



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA EN MECATRÓNICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO MECATRÓNICO**

**TEMA: “INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN
ALGORITMO DE CONTROL VISUAL BASADO EN POSICIÓN PARA
COMANDAR EL MOVIMIENTO DE UN BRAZO ROBÓTICO QUE
PERMITA LA ALIMENTACIÓN INDEPENDIENTE DE PERSONAS
AFECTADAS CON OSTEOARTRITIS”**

**AUTORES: AROCA REMACHE, KATHERINE ESTEFHANIA
JIMÉNEZ BONILLA, ARTURO FABIÁN**

DIRECTOR: ING. MENDOZA CHIPANTASI, DARÍO JOSÉ

LATACUNGA

2018



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

i

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, ***“INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL VISUAL BASADO EN POSICIÓN PARA COMANDAR EL MOVIMIENTO DE UN BRAZO ROBÓTICO QUE PERMITA LA ALIMENTACIÓN INDEPENDIENTE DE PERSONAS AFECTADAS CON OSTEOARTRITIS”*** fue realizado por los señores ***Aroca Remache, Katherine Estefhania y Jiménez Bonilla, Arturo Fabián***, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permite acreditar y autorizar para que lo sustente públicamente.

Latacunga, 02 de Agosto del 2018.

Ing. Darío José Mendoza Chipantasi

DIRECTOR


DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, ***Aroca Remache, Katherine Estefhania y Jiménez Bonilla, Arturo Fabián,*** declaramos que el contenido, ideas y criterios del trabajo de titulación: ***“INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL VISUAL BASADO EN POSICIÓN PARA COMANDAR EL MOVIMIENTO DE UN BRAZO ROBÓTICO QUE PERMITA LA ALIMENTACIÓN INDEPENDIENTE DE PERSONAS AFECTADAS CON OSTEOARTRITIS”*** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas. Consecuentemente el contenido de la investigación es veraz.

Latacunga, 02 de Agosto del 2018.



Katherine Estefhania Aroca Remache
C.C.: 1726752577



Arturo Fabián Jiménez Bonilla
C.C.: 0503797631



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

iii

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORIZACIÓN

Nosotros, **AROCA REMACHE, KATHERINE ESTEFHANIA Y JIMÉNEZ BONILLA, ARTURO FABIÁN**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL VISUAL BASADO EN POSICIÓN PARA COMANDAR EL MOVIMIENTO DE UN BRAZO ROBÓTICO QUE PERMITA LA ALIMENTACIÓN INDEPENDIENTE DE PERSONAS AFECTADAS CON OSTEOARTRITIS”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 02 de Agosto del 2018.

Katherine Estefhania Aroca Remache

C.C.: 1726752577

Arturo Fabián Jiménez Bonilla

C.C.: 0503797631

DEDICATORIA

Este trabajo de investigación lo dedico a Dios por darme la fuerza necesaria para resolver mis adversidades.

A mis padres y abuelitos por ser un gran ejemplo de amor, superación y esfuerzo. Cada uno de sus consejos me ha permitido obtener cada triunfo y ahora uno de los principales el ser una profesional.

A todos mis familiares y grandes amigos, sus palabras de motivación, alegría y apoyo han contribuido positivamente en mi vida.

Katherine

DEDICATORIA

Este trabajo de investigación quiero dedicar principalmente a Dios y a la Santísima Virgen María por guiarme en cada paso que doy y estar junto a mí en los momentos más difíciles.

A mis padres por siempre darme su apoyo y estar a mi lado cuando más los he necesitado ya que gracias a sus esfuerzos y amor he logrado cumplir mi sueño.

A mis hermanos por ser quienes, siempre me dieron palabras de aliento en los momentos que más lo necesitaba.

Arturo

AGRADECIMIENTOS

Agradezco a mis padres por darme el amor y apoyo incondicional para culminar mi carrera, a todos mis profesores quienes aportaron con sus enseñanzas y consejos, su conocimiento ha aportado favorablemente en mi formación profesional, a todos mis familiares, y amigos por todo su cariño, paciencia y colaboración.

Gracias a todos ustedes lo pude lograr.

Katherine

Quiero agradecer a mis padres y hermanos por ser quienes guiaron mi camino y gracias a ellos he logrado salir a delante y vencer las dificultades que la vida me ha puesto.

Agradezco a Dios por ser quien me guía en cada paso que doy y siempre me acompañado en los momentos más difíciles.

Arturo

ÍNDICE DE CONTENIDO

CARÁTULA

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN	iii
DEDICATORIA.....	iv
DEDICATORIA.....	v
AGRADECIMIENTOS	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE TABLAS	viii
RESUMEN.....	ix
ABSTRACT.....	x

CAPÍTULO I

FUNDAMENTO TEÓRICO

1.1. Antecedentes	1
1.2. Planteamiento del problema	3
1.3. Justificación e importancia.....	3
1.4. Objetivos.....	4
1.4.1. Objetivo General.....	4
1.4.2. Objetivos Específicos	4
1.5. Generalidades	6
1.5.1. La mano y la muñeca	6
1.5.2. Artrosis	6
1.5.3. Artrosis de la mano.....	6
1.6. Brazo robótico	7
1.7. Tipos y clasificación de los robots	9
1.7.1. Robot cartesiano	9
1.7.2. Robot cilíndrico	10

1.7.3. Robot esférico o Robot polar	10
1.7.4. Robot SCARA.....	11
1.8. Cinemática del Robot	11
1.8.1. Cinemática directa del brazo de un robot manipulador	11
1.9. Tecnología asistencial	13
1.10. Visión artificial.....	13
1.10.1. Conceptos Generales	13
1.10.2. Métodos de captación de las imágenes.....	14
1.11. Control visual.....	15
1.11.1. Visión Monocular	17
1.11.2. Eyeinhand System.....	18
1.11.3. Visión Estéreo	18
1.11.4. Cámaras mirando hacia el objeto	19
1.11.5. Sistema con Cámaras Redundantes	20
1.11.6. Control Visual Cinemático	20
1.11.7. Control Visual Dinámico	21
1.12. Detección facial usando Haar Cascades	22
1.13. OpenCV	24
1.13.1. Detección de Haar-cascada en OpenCV	24
1.14. Sistemas de reconocimiento de voz	25
1.14.1. Reconocimiento de voz	25
1.14.2. Arquitectura de un sistema de reconocimiento de voz	26
1.14.3. Extracción de características	26
1.14.4. Clasificador probabilístico	27
1.14.5. Tipos de sistemas de reconocimiento de voz que existen	27
1.14.6. Estilo del habla	28
1.14.7. Tamaño del vocabulario	29
1.14.8. Condiciones de trabajo	29
1.14.9. Aplicaciones	29
1.15. Sistema de pesaje	31
1.15.1. Pesaje estático	31

