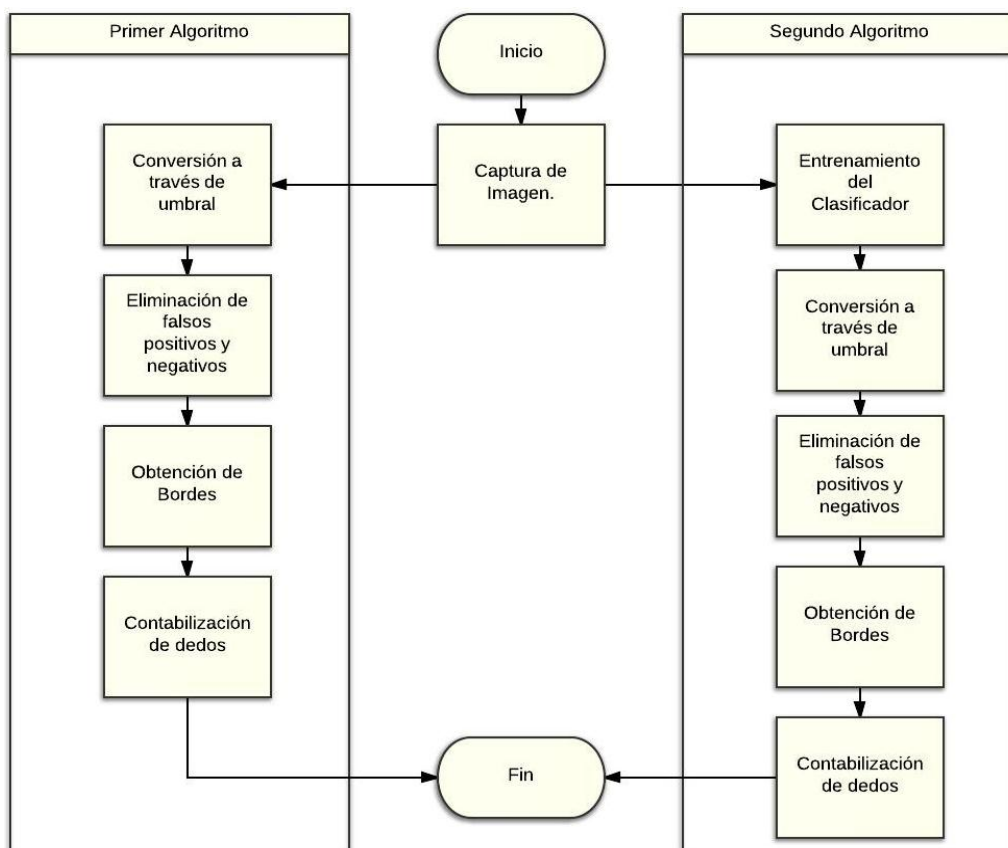


Figura 12. Algoritmos para la detección de la mano.

Una de las principales desventajas del primer algoritmo es que únicamente funciona en entornos controlados ya que en entornos no controlados los cambios de iluminación, los fondos variantes hacen que este tipo de algoritmo sea ineficiente, mientras que el segundo algoritmo es más eficiente pero requiere de altas prestaciones del equipo en el que se entrenará el clasificador.

4.4 Algoritmos para la detección de la mano

Cada uno de los pasos que se muestran en la Figura 12, se describen a continuación:

1. Captura de la imagen

Hace uso de dispositivos para la captación de imágenes o video en tiempo real, junto con el uso de las bibliotecas de visión artificial permiten clasificar los objetos que van a servir durante la ejecución del sistema.

2. Entrenamiento del clasificador

La librería OpenCV permite el entrenamiento de clasificadores que permiten la detección y el reconocimiento de casi cualquier clase de objetos, en la Figura 14, se presenta un algoritmo en que se hace uso de estos métodos para el reconocimiento del lenguaje de señas.

3. Conversión de imagen a través de un umbral

El sistema de visión por lo general utiliza un umbral binario para separar objetos del fondo. El valor apropiado de este umbral depende de la iluminación y las características de reflexión de los objetos, para ser eficaz en la separación de objetos de fondo y el objeto a seguir, es necesario que el objeto y el fondo tengan suficiente contraste [68]. Al ejecutar el algoritmo, se comprobarán todos los píxeles de la imagen y si el valor del píxel supera el umbral se etiquetará a un valor binario y si no es así, se le asignará el otro valor binario.

4. Eliminación de falsos positivos y negativos

Como el umbral que se selecciona para binarizar una imagen no es perfecto, los cambios de iluminación generan falsos positivos y negativos, los cuales no son más que pequeños grupos de píxeles que no pertenecen al objeto tampoco al fondo, en este paso se evitará que el reconocedor procese información que no pertenece a los objetos, a través del empleo de funciones de convoluciones.

5. Obtención de Bordes

La detección de bordes se basa en la detección de variaciones locales, que principalmente corresponden con los límites de un objeto en la imagen, pudiendo ser estas variaciones de color, intensidad, etc.

Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc., OpenCV trae consigo el filtro Canny para extraer los bordes de todos los objetos de la imagen capturada.

6. Contabilización de dedos

Una vez que se aplique la detección de bordes se podrá aplicar diferentes funciones de OpenCV como envolventes convexas la cual une todos los puntos que sobresalen de un objeto y los defectos de convexidad que son los puntos que no se encuentran sobre las envolventes convexas, siendo estos cada uno de los dedos de la mano.

4.5 Algoritmos para el reconocimiento del lenguaje de señas

OpenCV trae consigo algunas clases que permiten el entrenamiento de una red neuronal a través de la cual se podrá reconocer casi cualquier objeto, para el reconocimiento del alfabeto del lenguaje de señas se requiere únicamente de imágenes donde se muestran cada una de las posiciones que se realiza con las manos para representar una letra.

A continuación se muestra los algoritmos más utilizados en el proceso de detección de objetos.

1. Cascade Haar

Este tipo de clasificador fue desarrollado por Viola y Jones, está basado en árboles de decisión, Figura 13, combina un número elevado de reglas de baja precisión, para la obtención de un clasificador altamente confiable, es decir que se trata de una combinación lineal de clasificadores débiles para obtener clasificadores fuertes árboles de decisión [69].

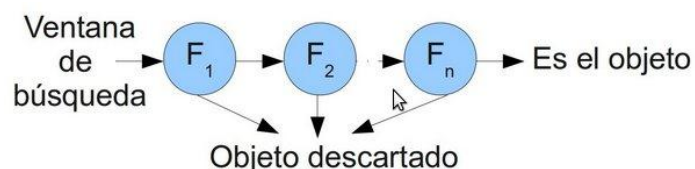


Figura 13. Representación del algoritmo Cascade Haar¹⁴.

¹⁴ Imagen tomada de OCW, Detección de Caras.

Para la creación de un clasificador de este tipo se emplea un conjunto de imágenes positivas y negativas para el proceso de entrenamiento así como para las pruebas, los pasos a realizar para crear el clasificador se describen a continuación [70–73]:

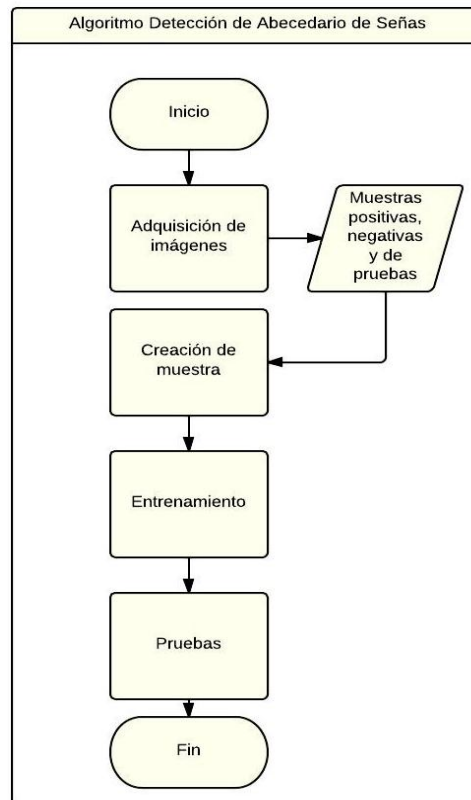


Figura 14. Algoritmo para la detección del abecedario de señas.

1. Adquisición de imágenes

Las imágenes empleadas para la clasificación son de dos tipos:

Muestras Negativas

Las muestras negativas son imágenes que no contiene el objeto a reconocer, puede usarse cualquier grupo de imágenes, se las lista en un archivo de texto plano, con el siguiente formato:

Ejemplo.

```
negativas-100.pgm
negativas-101.pgm
negativas-102.pgm
negativas-103.pgm
negativas-104.pgm
```


Muestras Positivas

Las muestras positivas son imágenes en las que aparece el objeto a reconocer, estas deben estar capturadas con distintos fondos e iluminación para que el clasificador se considere robusto.

Al igual que las muestras negativas se las debe listar, estas presentan algo más de información, ya que contienen las coordenadas exactas donde se encuentra el objeto a reconocer.

Ejemplo.

```
rawdata/100.bmp 1 47 2 200 197
rawdata/101.bmp 1 82 5 287 239
rawdata/102.bmp 1 161 5 239 242
rawdata/103.bmp 1 75 22 376 281
rawdata/104.bmp 1 77 10 351 301
```

Algo importante que se puede observar en los ejemplos es que tanto las muestras positivas como negativas deberán contener formatos en mapa de bits, para poder expresar las coordenadas correctas, algunos ejemplos de este tipo de formatos son: bmp, pgm, etc.

2. Crear muestra

La muestra se genera a través de la función `opencv_createsample`, a través de la que se emite un archivo vectorial (.vec) que contendrá las muestras de toda la información que servirá para el entrenamiento, los parámetros que recibe esta función son:

- **-info:** es la ubicación del archivo con el índice de las imágenes positivas.
- **-vec:** es el nombre del archivo de salida con la muestra generada.
- **-num:** cantidad de imágenes positivas.
- **-w:** ancho de la muestra de salida.
- **-h:** alto de la muestra de salida.

Ejemplo

```
opencv_createsamples -info positivas.txt -vec data/muestra.vec -num 1500 -w 20 -h 20
```

3. Entrenamiento

El entrenamiento se lo realiza a través de la función `opencv_traincascade` en la cual se integra la colección de muestras negativas y el archivo vectorial, de acuerdo a la

cantidad de imágenes el método tendrá una duración y un cierto número de etapas, concluido este proceso se creará un archivo `.xml` que será nuestro clasificador y el cual se lo podrá integrar a aplicaciones más complejas, a continuación se describen los parámetros que usa esta función:

- **-data:** directorio de salida para la generación del entrenamiento.
- **-vec:** archivo con muestras generado en la etapa anterior.
- **-bg:** archivo índice con las imágenes negativas.
- **-nstages:** número de etapas de entrenamiento.
- **-nsplit:** debilidad del clasificador.
- **-minhitrate:** mínimo rango de cada etapa.
- **-maxfalsealarm:** determina el rango máximo de falsas alarmas.
- **-npos:** número de imágenes positivas indexadas.
- **-nneg:** número de imágenes negativas indexadas.
- **-w:** ancho.
- **-h:** alto.
- **-mem:** memoria a utilizar en el proceso.

Ejemplo.

```
opencv-haartraining -data data/cascade -vec data/muestra.vec -bg negativas.txt -nstages 30 -nsplit 2 -minhitrate 0.999 -maxfalsealarm 0.5 -npos 1500 -nneg 5000 -w 20 -h 20 -mem 1300 -mode ALL
```

4. Pruebas

Las pruebas se las podrán realizar sobre un conjunto de imágenes que contengan o no el objeto a reconocer, se debe tener en cuenta que la creación de un clasificador robusto implica un número mayor de 3000 muestras positivas y 2000 negativas, es importante que el número de muestras positivas supere el número de las negativas.

2. Support Vector Machines (SVM)

El algoritmo de SVM permite realizar la separación entre datos para poder predecir resultados como en la Figura 15.

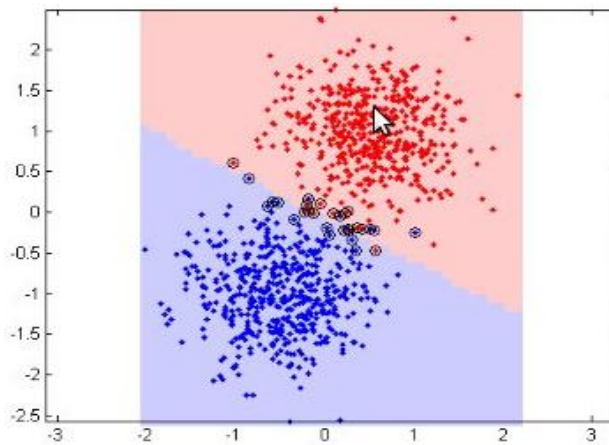


Figura 15. Representación del modelo SVM¹⁵.

Este algoritmo está compuesto por una serie de algoritmos que se encargan de encontrar una separación entre todo el conjunto de datos [74]. Dado un conjunto de ejemplos de entrenamiento (muestras) podemos etiquetar las clases manualmente y entrenar un clasificador SVM, para construir un modelo que prediga automáticamente la clase de una nueva muestra [75].

Para este caso, se dispondrá de un conjunto de datos de training o aprendizaje y otro conjunto de datos de testeo, el funcionamiento de los clasificadores SVM se realiza a través de un único clasificador, que realiza la predicción de una información de entrada en función del modelo que tiene aprendido.

En la Figura 16, se muestra cada uno de los pasos que se requieren para el entrenamiento de un clasificador SVM.

¹⁵ Imagen obtenida de “Multi-clasificación discriminativa por partes mediante códigos Correctores de Errores”.

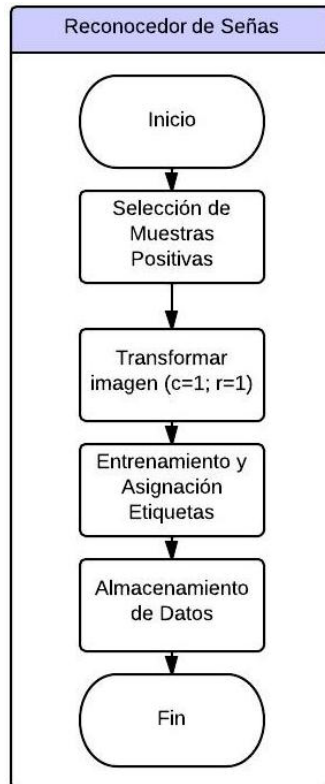


Figura 16. Reconocedor de Señas.

1. Selección de muestras positivas

Para el entrenamiento del clasificador se requiere de muestras que contengan las posiciones de las manos cuando se forman las distintas letras del abecedario, el número de imágenes que se debe tener por signo se puede fijar dependiendo de la cantidad de predicciones correctas que haga el clasificador.

2. Transformar imagen

Cada una de las imágenes a entrenar representan los vectores de clasificador es por ello que se deberán transformar cada una de ellas en una nueva imagen que tenga una única fila y una única columna.

3. Entrenamiento y asignación de etiquetas a cada clase

La información de aprendizaje consiste en un conjunto de datos de entrenamiento que se dividen en distintos tipos o clases y otro conjunto que serán los datos destinados a la comprobación del correcto funcionamiento del clasificador. Tanto en la fase de aprendizaje como en la de testeo, a cada clase se le asigna una etiqueta

que será su identificador del tipo. En función de ese identificador, se podrá distinguir la clase a la que pertenece el dato de entrada.

4. Almacenamiento de datos

Una vez que se haya entrenado el clasificador SVM se almacenará en un archivo XML para de esta manera utilizarlo cuando se lo requiera durante las predicciones.

3. Algoritmo KNN (K-Nearest Neighbor)

El KNN es un sistema de clasificación automática supervisado, para el entrenamiento de este tipo de clasificador se requiere de la selección de muestras, transformación de la imagen, la asignación de etiquetas a cada una de las clases, y el correspondiente almacenamiento, el procedimiento se lo realiza tal y como el algoritmo SVM, Figura 15.

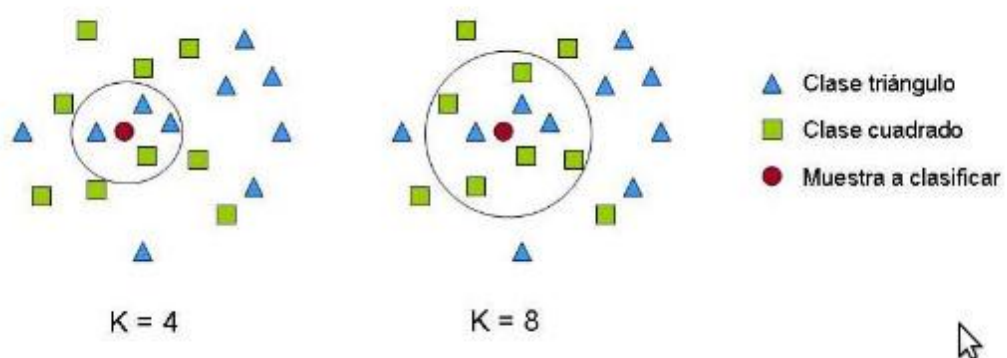


Figura 17. Clasificador KNN con K diferentes valores¹⁶.

La idea principal de este clasificador es muy simple. Partiendo de los vectores de características de cada una de las clases Figura 17, se calcula la distancia del vector de características a testear a los vectores de cada una de las clases existentes [76], el modelo que contenga el mayor número de muestras será la clase a la que pertenezca la muestra a clasificar.

Existen algunos problemas con la elección de este algoritmo a continuación se listan algunas sugerencias que se deben tomar en cuenta para la implementación [76]:

¹⁶ Imagen obtenida de "Reconocimiento de caras en entornos no controlados".

- **Elección incorrecta del parámetro k:** La elección del parámetro K del clasificador, tendrá que ser menor al número mínimo de muestras por clase, de no ser así el clasificador tomará muestras pertenecientes a otras clases.
- **Empate en la votación de Clases:** Una posible solución del clasificador es la localización de las muestras conflictivas y la eliminación de las mismas, también se puede dar el caso en que el clasificador adopte un valor extraño con el fin de señalar que la decisión por votación no ha dado un resultado.

Capítulo V. Metodologías de desarrollo

El empleo de una metodología durante todo el ciclo de desarrollo de sistemas es indispensable si se quiere garantizar el éxito de un proyecto, este capítulo inicia describiendo que son las metodologías ágiles seguido de la explicación de las metodologías que con mayor frecuencia son empleadas para el desarrollo de software, y concluye realizando una comparación entre dichas metodologías.

5.1 Metodologías de desarrollo ágil

Las metodologías ágiles surgen como una alternativa a las metodologías tradicionales de desarrollo, que se bloquean muy pronto en los detalles del proyecto y son menos capaces de ajustarse a las cambiantes necesidades de los clientes, del mercado y de los desafíos imprevistos que plantea la tecnología [77,78].

Hoy en día los entornos en los que se mueve el software es demasiado inestable y cambiante por lo que estas metodologías son ideales, en la reducción del tiempo de creación de sistemas sin dejar de lado la calidad.

Las características que presentan las metodologías de desarrollo ágil se describen a continuación [79]:

- Proceso iterativo e incremental.
- Mejora continua.
- Calidad desde el primer día.
- Priorización de requerimientos de acuerdo a su valor.
- Equipos dedicados y auto-gestionados.
- Colaboración continua con el cliente.
- Incorporar al cambio.
- Prácticas de desarrollo modernas.

5.1.1 Programación Extrema (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima

de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico [80].

Las iteraciones son relativamente cortas ya que se piensa que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se va a obtener y esto va a representar una mejor calidad del producto a largo plazo. Existe una fase de análisis inicial orientada a programar las iteraciones de desarrollo y cada iteración incluye diseño, codificación pruebas, fases superpuestas de tal manera que no se separen en el tiempo [81].

Algunas de las características de la metodología XP son [82]:

- Se basa en las pruebas realizadas a los principales procesos con el objetivo de detectar futuros errores.
- Se basa en la reutilización de código.
- Programación en pares, dos desarrolladores participan en un proyecto en una misma estación de trabajo.
- Lo fundamental en este tipo de metodología, es la comunicación entre los usuarios y los desarrolladores

5.1.2 SCRUM

Se centra principalmente a nivel de las personas y el equipo de desarrollo que construye el producto. Su objetivo es que los miembros del equipo trabajen juntos y de forma eficiente optimizando productos complejos y sofisticados.

- Empleada para proyectos con un alto grado de cambio de requerimientos.

Esta metodología Define un marco para la gestión de proyectos, está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante

iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [83].

Características que representan la metodología Scrum [84]:

- Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.
- Durante cada sprint, un periodo entre 15 y 30 días (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable).
- Requiere la fijación de fechas para realizar reuniones.
- Un principio clave de Scrum es el reconocimiento que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan.
- Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido.

5.1.3 Rapid Application Development (RAD)

RAD, fue definida por James Martin a principios de la década de 1980, es una metodología de desarrollo de software iterativa que enfatiza en el desarrollo extremadamente corto de un sistema basado en la construcción de prototipos y el uso de herramientas CASE, esta es una metodología que permite a las empresas desarrollar sistemas estratégicamente importantes de manera más rápida reduciendo a la vez costos de desarrollo y manteniendo la calidad [85,86]. Con este tipo de metodología el usuario del producto se encuentra relacionado en todas las fases de desarrollo, para de esta manera entender correctamente los requerimientos, delimitar el ámbito del proyecto, y reducir el tiempo de desarrollo no más de 60 a 90 días.