CAPÍTULO II

DISEÑO, SELECCIÓN E IMPLEMENTACIÓN DE LOS COMPONENTES DEL SISTEMA DE ALIMENTACIÓN

2.1. Criterios técnicos	32
2.2. Parámetros de diseño.....	33
2.3. Diseño y selección de componentes	34
2.3.1. Descripción de los componentes.....	34
2.4. Diseño de la estructura mecánica	50
2.4.1. Diseño de la estructura mecánica de soporte.....	51

CAPÍTULO III

DESARROLLO DEL ALGORITMO DEL SISTEMA DE ALIMENTACIÓN

3.1. Algoritmo de detección y seguimiento de la boca en Python.....	78
3.1.1. Bibliotecas	78
3.1.2. Clasificador en cascada.....	79
3.1.3. Algoritmo de detección del rostro	79
3.1.3. Algoritmo de seguimiento del rostro	80
3.1.4. Seguimiento de la boca	85
3.2. Implementación de los sensores y su programación	88
3.2.1. Acondicionamiento de los sensores	88
3.2.3. Funciones de mediciones de peso	90
3.3. Programación para el control del brazo robótico	91
3.3.1. Funciones manual y automática	93
3.3.2. Aplicación “SISTEMA DE ALIMENTACIÓN ARKAZ 2.0”.....	95
3.3.3. Funciones cuchara, devC	101
3.3.4. Funciones platoUno, platoDos, post y vaso.....	101
3.3.5. Función ik_arm	102
3.3.6. Función para la comunicación	106
3.3.7. Calibración del sistema para el modo automático	108

CAPÍTULO IV

IMPLEMENTACIÓN PRUEBAS Y RESULTADOS

4.1. Construcción e implementación de la estructura mecánica de soporte	112
4.1.1. Área de trabajo o zona de alimentación	113
4.1.2. Mecanismo de regulación.....	114
4.1.3. Base de accionamiento	115
4.2. Análisis, pruebas y resultados del sistema de alimentación	116
4.2.1. Análisis para el tiempo de alimentación del usuario	116
4.2.3. Porciones adecuadas para cada usuario.....	117
4.2.4. Pruebas de accionamiento y control por voz	118
4.2.5. Pruebas iniciales en el sistema de pesaje	119
4.2.6. Pruebas de levantamiento de cantidades con el brazo robótico.....	119
4.2.7. Pruebas del modo manual y modo automático.....	121
4.2.8. Comprobación de la hipótesis	129

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones	137
-------------------------	-----

5.2. Recomendaciones 140

REFERENCIAS BIBLIOGRÁFICAS 143

ANEXOS 162

Anexo A: Normas INEN

Anexo B: Manual de usuario del “SISTEMA DE ALIMENTACIÓN ARKAZ 2.0”

Anexo C: Instalación de paquetes

Anexo D: Video demostrativo

ÍNDICE DE TABLAS

Tabla 1	Escala de puntuación	32
Tabla 2	Características Físicas	35
Tabla 3	Especificaciones técnicas del brazo robótico	36
Tabla 4	Comparativa de las opciones para elegir el mejor Brazo Robótico	37
Tabla 5	Utensilios a ser usados.....	37
Tabla 6	Características de los pulsadores.....	38
Tabla 7	Características de la Raspberry Pi	40
Tabla 8	Características de la Beaglebone	41
Tabla 9	Comparativa de las opciones para elegir el computador	41
Tabla 10	HDMI LCD	42
Tabla 11	Pantalla de 7 pulgadas	42
Tabla 12	Comparativa de las opciones para elegir la pantalla	43
Tabla 13	Características del Arduino Uno	44
Tabla 14	Características técnicas de Arduino Mega	45
Tabla 15	Comparativa de las opciones para elegir la placa Arduino	45
Tabla 16	Celda de carga en viga.....	46
Tabla 17	Flexi Force y FSR.....	47
Tabla 18	Comparativa de las opciones para elegir el sensor de peso	48
Tabla 19	Especificaciones técnicas cámara FACECAM 1000X.....	49
Tabla 20	Especificaciones técnicas cámara FACECAM 315	49
Tabla 21	Comparativa de las opciones para elegir la cámara.....	50
Tabla 22	Elementos del área máxima de alimentación	50

Tabla 23	Valores de masa a levantar	56
Tabla 24	Datos para el dimensionamiento de la superficie de la altura de la mesa	59
Tabla 25	Número de dientes en el piñón para asegurar que no exista interferencia....	61
Tabla 26	Piñones cilíndricos con ángulo de presión de 20°	62
Tabla 27	Fórmulas de características de dientes de los engranes.....	64
Tabla 28	Valores de masa a levantar	65
Tabla 29	Factores de aplicación sugeridos, K_a	67
Tabla 30	Factores de tamaño que se sugieren, K_s	67
Tabla 31	Esfuerzo por flexión permisible aproximado	72
Tabla 32	Factor de confiabilidad, K_R	73
Tabla 33	Calibración de la coordenada z	111
Tabla 34	Elementos de la estructura mecánica de soporte	113
Tabla 35	Elementos de la estructura mecánica de soporte	114
Tabla 36	Elementos de la estructura mecánica de soporte	115
Tabla 37	Elementos de la estructura mecánica de soporte	116
Tabla 38	Peso de la cuchara vacía y con los diferentes alimentos	120
Tabla 39	Peso del vaso vacío y con diferentes pesos	120
Tabla 40	Prueba del plato de sopa del usuario uno	122
Tabla 41	Prueba del plato de sopa del usuario dos.....	122
Tabla 42	Prueba del plato de sopa del usuario tres	123
Tabla 43	Prueba del plato de sopa del usuario cuatro	123
Tabla 44	Prueba del plato de sopa del usuario cinco	124
Tabla 45	Porcentajes y tiempo de consumo de cada usuario	124

Tabla 46 Prueba del plato de sopa del usuario uno	125
Tabla 47 Prueba del plato de sopa del usuario dos.....	126
Tabla 48 Prueba del plato de sopa del usuario dos.....	126
Tabla 49 Prueba del plato de sopa del usuario cuatro	127
Tabla 50 Prueba del plato de sopa del usuario cinco	127
Tabla 51 Porcentajes y tiempo de consumo de cada usuario	128
Tabla 52 Porcentaje de consumo de alimentos sólidos.....	130
Tabla 53 Porcentaje de consumo de alimentos líquidos	132
Tabla 54 Porcentaje de consumo de alimentos sólidos.....	133
Tabla 55 Porcentaje de consumo de alimentos líquidos	135
Tabla 56 Tabla de t.....	136

ÍNDICE DE FIGURAS

Figura 1. Articulaciones afectadas con osteoartritis	7
Figura 2. Partes del brazo robótico basados en el brazo humano.....	8
Figura 3. Robot cartesiano	9
Figura 4. Robot cilíndrico.....	10
Figura 5. Robot esférico o Robot polar	10
Figura 6. Robot SCARA	11
Figura 7. La silla "GPS-ROMA"	13
Figura 8. Visión monocular	17
Figura 9. Visión estéreo.....	19
Figura 10. Visión estéreo con las cámaras colocadas en el efector	19
Figura 11. Control Visual Cinemático	21
Figura 12. Control Visual Dinámico	21
Figura 13. Imagen	22
Figura 14. Características del rostro.....	23
Figura 15. Detección de características de rostro en OpenCV.....	25
Figura 16. Arquitectura de un sistema de reconocimiento de voz	26
Figura 17. Partes del Brazo Robótico.....	35
Figura 18. Partes del Brazo robótico de 6 DOF	36
Figura 19. Pulsadores grandes interactivos.....	38
Figura 20. Raspberry Pi 3.....	39

Figura 21. Beaglebone	40
Figura 22. HDMI LCD de 7 pulgadas (B)	42
Figura 23. Pantalla de 7 pulgadas	43
Figura 24. Arduino Uno	44
Figura 25. Arduino Mega	45
Figura 26. Celda de carga de 5 kg	46
Figura 27. Flexi Force y FSR	47
Figura 28. FACECAM 1000X	48
Figura 29. FACECAM 315	49
Figura 30. Área máxima de alimentación	51
Figura 31. Terminología de dimensionamiento de una mesa	52
Figura 32. Espacio libre para el alojamiento de las piernas	53
Figura 33. Dimensiones del ancho y profundidad de la superficie de la mesa	53
Figura 34. Triángulo rectángulo con medidas reales	54
Figura 35. Dimensionamiento de ubicación de los utensilios de comida	55
Figura 36. Análisis de esfuerzo Von Mises	57
Figura 37. Factor de seguridad de la superficie	57
Figura 38. Altura de la superficie de la mesa	58
Figura 39. Puesto de trabajo sentado	59
Figura 40. Intervalo de altura de la mesa	60
Figura 42. Factor de proporción del piñón, C_{pf}	68
Figura 43. Factor de alineamiento del engranado, C_{ma}	69
Figura 44. Factor de espesor de la corona, K_B	70

Figura 45. Factor de dinámica, C_v y K_v	71
Figura 46. Factor de geometría J	71
Figura 47. Factor de resistencia flexionante por ciclos de esfuerzo Y_N	73
Figura 48. Ejes de ubicación	76
Figura 49. Diagrama de flujo del algoritmo	77
Figura 51. Región del ojo es más oscuro que las mejillas.....	79
Figura 52. Detección del rostro con Haar Cascade	80
Figura 53. Biblioteca Dlib.....	81
Figura 54. Inicialización de las variables	82
Figura 55. Transformación de variables	82
Figura 56. Inicio del seguidor.....	83
Figura 57. Actualización del seguidor.....	83
Figura 58. La calidad de seguidor (trackingQuality)	84
Figura 59. 68 puntos de coordenadas faciales.....	86
Figura 60. Dirección del algoritmo de seguimiento con los 68 puntos faciales.....	87
Figura 61. Índices para los puntos de referencia faciales de la boca	87
Figura 62. Extracción de las regiones de la boca	87
Figura 63. Detección de la boca con los 68 puntos faciales.....	88
Figura 64. Cálculo de las dimensiones de la región de la boca.....	88
Figura 65. Circuito de las celdas de carga.....	89
Figura 66. Galgas extenso métricas	89
Figura 67. Lectura de sensores	90
Figura 68. Promedio de voltaje leído de la Celda de Carga	90

Figura 69. Convertidor ADC	91
Figura 70. Biblioteca para el control del brazo robótico	91
Figura 71. Declaración de los servos	92
Figura 72. Comando para mover los servos.....	92
Figura 73. Función Home	93
Figura 74. Programación manual para la sopa.....	94
Figura 75. Áreas del plato.....	95
Figura 76. Bloque de verificación de activación del bluetooth del dispositivo móvil	97
Figura 77. Bloque de presentación de los dispositivos bluetooth disponibles	98
Figura 78. Bloque de selección del módulo bluetooth a utilizar	98
Figura 79. Bloque de activación del botón manual	99
Figura 80. Bloque de activación automático	99
Figura 81. Bloque de reseteo	99
Figura 82. Bloque que muestra el envío de comandos por voz.....	100
Figura 83. Bloque para cerrar la aplicación	101
Figura 84. Función cuchara	101
Figura 85. Activación de la función vaso	102
Figura 86. Componentes geométricos de la cinemática inversa	103
Figura 87. Angulo del brazo y antebrazo	104
Figura 88. Dimensiones de la imagen	106
Figura 89. Área del centro de la imagen.....	107
Figura 90. Dimensión de la horizontal del rostro	108
Figura 91. Diagrama del ángulo de visión, la distancia focal y el tamaño del sensor ..	108

Figura 92. Gráfica del ángulo en función de la distancia focal.....	109
Figura 93. Partes de la estructura mecánica de soporte	112
Figura 94. Área de alimentación.....	114
Figura 95. Mecanismo de regulación de altura.....	115
Figura 96. Base de accionamiento	116
Figura 97. Aplicación “SISTEMA DE ALIMENTACIÓN ARKAZ 2.0”	118
Figura 98. Pruebas de pesaje con platos de porcelana y aluminio.....	119
Figura 99. Pruebas de modo manual a los usuarios	121
Figura 100. Pruebas en el modo automático a los usuarios	125

RESUMEN

Este trabajo de titulación se enfoca a la investigación de la metodología más adecuada para el control de un brazo robótico de cinco grados de libertad, que permite la alimentación independiente de personas con osteoartritis, pues tienen dificultad al mover sus manos por el dolor y deformación causado en sus cartílagos. El conjunto está conformado por un sistema mecatrónico que consta de una parte mecánica; formada por una mesa de soporte con altura ajustable con espacio disponible para un brazo robótico asistencial; tres platos de comida; correspondientes a sopa, arroz, postre y un vaso de bebida con sorbete; la parte electrónica, con sensores de peso, pulsadores y cámara web, y su correspondiente parte de control, en donde se utiliza las tarjetas Raspberry, Arduino y una pantalla. El sistema posee comodidad de alimentación para el usuario, pues su accionamiento cuenta con dos modos, uno manual, comandado por pulsadores mediante movimientos programados, y uno automático activado por voz, con configuración "Eyeinhand" la misma que se encarga de generar la trayectoria en función de la cinemática del brazo. Los dos modos tanto el automático como el manual poseen también un control de pesaje para cada uno de los platos de comida.