Algunas de las características principales de esta metodología:

- Ciclo de desarrollo iterativo e incremental.
- Equipos Híbridos: gente motivada y muy versátil capaz de desempeñar diferentes roles durante el desarrollo del proyecto.
- Reutilización de componentes.
- Time-boxing, límites rígidos en los plazos de desarrollo.
- Utiliza el prototipado como modelo de aproximación a la solución final.
- Prototipación rápida: su objetivo es obtener en el menor tiempo posible el análisis, diseño e implementación con la ayuda de herramientas CASE.
- Generadores de Aplicación: los cuales permiten la automatización de porciones grandes de código a partir de definiciones generales o prototipos.

5.1.4 Diferencias entre Metodologías Ágiles

En la TABLA X, se realiza una comparación y se describen las características que diferencian las metodologías antes descritas [87,88]:

TABLA X. DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES.

SCRUM	XP	RAD
Las iteraciones de entrega son de dos a cuatro semanas.	Las iteraciones de entrega son de una a tres semanas.	Los ciclos de desarrollo son más pequeños ya que se emplea herramientas que facilitan la generación de código.
Los miembros del equipo trabajan en forma individual.	Los miembros del equipo trabajan en parejas.	Los encargados del proyecto, involucran a los usuarios del sistema.
Las tareas que se hayan realizado y que el propietario del producto haya mostrado su conformidad ya no se	Las tareas se van terminando aunque son susceptibles de ser modificadas durante el transcurso del proyecto,	Las fases realizadas pueden ser modificadas durante el transcurso del proyecto.

retocan, si funciona y está bien se aparta para realizar otras cosas.	incluso después que funcionen correctamente.	
Trata de seguir el orden de prioridades de tareas pero se puede cambiar si es mejor para el desarrollo de las tareas.	Siguen estrictamente el orden de prioridad de las tareas definidas por el cliente.	El orden de prioridades de las tareas puede ser modificado mostrando mayor flexibilidad.
Es una metodología de desarrollo ágil se basa más en la administración del proyecto.	Se centra en la programación o en la creación del proyecto.	Es una metodología de desarrollo ágil basado en la construcción de prototipos.

e. Materiales y Métodos

Para el desarrollo del Trabajo de Titulación se hizo necesario el empleo de métodos que garanticen la correcta ejecución de la aplicación desde los inicios hasta sus fases posteriores, a continuación se realiza la descripción de cada uno de los métodos empleados.

1. Métodos

- **Científico**

El empleo de este método sirvió para formular el presente Trabajo de Titulación, se siguió las diferentes etapas que conlleva el mismo, la observación nos permitió tener una idea principal del campo problemático que conlleva la enseñanza del lenguaje de señas, la formulación de hipótesis ayudo en la formación del tema propuesto para esta investigación, a través de la experimentación se desarrolló la aplicación con la ayuda del empleo de una metodología para el desarrollo, para finalmente en base a ella realizar las conclusiones referentes a la creación de la aplicación.

- **Estudio de Casos**

A través del estudio de casos se llevó a cabo la revisión literaria sobre las aplicaciones que hacen uso de la visión artificial, a partir de ello se revisaron casos de éxito como casos de fracaso en aplicaciones no industriales e industriales, como resultado de ello se obtuvo el análisis de los requerimientos tanto de hardware como software para la eficiente ejecución de un sistema con visión artificial.

- **Cualitativo**

Este método junto con la observación directa permitió obtener una comprensión profunda de cómo se encuentra la situación actual en la enseñanza del lenguaje de señas, así como permitió documentar y describir las acciones que se realizan para la interpretación del lenguaje de señas con el objetivo de implementar una aplicación para facilitar la enseñanza-aprendizaje en la asociación de sordos “Virgen del Cisne”.

- **Cuantitativo**

Permitió analizar e interpretar especialmente las tabulaciones que se realizó luego de la aplicación de técnicas como la encuesta a una muestra determinada de la población, para poder explicar la aceptación que el sistema tubo así como problemas que se pudieran presentar durante la ejecución.

Tipo de estudio

- **Estudio Documental:** el empleo de este tipo de estudio nos ayudó en la revisión documental, ya que constituye el primer paso durante el proceso de investigación, permitió la recolección y análisis de las fuentes bibliográficas que nos sirvieron como una fuente de respaldo para el marco teórico.

1.1 Metodología de Desarrollo

Para el desarrollo del software se ha optado por emplear una metodología de desarrollo ágil RAD (Rapid Application Development), ya que tanto los elementos como fases que constituyen esta metodología, la hacen ideal para la ejecución del mismo, principalmente permitiendo reducir el tiempo de desarrollo al reutilizar código que ha sido creado por la librería de desarrollo.

Se seleccionó ésta metodología puesto que proporciona mayor flexibilidad de cambios en cada una de sus fases, así como una mayor comprensión de los requerimientos con la elaboración de prototipos más efectivos, debido a una relación más cercana y constante con los usuarios finales del proyecto.

A continuación se describen las actividades realizadas en cada una de las fases de la metodología RAD:

- **Fase de planificación de Requisitos**

En esta fase se trabajó con el sub-coordinador de la asociación de sordos de Loja para analizar y acordar las necesidades para el desarrollo eficiente del sistema, para llevar a cabo estos procesos se emplearon técnicas de recolección de datos, las mismas que son descritas a continuación:

- **Observación directa:** para el empleo de esta técnica se asistió a una de las clases que son dictadas acerca del lenguaje de señas en la asociación, se pudo observar cómo se realizaba este proceso y cuáles eran las personas involucradas. La observación facilitó la comprensión de los procesos los mismos que fueron plasmados en los requerimientos del sistema (Ver Sección Resultados apartado 2.1 Planificación de requisitos).
- **Encuesta:** mediante este tipo de técnica se emplearon cuestionarios pre-elaborados y se seleccionaron las preguntas más convenientes, para conocer cuál es el tipo de lenguaje que se emplea en la enseñanza y los equipos de cómputo con los que cuenta la asociación para la ejecución del sistema (Ver Anexo 6. Encuesta para conocer el lenguaje de Señas que emplea en la enseñanza la Asociación de Sordos de Loja “Virgen del Cisne”).
- **La Entrevista:** la aplicación de esta técnica nos permitió obtener la información relevante a través del diálogo con la persona encargada de la asociación, de cómo se lleva a cabo cada uno de los procesos de enseñanza del lenguaje de señas (Ver Sección Resultados apartado 2.1.4 Diagramas de actividad).

Con la información obtenida se procedió a realizar el modelo de datos (Ver Sección Resultados apartado 2.2.1 Modelo de datos).

- **Fase de diseño del Usuario:**

Dados los requerimientos se realizaron los prototipos iniciales de pantallas que integran el sistema (Ver Anexo 8. Prototipos de pantallas iniciales.), estos prototipos fueron indicados a los usuarios los mismos que sugirieron realizar algunas modificaciones, este proceso se repitió en varias ocasiones, con los que finalmente se obtuvo prototipos claros y sencillos.

- **Fase de construcción rápida**

En base a la información obtenida en las fases anteriores y en la revisión literaria se procedió a codificar la aplicación, teniendo en cuenta las configuraciones de los entornos de desarrollo, los estándares de codificación, etc. (Ver Sección Resultados apartado 2.3 Construcción Rápida). Finalmente se realizaron pruebas las mismas que

fueron de tres tipos: Pruebas unitarias, Pruebas de cobertura, Pruebas en varios entornos (Ver Sección Resultados apartado 1.3.3 Pruebas),

- **Fase de Transición**

Finalmente esta fase de la metodología fue la encargada de poner en marcha el sistema en la asociación así como de brindar las correspondientes pruebas y capacitación a las personas que van hacer uso del sistema.

f. Resultados

Al realizar cada una de las actividades que complementan los objetivos se obtuvo los resultados que se describen a continuación:

Fase 1: Determinación de la Factibilidad para el desarrollo del sistema informático para el reconocimiento del lenguaje de señas.

Para el desarrollo de esta fase se realizó todo el estudio que conlleva la revisión literaria a continuación se describen los recursos como: herramientas, sistema operativo, librerías y casos de éxito que fueron empleados para el desarrollo del sistema.

1.2 Análisis revisión literaria

Con el uso de la Visión Artificial se han implementado un gran número de herramientas que han permitido solventar ciertos problemas, en la industria desde el control de calidad hasta aplicaciones no industriales como el control de tráfico, de las referencias revisadas los sistemas más relevantes y que han servido como base para otras aplicaciones son :

- **Sistema de Reconocimiento de Placas Vehiculares (ANPR):** ya que se lo ha implementado en diferentes países de América, Asia, Europa, África a continuación se muestran algunos de los países en los que se lo ha implementado, puesto que las aplicaciones han sido desarrolladas con soporte para diferentes lenguajes y caracteres alfanuméricos [89]:

TABLA XI. PAÍSES DONDE SE HA IMPLANTADO LOS SISTEMAS ANPR.

Brasil	Chile
México	Nueva Zelanda
Australia	Taiwan
Irán	Hong-Kong
España	Sudáfrica
USA	Hungría

- **Aplicaciones de control de calidad:** son muchas las empresas que han optado por esta tecnología, ya que reduce el tiempo y costo de los procesos que normalmente realizaría una persona encargada, en la inspección, monitorización y selección de un producto defectuoso. Actualmente se aplica la robótica con la visión artificial dando resultados inigualables, permitiendo así la automatización de empresas y procesos tales como [90]:
 - Envasado de productos [91].
 - Medición automática [92].
 - Guiado de Robots (Automoción) [93,94].

Existen una gran cantidad de aplicaciones desarrolladas a través de la técnica de Visión Artificial, en los campos industriales como control de calidad, control de procesos y no industriales como el control de tráfico, reconocimiento de gestos, todos estos proyectos actualmente se encuentran funcionando en muchas entidades, gracias a la existencia de múltiples herramientas, librerías y lenguajes que soportan esta tecnología, por consiguiente la construcción del proyecto “Visión Artificial para el reconocimiento del Lenguaje de Señas”, fue factible de realizar de acuerdo al análisis del conjunto de referencias bibliográficas.

1.3 Componentes hardware y software

De acuerdo a las características técnicas que se requieren para el desarrollo del sistema de visión artificial, resulta factible la realización, ya que se cuenta con los recursos necesarios, a continuación se describe cada una de las herramientas seleccionadas de acuerdo a las ventajas que ofrece para el desarrollo:

1.3.1 Software

Las herramientas software seleccionadas en base a la revisión bibliográfica se describen a continuación:

- **Sistema Operativo**

El sistema operativo que se seleccionó para el desarrollo del proyecto es Linux, ya que ofrece múltiples ventajas (Ver TABLA III. VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS OPERATIVOS MÁS CONOCIDOS.), frente a otros sistemas operativos, además de controladores de software libre y el consumo reducido de

recursos para la ejecución del sistema operativo, actualmente Linux comprende múltiples distribuciones sin embargo en esta ocasión se empleó la distribución de escritorio Ubuntu 12.04, ya que esta es la última de las versiones estables que cada 5 años sale y a la cual se la mantiene vigente brindándole el soporte adecuado.

- **Lenguaje de Programación**

Como el sistema estará funcionando en tiempo real consumirá varios recursos, uno de los más importantes es el tiempo de respuesta (velocidad), existen algunos inconvenientes de acuerdo a este tema:

- La imagen a ser procesada es de 635x461 pixeles, se la redujo a un tercio de su tamaño real, quedando de 211x153 pixeles, aun así el recorrido del bucle para encontrar las regiones de interés es de 32.283 iteraciones por frame.
- Se trabaja con un total de 50 frames por segundo.
- La mayoría de las librerías de visión artificial no brindan un soporte para el desarrollo web.

Es por ello que se optó por hacer la aplicación de escritorio, además de su fácil instalación para las personas que integran la asociación, uno de los lenguajes de programación que nos permite una mayor velocidad ya que es compilado y que además es multiplataforma es C++ (Ver TABLA IV), por ende es el lenguaje de seleccionado para el desarrollo del sistema.

- **Librerías de Visión Artificial**

La librería de visión artificial que se empleó fue OpenCV en su versión 2.4.8, las características que hizo que se seleccione esta librería son:

- Gran cantidad de sistemas que se han realizado en base a la misma.
- Extensa documentación.
- Gran cantidad de funciones que posee para la implementación de algoritmos propios.
- Compatibilidad con el lenguaje de programación antes seleccionado.

Finalmente al ser OpenCV una librería de código abierto obtendrá una máxima funcionalidad en el sistema operativo de Linux.

1.3.2 Hardware

A continuación se describen los requisitos hardware indispensable para el desarrollo del sistema y las características que poseen cada uno de ellos:

- **Cámara**

La cámara que se empleará para la captura de video es una Webcam normal, la que viene por defecto incluida en el equipo portátil.

- **Computadora**

Se empleará una computadora COMPAQ Presario CQ60 cuyas características se describen en la TABLA XII:

TABLA XII. CARACTERÍSTICAS DEL HARDWARE EMPLEADO PARA EL DESARROLLO.

Características	Especificación
Procesador	Intel® Core™2 Duo CPU T5800 @ 2.00GHz × 2
Memoria Ram	2 Gb
Disco Duro	50 Gb
Cámara web	HP CNF7047
Unidad CD-ROM	CD-ROM
Sistema Operativo	Ubuntu 12.04

El Hardware que se empleará está por encima de los requerimientos básicos (Ver TABLA IX. CARACTERÍSTICAS MÍNIMAS INSTALACIÓN DE S.O.), por lo que no debería haber ningún problema durante el desarrollo de la aplicación, así como en las pruebas. Además según la encuesta realizada al representante de la Asociación de Sordos de Loja “Virgen del Cisne” (Ver Anexo 6), la institución cuenta con varios equipos como televisores, proyectores y equipos de computación los cuales son usados para la formación de los estudiantes, y en los que se pondrá en funcionamiento el sistema.

De la misma manera los integrantes de esta Asociación cuentan con equipos portátiles donde se podrá instalar el software para realizar las diferentes prácticas en la enseñanza del lenguaje de señas.

1.4 Lenguaje de Señas empleado

El alfabeto manual o dactilológico es un sistema de representación simbólica o icónica de las letras de los alfabetos de las lenguas orales-escritas por medio de las manos. Dependiendo de los alfabetos o sistemas de escritura de cada país o comunidad, este lenguaje tendrá señas y códigos propios [65].

Es así que para América se ha establecido el lenguaje (ASL) Lenguaje de Signos Americano, sin embargo a nivel de países se han realizado ligeras variaciones dando lugar a lenguajes específicos como ejemplo tenemos el (LSM) Lenguaje de Signos Mexicano, Lenguaje de Signos Estadounidense, entre otros.

El lenguaje de señas que emplea la Asociación de Sordos de Loja “Virgen del Cisne” es el Lenguaje de señas Ecuatoriano (ESL) (Ver Anexo 6), para la enseñanza básica, se utiliza los números y el abecedario que están contenidos en este mismo lenguaje, los números y el alfabeto está formado por señas estáticas y dinámicas, las señas dinámicas muestran flechas de trayectoria que indican la dirección del movimiento (Ver Anexo 7. Lenguaje de Señas Ecuatoriano.).

Por consiguiente fue factible el desarrollo del sistema para el reconocimiento del lenguaje de señas ya que se tuvo tanto los recursos hardware como software, así como gran cantidad de información, además se contó con el apoyo de todas las personas que hacen la Asociación de Sordos de Loja “Virgen del Cisne”.

Fase 2: Diseño e Implementación del sistema

Antes de la implementación del sistema la elección de una metodología de desarrollo se hizo necesaria, con el propósito de garantizar la calidad del producto final, el cumplimiento de los tiempos estipulados, así como la ordenada elaboración, la metodología empleada en el desarrollo del sistema, gracias a las ventajas que ofrece fue RAD (Ver Diferencias entre Metodologías Ágiles), ya que el tiempo designado para el desarrollo es corto además se requiere de una constante interrelación con los

usuarios ya que no se está familiarizado con el lenguaje de señas, a continuación se describe cada una de las fases empleadas para el desarrollo del sistema:

1.1 Planificación de requisitos

En esta fase se procedió a realizar un análisis de los procesos que se realizan para la enseñanza del lenguaje de señas y así obtener los requerimientos, conceptos y objetos relacionados con la misma.

Para el logro de esta fase se aplicaron algunas técnicas de recolección de información, especialmente la observación directa (Ver Anexo 3. Guía de Observación directa.), y algunas otras como la entrevista y la encuesta.

A través de la observación directa se pudo apreciar cómo se realiza la enseñanza del lenguaje de señas a personas que poseen discapacidad auditiva, así como para personas que desean instruirse en este tipo de lenguaje, luego de aplicada una entrevista al Lic. Cristian Giler Sub-Coordinador de la Asociación de Sordos de Loja “Virgen del Cisne”, se ha logrado extraer una lista de requerimientos funcionales y no funcionales, en donde se procedió a verificar su consistencia y redundancia dando como resultado los Requisitos Funcionales y No Funcionales que se describen en las TABLA XIII y TABLA XIV:

1.1.1 Requerimientos Funcionales:

El sistema permitirá:

TABLA XIII. REQUERIMIENTOS FUNCIONALES.

Ref.	Descripción	Categoría	Técnica de Recolección
RF001	Seleccionar un umbral adecuado para el seguimiento de un color.	Evidente	Entrevista
RF002	Reconocer la mano del usuario.	Evidente	Observación (Ver Anexo 3. Guía de Observación directa.)
RF003	Realizar un seguimiento de la mano del usuario.	Oculto	Entrevista

RF004	Obtención de imágenes de la mano del usuario para la clasificación.	Oculto	Observación (Ver Anexo 3. Guía de Observación directa.)
RF005	Reconocer cada uno de los signos del alfabeto de señas.	Evidente	Entrevista
RF006	Reconocer cada uno de los signos de los números.	Evidente	Entrevista
RF007	Formación de palabras en base a los signos representados en el lenguaje de señas.	Evidente	Observación (Ver Anexo 3. Guía de Observación directa.)

1.1.2 Requerimientos No Funcionales:

El sistema deberá:

TABLA XIV. REQUERIMIENTOS NO FUNCIONALES.

Ref.	Descripción	Categoría	Técnica de Recolección
RN001	No ser demasiado complejo en su utilización.	Evidente	Encuesta (Ver Anexo 5, pregunta número 6)
RN002	Tener una interfaz de usuario amigable.	Evidente	Entrevista
RN003	Proporcionar ayuda en la pantalla que se encuentre.	evidente	Entrevista
RN004	Brindar tiempos de respuesta aceptables.	oculto	Entrevista

1.1.3 Lista de Entidades a desarrollar

En la TABLA XV, se muestra un conjunto de entidades con mayor tendencia a ser implementadas, todas ellas con su correspondiente definición que nos permite tener una comprensión mejor de la forma en que cada una opera.

TABLA XV. LISTA DE ENTIDADES A DESARROLLAR.

Entidad	Descripción
Persona	Entidad principal que hará uso del sistema a desarrollar.
Color	Tez de una persona específica.
Dispositivo	Accesorio que permite la captura y el procesamiento de las imágenes.
Procesamiento	Funciones principales para la segmentación de las imágenes.
Reconocedor	Entidad que se encarga de clasificar un objeto.
Docente	Persona encargada de la enseñanza del Lenguaje de Señas.
Estudiante	Persona que tiene como objetivo el aprendizaje del Lenguaje de Señas.
Seña	Representación de un carácter, número o palabra.

1.1.4 Diagramas de actividad.

Luego de emplear la observación directa, junto con una entrevista al subcoordinador de la asociación, para constatar el proceso que se lleva para la enseñanza del lenguaje de señas, se plasmó el diagrama de actividad Figura 18:

1.1.4.1 Descripción del diagrama de actividad.

A continuación se realiza la descripción del diagrama de actividad que se emplea para la enseñanza del lenguaje de señas:

1. El docente es el que inicia este proceso, realiza la seña, indica la posición que debe tener la mano e indica el significado de la misma.

2. Los estudiantes repiten la posición de la mano y el significado de la seña realizada.
3. Al terminar de repetir el docente revisa de forma individual la ejecución de la seña, en caso de ser necesario repite y corrige la posición de la mano.
4. Una vez que todos los estudiantes realizan correctamente las letras del abecedario, el docente aumenta la complejidad, formando palabras y realizando oraciones completas.