PALABRAS CLAVE:

- **OSTEOARTRITIS**
- **SISTEMAS DE CONTROL**
- **BRAZO ROBÓTICO**
- **ROBÓTICA**

ABSTRACT

This work of titration focuses on the investigation of the most appropriate methodology for the control of a robotic arm of five degrees of freedom, which allows the independent feeding of people with osteoarthritis, since they have difficulty moving their hands because of the pain and deformation caused in their cartilages. The set consists of a mechatronic system consisting of a mechanical part; formed by a support table with adjustable height with space available for a robotic assistance arm; three plates of food; corresponding to soup, rice, dessert and a glass of drink with sorbet; the electronic part, with weight sensors, push buttons and webcam, and its corresponding control part, where the Raspberry, Arduino and a screen are used. The system has power supply for the user, since its operation has two modes, one manual, controlled by pushbuttons by means of programmed movements, and one automatic activated by voice, with "Eyeinhand" configuration, which is in charge of generating the trajectory in function of arm kinematics. The two modes, both automatic and manual, also have a weighing control for each of the food plates.

KEYWORDS:

- **OSTEOARTRITIS**
- **CONTROL SYSTEMS**
- **ROBOTIC ARM**
- **ROBOTICS**

CAPÍTULO I

1.FUNDAMENTO TEÓRICO

1.1. Antecedentes

El presidente de la Sociedad de Geriátría del Ecuador, afirma que la artrosis afecta al 60 por ciento de personas mayores de 65 años, según cifras mundiales, esto debido a la degeneración de las articulaciones (Wilmot, 2017). Entre las consecuencias que se presentan con más frecuencia están las lesiones musculares, que se deben a caídas, aumento de peso y sedentarismo, lo que afecta sustentablemente la calidad de vida de los adultos mayores (Pérez Rodríguez, Domínguez Sosa, & González Baños, 2014).

Uno de los casos más comunes de artrosis es la osteoartritis o denominada artrosis de las manos, enfermedad que afecta directamente a las articulaciones de la muñeca, la mano y los dedos, causando rigidez, inflamación, dolor, deformidad, y debilidad; lo cual produce que la persona no pueda sujetar y tomar los objetos para realizar con normalidad las actividades diarias como: abrir un frasco, dar la vuelta a una llave o coger una cuchara.

Esta enfermedad afecta principalmente al género femenino y está relacionado con actividades manuales realizadas durante periodos prolongados, la que suele comenzar durante la edad de 40-50 años, y periodos más tardíos (Artrosis, 2014).

Debido a los efectos causados por esta enfermedad, especialmente la degeneración y dolor en los cartílagos de sus articulaciones, es necesario la intervención de un asistente que le ayude con sus actividades diarias, como lo es el poder alimentarse.

Dentro de las discapacidades físicas de una persona, la capacidad que se dificulta en mayor proporción es la independencia a la hora de su alimentación, es por ello que se han desarrollado proyectos de tecnología asistida como BrainGate, el mismo que se describe como un sistema de implante cerebral, para ayudar a aquellos que perdieron el control de sus extremidades, u otras funciones corporales, como pacientes con Esclerosis lateral amiotrófica o lesión de la médula espinal, que por medio de un sensor implantado dentro del cerebro, supervisa la actividad cerebral del paciente y convierte la intención del usuario en comandos para la computadora. Además, existen otras variantes como iCRAFT el mismo que fue desarrollado por investigadores de la Universidad de Northeastern en Boston (EE.UU.), que consiste en tecnología de alimentación robótica controlada por los ojos y otros proyectos que para su funcionamiento usan pulsadores.

En la actualidad la robótica ha ampliado su campo tanto en la parte industrial como en la salud en la cual se encuentran instrumentos de laboratorio y de experimentación quirúrgica que contienen manipuladores basados en brazos robóticos, los mismos que son usados también dentro de la robótica de servicio para ayudar a pacientes con discapacidades físicas. El control de posición por medio de visión, es una técnica que utiliza la información de realimentación extraída de un sensor de imágenes (retroalimentación visual) para controlar el movimiento de un robot, en la actualidad esta tecnología es ampliamente utilizada (Mancilla, 2016).

1.2. Planteamiento del problema

La osteoartritis es una enfermedad degenerativa del cartílago de la mano, que se desarrolla con la presencia de nódulos óseos en las articulaciones de los dedos, los cuales producen inflamación y dolores profundos. En Ecuador existe una gran incidencia de esta enfermedad, es por ello que es necesario que otro individuo asista en actividades simples como la alimentación de la persona; en países del primer mundo para evitar la intervención de otro individuo se han creado manipuladores con tecnología sofisticada pero no accesible a personas de escasos recursos debido a su alto costo.

En la página "<https://meetobi.com>" existe un brazo robótico de alimentación con características similares a un precio de \$4 500 exceptuando los valores de impuestos y de envío. En el país no existe un manipulador que permita la alimentación de pacientes, es por ello que se hace factible la realización de esta investigación, y con ello permitir un mejor estilo de vida a las personas que padecen de esta enfermedad.

1.3. Justificación e importancia

En el Hospital General de la ciudad de Latacunga existen 100 pacientes que presentan osteoartritis, esta enfermedad impide a las personas efectuar labores cotidianas, una de ellas su propia alimentación, para lo se hace necesario la ayuda de un asistente.

Dentro de la mayor parte de países del primer mundo existen los denominados brazos con tecnología asistencial, los que permiten realizar labores como la alimentación a un paciente, pero en nuestro país debido a su alto costo no son adquiridos por los hospitales públicos y menos aún por las personas afectadas por osteoartritis.

Con la realización de este proyecto se pretende que las personas con problemas de osteoartritis adquieran un sistema de bajo costo comparado con los ya dispuestos en el mercado, siendo fácil de operar y además brindando independencia que presta su servicio a la hora de alimentarse, pues el brazo permitirá el consumo de los alimentos necesarios, sin sufrir dolencias y molestias en las articulaciones de sus manos.

El proyecto consta de un sistema de visión artificial, para estimar la posición del efector final del brazo robótico con respecto a la boca del usuario, el mismo que adquirirá información de la cámara para realizar el seguimiento del manipulador a la boca y así la persona pueda tomar el alimento.

Cabe recalcar que este sistema se lo puede utilizar si el usuario dispone de otros problemas de discapacidad en sus manos, ya sea amputación de las mismas o de las extremidades superiores, lesiones, fracturas, enfermedades de las manos y otros problemas a fines.

1.4. Objetivos

1.4.1. Objetivo General

- Investigar, diseñar e implementar un algoritmo de Control Visual Basado en Posición para comandar el movimiento de un brazo robótico que permita la alimentación independiente de personas afectadas con osteoartritis.

1.4.2. Objetivos Específicos

- Investigar, recopilar información y requerimientos sobre robots asistenciales que permiten la alimentación de personas con discapacidad, y sistemas de Control Visual Basado en Posición.

- Diseñar y construir la estructura mecánica del sistema, en la que el usuario podrá regular la altura de la misma según su necesidad.
- Investigar y diseñar el algoritmo de Control Visual basado en Posición para comandar las trayectorias del brazo robótico.
- Diseñar e implementar el sistema electrónico semiautomático y automático, para que el modelo ofrezca mayor movilidad y eficiencia.
- Realizar pruebas de funcionamiento y movilidad del prototipo de brazo robótico para la alimentación de personas con osteoartritis.

1.5. Generalidades

1.5.1. La mano y la muñeca

La mano posee una gran movilidad en los dedos, sobre los que actúan complejos sistemas de tendones, por ejemplo, las manos de un pianista. Esto por la disposición de la columna del pulgar, la que permite oponerse a los otros dedos, permitiendo que la mano pueda realizar todo tipo de movimientos sencillos, como agarrar una fruta, o más difíciles como sostener a un individuo (Fisioterapia, 2017).

1.5.2. Artrosis

La artrosis es una enfermedad que daña el cartílago articular. Las articulaciones son parte del esqueleto que ayudan a realizar el movimiento del cuerpo y las cuales se forman por la unión de dos huesos en la cápsula articular, las mismas que se recubren con cartílago. Si el cartílago articular se daña, empieza a producir dolor, incluso falta de movilidad (Torbert, 2016).

1.5.3. Artrosis de la mano

Las personas que padecen esta enfermedad tienen con frecuencia dolor en las articulaciones lo que produce una limitación en sus movimientos. Las mujeres son las más afectadas con esta enfermedad y que suele comenzar a la edad de cuarenta a cincuenta años. Las articulaciones que se afectan con más frecuencia son (figura 1):

- En el círculo celeste se puede observar la articulación trapecio-metacarpiana.

- En el círculo azul se puede observar la articulación de metacarpofalángica.
- En el círculo verde claro se puede ver la articulación de interfalángica proximal.
- En el círculo verde oscuro se observa la articulación de interfalángica distal.

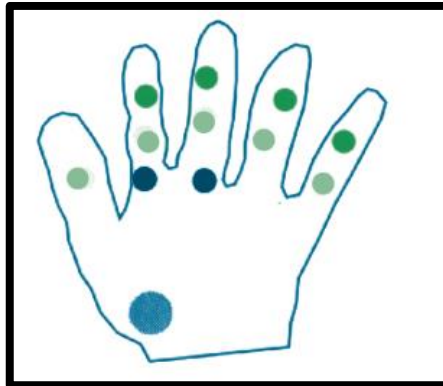


Figura 1. Articulaciones afectadas con osteoartritis
Fuente: (Fisioterapia ayllón, 2012)

Síntomas

El dolor es el primer síntoma que viene, cuando se genera movimiento por alguna actividad, el dolor es pequeño al principio, pero si la actividad que se realiza es por un largo periodo vuelve el dolor. Debido a las alteraciones de los ligamentos, tendones y músculos se ocasiona la deformidad en las articulaciones, y con ello nace la dificultad de mover los dedos (Fisioterapia ayllón, 2012).

1.6. Brazo robótico

Los manipuladores robóticos son conocidos también como brazos de robot debido al parecido con la extremidad superior de las personas. Las partes del brazo robótico o brazo robot son: cuerpo, brazo, muñeca y elemento o actuador final. Se puede destacar en primera instancia que son diseñados para mover materiales, piezas, herramientas o dispositivos especializados (González V. , Robot Industriales, 2005).

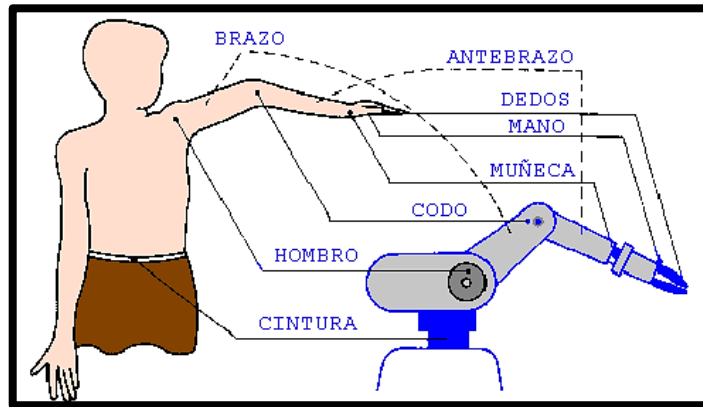


Figura 2. Partes del brazo robótico basados en el brazo humano

Fuente: (González V. , Robot Industriales, 2005)

Un brazo robótico industrial incluye a dispositivos como sensores que son necesarios para que el robot pueda hacer sus operaciones, así como la secuenciación o monitoreo de las zonas de comunicación. Los brazos robots se usan en su mayor parte para actividades inseguras, peligrosas e incluso repetitivas para el operador. Estos robots realizan diversas funciones, como el manejo de materiales, montaje, soldadura, carga y descarga de herramienta, pintura, pulverización, la mayoría de los manipuladores operan con la técnica de enseñanza y repetición (Senmache Torres, 2008). En este modo, el programador utiliza una consola para enseñar al robot la tarea que debe realizar. Las velocidades del robot durante estas sesiones de programación son lentas.

Esta técnica incluye consideraciones de seguridad necesarias para operar el robot correctamente y utilizarlo automáticamente junto con otros equipos periféricos, esto se aplica a los sistemas robóticos industriales fijos únicamente, pero existen sistemas que están excluidos de esta técnica.

Un brazo robótico es un mecanismo mecánico, el cual se programa, con funciones similares a las que realiza un brazo humano (Robotica, 2015). Las partes de los

manipuladores se conectan con ayuda de articulaciones las mismas que generan los movimientos: rotacional como los de un robot articulado, movimiento traslacional o desplazamiento lineal (González V. , Control y Robótica, 2014).

1.7. Tipos y clasificación de los robots

Los brazos robóticos son de diferentes de tamaños, formas y configuraciones; los mismos que están diseñados y fabricados con diferentes configuraciones. Estos diseños dependen del área, volumen de trabajo un ejemplo es el alcance que se necesita en el espacio laboral.

1.7.1. Robot cartesiano

Este manipulador tiene ejes que son iguales a los ejes cartesianos por lo que tienen tres articulaciones prismáticas (Santiago, 2015).