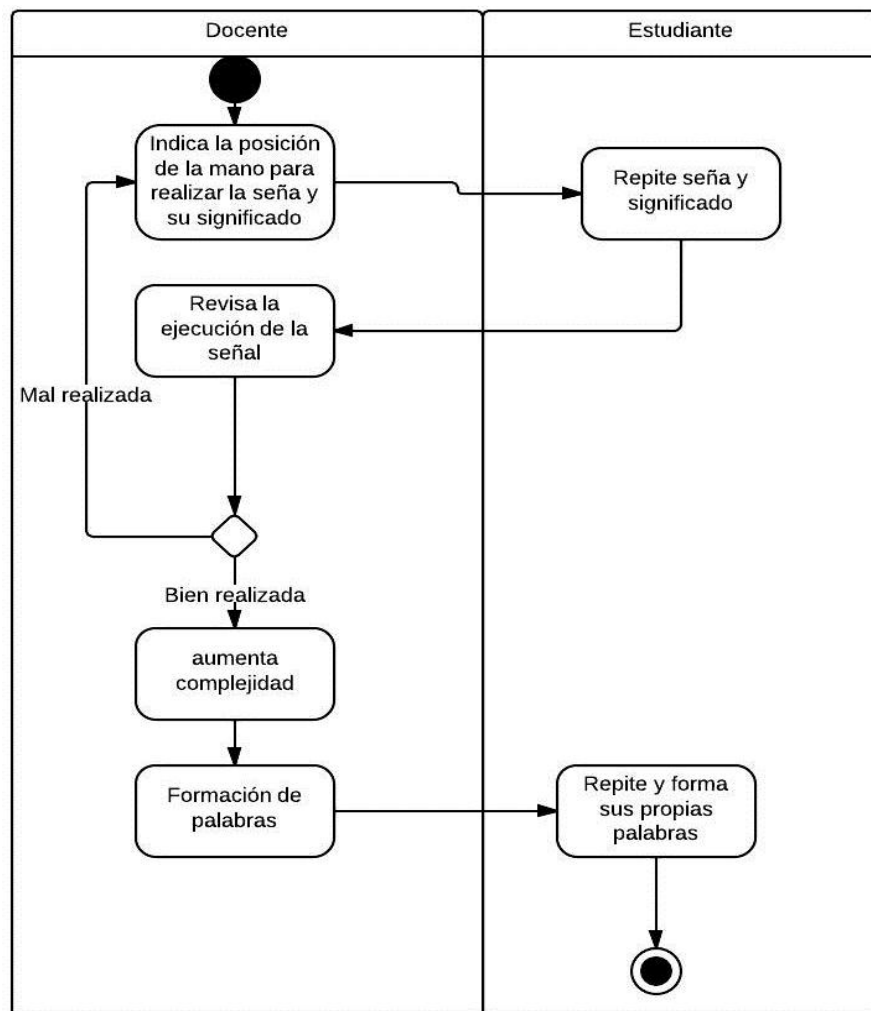


Figura 18. Diagrama de actividades en el proceso de aprendizaje del lenguaje de señas.

1.2 Diseño del usuario.

Con el uso de la lista de entidades a desarrollar se realizó el modelo del dominio Figura 19, obteniendo así para cada una de las clases sus respectivos métodos y atributos, además se refinaron dichas clases y sus relaciones, quedando de la siguiente manera:

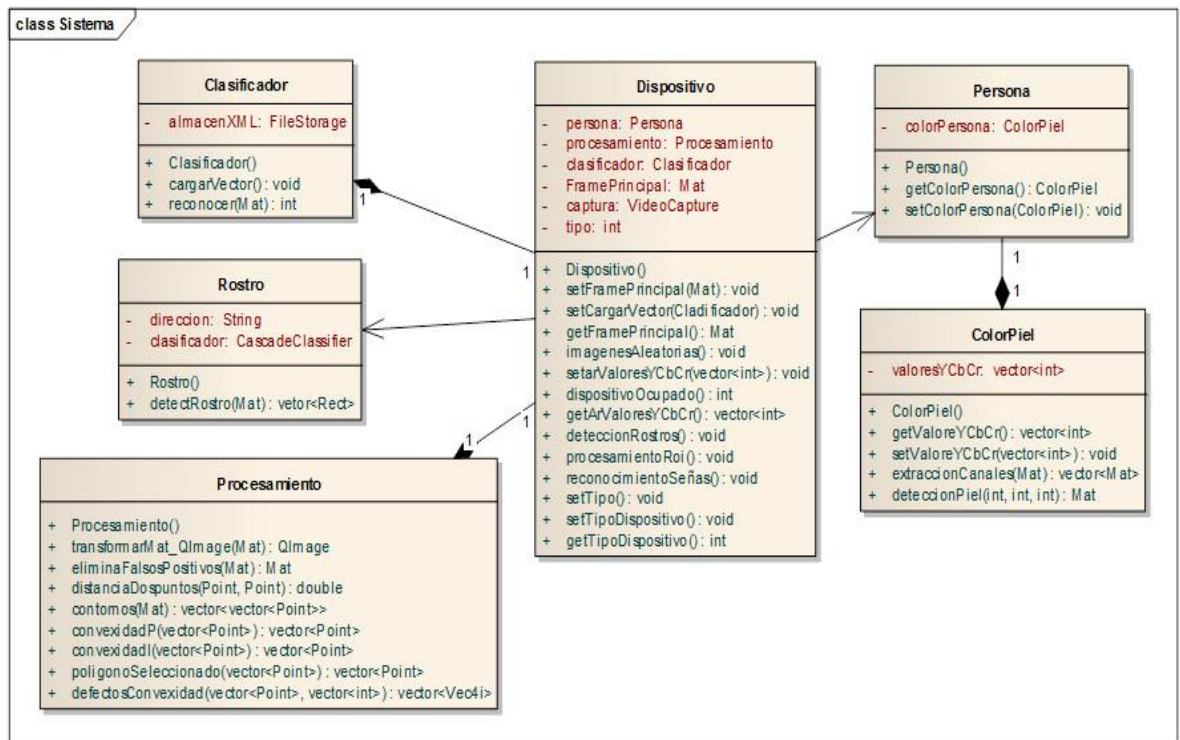


Figura 19. Modelo del dominio.

- **Persona:** Entidad que va hacer uso del sistema para el aprendizaje y detección del lenguaje de señas.
- **Rostro:** Entidad que permite la detección del rostro de una persona para su posterior eliminación, de manera que se evite un procesamiento innecesario.
- **Mano:** Clase que emite señales en base a posiciones diferentes.
- **ColorPiel:** Clase que contiene los valores que representan el color de piel de una persona específica.
- **Clasificador:** Entidad que permite la predicción de cada uno de los signos realizados.

- **Procesamiento:** Clase que contiene los procesos básicos para realizar la segmentación de la imagen inicial, para conseguir únicamente la extracción de la mano.
- **Dispositivo:** Entidad que permite la captura de imágenes en tiempo real.

1.2.1 Modelo de datos

No se empleó una base de datos, se hizo uso de un archivo de almacenamiento xml que permite guardar cada una de las muestras con la que se compara las imágenes extraídas del video en tiempo real, algunas de las razones por las que no se hizo uso de la misma se describen en la TABLA XVI:

TABLA XVI. VENTAJAS NO USAR UNA BASE DE DATOS.

Nº	Descripción
1	No se tendrá que instalar una base de datos donde corra el sistema.
2	Se reduce el tiempo ya que no se tendrá que hacer consultas a la base de datos.
3	La cantidad de caracteres a almacenar es muy grande.
4	Únicamente se almacenarán dos archivos.

1.2.2 Flujos de ventanas principales del sistema

Ya con la lista de requerimientos se procedió a realizar los prototipos rápidos de pantalla iniciales (ver Anexo 8), estos fueron presentados a los usuarios con el objetivo de obtener sugerencias de manera que se obtenga un prototipado claro y sencillo.

Para el desarrollo de las interfaces del sistema para el reconocimiento de señas en tiempo real se hizo necesario el empleo de cuatro hilos de ejecución de forma paralela, los dos primeros se ejecutan cada 20 milisegundos el tercero se ejecuta cada medio segundo y el último cada vez que el usuario requiera de la selección del color de piel.



Figura 20. Pantalla principal para el entrenamiento.

- El primer hilo se lo empleó para la captura de video en tiempo real, con el objetivo de realizar el procesamiento para la extracción de características que correspondan a la mano, la reducción del tamaño de video de entrada, ya que se hace necesario para reducir el tiempo durante la ejecución del algoritmo en cada frame capturado, de la misma manera se realiza la detección del rostro de la persona que se encuentra frente al ordenador y se la extrae para evitar procesar dicha sección de la imagen en busca de características que únicamente posee la mano.



Figura 21. Pantalla segmentación del color.

- La ejecución del segundo hilo, es la encargada de enviar información de la imagen en tiempo real al frame para la captura de los valores que representan el color de piel del usuario, de este frame se extrae cada uno de los canales del video y se le aplica la selección de un umbral adecuado, con el objetivo de clasificar únicamente el color de piel, cuando ya se haya seleccionado un umbral adecuado por cada uno de los canales de las imágenes, se vuelven a combinar y se elimina el ruido o falsos positivos y negativos que pueda contener la imagen, estos valores son devueltos al frame de reconocimiento de señas y almacenados en el frame principal para evitar que estos se pierdan, de manera que cuando se vuelva a abrir la pantalla de entrenamiento o traducción en tiempo real, el reconocedor de señas mantenga cargado el color de piel del usuario.



Figura 22, Pantalla de traducción del lenguaje.

- Una vez que el dispositivo se encuentra únicamente siguiendo el color de piel del usuario se pone en ejecución el tercer hilo el cual permite la extracción de características que representan la mano del usuario, se encuentran los contornos de cada uno de los objetos que aparecen en el video capturado, si los contornos son mayores a cierto número de píxeles se los toma como una región de interés caso contrario se lo considera ruido, una vez extraída la región de interés se busca el centro del objeto contenido en la misma, y la convexidad, ya que esta representa cada uno de los dedos de la mano, finalmente se calcula la distancia del centro de la mano a cada uno de los de los puntos de convexidad (dedos), y la mayor distancia de ellas la empleamos como radio para la extracción de una subregión de interés, esta será la mano del usuario.



Figura 23. Captura de la mano para la clasificación.

- Finalmente el cuarto hilo se ejecuta cada segundo y es el que permite pasar la ROI, al clasificador para generar la predicción de la imagen capturada en tiempo real Figura 23, contra el conjunto de imágenes que se encuentran almacenadas en los archivos xml (Vector de Características) generados por los clasificadores SVM y KNN, y que sirven como modelos de clasificación.

1.3 Construcción Rápida

En esta fase de la metodología se realiza la implementación de los algoritmos para el reconocimiento de la mano así como para la clasificación de las señas, a continuación describe los procedimientos que se emplearon para el desarrollo:

1.3.1 Construcción del Reconocedor de la mano

Para el desarrollo de esta fase se procedió a configurar OpenCV con el entorno de desarrollo QT-Creator (Ver Anexo 9), para generar e implementar el código del modelo de dominio.

De los algoritmos que se analizaron para la detección de la mano, únicamente se emplearon algunos pasos, puesto que en la reproducción de los algoritmos se presentaron problemas ya que los mismos están indicados de forma general, a más de ello se hace indispensable el uso de dispositivos sumamente costosos como (Kinect) [95, 96], o lugares completamente controlados que mantengan un fondo de color uniforme, y donde no aparezca ningún otro objeto que no sea las manos de la persona que manipula el sistema [97, 98].

Es por ello que se optó por la construcción de un algoritmo para el reconocimiento de las manos, a continuación se muestra el algoritmo Figura 24, y parte del código de implementación:

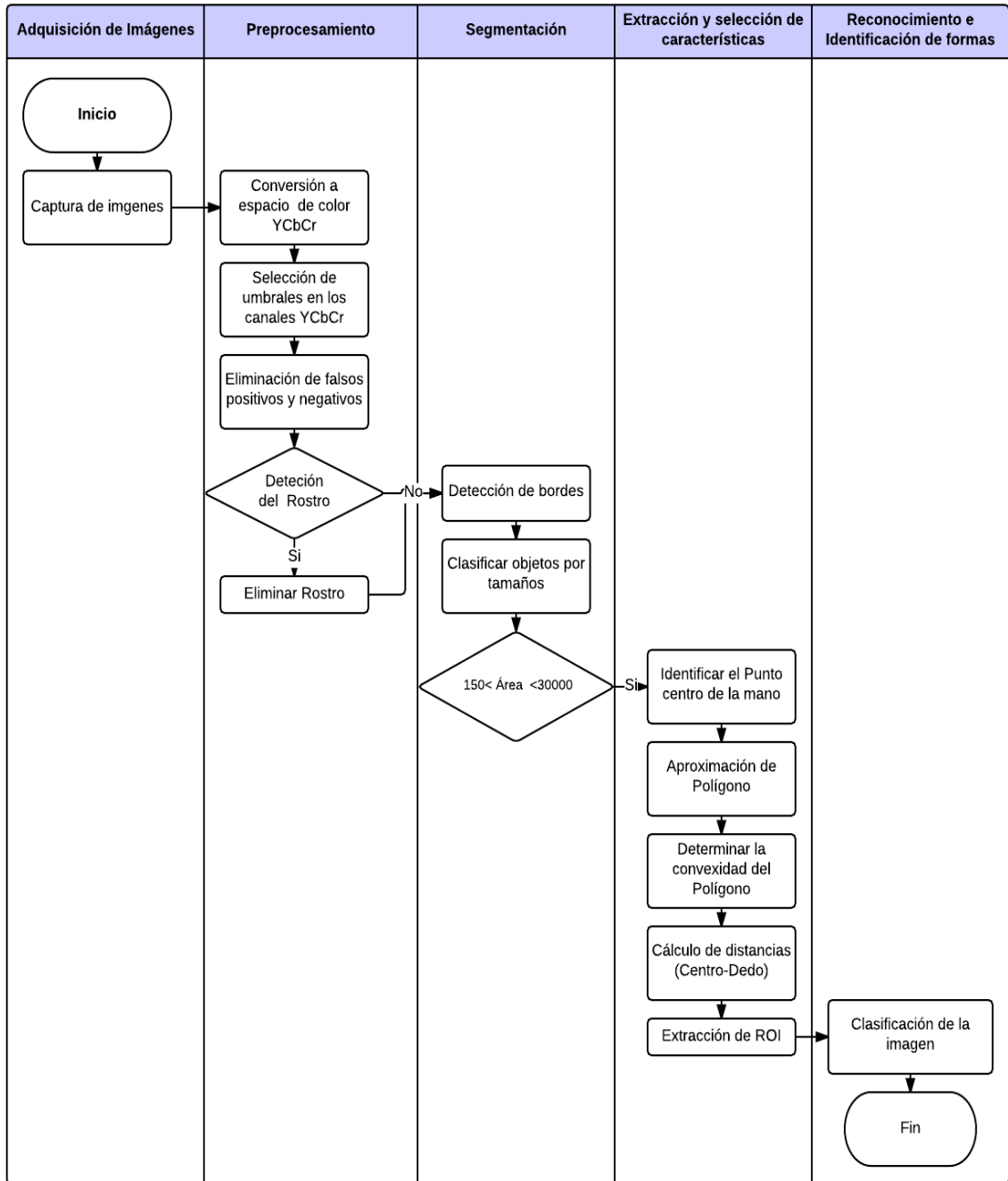


Figura 24. Algoritmo para reconocer la mano.

Para la implementación del algoritmo del reconocimiento de las manos, se emplearon las funciones de OpenCV descritas a continuación:

TABLA XVII. FUNCIONES EMPLEADAS EN LA CONSTRUCCIÓN DEL ALGORITMO DE RECONOCIMIENTO DE LA MANO.

Fases	Métodos
Captura de Imágenes	VideoCapture(int device); <ul style="list-style-type: none"> • device: permite la captura desde un dispositivo, cero es la webcam, dispositivo por defecto.
Conversión de Color	cvtColor(InputArray src, OutputArray dst, int code, int dstCn=0); <ul style="list-style-type: none"> • src: imagen de entrada. • dst: imagen de salida (transformada). • code: código del espacio de color. • dstCn: número de canales de la imagen de salida.
Eliminación de Falso positivos y negativos	dilate(InputArray src, OutputArray dst, InputArray element); erode(InputArray src, OutputArray dst, InputArray element); <ul style="list-style-type: none"> • src: imagen de ingreso. • dst: imagen de salida. • element: estructura del elemento usado para la erosión, Mat (3x3), estructural cuadrada, o rectangular.
Detección de Rostro	CascadeClassifier::detectMultiScale(const Mat image, vector<Rect>objects, double scaleFactor=1.1, int minNeighbors=3, Size minSize=Size(), Size maxSize=Size()); <ul style="list-style-type: none"> • objects: vector que contiene los objetos detectados. • scaleFactor: especifica la escala de la imagen. • minNeighbors: especifica el número de vecinos que puede tener los objetos detectados. • minSize: tamaño mínimo del objeto a detectar. • maxSize: tamaño máximo del objeto a detectar.
Detección de Bordos	findContours(InputOutputArray image, OutputArrayOfArrays contours, int mode, int method, Point offset=Point()); <ul style="list-style-type: none"> • image: imagen de entrada. • contours: vector de vectores que contendrá los puntos de los contornos. • mode: modo que permite la selección de contornos. • method: método para la aproximación de los

	<p>contornos.</p> <ul style="list-style-type: none"> • offset: desplazamiento de cada punto del contorno.
Clasificación de Objetos por tamaño	<p>contourArea(InputArray contour, bool oriented=false);</p> <ul style="list-style-type: none"> • contour: contorno al que se desea calcular el área.
Punto centro de la mano	<p>minMaxLoc(InputArray src, double* minVal, double* maxVal=0, Point* min-Loc=0, Point* maxLoc=0);</p> <ul style="list-style-type: none"> • src: imagen de entrada. • minVal: variable que almacena la mínima distancia. • maxVal: variable que almacena la máxima distancia. • minLoc: variable que almacena el punto mínimo. • maxLoc: variable que almacena el punto máximo.
Convexidad	<p>convexHull(InputArray points, OutputArray hull, bool clockwise=false);</p> <ul style="list-style-type: none"> • points: imagen de entrada. • hull: imagen de salida. • clockwise: orientación para la selección de la convexidad.
Distancia Centro-Dedo	<p>Procesamiento:distanciaDospuntos(Point p1, Point p2);</p> <ul style="list-style-type: none"> • p1: punto inicial. • p2: punto final.
Clasificación de la señas	<p>La descripción de los algoritmos empleados para esta fase se describe en el apartado 1.3.2 Construcción del Reconocedor del Alfabeto de señas.</p>

El siguiente bloque de código contiene la implementación de todo el algoritmo para el reconocimiento de la mano,

TABLA XVIII. IMPLEMENTACIÓN DEL ALGORITMO PARA RECONOCER LA MANO.

Implementación del algoritmo para reconocer la mano.
<pre>void Dispositivo::procesamientoRoi() { //Dibujamos los contornos si son mayores a 5000 pixeles for (size_t cont = 0; cont < arContorno.size(); ++cont) {</pre>

```

if (contourArea(arContorno[cont])*3>150 && contourArea(arContorno[cont])*3<30000 ){
//Región de interés que son las manos
rEncierraManos=boundingRect(arContorno[cont]);
mRegionInteres=mElimiRostro(rEncierraManos);
Mat imagenAux(mRegionInteres.size(), CV_32FC1 );
//Recorre pixel a pixel para encontrar los pixeles del centro del objeto
for( int j = 0; j < imagenAux.rows; j=j+1 ){
for( int i = 0; i < imagenAux.cols; i=i+1 ){
imagenAux.at<float>(j,i) = pointPolygonTest( arContorno[cont],
Point2f(i+rEncierraManos.x,j+rEncierraManos.y), true );
}
}
//Selecciona y dibuja el pixel con menor valor que será el centro
minMaxLoc( imagenAux, &dMinVal, &dMaxVal, &pMincentro, &pMaxcentro, Mat() );
pCentro=pMaxcentro+Point(rEncierraManos.x,rEncierraManos.y);
centroDeslizado=pCentro;
//Dibuja un polígono con los puntos de la mano para evitar que sean muchos los puntos
que representan los dedos
arPoligonoMano[cont]=mMano.poligonoSeleccionado(arContorno[cont]);
//Encuentra y dibuja la convexidad del contorno de las manos
arPuntosConvexidadCont[cont]=mMano.convexidadP(arContorno[cont]);
//Encuentra y dibuja la convexidad del polígono que envuelve los contornos de las
manos
arConvexidadPuntosPol[cont]=mMano.convexidadP(arPoligonoMano[cont]);
arConvexidadIntPol[cont]=mMano.convexidadI(arPoligonoMano[cont]);
if (arConvexidadPuntosPol[cont].size())>3) {
//Recorre cada uno de los puntos de la convexidad del polígono
for (size_t convPoli = 0; convPoli < arConvexidadPuntosPol[cont].size(); ++convPoli){
//Encuentra y dibuja los defectos de convexidad de las manos
arDefectos=mMano.defectosConvexidad(arPoligonoMano[cont],arConvexidadIntPol[con
t]);
//Dibuja los puntos de los defectos de convexidad del polígono
for (size_t defec = 0; defec < arDefectos.size(); ++defec) {
int defect=arDefectos[defec][2];

```

```

    Point puntoDefect( arPoligonoMano[cont][defect] );
  }
}
}

```

1.3.2 Construcción del Reconocedor del Alfabeto de señas.

Para la construcción del reconocedor del alfabeto de señas se entrenó los clasificadores SVM (Máquina de Soporte Vectorial) y el KNN (K-vecino más cercano), descritos en el Capítulo IV de la revisión bibliográfica, ya que el entrenamiento Cascadas Haar, requiere de una gran cantidad de tiempo [74], muestras [99] y un equipo de cómputo con características potentes en cuanto a procesador y memoria.

El conjunto de muestras empleado para el entrenamiento de los clasificadores SVM y KNN fue de 15, 30 y 50 muestras por signo, a continuación se presenta la implementación así como las pruebas realizadas a cada uno de los clasificadores:

TABLA XIX. IMPLEMENTACIÓN DEL CLASIFICADOR KNN.

Implementación del Clasificador KNN
<pre> //Conversión de la imagen para la predicción cv::resize(mImage,mImage,cvSize(50,100),0,0,CV_INTER_AREA); //Entrenamiento del clasificador KNearest knn(mFNN_N_Datos,mFNN_N_Clases,Mat(),false,3); //Transformación de la imagen en otra imagen equivalente de una sola fila y columna mp= mImage.reshape(1, 1); mp.convertTo(mp, CV_32FC1); //Predicción del tipo de signo perteneciente a la imagen dada como parámetro iFnn= (int)knn.find_nearest(mp,3); </pre>

TABLA XX. IMPLEMENTACIÓN DEL CLASIFICADOR SVM.

Implementación del Clasificador SVM
<pre>//Conversión de la imagen para la predicción cv::resize(mlmage,mlImage,cvSize(50,100),0,0,CV_INTER_AREA); //Transformación de la imagen en otra imagen equivalente de una sola fila y columna mp= mlmage.reshape(1, 1); mp.convertTo(mp, CV_32FC1); //Predicción del tipo de signo perteneciente a la imagen dada como parámetro iSvm=(int)svmClassifierNumeros.predict(mp);</pre>

Para probar la precisión de los clasificador SVM y KNN, se empleó un conjunto de 50 imágenes que contienen señas de números y letras respectivamente, los resultados emitidos por cada clasificador se describen a continuación:

TABLA XXI. PORCENTAJE DE PREDICCIÓN CLASIFICADOR SVM.

Muestras	Imágenes	Reconocidas	Porcentaje
15	Números	44	88 %
	Letras	42	84 %
30	Números	46	92 %
	Letras	45	90 %
50	Números	44	88 %
	Letras	40	80 %

TABLA XXII. PORCENTAJE DE PREDICCIÓN CLASIFICADOR KNN.

Muestras	Imágenes	Reconocidas	Porcentaje
15	Números	45	90 %
	Letras	42	84 %
30	Números	46	92 %
	Letras	46	92 %
50	Números	40	80 %
	Letras	41	82 %

En la primera fila de la TABLA XXI y TABLA XXII, se muestra el empleo de 15 muestras por signo, a través de las cuales se obtuvo un porcentaje de 86 y 87 % de aciertos en la clasificación, razón por la cual se duplicó el número de muestras con el objetivo de mejorar la clasificación, el nuevo porcentaje promedio obtenido fue de 91 y 92 % de aciertos, se empleó una tercera muestra con un número mayor al total de muestras anteriores, esta vez el clasificador tuvo un sobre-entrenamiento ya que el porcentaje obtenido es menor al esperado a más de ello, los archivos generados para la clasificación aumentaron proporcionalmente el tamaño, ya que el peso depende del número de muestras, por esta razón se decidió emplear un conjunto de 30 muestras por signo para la clasificación del alfabeto del Lenguaje de señas.

El porcentaje de error de los dos clasificadores es de 8 y 10 % respectivamente, para disminuir el porcentaje de error durante la predicción se combinó los dos algoritmos, la implementación para este tipo de clasificador se describe en el siguiente bloque de código:

TABLA XXIII. ALGORITMO PARA RECONOCER EL ALFABETO DE SEÑAS.

Implementación del algoritmo para reconocer el alfabeto de señas

```

int Clasificador::reconocer(Mat mImage, int iNumLet)
{
    //Conversión de la imagen para la predicción
    cv::resize(mImage,mImage,cvSize(50,100),0,0,CV_INTER_AREA);
    //Entrenamiento del clasificador
    KNearest knn(mFNN_N_Datos,mFNN_N_Clases,Mat(),false,3);
    //Transformación de la imagen a una imagen equivalente de una sola      fila y
columna
    mp= mImage.reshape(1, 1);
    mp.convertTo(mp, CV_32FC1);
    //Predicción del tipo de signo perteneciente a la imagen dada como parámetro
    iFnn= (int)knn.find_nearest(mp,3);
    //Predicción del tipo de signo perteneciente a la imagen dada como parámetro
    iSvm=(int)svmClassifierNumeros.predict( mp );
    //Si los dos clasificadores coinciden se devuelve el tipo de imagen al que pertenece el
parámetro
    if (iSvm==iFnn){
        return iSvm;
    }
}
    
```

Para las pruebas de este nuevo clasificador se empleó un conjunto de 150 imágenes que contienen señas de números y letras, los resultados emitidos por este clasificador se describen a continuación:

TABLA XXIV. PORCENTAJE DE PREDICCIÓN CLASIFICADOR SVM Y KNN.

Imágenes	Reconocidas	Porcentaje
150	145	96.66%

El porcentaje de error que maneja este nuevo clasificador es de 3.33 %, la combinación de los clasificadores mejora notablemente el porcentaje de predicción. La importancia de este tipo de clasificador radica en la conversión de la imagen a una sola fila y columna y en la asignación de la clase a la que pertenece.

1.3.3 Pruebas

Para mantener un adecuado orden en la programación de todas las clases, atributos y métodos se usó un estándar de codificación en el lenguaje C++, (ver Anexo 10), en donde se define cada una de las reglas para la declaración de variables, clases, funciones, etc, de manera que cualquier persona ajena al proyecto pueda hacer uso del mismo.

Para la generación de pruebas de caja blanca se emplearon dos herramientas que sirvieron para hacer las pruebas unitarias y pruebas de cobertura:

- **Unittest++:** es un framework para realizar pruebas unitarias en C++, fue diseñado para hacer el desarrollo de pruebas bajo una amplia variedad de plataformas, posee algunas características importantes como la simplicidad, portabilidad, velocidad y tamaño compacto que hacen que esta sea una de las librerías más utilizadas en la actualidad [100]. Posee una licencia de software libre a través de la cual se puede copiar, distribuir y modificar.
- **Gcov:** es una librería para realizar pruebas de cobertura, en los reportes que son emitidos por esta herramienta se pueden obtener detalles tales como [101]:
 - Frecuencia con que cada línea de código se ejecuta.
 - Líneas de código que son realmente ejecutadas.

Gcov trabaja junto con Lcov que es una herramienta gráfica que permite la visualización de las pruebas de cobertura. Recopila todos los datos generados por gcov y con esta información genera páginas HTML que contiene el código fuente junto con la información de cobertura [102,68,67].

1.3.3.1 Pruebas de Caja Blanca

A través de esta fase se realizaron pruebas individuales y colectivas del software, con el objetivo de encontrar vulnerabilidades que el sistema pueda tener de acuerdo a

diferentes entradas, de manera que el sistema soporte el ingreso de datos erróneos o inesperados, para garantizar la calidad y rendimiento del mismo.

1.3.3.1.1 Pruebas Unitarias

Las pruebas unitarias se realizaron únicamente a los métodos de las clases que tienen como tipo de retornos, valores que pueden ser predecidos, para la respectiva comparación entre los resultados emitidos, la mayoría de los métodos desarrollados para el reconocimiento del lenguaje de señas tienen como tipo de retorno vectores de puntos los cuales son dependientes de la distancia, la luminosidad, entre otros factores es por ello que a este tipo de métodos no se pudo realizar las pruebas unitarias correspondientes.

A continuación se describe los métodos a los que sí se pudo realizar las pruebas unitarias:

Clase: Clasificador.

Método: Reconocer

Este método es el responsable de predecir a qué clase pertenece cada una de las imágenes recibidas como parámetro, obtenida a través de la captura de video, para la realización de las pruebas unitarias se tendrá en cuenta algunas condiciones que tendrán que ser evaluadas.

- En caso de predecir números los resultados tendrán que estar en un rango de 1 a 10, en caso de predecir letras el resultado tendrá que estar en un rango de 1 a 26.
- Los tamaños de las imágenes deberán ser para el reconocimiento de números 50x100 pixeles y para el reconocimiento de las letras 50x50 pixeles de ancho y alto.

Tomando en cuenta estas consideraciones se realizaron las pruebas empleando diferentes muestras quedando los resultados emitidos por la prueba de la siguiente manera:

TABLA XXV. IMPLEMENTACIÓN DE LAS PRUEBAS UNITARIAS.

Implementación de las pruebas unitarias.

```

TEST(PruebaReconocedor){
for(int i=1; i<= 20; i++){
    //Almacenamiento de la dirección de la imagen
    nombre=":/n/"+QString::number(i)+".jpg";
    //Conversión de la imagen QImage a Mat
    QImage aux(nombre);
    Mat      mImagen(aux.height(),aux.width(),CV_8UC1,      (uchar*)aux.bits(),
aux.bytesPerLine());
    if (i<=5) {
        //Cambio de tamaño de la imagen
        cv::resize(mImagen,mImagen,cvSize(10,10),0,0,CV_INTER_AREA);
        valor = (int)rReconocedor->reconocer(mImagen,1);
        cout<<"Imagen "<<i<<".jpg"<<" representa el Número "<<valor<<"\n";
        // CHECK_EQUAL(valor,i);
    }
    if(i>=6 && i<=13){
        //Cambio de tamaño de la imagen
        cv::resize(mImagen,mImagen,cvSize(10,10),0,0,CV_INTER_AREA);
        valor = (int)rReconocedor->reconocer(mImagen,2);
        cout<<"Imagen "<<i<<".jpg"<<" representa la letra "<<valor<<"\n";
    if (i==6) {
        CHECK_EQUAL(valor,1);
    }else if (i==7) {
        CHECK_EQUAL(valor,2);
    }else if (i==8) {
        CHECK_EQUAL(valor,3);
    }
    }
}
}

```

Salida Generada por la prueba.

```
Salida
Números 1 2 3 4 5
Letras 1 2 3 10 13 19 21 23
No Números 0 0 0 0 0 0
No Letras -1079492468 -1079492468 -1079492468 -1079492468 -1079492468 -
1079492468 -1079492468
Success: 1 tests passed.
Test time: 2.35 seconds.
```

Al finalizar la prueba, se realizó las correcciones necesarias de manera que el sistema pueda aceptar:

1. Diferentes tamaños de las imágenes

El reconocedor acepto todas las imágenes de prueba, ya que internamente existe un cambio de tamaño dependiendo de la ventana que es enviada la imagen a testear.

2. Imágenes con señas de números y letras

Se completó esta prueba al 100%, ya que se hace uso de dos clasificadores para la predicción de las clases a las que pertenece la imagen capturada.

3. Imágenes que no contienen señales especificadas en el alfabeto de señas.

Durante el reconocimiento los clasificadores toman valores que se encuentran fuera del rango previsto para los números y para las letras, estos tipos de retorno no son tomados en cuenta al momento de seleccionar y presentar un caracter.

1.3.3.1.2 Pruebas de Cobertura

Una vez realizadas las pruebas unitarias se procedió a realizar las pruebas de cobertura con el objetivo de conocer los porcentajes de uso de cada una de las clases, métodos y líneas de código, para la respectiva optimización, ya sea con el mejoramiento o la eliminación de código sin utilizar.

El resultado obtenido al ejecutar la prueba de cobertura Figura 25, se describe a continuación:

LCOV - code coverage report

Current view: top level - /home/luigi/LenguajeSeñasEcuatoriano/LSE/modelo Test: coverage.info Date: 2014-08-11	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Hit</th> <th>Total</th> <th>Coverage</th> </tr> </thead> <tbody> <tr> <td>Lines:</td> <td style="text-align: center;">431</td> <td style="text-align: center;">481</td> <td style="text-align: center;">89.6 %</td> </tr> <tr> <td>Functions:</td> <td style="text-align: center;">69</td> <td style="text-align: center;">69</td> <td style="text-align: center;">100.0 %</td> </tr> <tr> <td>Branches:</td> <td style="text-align: center;">452</td> <td style="text-align: center;">924</td> <td style="text-align: center;">48.9 %</td> </tr> </tbody> </table>		Hit	Total	Coverage	Lines:	431	481	89.6 %	Functions:	69	69	100.0 %	Branches:	452	924	48.9 %
	Hit	Total	Coverage														
Lines:	431	481	89.6 %														
Functions:	69	69	100.0 %														
Branches:	452	924	48.9 %														

Filename	Line Coverage ↕		Functions ↕		Branches ↕		
colorpiel.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	23 / 23	100.0 %	6 / 6	50.0 %	31 / 62
colorpiel.h	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	1 / 1	100.0 %	1 / 1	-	0 / 0
dispositivo.cpp	<div style="width: 85%; height: 10px; background-color: yellow;"></div>	85.0 %	284 / 334	100.0 %	32 / 32	47.7 %	332 / 696
main.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	9 / 9	100.0 %	2 / 2	50.0 %	8 / 16
mano.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	20 / 20	100.0 %	7 / 7	50.0 %	12 / 24
persona.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	6 / 6	100.0 %	4 / 4	-	0 / 0
persona.h	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	1 / 1	100.0 %	1 / 1	-	0 / 0
procesamiento.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	25 / 25	100.0 %	5 / 5	54.2 %	26 / 48
reconocedor.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	51 / 51	100.0 %	5 / 5	56.1 %	37 / 66
reconocedor.h	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	1 / 1	100.0 %	2 / 2	50.0 %	2 / 4
rostro.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	9 / 9	100.0 %	3 / 3	50.0 %	3 / 6
rostro.h	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	1 / 1	100.0 %	1 / 1	50.0 %	1 / 2

Generated by: [LCOV version 1.9](#)

(a) Pruebas de cobertura del Controlador

LCOV - code coverage report

Current view: top level - /home/luigi/LenguajeSeñasEcuatoriano/LSE/controlador Test: coverage.info Date: 2014-08-11	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Hit</th> <th>Total</th> <th>Coverage</th> </tr> </thead> <tbody> <tr> <td>Lines:</td> <td style="text-align: center;">357</td> <td style="text-align: center;">388</td> <td style="text-align: center;">92.0 %</td> </tr> <tr> <td>Functions:</td> <td style="text-align: center;">48</td> <td style="text-align: center;">48</td> <td style="text-align: center;">100.0 %</td> </tr> <tr> <td>Branches:</td> <td style="text-align: center;">370</td> <td style="text-align: center;">764</td> <td style="text-align: center;">48.4 %</td> </tr> </tbody> </table>		Hit	Total	Coverage	Lines:	357	388	92.0 %	Functions:	48	48	100.0 %	Branches:	370	764	48.4 %
	Hit	Total	Coverage														
Lines:	357	388	92.0 %														
Functions:	48	48	100.0 %														
Branches:	370	764	48.4 %														

Filename	Line Coverage ↕		Functions ↕		Branches ↕		
acercade.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	13 / 13	100.0 %	5 / 5	50.0 %	2 / 4
ayuda.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	14 / 14	100.0 %	5 / 5	50.0 %	6 / 12
entrenamiento.cpp	<div style="width: 82.3%; height: 10px; background-color: yellow;"></div>	82.3 %	130 / 158	100.0 %	13 / 13	47.6 %	235 / 494
principal.cpp	<div style="width: 97.5%; height: 10px; background-color: green;"></div>	97.5 %	115 / 118	100.0 %	10 / 10	50.0 %	32 / 64
selectorcolorpiel.cpp	<div style="width: 100%; height: 10px; background-color: green;"></div>	100.0 %	85 / 85	100.0 %	15 / 15	50.0 %	95 / 190

Generated by: [LCOV version 1.9](#)

(b) Prueba de cobertura del Modelo

Figura 25. Pruebas de cobertura (a) modelo, (b) controlador.

En la prueba de cobertura se pueden observar cuatro columnas, cada una con un determinado porcentaje de uso, descritos de la siguiente manera:

- **Filename:** Representa el nombre de cada una de las clases empleadas en el proyecto.
- **Line Coverage:** Indica el número de líneas de código empleadas durante la prueba de cobertura.
- **Functions:** Muestra el número de funciones que se han empleado al ejecutar la prueba.
- **Branches:** Representa las ramificaciones o tomas de decisión empleadas durante la ejecución.

Las clases que se encuentran resaltadas con un color verde poseen un porcentaje de cobertura elevado superior a 90%, es por ello que se considera que su código se encuentra optimizado al máximo y que se ejecuta completamente.

El color rojo de la columna Branches, indica que no todas las condiciones que se encuentran en los métodos se cumplen, es por ello que se encuentran con un porcentaje inferior a 60%.

El porcentaje total de cobertura del proyecto se describe en la siguiente tabla:

TABLA XXVI. COBERTURA DEL PROYECTO.

	Líneas de código	Funciones
Modelo	89 %	100 %
Controlador	92 %	100 %

En los dos casos el porcentaje de branches o ramificaciones es menor a 50 %, ya que por lo general los métodos están compuestos por bucles condicionales if's, que contienen dos estados un verdadero y un falso. Los porcentajes de la cobertura son aceptables ya que se encuentran por encima del porcentaje límite que utiliza la librería gcov para la clasificación [103]:

TABLA XXVII. NIVELES DE COBERTURA.

90-100 % Cobertura Total
75-89 % Cobertura Media
0-74 % Cobertura Baja

1.3.3.2 Pruebas de Caja Negra

Al realizar las pruebas de caja negra a todo el sistema, se pudo constatar algunos inconvenientes en módulos que son de vital importancia para la ejecución del mismo:

1.3.3.2.1 Entrenamiento con Webcam y Formación de Palabras

Las salidas que producen estos módulos del sistema es la captura de las imágenes en tiempo real, la respuesta que tiene el sistema en la identificación del lenguaje de señas es demasiado tardía, es por ello que se optó por la división de funciones, reconocimiento de rostro (extracción), procesamiento de las regiones de interés y finalmente el reconocimiento de la seña, para la asignación de cada una de ellas a un hilo diferente que se ejecuta de forma paralela, ayudando así a mejorar la velocidad de procesamiento.

1.3.3.2.2 Verificación de datos ingresados en los entrenamientos

Este proceso permitía que se ingresen números de varios dígitos así como palabras completas, se agregaron controles para ingresar únicamente un número de un solo dígito y palabras de dos caracteres ya que en el alfabeto de señas se emplean las letras “ch”, “ll”, “rr”, con los cambios realizados este proceso verifica que los datos que se ingresan para conocer el signo o para realizar la comparación entre signo y caracter sean los adecuados, por ende no deben ser cadenas vacías, números cuando se pida una letra y viceversa, las respuestas del sistema a estos inconvenientes se muestran en la Figura 26.



Figura 26. Identificación de errores de ingreso.

1.3.3.2.3 Pruebas en diferentes Entornos

La mayoría de los métodos desarrollados para el reconocimiento del lenguaje de señas, son dependientes de la distancia, la luminosidad, entre otros factores es por ello que ha este tipo de métodos se los evaluó en diferentes lugares y circunstancias para determinar los entornos donde es factible la ejecución.

1. **Iluminación:** Con la iluminación adecuada el sistema puede segmentar la imagen de captura, y seguir únicamente el color asignado, por ende los resultados que emite durante la clasificación son aceptables Figura 27, mientras que con una iluminación inadecuada o la existencia de colores con demasiada intensidad el sistema no puede segmentar la imagen por el color, seleccionando para la fase de clasificación objetos que se encuentran próximos al umbral elegido, de esta manera el sistema emite resultados erróneos o simplemente no presenta resultados, ya que la no existen coincidencias de la imagen obtenida con los clasificadores Figura 28.



Figura 27. Prueba con Iluminación Adecuada.



Figura 28. Prueba con Iluminación inadecuada.

2. **Toma Nocturna:** El algoritmo de reconocimiento de señas se ve afectado cuando existe poca luminosidad o las fuentes de iluminación son escasas, la selección de cada uno de los valores que componen el umbral se hace dificultosa, ya que no existen colores definidos en el entorno, por ende el reconocedor no podrá seguir un objeto específico como se muestra en la Figura 29, y la predicción no se realiza por no coincidir con las clases que representan las letras o números, y en caso de suceder lo contrario las letras o números que emite el sistema luego del reconocimiento no corresponden a las señas realizadas por el usuario, que se encuentra manipulando el sistema.



Figura 29. Prueba nocturna.

3. **Distancia entre el usuario y la cámara:** la distancia adecuada para el reconocimiento del lenguaje de las señas, se encuentra en un rango de 60 a 100 cm, si este valor se ve afectado con exceso o se limita el valor de las distancias establecidas, el reconocimiento de las señas no se realiza, a continuación se detallan los inconvenientes que hacen que el sistema no funcione adecuadamente:

- Con una distancia inferior a la establecida, la imagen procesada como ROI, contiene un mayor número de filas y columnas lo que provoca que el sistema tiende a demorarse en la clasificación de la seña.
- Con un valor que excede la distancia establecida, aunque se detecte el color y se segmente la imagen, la clasificación no se logra, ya que el número de píxeles no supera el rango de la condición para la extracción de la ROI, en la Figura 30, se indica la ejecución de esta prueba.



Figura 30. Prueba con distancia mayor a 1m.

1.3.4 Implantar la aplicación con la que el docente contará para realizar las actividades de aprendizaje

Antes de la implantación la capacitación se hace necesaria para ello se ha desarrollado un plan que se puso en marcha (Ver Anexo 11), en él se detalla cada una de las fases y los recursos que se emplearon.

Para realizar las pruebas con los usuarios finales se realizó una remasterización del sistema operativo Ubuntu, en el que la aplicación se ejecute al arrancar el mismo, ya

que para la versión de Windows las últimas versiones de OpenCV no traen el soporte correspondiente haciendo que no se pueda ejecutar la aplicación bajo esta plataforma.

Luego de realizar las pruebas se procedió a realizar una encuesta al docente (Ver Anexo 13. Encuesta funcionalidad del software.), así como a los estudiantes, para conocer el nivel de aceptación del sistema, de la misma manera conocer si el sistema ha sufrido errores durante su manipulación.

A continuación se presenta el análisis de los resultados obtenidos para cada una de las preguntas realizadas en la encuesta al docente así como a los estudiantes:

1.3.4.1 Encuesta Realizada a los estudiantes

Las respuestas a las preguntas planteadas en la encuesta a los estudiantes se describen a continuación:

¿Tuvo alguna dificultad al ingresar al sistema?