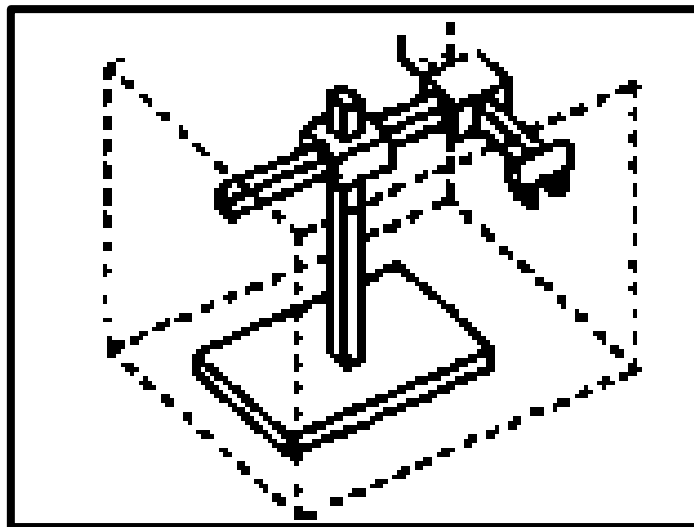


Figura 3. Robot cartesiano

Fuente: (United States Department of Labor, 2015)

1.7.2. Robot cilíndrico

Sus ejes forman un sistema de coordenadas cilíndricas, a este manipulador se lo utiliza en operaciones de ensamblaje y soldadura.

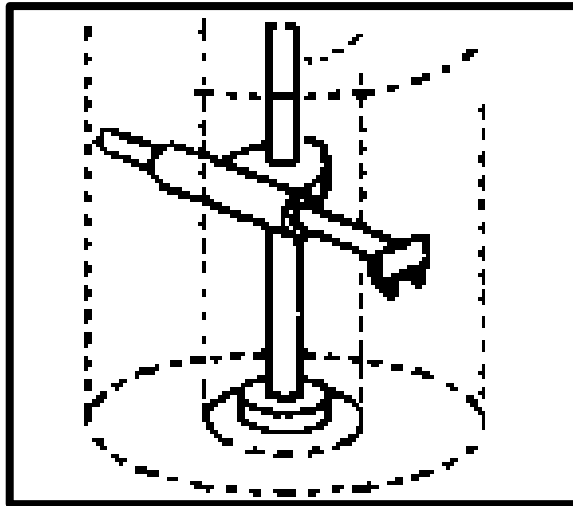


Figura 4. Robot cilíndrico

Fuente: (United States Department of Labor, 2015)

1.7.3. Robot esférico o Robot polar

Los ejes están formados por un sistema de coordenadas polar, y su uso radica en actividades de soldadura y manipulación de herramientas.

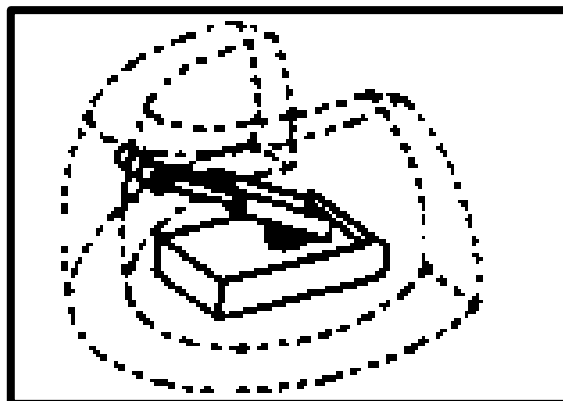


Figura 5. Robot esférico o Robot polar

Fuente: (United States Department of Labor, 2015)

1.7.4. Robot SCARA

Este robot tiene dos articulaciones rotatorias paralelas, con el objetivo de dar mayor movilidad en el plano y es usado para trabajos de tomar y colocar, es por ello que entre sus aplicaciones están las operaciones de ensamblado y manipulación de herramientas (Villa, 2013).

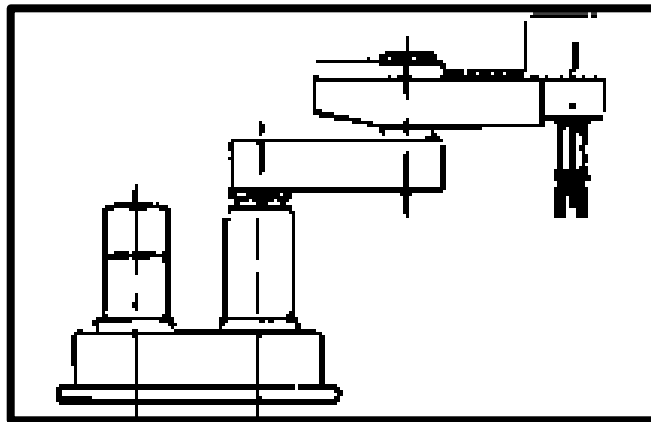


Figura 6. Robot SCARA

Fuente: (United States Department of Labor, 2015)

1.8. Cinemática del Robot

Denominada como el estudio del movimiento espacial del brazo robótico, como también las relaciones que hay entre la posición y la orientación del actuador final (López Apostolovich, 2009).

1.8.1. Cinemática directa del brazo de un robot manipulador

Para encontrar el extremo final del robot se usa una matriz de transformación de coordenadas de referencia, que cambia en coordenadas de movimiento al manipulador, esto es denominado como cinemática directa (Esparza, Carvajal, & García, 2010).

Así se usa matrices de transformación homogénea en donde se realizan operaciones de rotación y traslación (Chávez, 2008):

$$T = \left[\begin{array}{cc} \text{matriz de rotación} & \text{vector posición} \\ f_{1 \times 3} & \text{escalado} \end{array} \right] = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde n, o, a, son vectores ortogonales unitarios y p es un vector que da la posición origen del sistema en relación a las coordenadas actuales. La matriz de transformación homogénea permite:

- Tener un sistema de referencia al cual se rota y traslada al vector.
- Transformar un vector expresado en coordenadas con respecto al sistema móvil, a su equivalente en coordenadas con respecto del sistema fijo (Aranda Aboy, 2009).

1.8.2. Cinemática Inversa

Es la configuración que debe tomar el robot para tener una orientación y posición requerida por el usuario. Contiene coordenadas articulares del robot $q = [q_1, q_2, \dots, q_n]^T$, para que su extremo se posicione y oriente según una localización espacial determinada (Yumpu, 2017). Con la que se puede encontrar una solución cerrada, por la relación matemática:

$$q_k = f_x(x, y, z, \alpha, \beta, \gamma)$$

$$k = 1 \dots n \quad (\text{GDL})$$

1.9. Tecnología asistencial

La tecnología asistencial agrupa a los equipos, dispositivos, programas que se usan para mejorar la calidad de vida de pacientes que poseen algún tipo de discapacidad, pues ayudan con asistencia y rehabilitación basados en tecnología computacional.

Entre sus ventajas se presenta la máxima libertad de movimientos con el mínimo espacio. Facilitando la movilidad de las personas tanto en el trabajo, en la vivienda, como en espacios de ocio o culturales. Un ejemplo radica en la silla "GPS-ROMA" usada para entornos exteriores conocidos, de esta forma mejora la movilidad de personas que tienen discapacidades físicas (Retro Informática, 2015).

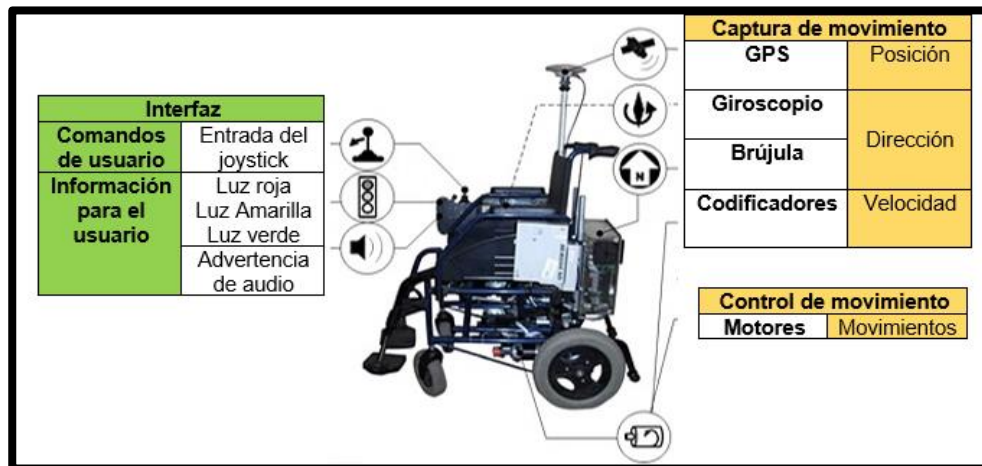


Figura 7. La silla "GPS-ROMA"

Fuente: (Retro Informática, 2015)

1.10. Visión artificial

1.10.1. Conceptos Generales

Se puede definir la "Visión Artificial" como una rama de la "Inteligencia Artificial" en la que se obtiene información por imágenes digitales (Scribd, 2017).

La visión artificial está formada por un conjunto de procesos destinados a realizar el análisis de imágenes digitales iniciando de la obtención de imágenes, memorizar la información, procesar y dar a entender los resultados (Flores Pérez, 2017).

Con la visión artificial se pueden:

- Evitar realizar inspecciones de tareas repetitivas automatizándolas.
- Hacer controles de calidad en procesos que no se podían porque eran muy complejos para los operadores.
- Realizar inspecciones de productos sin tener contacto físico.
- Encontrar las coordenadas exactas de un objeto determinado.
- Realizar mediciones tridimensionales.

1.10.2. Métodos de captación de las imágenes

Entre los métodos de captación de imágenes se presentan:

- **Digital.** La función que se tiene después de diversos muestreos realizados en diferentes intervalos de tiempo. Las mismas que son manipuladas mediante un computador.
- **Píxel.** Una imagen digital es un tipo de cuadrícula; donde cada elemento de esa cuadrícula se llama Píxel (Eitudela, 2010).
- **Nivel de grises.** Cuando se digitaliza una imagen, cada punto cuantificado corresponde a la intensidad del brillo en la escena.
- **Imagen binaria.** Es la imagen presenta solo negro y blanco.

1.11. Control visual

La utilización de la visión por computador para controlar un robot se ha denominado en la literatura como visual servoing, control visual de robots, siendo este el término anglosajón propuesto por Hill y Park (Pari Pinto, 2015). La inclusión de la visión en sistemas robóticos tiene como principal objetivo el aumento de la flexibilidad y la precisión de estos sistemas. Uno de los primeros trabajos donde se utilizó la visión para el control de robots fue presentado por Shirai e Inoue hace más de tres décadas. El papel de la visión por computador es el de ofrecer al robot el estado de su entorno de trabajo, para que esta información pueda ser utilizada para su control. En el caso de robots manipuladores, se encuentran en su entorno de trabajo uno o varios objetos a manipular (Guitar, 2014).

La información del entorno de trabajo del robot es obtenida por la imagen, información visual, que puede ser usada para el control de dos formas diferentes. La primera de ellas se denomina Open-Loop-Robot-Control, control en lazo abierto, y se basa en la separación de la parte relacionada con la visión de la parte del control. La designación de control en lazo abierto surge debido a la no utilización de información visual durante el movimiento, puesto que el lazo de realimentación se cierra una sola vez al principio y queda abierto en el resto del proceso. Se puede afirmar que, según lo dicho, la posición inicial y la deseada del robot.

Se determina off - line a través de la información visual, sabiendo que el movimiento del robot es "a ciegas" puesto que el control se realiza únicamente en base a los sensores de articulación (Escalona, 2014). Las posiciones iniciales y finales se determinan a través

de algoritmos de estimación de posición, siendo necesario calibrar la cámara y obtener el modelo del objeto (Gómez Molina & Ríos Velásquez, 2009).

La segunda forma, la cual se denomina convencionalmente visual servoing, en este caso, se utiliza habitualmente un control interno del robot, así como la realimentación de todas las variables de visión (ADINGOR, 2015). La utilización directa de la información visual implica tener como variables controladas las coordenadas 2D del plano de la imagen; en cuanto a la utilización indirecta de información visual, esta implica la extracción de características relevantes del objeto a partir de las coordenadas 2D del plano de la imagen (González Sánchez).

En los robots manipuladores controlados por visión, generalmente se utilizan medidas visuales obtenidas a través de características de la imagen. La obtención de estas medidas depende de la ubicación del objeto, del número de cámaras utilizadas, de su configuración respecto al robot, y de su calibración. Esta información puede obtenerse directamente (basándose en las características de la imagen del objeto en el plano) o indirectamente de la imagen, basándose en la estimación de la posición tridimensional del objeto respecto a la cámara, y la estimación del movimiento de la cámara (López Puigdollers, Cabeza Laguna, & Villanueva Larre, 2016). La utilización en conjunto de los dos tipos de información visual descritas, permiten la obtención de varios grupos distintos de medidas del objeto.

Según el número de cámaras el sistema puede ser monocular (una cámara), visión estéreo (dos cámaras combinadas entre sí) y sistema de cámaras redundantes (más de dos cámaras redundantes entre sí). Si la configuración de la cámara se encuentra en el elemento terminal del robot y mira hacia el objeto se denomina eye in hand; mientras que

si la cámara se encuentra en el espacio y mira hacia el elemento terminal del robot y al objeto simultáneamente se llama eye to hand.

1.11.1. Visión Monocular

Un sistema de visión monocular utiliza solamente una cámara, que puede ser colocada en el efector final del robot mirando hacia el objeto, o en el espacio mirando simultáneamente hacia el efector final del robot y hacia el objeto (Capek, 2010).

La principal ventaja de este tipo de sistema es que minimiza el tiempo de procesamiento necesario para extraer las informaciones visuales necesarias (Calero, 2010), siendo otra ventaja su coste. No obstante, tiene la desventaja de que no es posible determinar la profundidad, distancia entre la cámara y el objeto, de forma exacta siendo necesaria su estimación utilizando un modelo del objeto, o utilizando otro tipo de métodos durante el control.