Si () No ()

En caso de existir dificultad, descríbala:

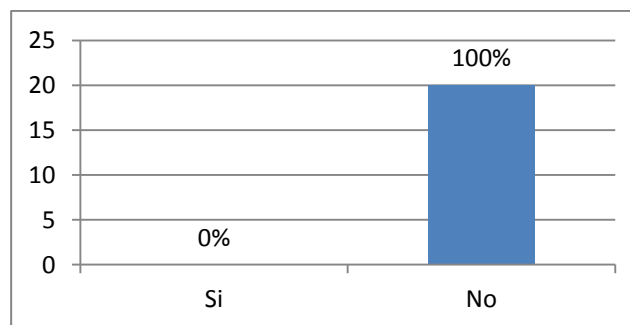


Figura 31. Resultados pregunta 1.

La interpretación de esta pregunta nos permite asegurar que los estudiantes encuestados no tuvieron ningún problema con el ingreso al sistema para el reconocimiento del lenguaje de señas, esto demuestra que el empleo del sistema no es complejo para su utilización.

Considera que el sistema es:

- Lento []
- Eficiente []
- Rápido []
- Complejo []
- Confiable []

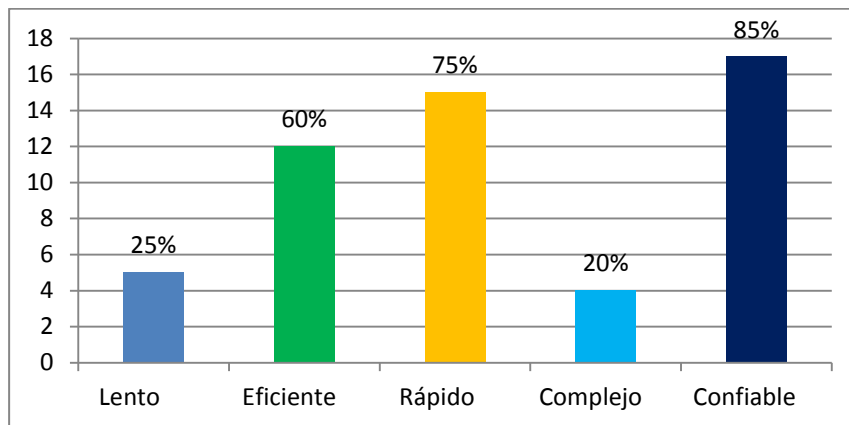


Figura 32. Resultados pregunta 2.

En la presente interrogante, los resultados de cada una de las opciones de respuesta se encuentran divididas sin embargo las opciones que mayoritariamente prevalecen son las que califican al sistema como rápido y confiable, de esta manera la mayoría de los estudiantes encuestados afirma que estas características corresponden a la aplicación realizada frente a un conjunto dividido que opinan lo contrario.

¿Se presentaron problemas al ejecutar alguno de los siguientes procesos?

Proceso	Si ()	No ()
Conocimiento del abecedario de señas.	Si ()	No ()
Entrenamiento del abecedario de señas.	Si ()	No ()
Entrenamiento en tiempo real.	Si ()	No ()
Formación de palabras.	Si ()	No ()

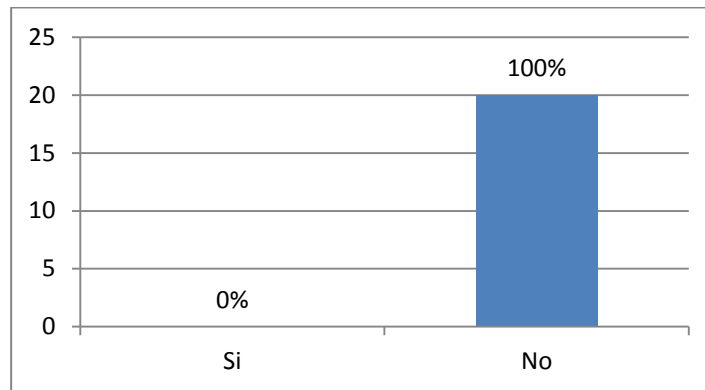


Figura 33. Resultados pregunta 3.

Según los resultados obtenidos para esta pregunta todos los estudiantes afirmaron no tener ningún problema al ejecutar las diferentes funcionalidades del sistema.

Considera que el sistema posee una interfaz:

- Fácil de utilizar []
- Compleja []
- Agradable []
- Complicada []

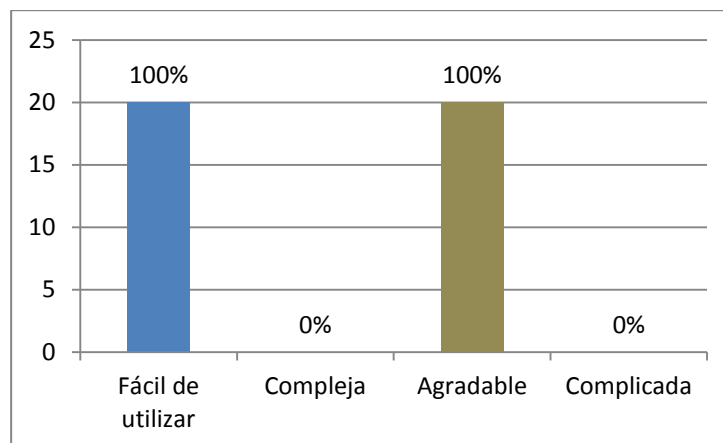


Figura 34. Resultados pregunta 4.

Todos los estudiantes aseguran en esta pregunta que la interfaz gráfica de usuario es agradable y que además es fácil de utilizar, permitiendo de esta manera tener un rápido acceso a las diferentes funcionalidades del sistema.

¿Cree que el diseño del sistema es acorde a las necesidades que presenta el aprendizaje del lenguaje de señas?

Si () No ()

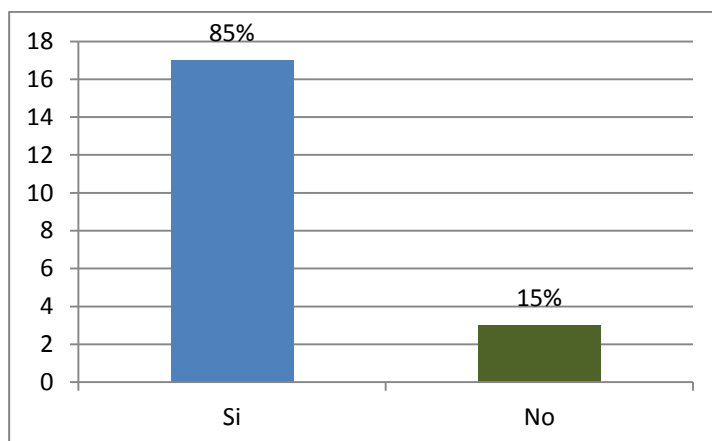


Figura 35. Resultados pregunta 5.

Dentro de las respuestas contenidas para esta pregunta la primera y que mayor porcentaje tiene afirma que los estudiantes creen que el sistema sirve para la enseñanza del lenguaje de señas en el nivel básico ya que no se requiere de un profesor, que lo supervise en la ubicación de las manos así como en la muestra de las señas, mientras que la respuesta restante aseguró que el docente es indispensable en la enseñanza.

1.3.4.2 Encuesta Realizada al Docente

La encuesta se la realizó al Subcoordinador de la Asociación de Sordos de Loja “Virgen del Cisne”, ya que también es un docente que labora en la institución.

¿Cómo se ha evidenciado la utilización del sistema en los estudiantes?

- Mayor Nivel de conversación. []
- Rapidez en la realización de señas. []
- Mejor comprensión. [x]

El factor en el cual se ha visto mayor desarrollo en los estudiantes es la comprensión, los demás factores como el nivel de conversación y rapidez en la realización de las señas son complementarios al mismo.

¿El empleo del sistema ha facilitado el entrenamiento del lenguaje de señas?

Si, ya que ha ayudado a los niños al mejoramiento del aprendizaje puesto que se lo puede realizar en cualquier parte donde se disponga de un ordenador, evitando que únicamente el docente sea quien indique las posiciones de las letras del abecedario.

¿De qué manera ha ayudado la implantación del sistema, en la comunicación entre maestros, estudiantes, y demás integrantes de la asociación?

Mejoramiento del aprendizaje en los jóvenes y niños que aún no conocen del Lenguaje de Señas Ecuatoriano, además el sistema cuenta con un módulo de traducción con el que se podría realizar la comunicación entre con y sin discapacidad auditiva.

¿El sistema implantado ha contribuido a mejorar el nivel de aprendizaje de los estudiantes?

Si sobre todo a quienes están empezando con el lenguaje de señas, los estudiantes jóvenes son quienes más uso le dan al sistema, ya que están relacionados con la tecnología.

La ayuda del sistema permite tratar con mayor rapidez las temáticas en la enseñanza del nivel básico.

Sí, es evidente la reducción del tiempo de enseñanza del abecedario así como la formación de palabras básicas, ya que se puede emplear al software para que realice esta función, y únicamente se tiene que enseñar a manejar el sistema.

De acuerdo con los resultados obtenidos luego de la tabulación de las encuestas el sistema para el reconocimiento del lenguaje de señas ha mejorado el aprendizaje de los alumnos de la asociación de sordos de Loja "Virgen del Cisne", reduciendo el tiempo en que los docentes dedican a cada uno de los estudiantes en la enseñanza, ya que únicamente se limita a indicar el funcionamiento del sistema, el empleo del mismo no es complejo ya que dispone de interfaces gráficas simples y amigables que permiten que los estudiantes rápidamente lo dominen, viendo un mayor aprendizaje del lenguaje de señas por parte de los estudiantes más jóvenes ya que estos se adaptan fácilmente al empleo de la tecnología.

g. Discusión

1. Desarrollo de la propuesta alternativa

Luego de culminar con el desarrollo del Trabajo de Titulación denominado “**Visión Artificial aplicada para el reconocimiento del lenguaje de señas**”, se procede a realizar la evaluación de cada uno de los objetivos con el propósito de determinar si fueron o no cumplidos, a continuación se describe cada uno de ellos:

- **Objetivo Específico 1:** Seleccionar casos de éxito que hagan uso de la visión artificial para determinar cuán factible es realizar el sistema informático para el reconocimiento del lenguaje de señas.

La existencia de múltiples sistemas que hacen uso de la visión artificial han contribuido en la determinación de la factibilidad, en la construcción del sistema para el reconocimiento del alfabeto de señas, toda esta información fue seleccionada y clasificada empleando un gestor de contenido denominado Zotero, con el objetivo de tenerla siempre disponible y de esta manera poderla reutilizarla.

- **Objetivo Específico 2:** Realizar el análisis de los requisitos necesarios para llevar a cabo la construcción del sistema.

Este objetivo al igual que el primero hacen uso de la revisión literaria, para el análisis de los requisitos, se llevó a cabo una encuesta para conocer si los equipos con los que cuenta la asociación poseen las características básicas para la ejecución del sistema, los resultados obtenidos muestran que poseen dispositivos actuales con características potentes (ver Anexo 2. Encuesta para conocer el lenguaje de Señas que emplea en la enseñanza la Asociación de Sordos de Loja “Virgen del Cisne”.) que permitirán la ejecución normal de la aplicación. El análisis de factibilidad se lo realizó en base al Hardware y Software disponibles (Ver Sección Resultados apartado 1.1 Componentes hardware y software).

- **Objetivo Específico 3:** Analizar las diferentes bibliotecas existentes para visión artificial, y seleccionar la que más se adapte a los requerimientos del sistema a desarrollar.

Para el logro eficiente de éste objetivo, se probaron cada una de las bibliotecas descritas en la revisión literaria (Ver Librerías de desarrollo), junto a varios métodos básicos, de acuerdo a las tablas de funciones o métodos que están descritas en cada una de ellas, esto con el fin de conocer la facilidad de manejo, documentación, instalación y soporte que poseen.

Luego de este procedimiento se realizó la selección de la librería de visión artificial OpenCV, por la cantidad de aplicaciones realizadas en base a ella, la extensa documentación que posee y la facilidad de integración e implementación.

- **Objetivo Específico 4:** Diseño, Implementación e implantación del sistema.

Para el desarrollo de este objetivo se empleó la metodología RAD, y cada una de sus fases fue implementada de forma detallada (ver Sección Resultados 2. Diseño e Implementación del sistema), finalmente se procedió a la capacitación de los alumnos pertenecientes a la asociación e implantación del sistema con la que contarán para el aprendizaje del lenguaje de señas.

2. Valoración técnica económica y ambiental

Para conocer si el sistema fue factible de realizar, se hizo una valoración considerando tres aspectos esenciales en lo técnico, económico y social, a continuación son descritos cada uno de ellos.

2.1 Valoración técnica-económica

El Trabajo de Titulación hizo necesario el empleo de varios conjuntos de materiales, a continuación se describe cada uno de ellos además de los costos del talento humano, técnicos, tecnológicos y servicios básicos necesarios:

2.1.1 Talento Humano

El proyecto fue planificado para ser desarrollado por un investigador, con la asesoría de un docente, el tiempo estimado para el desarrollo del proyecto es de seis meses en el que se emplearán 8 horas diarias, en la TABLA XXVIII, se describe el número total de horas empleado y el costo por hora.

TABLA XXVIII. TALENTO HUMANO.

Investigador	Nº de Horas	V. Unitario	Total
Luis Sivisapa	960	\$5.00	\$4800.00
Docente			
Ing. Henry Paz	20	\$00 .00	\$00.00
Subtotal			\$4800.00

2.1.2 Recursos Materiales

Durante el proyecto se hizo necesario el empleo de diversos suministros de oficina como hojas, perfiles, impresiones, copias, anillados en la TABLA XXIX, se describe cada uno de ellos:

TABLA XXIX. RECURSOS MATERIALES.

	Cantidad	V. Unitario	Total
Copias	200	\$0.05	\$10.00
Anillado	7	\$2.00	\$14.00
Empastado	3	\$7.00	\$21.00
Perfiles	8	\$0.60	\$4.80
Impresiones	800	\$0.15	\$120.00
Subtotal			\$169.80

2.1.3 Recursos Técnicos

Para el desarrollo de la aplicación se emplearon recursos hardware, como un ordenador que disponga de una cámara de manera que permita captar imágenes en tiempo real, así como recursos software en los cuales se incluye el lenguaje de programación C++, librerías de visión artificial así como algunas otras herramientas que nos permitirán realizar toda la documentación, en la TABLA XXX, se describen todos estos recursos.

TABLA XXX. RECURSOS TÉCNICOS.

Hardware	Cantidad	Depreciación	Total
Portátil	1	\$400.00	\$400.00
Impresora	1	\$120.00	\$120.00
Software	Cantidad	V. Unit	Total
Paquete de Libre Office	1	\$10.00	\$10 .00
Latex	1	\$10.00	\$10.00
OpenCV	1	\$10.00	\$10.00
Subtotal			\$550.00

2.1.4 Servicios Básicos

En la TABLA XXXI, se describe el costo de los servicios básicos utilizados para el desarrollo del presente proyecto, algunos de ellos como energía eléctrica, teléfono e internet como medio de consulta, de la misma manera el transporte para asistir a cada una de las asesorías.

TABLA XXXI. SERVICIOS BÁSICOS.

	Valor/mes	Total
Energía Eléctrica	\$20.00	\$120.00
Teléfono	\$7.00	\$42.00
Internet	\$20.00	\$120.00
Transporte	\$50.00	\$300.00
	Subtotal	\$582.00

El presupuesto total que se empleó para emprender con este tipo de proyecto se detalla en la TABLA XXXII, que se presenta a continuación:

TABLA XXXII. PRESUPUESTO TOTAL.

	Total	
Talento Humano	\$4800.00	
Recursos Económicos	\$96.20	
Recursos Técnicos	\$580.00	
Servicios Básicos	\$582.00	
	Subtotal	\$6782.05
	Imprevistos 5%	\$316.41
	Total PFC	\$6644.61

2.2 Valoración ambiental

El Trabajo de Titulación es totalmente factible ya que la asociación de sordos de Loja “Virgen del Cisne”, cuenta con los recursos hardware necesario y un buen estado de los mismos para la ejecución de la aplicación, con el empleo de este sistema se evita gasto innecesario de tinta y de papel ya que las señales de video son captadas a través de una cámara USB o cámara web y todos los resultados emitidos luego del procesamiento de las imágenes y video en tiempo real, son observados mediante un dispositivo de salida como el monitor de la computadora donde está ejecutándose el sistema, siendo esta una de las formas en que se contribuirá en gran medida al cuidado del medio ambiente.

h. Conclusiones

Al finalizar con el desarrollo del sistema para el reconocimiento del abecedario del lenguaje de señas se pudo llegar a las siguientes conclusiones:

- El empleo de una metodología de desarrollo ágil facilita la construcción de sistemas de visión artificial ya que permite una mayor interacción con los usuarios y flexibilidad de cambios durante todo el ciclo de desarrollo.
- La combinación de algoritmos KNN y SVM para el reconocimiento de las señas, permite la reducción del porcentaje de error durante la clasificación.
- Los cambios de iluminación se deben controlar sobre todo para la construcción de aplicaciones no industriales, ya que de ello depende el éxito o fracaso de los resultados que emita el sistema.
- EL sistema de **“Visión Artificial aplicada para el reconocimiento del lenguaje de señas”**, se desarrolló con el empleo de herramientas de software libre, lo que permitió reducir el costo de la aplicación sin perder la calidad ni la robustez.
- OpenCV cuenta con una extensa documentación así como la fácil integración en varios lenguajes de programación, lo que facilita su uso en la implementación de este tipo de sistemas.
- El desarrollo del presente sistema facilitó la enseñanza, evitando que el docente revise cada una de las señas realizadas a cada uno de los estudiantes, ya que el docente únicamente explica cómo se debe ejecutar el sistema y sus diversas funcionalidades, evidenciándose mayor interés en los estudiantes jóvenes al estar más relacionados con el uso de las tecnologías.

i. Recomendaciones

- Antes de empezar a utilizar una herramienta de visión artificial es necesario indagar varias alternativas, con el propósito de elegir la adecuada para un determinado proyecto.
- Para el mejoramiento de la velocidad en la construcción de sistemas que hagan uso de la visión artificial se recomienda:
 - Reducir el tamaño de las imágenes de entrada y una vez realizado el respectivo procesamiento de la región de interés, transformarlas a su tamaño normal.
 - Dividir por hilos de ejecución los métodos que hagan uso múltiples funciones de la herramienta de visión artificial.
 - Hacer uso de un IDE como QT-Creator ya que nos facilita la generación de interfaces gráficas así como la integración de OpenCV con el lenguaje de programación seleccionado.
- Antes de la ejecución de cualquier sistema que haga uso de la visión artificial se debe tener en cuenta una buena fuente de iluminación y un ambiente adecuado.
- Para posibles ampliaciones de funcionalidades del sistema, se sugiere continuar con la utilización de los estándares de programación que fueron implementados, de tal manera que mantengan los estilos de las clases, funciones, objetos, etc.

Trabajos Futuros

La ejecución del Trabajo de Titulación abre nuevas líneas de investigación relacionadas con mejoras en el sistema o nuevas aplicaciones en base al mismo:

- Implementar un traductor del Lenguaje de Señas que comprenda la clasificación de palabras enteras, mediante el empleo de una sola seña o de señas que contengan movimientos complejos.
- Implementar un sistema que permita la generación de cursos personalizados, de acuerdo al nivel de conocimientos o tipo de aprendizaje de cada alumno.
- Hacer uso de la inteligencia artificial para implementar un avatar que permita entablar diálogos con los alumnos.
- Migrar la aplicación para la traducción del Lenguaje de Señas a dispositivos móviles.

j. Bibliografía

- [1] Esther Alicia Amate, Armando J Vásquez, *Discapacidad: lo que todos debemos saber*. Panamerican health org., 2006.
- [2] La Jornada, «Crean científicos del Cinvestav lentes con inteligencia artificial para invidentes.», nov-2014. [En línea]. Disponible en: <http://www.jornada.unam.mx/ultimas/2014/05/16/crean-cientificos-del-cinvestav-lentes-con-inteligencia-artificial-para-invidentes-4412.html>. [Accedido: 09-nov-2014].
- [3] C. E. G. Alicia Sarabia Sánchez, «Clasificaciones de la OMS sobre la discapacidad». [En línea]. Disponible en: <http://www.sld.cu/galerias/pdf/sitios/rehabilitacion-temprana/clasificacionesomsdiscapacidad.pdf>. [Accedido: 14-oct-2014].
- [4] Dr. Robert L. Schalock, «Hacia Una nueva Concepción de la Discapacidad». [En línea]. Disponible en: <https://campus.usal.es/~inico/investigacion/jornadas/jornada3/actas/conf6.pdf>. [Accedido: 14-oct-2014].
- [5] Universia España, «Tipos de Discapacidad». [En línea]. Disponible en: <http://universitarios.universia.es/voluntariado/discapacidad/>. [Accedido: 14-oct-2014].
- [6] Conadis, «Datos Generales por Provincia», 18-oct-2013. [En línea]. Disponible en: http://www.conadis.gob.ec/index.php?option=com_wrapper&view=wrapper&Itemid=69.
- [7] Conadis, «Conadis Registro Nacional Discapacidades». [En línea]. Disponible en: http://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2014/03/conadis_registro_nacional_discapacidades.pdf. [Accedido: 20-oct-2014].
- [8] Bernal Gavilanes Dione Casandra, «Percepciones y actitudes sociales sobre la sexualidad de las personas con discapacidad en el Ecuador». [En línea]. Disponible en: <http://dspace.utpl.edu.ec/bitstream/123456789/5656/1/Bernal%20Gavilanes%20Dione%20Casandra.pdf>. [Accedido: 14-oct-2014].
- [9] Conadis, «Ley Orgánica de Discapacidades», 18-oct-2013. [En línea]. Disponible en: <http://www.plataformaconadis.gob.ec/ley.html>.

- [10]Ciencia Popular, «La Inteligencia». [En línea]. Disponible en: <http://www.cienciapopular.com/biologia-y-fosiles/la-inteligencia>. [Accedido: 14-oct-2014].
- [11]Inteligencia Artificial, «Inteligencia Artificial: Concepto, Características y Metodologías de La Inteligencia Artificial». [En línea]. Disponible en: <http://inteligenciaartificialudb.blogspot.com/2008/01/concepto-caractersticas-y-metodologas.html>. [Accedido: 14-oct-2014].
- [12]J. Delgado, «Redes Neuronales», 18-oct-2013. [En línea]. Disponible en: <http://goo.gl/g5KxF>.
- [13]A. Serrano, «Redes Neuronales Artificiales». [En línea]. Disponible en: http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf. [Accedido: 14-oct-2014].
- [14]Wiiibooks, «Ingeniería del conocimiento», 18-oct-2013. [En línea]. Disponible en: http://es.wikibooks.org/wiki/Ingenier%C3%ADa_del_conocimiento/Versi%C3%B3n_para_imprimir. [Accedido: 14-oct-2014].
- [15]Universidad de Sevilla, «Introducción a la Ingeniería del Conocimiento», 18-oct-2013. [En línea]. Disponible en: <http://www.cs.us.es/blogs/iic2012/files/2012/02/IIC-Teoria1.pdf>. [Accedido: 14-oct-2014].
- [16]Maria Isabel Alfonso Galipienso, «Inteligencia artificial: modelos, técnicas y áreas de aplicación», 18-oct-2013. [En línea]. Disponible en: http://books.google.com.ec/books?id=_spC6S7UfZgC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. [Accedido: 14-oct-2014].
- [17]Universidad Nacional del Nordeste, «Minería de Datos», 18-oct-2013. [En línea]. Disponible en: http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Mineria_Datos_Vallejos.pdf. [Accedido: 14-oct-2014].
- [18]Viaani Lily Álvarez Prados, «Bases de Datos Espacio–Temporales». [En línea]. Disponible en: <http://www.sisman.utm.edu.ec/libros/FACULTAD%20DE%20CIENCIAS%20INFORMATICAS/CARRERA%20DE%20INGENIER%3%8DA%20DE%20SISTEMAS%20INFORMATICOS/ELECTIVAS/Spatial%20Databases/Alvarez%20Prados.pdf>.
- [19]Universidad de Corruña, «Visión artificial e interacción sin mandos», 18-oct-2013. [En línea]. Disponible en:

- <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/VisionArtificial/index.html>. [Accedido: 14-oct-2014].
- [20] Aranguren Zapata Alejandro y Vela Asin Tiffany, «Sistema de seguimiento de objetos mediante procesamiento digital de imágenes aplicado al control de robots autónomos». [En línea]. Disponible en: http://repositorioacademico.upc.edu.pe/upc/bitstream/10757/305116/1/aranguren_zapata-delfos.pdf. [Accedido: 14-oct-2014].
- [21] OpenCV, «OpenCV Tutorials». [En línea]. Disponible en: <http://docs.opencv.org/doc/tutorials/tutorials.html>. [Accedido: 15-oct-2014].
- [22] Universidad Nacional de Quilmes, «Iluminación para las aplicaciones de Visión Artificial». [En línea]. Disponible en: <http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Tipos%20de%20Iluminacion%20B3n.pdf>. [Accedido: 14-oct-2014].
- [23] Universidad Politécnica Salesiana, «Visión Artificial». [En línea]. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/220/2/Capitulo%201.pdf>. [Accedido: 14-oct-2014].
- [24] Mónico Manuel Ponce González, «Visión por Computador para UAS». [En línea]. Disponible en: http://oa.upm.es/14316/2/PFC_MONICO_MANUEL_PONCE_GONZALEZ_B.pdf. [Accedido: 14-oct-2014].
- [25] Francisco Javier García Fernández, «Reconocimiento de Objetos en una cocina con Webcam». [En línea]. Disponible en: http://e-archivo.uc3m.es/bitstream/handle/10016/10007/PFC_FranciscoJavier_Garcia_Fernandez.pdf?sequence=3. [Accedido: 14-oct-2014].
- [26] U. E. Gobierno de España, «“Aplicación práctica de la visión artificial en el control de procesos industriales”.» [En línea]. Disponible en: http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf.
- [27] Cindy González, Mabel López y Diego González, «“Desarrollo de un sistema de Visión Artificial para el reconocimiento de placas en vehículos particulares”.» [En línea]. Disponible en: <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/43331.pdf>.
- [28] Helen Rodríguez, «Detección y extracción de placas de vehículos en señales de video».

- [29]Richard Gutiérrez, Ma. Fernanda Frydson y Phd. Boris Vintimilla, «“Aplicación de Visión por Computador para el Reconocimiento Automático de Placas Vehiculares utilizando OCR's Convencionales”», 11-oct-2014. [En línea]. Disponible en: http://www.dspace.espol.edu.ec/bitstream/123456789/19074/1/Paper_Gutierrez_Frydson_Vintimilla.pdf. [Accedido: 20-oct-2014].
- [30]Vivianne A. Mahecha y Julián Quiroga Sepúlveda, «“Sistema de Reconocimiento y Lectura de Placas de Vehículos en Movimiento ”.» [En línea]. Disponible en: [pwp.etb.net.co/gaquirogar/files/STSIVA08\(Placas\).pdf](http://pwp.etb.net.co/gaquirogar/files/STSIVA08(Placas).pdf).
- [31]Carlos Choez, Steve Salas y , Boris X. Vintimilla, «“Normalización de imágenes de placas vehiculares a través de corrección geométrica”.» [En línea]. Disponible en: http://www.dspace.espol.edu.ec/bitstream/123456789/24533/1/FIEC_Normalizaci%C3%B3n%20de%20im%C3%A1genes%20de%20placas%20vehiculares%20a%20trav%C3%A9s%20de%20correcciones%20geom%C3%A9tricas.pdf.
- [32]Cristian Andrés Tasiguano Pozo, «“Desarrollo de algoritmos para el reconocimiento de placas de vehículos”.» [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/3867/1/CD-3672.pdf>.
- [33]Eusko Jauraritzta, «“OCR: Tecnología para el Reconocimiento Óptico de Caracteres en una imagen”.» .
- [34]Antonio Ignacio Betancor Pérez, «“Sistema De Reconocimiento de Matrículas Basado en Visión Artificial para Control de Acceso”.» [En línea]. Disponible en: <http://repositorio.bib.upct.es/dspace/bitstream/10317/725/1/pfm21.pdf>.
- [35]Kineo, «“Reconocimiento Automático del Número de Placa”.» .
- [36]Enrique Antón López, «“Reconocimiento automático de lenguaje de signos:Lenguaje ASL”.» [En línea]. Disponible en: <http://www.maia.ub.es/~sergio/linked/enrique09.pdf>.
- [37]Beatriz Nathalia Serrato, «“Reconocimiento del Lenguaje de Gestos manuales Alfabéticos mediante Visión Artificial”.» [En línea]. Disponible en: <http://www.iiis.org/CDs2008/CD2009CSC/CISCI2009/PapersPdf/C198VV.pdf>.
- [38]Ernesto de la Rocha Gómez, «“Mejora del Tráfico en un cruce regulado por semáforos mediante un sistema basado en Visión Artificial”.» [En línea]. Disponible en: <http://www.iit.upcomillas.es/pfc/resumenes/4a40baa5a5a7c.pdf>.
- [39]Universitat de les Illes Balears, «“Aplicaciones de la la Visión Artificial”.» [En línea]. Disponible en: http://dmi.uib.es/~ygonzalez/VI/Material_del_Curso/Teoria/Aplicaciones_VC.PDF.

- [40]Redalyc.org, «“Estacion de Control de calidad por Visión Artificial para un centro de manufactura integrada por computador (CIM)”» [En línea]. Disponible en: <http://www.redalyc.org/pdf/477/47711103.pdf>.
- [41]Hugo Sánchez y Ana Morales, «“Sistema de Visión Artificial para la inspección seleccion y control de calidad de fresas”» .
- [42]Victor Vargas, «“Sistema de Visión Artificial para el control de piezas cromadas”» [En línea]. Disponible en: <http://tesis.bnct.ipn.mx/dspace/bitstream/123456789/7250/1/68.pdf>.
- [43]Castro Alfredo C., Schugurensky Carlos y Kuchen Benjamín, «“Detección de anomalías en envases de vidrio mediante Visión Artificial”» [En línea]. Disponible en: http://www.inaut.unsj.edu.ar/Files/Cc1042_96.pdf.
- [44]Ledda I. Larcher, Pedro M. Juárez, Ana I Ruggeri, Enrique M. Biasoni, Carlos A. Cattaneo y Gustavo A. Villalba, «“Ponderacion de calidad en frutas usando técnicas de visión artificial para la estimación de daños”» [En línea]. Disponible en: <http://www.cimec.org.ar/ojs/index.php/mc/article/viewFile/4498/4428>.
- [45]Córdova Alfredo y Filorio Aldo, «Programación Inteligente de un brazo de robot, controlado por visión, derivado del proyecto de Investigación CGPI 20040406». [En línea]. Disponible en: <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/25/1/Documento%20de%20tesisfilorio.pdf>. [Accedido: 14-oct-2014].
- [46]Wikispaces, «Opensuse - ventajas y desventajas de los sistemas operativos». [En línea]. Disponible en: <http://opensuse.wikispaces.com/VENTAJAS+Y+DESVENTAJAS+DE+LOS+SISTEMAS+OPERATIVOS>. [Accedido: 14-oct-2014].
- [47]Francisco Olcina Grande, «Análisis, diseño e implantación de un sistema de servidores departamental basado en máquinas virtuales».
- [48]Alcibar Carmen, «Reconocimiento de imágenes en tiempo real aplicando visión robótica para el laboratorio de sistemas lógicos microprogramables de la facultad de ciencias informáticas».
- [49]Francisco Javier Romero Galey, «Integración de sistema de visión artificial y robot en aplicación tipo pick&place».
- [50]Universidad de Valladolid, «Características de Java como lenguaje de programación». [En línea]. Disponible en: <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>. [Accedido: 14-oct-2014].

- [51]Deltalinuxer, «En el campo de batalla: C y Java». [En línea]. Disponible en: <http://deltalinuxer.wordpress.com/2007/07/10/en-el-campo-de-batalla-c-y-java/>. [Accedido: 14-oct-2014].
- [52]Ph.D. Carlos Óscar Sánchez Sorzano, «Curso de utilización práctica de Matlab». [En línea]. Disponible en: <http://biocomp.cnb.csic.es/~coss/Docencia/MATLAB/ApuntesGrande.pdf>. [Accedido: 14-oct-2014].
- [53]Francisco Cifuentes, «Comparativa plataformas Java - .Net». [En línea]. Disponible en: <http://www.alcaldiabogota.gov.co/docsig/documentos/Solicitudes/2213300GS006.pdf>. [Accedido: 14-oct-2014].
- [54]Celia Castillo, «Python». [En línea]. Disponible en: <http://www.it.uc3m.es/spickin/docencia/comsoft/presentations/spanish/doc/Python.pdf>. [Accedido: 14-oct-2014].
- [55]Miguel Cazorla, «JavaVis: Una librería para visión artificial en Java». [En línea]. Disponible en: <http://rua.ua.es/dspace/bitstream/10045/12363/6/Jenui01-1.pdf>. [Accedido: 14-oct-2014].
- [56]Fernando Ortiz Renilla, «Creación de mapas visuales panorámicos mediante técnicas de visión artificial». [En línea]. Disponible en: http://e-archivo.uc3m.es:8080/bitstream/handle/10016/15231/pfc_fernando_ortiz_renilla_2012.pdf?sequence=8. [Accedido: 15-oct-2014].
- [57]Alejandro Furfaro, «Manejo de Bibliotecas Opencv». 01-oct-2210.
- [58]Ing. Francisco Carlos Calderón, «Opencv». [En línea]. Disponible en: <http://opencvjaveriana.wikispaces.com/file/view/OpenCV-1.pdf>. [Accedido: 15-oct-2014].
- [59]Wikispace, «Sistemas de visión artificial». [En línea]. Disponible en: <http://proyectojefer.wikispaces.com/file/view/Visi%C3%B3n+artificial.pdf>. [Accedido: 15-oct-2014].
- [60]MathWorks, «Image Processing Toolbox - MATLAB». [En línea]. Disponible en: http://www.mathworks.com/products/image/index.html?s_tid=gn_loc_drop. [Accedido: 14-oct-2014].
- [61]Ingeniería Superior de Informática, «Fundamentos de la “Image Processing toolbox” de Matlab». [En línea]. Disponible en: <http://serdis.dis.ulpgc.es/~ii->

- vpc/MatDocen/notas_practicas/MATLAB/imgtool/images_tb.pdf. [Accedido: 15-oct-2014].
- [62] José Luis Pérez Yubero, «Técnicas de Tracking y Autocalibración», 2001. [En línea]. Disponible en: <http://www.elai.upm.es/webantigua/spain/Investiga/GCII/personal/jyubero/main63.pdf>. [Accedido: 15-oct-2014].
- [63] Debian.org, «Cumplir los requisitos mínimos de hardware». [En línea]. Disponible en: <http://www.debian.org/releases/stable/i386/ch03s04.html.es>. [Accedido: 14-oct-2014].
- [64] Microsoft Support, «Requisitos del sistema para los sistemas operativos Windows XP». [En línea]. Disponible en: <http://support.microsoft.com/kb/314865/es>. [Accedido: 14-oct-2014].
- [65] Sebastian Bronte Palacios, «Sistema de detección y reconocimiento facial de conductores mediante sistemas de visión computacional».
- [66] Lic. María Esther García Chang, «Diseño e implementación de una herramienta de detección facial». [En línea]. Disponible en: <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/6111/1/DISENOIMPLEMENTACIONAL.pdf>. [Accedido: 15-oct-2014].
- [67] Arranz Aranda, «Interacción persona-computador basada en el reconocimiento visual de manos Interacción persona-computador basada en el reconocimiento visual de manos».
- [68] Walter Charles Sousa Seiffert Simões, «Visión por computador para manos a base de reconocimiento de gestos para la interacción con los sistemas operativos de escritorio Windows y Linux».
- [69] Universidad de Andalucía, «OCW, Detección de Caras». [En línea]. Disponible en: http://ocw.unia.es/ciencias-tecnologicas/tecnologia-del-ocio/materiales-basicos-folder/html/B2_U3/deteccion_de_caras.html/skinless_view. [Accedido: 15-oct-2014].
- [70] Naotoshi Seo, «OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features)». [En línea]. Disponible en: <http://note.sonots.com/SciSoftware/haartraining.html>. [Accedido: 15-oct-2014].
- [71] OpenCV, «Cascade Classifier Training». [En línea]. Disponible en: http://docs.opencv.org/doc/user_guide/ug_traincascade.html. [Accedido: 15-oct-2014].

- [72]Solid-ec.org, «Entrenar Opencv». [En línea]. Disponible en: <http://solid-ec.org/?q=node/104>. [Accedido: 15-oct-2014].
- [73]Modesto Fernando Castrilló Santana, «Tutorial Viola-Jones». [En línea]. Disponible en: <http://gias720.dis.ulpgc.es/Gias/MODESTO/TutorialViolaJones.pdf>. [Accedido: 15-oct-2014].
- [74]Juan Carlos Ortega Pérez, «Multi-clasificación discriminativa por partes mediante códigos Correctores de Errores».
- [75]Julian Dondero, «Modelos activos de apariencia y máquinas de soporte vestorial para reconocimiento de expresiones faciales en tiempo real».
- [76]Borja Fernández, «Reconocimiento de caras en entornos no controlados», 20-jun-2012. [En línea]. Disponible en: http://upcommons.upc.edu/pfc/bitstream/2099.1/15642/4/Reconocimiento_de_caras_en_entornos_no_controlados,_Borja_Pelaez_Fernandez.pdf. [Accedido: 15-oct-2014].
- [77]Leslie Ekas, «Beneficios de las metodologías ágiles en el desarrollo de sistemas de software.» [En línea]. Disponible en: https://www.ibm.com/developerworks/community/blogs/rationalspain/entry/5_beneficios_de_las_metodolog_C3_ADAs__C3_A1giles_en_el_desarrollo_de_sistemas_de_software5?lang=en. [Accedido: 25-oct-2014].
- [78]Manuel Trigas Gallego, «Gestión de proyectos informáticos». [En línea]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>. [Accedido: 25-oct-2014].
- [79]Desarrollo ágil, «Metodologías de desarrollo ágil.» [En línea]. Disponible en: http://www.dosideas.com/wiki/Desarrollo_Agil_De_Software. [Accedido: 25-oct-2014].
- [80]Universidad Unión Bolivariana, «XP - Extreme Programing Ingenieria de Software». [En línea]. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html. [Accedido: 25-oct-2014].
- [81]Javeriana, «Programación-Extrema». [En línea]. Disponible en: <http://programacion-extrema.wikispaces.com/5.+Ciclo+de+vida+y+fases>. [Accedido: 25-oct-2014].
- [82]EcuRed, «EXtreme Programming». [En línea]. Disponible en: http://www.ecured.cu/index.php/EXtreme_Programming. [Accedido: 25-oct-2014].

- [83]Patricio Letelier, «Metodologías ágiles para el desarrollo de software.» [En línea].
Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. [Accedido: 25-oct-2014].
- [84]EcuRed, «Metodología Scrum». .
- [85]Julio César Rueda Chacon, «Aplicación de la metodología de aplicaciones basado en el estándar J2EE», 2006.
- [86]Consejo Profesional de Ciencias Informáticas de la Provincia de Buenos Aires, «RAD2PY: Plataforma Python para el Desarrollo Rápido de Aplicaciones bajo un Proceso de Software Personal». [En línea]. Disponible en: http://www.cpciba.org.ar/modulos/articulos/popup_articulos.php?articulo_id=1183&bd=cpciba&PHPSESSID=dd57fe5b0ff2ecb8a59d3953a7d6c771.
- [87]Islavisual, «Diferencias entre scrum y xp». [En línea]. Disponible en: http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php. [Accedido: 25-oct-2014].
- [88]Luis Miranda, «Metodologías de Desarrollo de Software: Ventajas y Desventajas de RAD». [En línea]. Disponible en: <http://desarrollodefww.blogspot.com/2012/10/ventajas-y-desventajas-de-rad.html>. [Accedido: 25-oct-2014].
- [89]Kineo Ingeniería informática y electrónica, «“Técnicas de Gestión del Tráfico”.» [En línea]. Disponible en: https://www.interempresas.net/FeriaVirtual/Catalogos_y_documentos/199309/Reconocimiento_Automatico_Numero_Placa_TGT.pdf.
- [90]JasVsio, «Aplicaciones de la Visión Artificial». [En línea]. Disponible en: <http://www.jasvisio.com/vision-artificial-aplicada-industria-alimentacion.html>. [Accedido: 21-ene-2014].
- [91]Guarnes Miguel, «Inspección automática de impresiones sobre envases cilíndricos empleando visión artificial.» [En línea]. Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/22986/Documento_completo.pdf?sequence=1. [Accedido: 24-oct-2014].
- [92]John Forero, «Medición automatizada de piezas torneadas usando visión artificial.» [En línea]. Disponible en: <http://oaji.net/articles/597-1403541310.pdf>. [Accedido: 24-oct-2014].
- [93]Manuel Suárez Martín, «Sistema de guiado de robots por visión artificial para el marcado de motores.» [En línea]. Disponible en:

- http://riunet.upv.es/bitstream/handle/10251/40177/Tesina_Master_Manuel.pdf?sequence=1. [Accedido: 24-oct-2014].
- [94] Lía García Pérez, «Navegación autónoma de robots en agricultura: un modelo de agentes.» [En línea]. Disponible en: <http://eprints.ucm.es/5277/1/T27198.pdf>. [Accedido: 24-oct-2014].
- [95] Heng Du TszHand To, «Hand gesture recognition usin Kinect». [En línea]. Disponible en: <http://iss.bu.edu/data/jkonrad/reports/HDTT11-04buece.pdf>. [Accedido: 15-oct-2014].
- [96] Shashank Sonkar, «Hand gesture recognition using Microsoft Kinect». [En línea]. Disponible en: <http://www.cse.iitk.ac.in/users/kakshay/docs/Kinect.pdf>. [Accedido: 15-oct-2014].
- [97] Vaughn M. Segers, «Real-Time gesture recognition using Eigenvectors».
- [98] Cristina Manresa, «Real-Time hand tracking and gesture recognition for human-computer interaction», 2000.
- [99] Andrei Dragolici, «Detección, localización e identificación biométrica de caras en imagenes: métodos y evaluación en el marco NIST-MBGG».
- [100] UnitTest++, «UnitTest++». [En línea]. Disponible en: <http://unittest-cpp.sourceforge.net/>. [Accedido: 15-oct-2014].
- [101] Gcov, «Gcov - Using the GNU Compiler Collection (GCC)». [En línea]. Disponible en: <https://gcc.gnu.org/onlinedocs/gcc/Gcov.html>. [Accedido: 15-oct-2014].
- [102] Lcov, «Linux Test Project - Coverage » Lcov». [En línea]. Disponible en: <http://ltp.sourceforge.net/coverage/lcov/lcovrc.5.php>. [Accedido: 20-oct-2014].
- [103] Lcov, «Linux Test Project - Coverage » Lcov». [En línea]. Disponible en: <http://ltp.sourceforge.net/coverage/lcov.php>. [Accedido: 20-oct-2014].
- [104] OpenCV, «Installation in Linux». [En línea]. Disponible en: http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html. [Accedido: 15-oct-2014].
- [105] Advanced Software Production Line, «Estándares de codificación». [En línea]. Disponible en: <http://www.aspl.es/fact/files/aspl-fact/estandares-node2.html>. [Accedido: 15-oct-2014].
- [106] Universidad Jesuita de Guadalajara, «Estándar para la Codificación en lenguaje C++». [En línea]. Disponible en: <http://progra.iteso.mx/estandares/estandar%20codificacion%20c++estandardcodificacion.pdf>.

- [107] gnu.org, «General Public License». [En línea]. Disponible en:
<http://www.gnu.org/licenses/gpl.txt>. [Accedido: 15-oct-2014].
- [108] Creative Commons, «Creative Commons». [En línea]. Disponible en:
<https://creativecommons.org/licenses/?lang=es>. [Accedido: 15-oct-2014].

k. Anexos

Anexo 1. Solicitud de permiso para el desarrollo del sistema en la Asociación de Sordos de Loja "Virgen del Cisne".

Loja, 10 de noviembre del 2013

Lic. Cristian Giler
Sub-Coordinator de la Asociación de Sordos "Virgen del Cisne".

De mis consideraciones:

Por medio de la presente, yo, **Luis Germán Sivisapa Aguilera**, portador de la cédula de identidad N° **1104746936**, egresado de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, recorro ante usted para solicitarle el permiso correspondiente, en el centro al cual usted dirige, mi objetivo a mediano plazo es desarrollar para este, el proyecto de Fin de Carrera denominado "**Visión Artificial para el Reconocimiento del Lenguaje de señas**".

Por la favorable atención que se digne dar a la presente reciba mis más sinceros agradecimientos.

Atentamente:

.....
Luis Germán Sivisapa Aguilera
C.I. 1104746936

10-11-03

Anexo 2. Certificación de Traducción.

DRA. MG. SC. ROCIO ESPERANZA PEÑARANDA REQUELME,

CERTIFICA:

Haber realizado la traducción del idioma Español al Inglés, del Resumen de la Tesis **“VISIÓN ARTIFICIAL APLICADA PARA EL RECONOCIMIENTO DEL LENGUAJE DE SEÑAS”**; cuya autoría es del estudiante **LUIS GERMAN SIVISAPA AGUILERA**.- Lo certifico en honor a la verdad. Loja, 27 de Octubre de 2014.-



Dra. Mg. Sc. Rocío Esperanza Peñaranda Requelme
C.I. Nro. 1102050299

Anexo 3. Guía de Observación directa.

Luego de la asistencia a una de las clases en las que se instruye el lenguaje de señas se pudo apreciar cómo se realizaba este proceso, a continuación se presenta la TABLA XXXIII, en la cual se describe el proceso de enseñanza del abecedario de señas.

TABLA XXXIII. GUÍA DE OBSERVACIÓN DIRECTA.

Proyecto: "Visión Artificial aplicada para el reconocimiento del Lenguaje de Señas"		
Observador: Luis Germán Sivisapa Aguilera		
Lugar: Asociación de Sordos de Loja "Virgen del Cisne"		
Objetivos: Determinar cómo se realiza el proceso de enseñanza del lenguaje de señas.		
TEMA	DESCRIPCIÓN	REQUISITOS
Enseñanza del lenguaje de señas.	El docente encargado indica la posición de la mano para realizar la seña y explica el significado de la misma.	RF002
	EL estudiante repite la posición de la seña y el significado. El docente realiza la verificación de la ejecución de la seña de forma individual por cada alumno.	RF004
	Cuando se ha realizado la explicación de todo el abecedario de señas se procede al incremento de la complejidad realizando la formación de palabras.	RF007

Anexo 4: Ley Orgánica de Discapacidades en el Ecuador.

A continuación se describen algunos artículos del segundo capítulo de la ley Orgánica de Discapacitados en el Ecuador que hace referencia a la inclusión de las personas con discapacidad en la educación, así como los medios y las actividades que se pueden realizar para que se cumpla dicha disposición [9].

Capítulo Segundo

DE LOS DERECHOS A LAS PERSONAS CON DISCAPACIDAD

Sección Tercera

DE LA EDUCACIÓN

Art. 27.- Derecho a la educación.- El estado procurará que las personas con discapacidad puedan acceder, permanecer y culminar, dentro del Sistema Nacional de Educación y del Sistema de Educación Superior, sus estudios, para obtener educación, formación y/o capacitación, asistiendo a clases en un establecimiento educativo especializado o en un establecimiento de educación escolarizada, según el caso.

Art. 28.- Educación inclusiva.- La autoridad educativa nacional implementará las medidas pertinentes, para promover la inclusión de estudiantes con necesidades educativas especiales que requieran apoyos técnico-tecnológicos y humanos, tales como personal especializado, temporales o permanentes y/o adaptaciones curriculares y de accesibilidad física, comunicacional y espacios de aprendizaje, en un establecimiento de educación escolarizada.

Art. 30.- Educación especial y específica.- El Consejo Nacional de Igualdad de Discapacidades coordinará con las respectivas autoridades competentes en materia de educación, el diseño, la elaboración y la ejecución de los programas de educación, formación y desarrollo progresivo del recurso humano necesario para brindar la atención integral a las personas con discapacidad, procurando la igualdad de oportunidades para su integración social.

Art. 32.- Enseñanza de mecanismos, medios, formas e instrumentos de comunicación.- La autoridad educativa nacional velará y supervisará que en los establecimientos educativos públicos y privados, se implemente la enseñanza de los diversos mecanismos, medios, formas e instrumentos de comunicación para las personas con discapacidad, según su necesidad.

Art. 33.- Accesibilidad a la educación.- La autoridad educativa nacional en el marco de su competencia, vigilará y supervisará, en coordinación con los gobiernos autónomos descentralizados, que las instituciones educativas escolarizadas y no escolarizadas, especial y de educación superior, públicas y privadas, cuenten con infraestructura, diseño universal, adaptaciones físicas, ayudas técnicas y tecnológicas para las personas con discapacidad; adaptación curricular; participación permanente de guías intérpretes, según la necesidad y otras medidas de apoyo personalizadas y efectivas que fomenten el desarrollo académico y social de las personas con discapacidad.

Art. 34.- Equipos multidisciplinarios especializados.- La autoridad educativa nacional garantizará en todos sus niveles la implementación de equipos multidisciplinarios especializados en materia de discapacidades, quienes deberán realizar la evaluación, seguimiento y asesoría para la efectiva inclusión, permanencia y promoción de las personas con discapacidad dentro del sistema educativo nacional.

Art. 35.- Educación co-participativa.- La autoridad educativa nacional y los centros educativos inclusivos, especiales y regulares, deberán involucrar como parte de la comunidad educativa a la familia y/o a las personas que tengan bajo su responsabilidad y/o cuidado a personas con discapacidad, en la participación de los procesos educativos y formativos, desarrollados en el área de discapacidades.

Art. 36.- Inclusión étnica y cultural.- La autoridad educativa nacional velará que las personas con discapacidad tengan la oportunidad de desarrollar los procesos educativos y formativos dentro de sus comunidades de origen, fomentando su inclusión étnico-cultural y comunitaria de forma integral.

Art. 38.- Becas.- Aquellas personas con discapacidad en cuya localidad no exista un establecimiento educativo público con servicios adecuados para atender a sus necesidades educativas especiales podrán recibir del Instituto Ecuatoriano de Crédito Educativo, becas y créditos educativos, a fin de que asistan a una institución educativa particular o fiscomisional que sí ofrezca los servicios adecuados, de conformidad con la normativa específica que se expida para el efecto.

Art. 40.- Difusión en el ámbito de educación superior.- La Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación, asegurará que en todas las instituciones de educación superior se transversalice el conocimiento del tema de la discapacidad dentro de las mallas curriculares de las diversas carreras y programas académicos, dirigidos a la inclusión de las personas con discapacidad y a la formación humana de las y los futuros profesionales.

Anexo 5. Encuesta Situación Actual de la Asociación de Sordos de Loja “Virgen del Cisne”.

La presente encuesta se realizó al Lic. Cristian Giler, quien es el encargado de la Asociación de Sordos de Loja “Virgen del Cisne”, dicha encuesta nos ayudó en la recolección de los problemas que conlleva el aprendizaje del lenguaje de señas así como la detección de algunas ventajas que tendría la construcción de un software que ayude en el proceso de enseñanza, a continuación se describe cada una de las respuestas a las preguntas planteadas.



UNIVERSIDAD NACIONAL DE LOJA ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

La presente encuesta tiene el propósito de recabar datos importantes para iniciar con el Proyecto de Fin de Carrera (PFC), motivo por el cual solicitamos a usted de forma muy comedida y respetuosa se digne contestar las siguientes preguntas:

ENCUESTA

1. ¿Qué métodos y técnicas se emplean actualmente en la enseñanza del Lenguaje de Señas?

Enumérelas:

Método Incremental: empezamos con cosas básicas para luego ir incrementando en complejidad.

Técnica:

- Expresión del cuerpo de acuerdo a las características de los objetos.
- Memorística.

2. ¿Qué problemas se presentan en el aprendizaje de los estudiantes al momento de aplicar un método o técnica de enseñanza?

Enumérelas:

- Problema en lectura y en la comprensión, son inestables e inseguros.
- Coordinación en la formación de los signos.
- Dificultad al memorizar palabras extensas.
- Confusión para reconocer signos.

3. ¿Cuál es el proceso realizado para medir los resultados de aprendizaje de los estudiantes?

De acuerdo a la combinación de señas que realizan.

4. ¿Cuenta actualmente con un sistema que ayude en la enseñanza del lenguaje de Señas?

Si (x) No ()

Por qué:

Diccionario de lengua de señas el cual fue realizado en colaboración con otras instituciones.

5. ¿De qué manera cree usted que con la ayuda de un Sistema Informático mejoraría los procesos de enseñanza del lenguaje de señas?

Mediante un intérprete computarizado se ayudaría a mejorar el aprendizaje básico del lenguaje de señas.

6. ¿Al desarrollar un Sistema Informático para mejorar los procesos de enseñanza del Lenguaje de Señas, qué características considera que el mismo debe poseer?

Enumérelas:

- Fácil de usar
- Que sea básico para el aprendizaje en los niños.
- Modelo computarizado de un intérprete.

Anexo 6. Encuesta para conocer el lenguaje de Señas que emplea en la enseñanza la Asociación de Sordos de Loja “Virgen del Cisne”.

Con el objetivo de conocer el lenguaje de signos que emplea la Asociación de Sordos en la enseñanza del Lenguaje de señas, se realizó una encuesta en la que además se planteó algunas preguntas con el propósito de saber si poseen equipos con las características suficientes donde se ponga en marcha el sistema a desarrollar.



UNIVERSIDAD NACIONAL DE LOJA ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

El objetivo de esta encuesta, es recolectar información que nos servirá durante el desarrollo del sistema para la enseñanza básica del lenguaje de señas, es por ello que le rogamos se digne contestar las siguientes preguntas:

ENCUESTA

¿Cuál es el Alfabeto que utiliza la institución en la enseñanza del lenguaje de señas?

() Lenguaje de Signos Americanos

() Lenguaje de Signos Español

(X) Otros

Lenguaje de Señas Ecuatoriano.

¿En la enseñanza del alfabeto de Señas se utilizan las dos manos?

Si (X) No ()

¿Por qué?

Existen más de 5.000 palabras y sería difícil sacar esa cantidad con una sola mano, sin embargo para la enseñanza del abecedario, los números y cosas básicas se lo realiza con una sola mano.

¿Cuenta la institución con equipos tecnológicos donde se pueda hacer uso de un sistema que haga uso de la Visión Artificial?

Si, actualmente la institución cuenta con:

- Computadores,
- Televisores y
- Proyector.

¿Cuáles son las características que poseen los equipos?

Características	Especificación
Procesador	Core i3 3.3 GHz
Memoria RAM	Ram 2 Gb
Disco Duro	500Gb
Cámara web	Webcam Genius eyes 110
Unidad CD-ROM	CD-ROM
Sistema Operativo	Windows 7

Anexo 7. Lenguaje de Señas Ecuatoriano.

Dentro del aprendizaje básico del lenguaje de señas, en la Asociación de Sordos de Loja “Virgen del Cisne”, está integrado el aprendizaje de los números así como el abecedario del Lenguaje de Señas Ecuatoriano, la Figura 36 y la Figura 37, son páginas del libro guía utilizado en la enseñanza, en él se describe cómo realizar cada uno de los números así como las letras del abecedario, con las que se podrá ayudar a formar palabras o frases complejas que permitirá la comunicación con personas con discapacidad auditiva.



ABECEDARIO

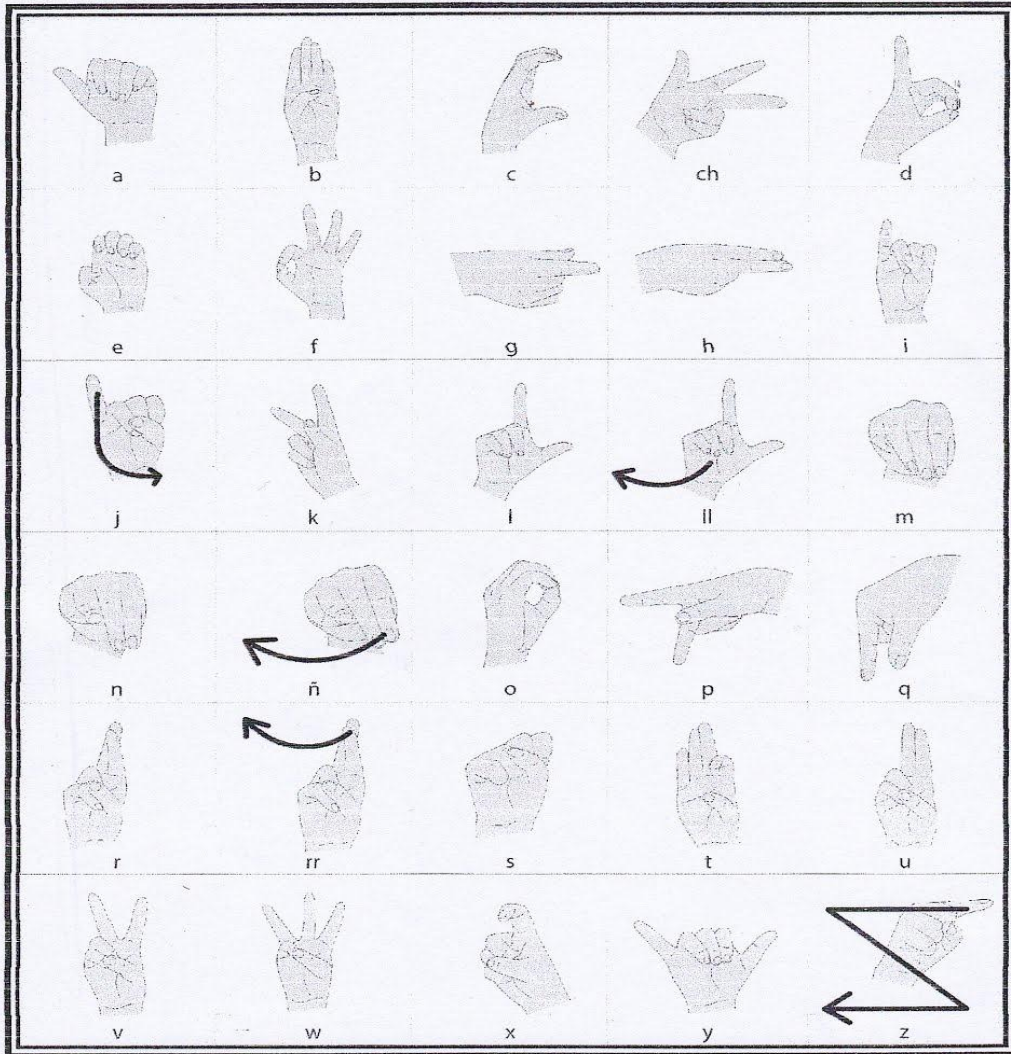


Figura 36. Abecedario del Lenguaje de Señas Ecuatoriana.

Tema: NÚMEROS

número 	uno 	dos
tres 	cuatro 	cinco
seis 	siete 	ocho
nueve 	diez 	once



Figura 37. Números del Lenguaje de Señas Ecuatoriana.

Anexo 8. Prototipos de pantallas iniciales.

El prototipo inicial de pantallas fue presentado a los usuarios con el objetivo de recibir algunas recomendaciones para realizar cambios en las mismas, los resultados finales de estos prototipos se pueden evidenciar en las capturas de pantallas finales (observar en Flujos de ventanas principales del sistema).

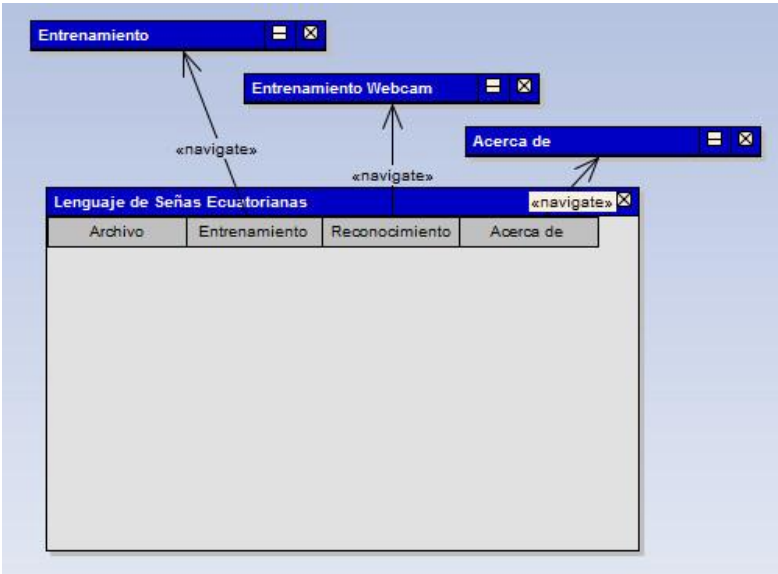
Pantalla	Funcionalidad
	<p>Esta pantalla recibe al usuario y a través de la misma se podrá dar continuidad a todo el proceso de entrenamiento con imágenes y entrenamiento en tiempo real.</p>

Figura 38. Prototipo pantalla Principal.

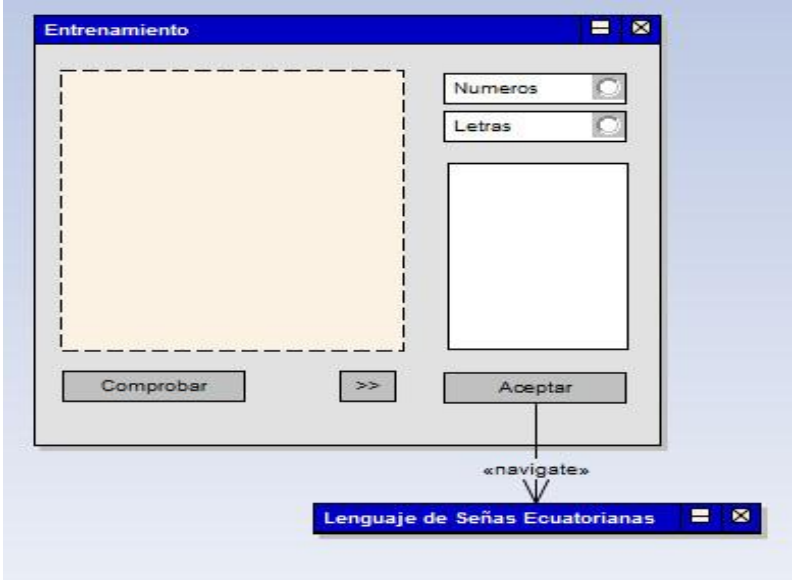
Pantalla	Funcionalidad
	<p>Esta pantalla permite realizar un entrenamiento a través de imágenes estáticas de los números y las letras.</p>

Figura 39. Prototipo pantalla Entrenamiento con Imágenes.

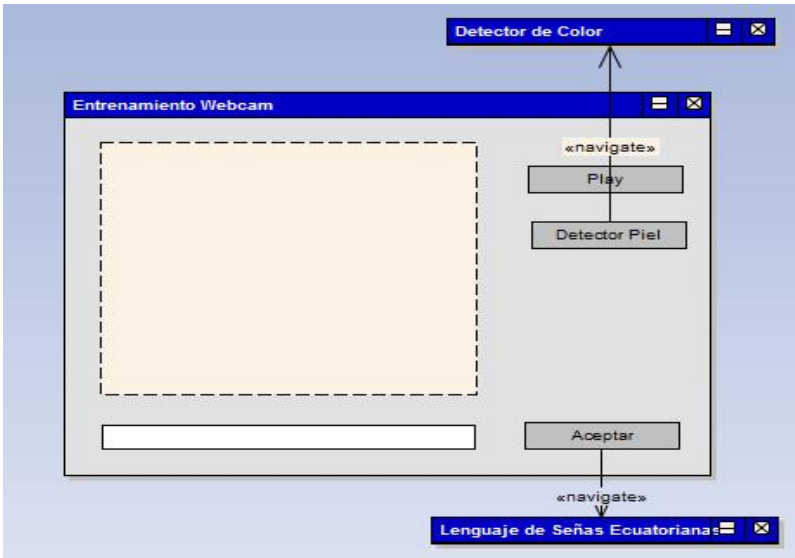
Pantalla	Funcionalidad
	<p>El objetivo de esta pantalla es el permitir un entrenamiento del lenguaje de signos y los números en tiempo real.</p>

Figura 40. Prototipo pantalla Entrenamiento con Webcam.

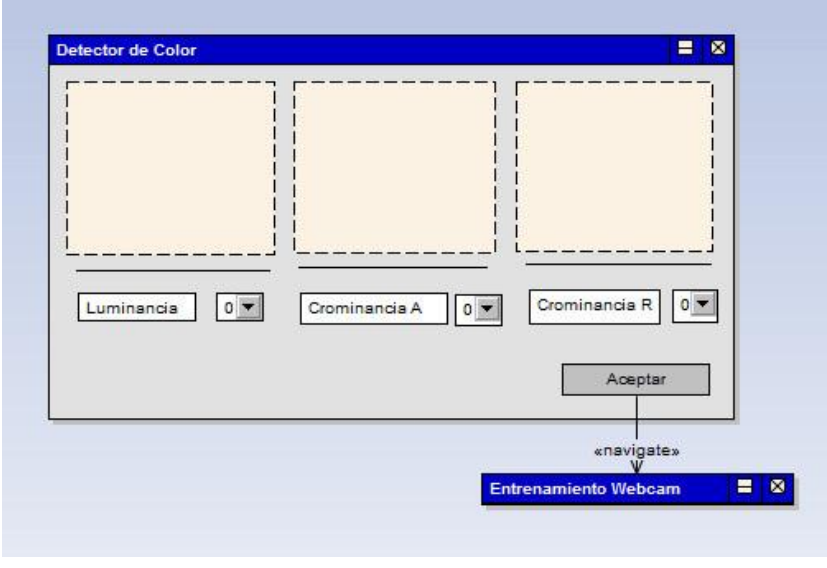
Pantalla	Funcionalidad
	<p>EL objetivo de esta pantalla es recolectar información relevante en el proceso de umbralización de la imagen.</p>

Figura 41. Prototipo pantalla Detección de Piel.


Pantalla	Funcionalidad
	<p>En esta pantalla se muestra información principal acerca del software, como su licencia y objetivo.</p>

Figura 42. Prototipo pantalla Acerca de...

Anexo 9. Instalación y Configuración de OpenCV.

Para el empleo de OpenCV durante el proceso de desarrollo se requiere de la instalación, para ello se puede seguir algunos pasos sencillos que se describen desde su página oficial [104], o bien se puede seguir los pasos que se detallan a continuación:

1. Actualización de Repositorios

```
sudo apt-get update
```

2. Instalación de Dependencias

La Instalación de paquetes adicionales es muy importante ya que permiten la ejecución de OpenCV, algunas de las dependencias se listan a continuación [104]:

- GCC 4.4.x
- build-essential
- CMake 2.6
- pkg-config
- Python 2.6, Numpy 1.5
- ffmpeg códec

Para la instalación se emplea el siguiente comando:

```
sudo apt-get install build-essential libgtk2.0-dev libjpeg-dev libtiff4-dev libjasper-dev libopenexr-dev cmake python-dev python-numpy python-tk libtbb-dev libeigen2-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev libqt4-dev libqt4-opengl-dev sphinx-common texlive-latex-extra libv4l-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev
```

3. Descarga e instalación

Se descarga el paquete de OpenCV desde su página oficial, dependiendo del tipo de plataforma que utilizaremos, se descomprime el paquete descargado, una vez dentro de dicho paquete se crea una carpeta (build), en esta carpeta es donde se crearan todos los binarios que permiten la ejecución de OpenCV, dentro de build se deberá preparar los archivos, compilar y finalmente instalar OpenCV a través de los siguientes comandos

```
tar -xvf OpenCV-2.4.x.tar.bz2
cd OpenCV-2.4.x
mkdir build
cd build

cmake -D WITH_TBB=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D
WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON -D
INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D
WITH_OPENGL=ON ..
make
make install
```

4. Configuración de OpenCV en el Sistema Operativo

Para la configuración de OpenCV en el S.O. se debe modificar el archivo de configuración de OpenCV:

```
sudo gedit /etc/ld.so.conf.d/opencv.conf
```

Añadiendo la dirección donde se encuentran almacenadas las librerías que se terminaron de instalar como se indican en la Figura 43:

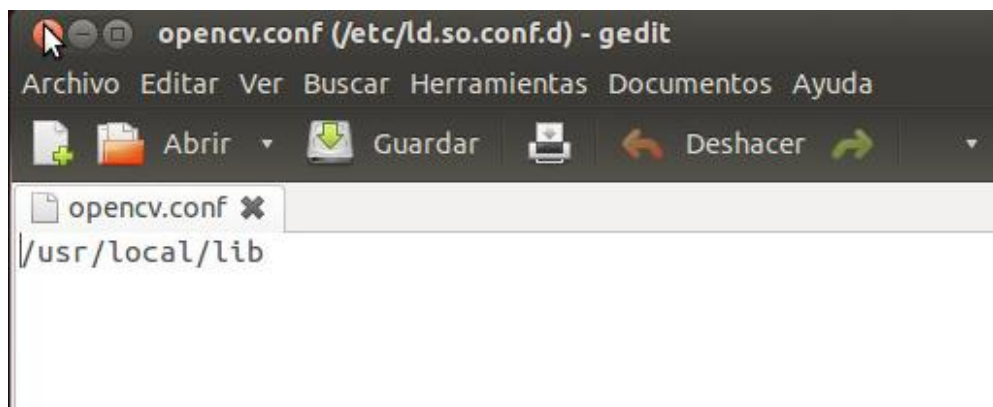


Figura 43. Configuración Opencv.conf

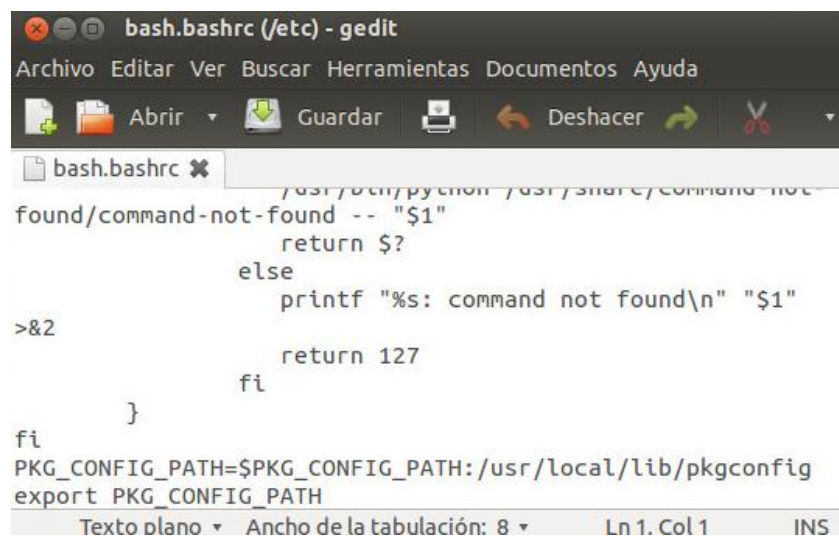
Una vez que se haya modificado el archivo este deberá ser guardado y actualizadas las librerías del sistema a través del comando:

```
sudo ldconfig
```

Para la que el sistema operativo conozca de la instalación de OpenCV se debe modificar el PATH añadiendo todas las librerías que se acaban de instalar Figura 44, a través de:

```
sudo gedit /etc/bash.bashrc
```

```
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig  
export PKG_CONFIG_PATH
```



```
bash.bashrc (/etc) - gedit  
Archivo Editar Ver Buscar Herramientas Documentos Ayuda  
Abrir Guardar Deshacer  
bash.bashrc x  
found/command-not-found -- "$1"  
    return $?  
else  
    printf "%s: command not found\n" "$1"  
>&2  
    return 127  
fi  
}  
fi  
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig  
export PKG_CONFIG_PATH  
Texto plano Ancho de la tabulación: 8 Ln 1. Col 1 INS
```

Figura 44. Agregar OpenCV al path.

Concluido con este procedimiento se reinicia el ordenador y de esta forma estará instalado OpenCV en nuestro sistema.

5. Configuración de Opencv en Qt-Creator

En Qt-Creator la inclusión de las librerías de OpenCV se las realiza a través del archivo de configuración (.pro) que se genera al momento de crear un proyecto, un ejemplo de como de se debe incluir dichas librerías es el siguiente:

```
INCLUDEPATH += /usr/local/include/opencv2  
INCLUDEPATH += /usr/local/include/opencv  
LIBS += -L/usr/local/lib  
LIBS += -lopencv_core
```

```
LIBS += -lopencv_imgproc
LIBS += -lopencv_highgui
LIBS += -lopencv_ml
LIBS += -lopencv_video
LIBS += -lopencv_features2d
LIBS += -lopencv_calib3d
LIBS += -lopencv_objdetect
LIBS += -lopencv_contrib
LIBS += -lopencv_legacy
LIBS += -lopencv_flann
LIBS += -lopencv_nonfree
```

Anexo 10. Estándar para Codificación en lenguaje C++.

La forma de escribir código es propia de cada programador y completamente diferente a la forma de cualquier otro [105]. El empleo de un estándar para la codificación facilita agregar nuevas características o modificar las existentes, ya que se maneja un lenguaje común con personas ajenas al desarrollo, para este proyecto se reutiliza un estándar para la codificación realizado por la Universidad Jesuita de Guadalajara [106], en la cual se definen las reglas para la declaración de variables, punteros, métodos, etc.

A continuación se presenta el documento con dichas convenciones:

Estándar para Codificación en lenguaje C++

INTRODUCCIÓN

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

A continuación se presentan algunas de estas convenciones para la programación en lenguaje C++.

- Comentarios
- Cada programa deberá comenzar con un comentario que incluya:
 - Autor
 - Fecha
 - Objetivo, o problema que resuelve el programa
 - Fecha de creación y bitácora de versiones con las dos últimas fechas de modificación.
 - Algoritmo.

Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.
- Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- Explicación de uso de argumentos (parámetros) no obvios.
- Explicación de uso de valores devueltos (de retorno).

Nombres de identificadores.

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que éstos tengan una longitud razonable. Si usa abreviaturas deben manejar la misma lógica en todo el programa.
- Evitar identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.
- Cada identificador de función, variable o procedimiento deberá ser precedido por la abreviación del tipo de dato de que es la variable, o si se trata de una función o procedimiento del tipo de dato que regresa.

Identificadores de variables

- Comenzarán siempre con la primera letra minúscula correspondiente a su tipo de dato.
- Para distinguir palabras dentro del nombre deberá emplearse una letra mayúscula o un guión bajo (_), sin mezclar ambas formas en un mismo programa.

Ejemplos:

temperaturaDeVapor

temperatura_de_vapor

- Identificadores de punteros (apuntadores). Su nombre deberá comenzar con la letra p.

Ejemplo:

pAlumno

- Dónde pAlumno es un puntero que podrá tener la dirección del lugar donde se almacena información de un alumno.
- Identificadores de variables dimensionadas (arreglos, matrices). Su nombre deberá comenzar con las letras ar.

Ejemplo:

arAlumnos

Donde arAlumnos es un arreglo de datos para guardar información de alumnos.

- Identificadores de datos constantes. Serán declaradas en letras mayúsculas.

Ejemplo:

const IVA = 0.15;

- Identificadores de funciones. La primera letra deberá ser mayúscula.

Ejemplo:

void vFuncion();

- Identificadores de tipos definidos por el usuario. La primera letra será mayúscula.

Ejemplo:

class Clase o struct Estructura

Organización Visual del Programa

Generales

- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- Sangrías
- Las sangrías tendrán una longitud de tres espacios.
- Para las llaves que definen el cuerpo de una función, sangre un nivel.

Ejemplo:

```
void Funcion ( )  
{  
  //Instrucciones de la función  
}
```

- Sangre las instrucciones del cuerpo de cada estructura de control.

Ejemplo:

```
for (int x = 0; x < 5; x++)  
{  
  //Instrucciones a ejecutar  
}
```

- Trate de evitar codificar más de tres niveles de sangrado.
- Insertar una línea en blanco antes y después de una declaración de datos que aparezca entre instrucciones ejecutables.

```
a = b + c;  
// línea en blanco  
int f; //declaración entre instrucciones  
// línea en blanco  
f = a;
```

- Las declaraciones de datos dentro de una función, deberán ir al inicio y separadas de las instrucciones ejecutables de la función por medio de una línea en blanco.
- Deben incluirse espacios en ambos lados de los operadores binarios.

Ejemplo:

```
y = 50 + 15 - x;
```

- Es posible distribuir una instrucción grande sobre varias líneas. Si lo hace, seleccione puntos de ruptura que tengan sentido, como después de una coma en el caso de una lista, o después de un operador en el caso de una expresión larga. Sangre todas las líneas subsecuentes.

Ejemplo:

```
cout << "Ejemplo de ruptura de una instrucción en más de una"
<< " línea de comandos";
```

- Los operadores unarios (++ , -- , etc.) deben ponerse junto a sus operandos, sin espacios intermedios.
- Antes y después de cada estructura de control deberá poner una línea en blanco.
- Paréntesis
- Para hacer más clara una expresión, es aceptable agregarle paréntesis innecesarios. Dichos paréntesis se llaman paréntesis redundantes.

Anexo 11. Plan de Capacitación e Instalación.

Plan para la Implantación y capacitación

1. Título

Plan para la instalación y capacitación del Sistema de Visión Artificial para el Reconocimiento del Lenguaje de señas en la Asociación de sordos de Loja “Virgen del Cisne”.

2. Objetivo General

Capacitar a las personas involucradas en la enseñanza así como en el aprendizaje de la Asociación de Sordos de Loja “Virgen del Cisne”, en la instalación y el manejo de todos los procesos y funcionalidades del sistema de Visión Artificial para el Reconocimiento del Lenguaje de Señas.

3. Objetivos específicos

- Ayudar en la solución de problemas que se puedan presentar en la instalación del sistema.
- Proporcionar orientación durante la ejecución del sistema.

4. Estrategias

Las estrategias que se emplean para la capacitación de los docentes y estudiantes es:

- **Metodología de exposición y diálogo.**

Con la ayuda de material digital (diapositivas), se pretende indicar como se realiza la manipulación del sistema, así como su instalación.

5. Logística

Para cumplir con el objetivo propuesto se realizarán las coordinaciones necesarias para que dicha capacitación sea en horas laborables de acuerdo a la asociación, es importante que cada integrante de la asociación traiga su equipo de cómputo en caso de disponer, de la misma manera se harán los trámites necesarios para ofrecer:

- Un salón de clases dotado de ayudas audiovisuales (proyector).
- Exposiciones por parte del tutor, acerca de la instalación así como de la ejecución del sistema.

6. Ejecución de Actividades

TABLA XXXIV. CRONOGRAMA DEL PLAN DE INSTALACIÓN Y CAPACITACIÓN.

Actividad	Participantes	Fecha
Instalación del sistema.	Docentes, Estudiantes	11-10-2014
Capacitación del sistema. Entrenamiento con imágenes. <ul style="list-style-type: none"> ○ Conocimiento de los signos. ○ Realizar signo. Entrenamiento en tiempo real. <ul style="list-style-type: none"> ○ Realizar signo. ○ Formación de palabras. 	Docentes, Estudiantes	11-10-2014

7. Verificación de la ejecución. (FEEDBACK)

Luego de haber realizado las actividades planteadas, se hará uso de:

- Una encuesta a los estudiantes y al docente para verificar la aceptación y de la misma manera conocer si ha existido un mejoramiento en el aprendizaje a través del empleo del sistema para el reconocimiento del Lenguaje de Señas.

Anexo 12. Certificación de Validación y pruebas realizadas.

Certificado entregado por el Lic. Cristian Giler subcoordinador de la asociación de sordos de Loja "Virgen del Cisne", luego de la correspondiente capacitación del sistema para el aprendizaje del lenguaje de señas.

Loja, 11 de Octubre del 2014

Lic.
Cristian Giler
SUB-COORDINADOR DE LA ASOCIACIÓN DE SORDOS DE LOJA "VIRGEN DEL CISNE"

CERTIFICA:

Que las pruebas y capacitaciones correspondientes al programa "**Visión Artificial aplicada para el reconocimiento del Lenguaje de Señas**", relacionado con la enseñanza del lenguaje de señas en el nivel básico de aprendizaje, se realizó en la asociación de sordos de Loja "Virgen del Cisne", con la supervisión del señor tesista Luis Germán Sivisapa Aguilera, a los docentes así como a los estudiantes.

Es todo en cuanto puedo decir en honor a la verdad. Los interesados pueden hacer uso del mismo.

Atentamente;


Lic. Cristian Giler
Sub-coordinador de la Asociación de Sordos de Loja
"Virgen del Cisne".

Anexo 13. Encuesta funcionalidad del software.

La presente encuesta se la realizó al docente encargado de la enseñanza del Lenguaje de Señas, con el propósito de conocer cómo ayuda el sistema en la enseñanza luego de su correspondiente implantación.



UNIVERSIDAD NACIONAL DE LOJA ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

La presente encuesta tiene el propósito de recabar datos importantes sobre el nivel de aceptación del sistema para el aprendizaje del Lenguaje de Señas, motivo por el cual solicitamos a usted de forma muy respetuosa se digne contestar las siguientes preguntas:

ENCUESTA ACEPTACIÓN DEL SISTEMA

¿Cómo se ha evidenciado la utilización del sistema en los estudiantes?

- Mayor Nivel de conversación. []
- Rapidez en la realización de señas. []
- Mejor comprensión. [X]

Otro:.....
.....

¿El empleo del sistema ha facilitado el entrenamiento del lenguaje de señas?

Si (X) No ()

Por qué: *Ayuda a los niños a un mejor Aprendizaje*
.....
.....

¿De qué manera ha ayudado la implantación del sistema, en la comunicación entre maestros y estudiantes, y demás integrantes de la asociación?

En los jóvenes y niñas que aún no saben la LSEC el sistema es bueno ya que están empezando con el ABC,

¿El sistema implantado ha contribuido a mejorar el nivel de aprendizaje de los estudiantes?


Si (X) No ()

Por qué: están empezando con el ABC,

La ayuda del sistema permite tratar con mayor rapidez las temáticas en la enseñanza del nivel básico.

Si (X) No ()

Por qué: en la mayoría de palabras básicas consta el movimiento de la mano incluida la letra del ABC,

Firma: 

Le agradecemos por el tiempo invertido en el desarrollo de la encuesta.

Anexo 14. Licencias Aplicadas al sistema.

El proyecto consta de dos tipos de licencia uno para toda la documentación y otra para el código fuente del sistema.

Licencia General Public License

Este tipo de licencia es el más utilizado en el mundo del software ya que garantiza a los usuarios finales la libertad de usar, estudiar, compartir (copiar) y modificar el software.

El código del proyecto de Visión Artificial aplicada para el reconocimiento del Lenguaje de Señas, se encuentra bajo la licencia GNU GPL versión 3.

Esta licencia incluye algunos términos y condiciones para la reproducción, distribución y modificación del software [107]:

- Definiciones.
- Código fuente.
- Permisos básicos.
- Protección de los derechos legales de los usuarios frente a la Ley Anti-Evasión.
- Transmisión de copias exactas.
- Transmisión de versiones modificadas del código fuente.
- Transmisión de códigos que no son códigos fuente.
- Términos adicionales.
- Cancelación.
- Aceptación innecesaria para la posesión de copias.
- Traspaso automático de licencia a destinatarios subsiguientes.
- Patentes.
- Protección de la libertad de terceros.
- Uso conjunto con la Licencia Pública General Affero de GNU.
- Revisiones de esta Licencia.
- Ausencia de garantías.
- Limitación de la responsabilidad.
- Interpretación de las secciones 15 y 16.

Una copia de esta licencia se puede apreciar desde el siguiente enlace:

<http://www.gnu.org/licenses/gpl.txt>

Licencia Creative Commons

Las licencias y herramientas de derechos de autor Creative Commons, genera un equilibrio dentro del escenario tradicional de "todos los derechos reservados" que crean las leyes de propiedad intelectual.

Cada licencia ayuda a los creadores a mantener sus derechos de autor al mismo tiempo que permiten a otros copiar, distribuir, y hacer algunos usos de su obra, al menos de forma no comercial, funcionan alrededor del mundo y duran tanto tiempo como sea aplicable el derecho de autor [108].

El proyecto de Visión Artificial aplicada para el reconocimiento del Lenguaje de Señas, se encuentra bajo esta licencia:

Atribución-NoComercial-CompartirIgual CC BY-NC-SA

Esta licencia permite a otros distribuir, mezclar, retocar, y crear a partir de la obra de modo no comercial, siempre y cuando le den crédito y licencien sus nuevas creaciones bajo las mismas condiciones.



Trabajo de Titulación by Luis Svisapa is licensed under a
[Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License](https://creativecommons.org/licenses/by-nc-sa/4.0/).