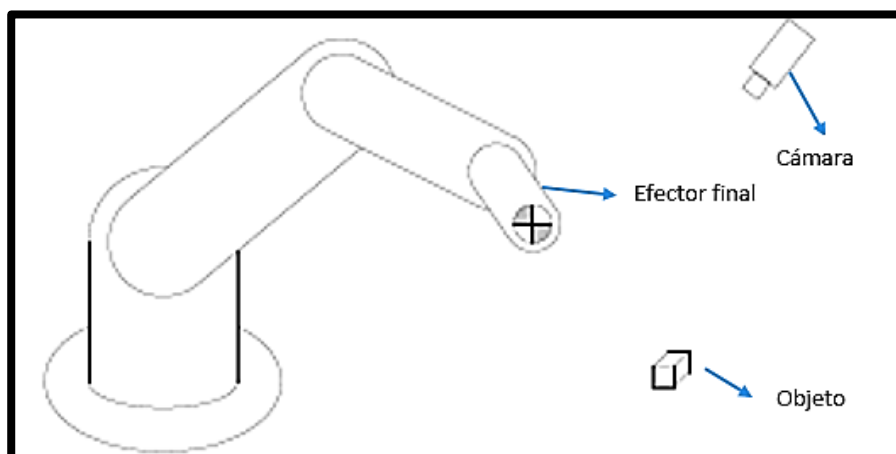


Figura 8. Visión monocular

Fuente: (García Gómez, 2010)

1.11.2. Eyeinhand System

Un sistema de Eyeinhand es un robot equipado con una cámara de autoenfoco. Con esta configuración, las aplicaciones típicas son conservar un determinado objeto dentro de la imagen.

Como ejemplo de posiciones definidas en imágenes diferentes en el trabajo de Remazeilles, basado en una base de datos de imágenes (memoria visual).

Este sistema requiere una calibración para poder controlar variables como la posición del objeto respecto al efector final del robot.

1.11.3. Visión Estéreo

Un sistema de visión estereo está constituido por dos cámaras ligadas rígidamente con la finalidad de obtener dos imágenes del mismo objeto.

Este sistema de visión es bastante útil cuando se pretende obtener información tridimensional, por ejemplo, de un determinado objeto dentro del campo de visión del par estereo.

A continuación, se presentan las configuraciones más habituales de este sistema de visión, que son idénticas a las de la visión monocular, como son las cámaras mirando hacia el objeto y hacia el efector final del robot simultáneamente (Romero Velásquez, Olgún Díaz, & Morales Díaz, 2009).

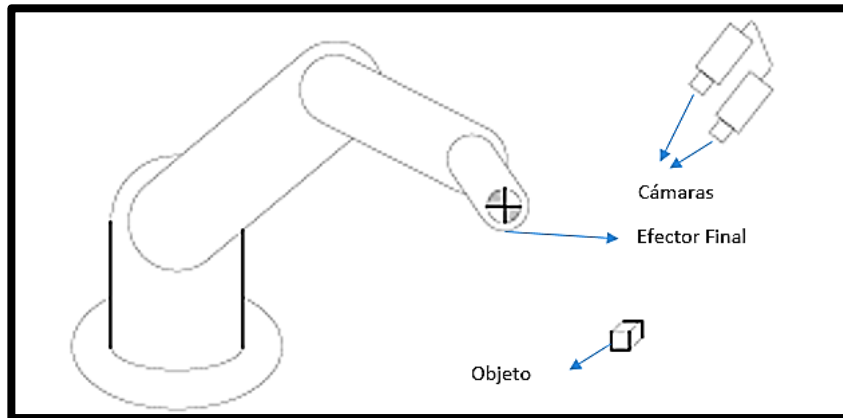


Figura 9. Visión estéreo

Fuente: (García Gómez, 2010)

1.11.4. Cámaras mirando hacia el objeto

La utilización de un par estéreo ligado rígidamente al efector final, Eyeinhand, no es muy habitual, debido al hecho de que este tipo de sistemas están asociados a un mayor volumen (respecto a la visión monocular) lo que dificulta otro tipo de aplicaciones, tal como la manipulación de objetos. Con todo esto, la miniaturización cada vez mayor de los sistemas de visión tiene a solucionar este obstáculo.

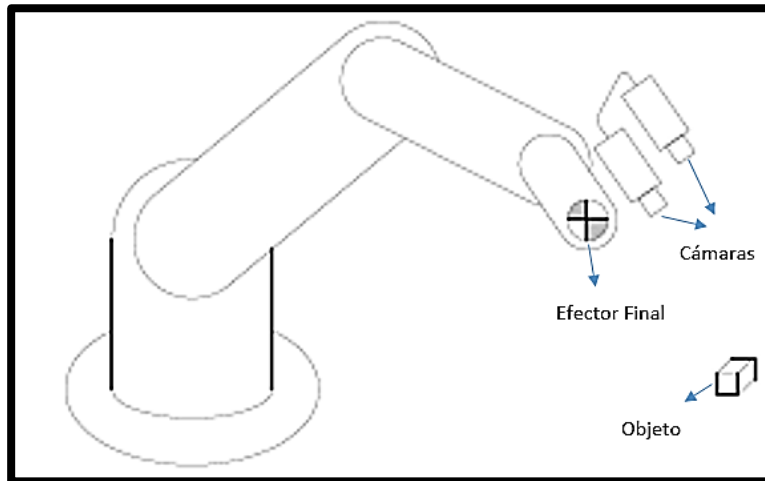


Figura 10. Visión estéreo con las cámaras colocadas en el efector

Fuente: (García Gómez, 2010)

1.11.5. Sistema con Cámaras Redundantes

Cuando se utilizan más de dos cámaras, los sistemas se definen como redundantes. No obstante, el matching entre las distintas imágenes de la escena que se va a visualizar es un procedimiento complejo, pesado computacionalmente y con costes más elevados. Como consecuencia de estos hechos, la utilización de estos sistemas en el control visual de robots es poco frecuente. Se nota que en este tipo de sistemas también se incluye la combinación de la visión estéreo y la visión monocular, como es el caso, por ejemplo, de un robot con una cámara en el efector final y un par estéreo que visualice el robot y el objeto (López Germán, 2014).

1.11.6. Control Visual Cinemático

En este tipo de control la cámara proporciona una señal de referencia ya procesada e información que proporciona la cámara, con ello el controlador proporciona al robot una referencia en velocidad para que este alcance dicha señal. La referencia en velocidad se puede definir en coordenadas de articulación, velocidades articulares (joint velocities), o en coordenadas cartesianas, torsor de velocidad (velocity screw). A continuación, se presentan las estructuras de control visual cinemático existentes en la literatura, llamadas: control visual cinemático basado en imagen. También existe en la literatura el denominado control visual basado en el movimiento, donde se controlan los movimientos de la cámara.

Este tipo de control visual se basa en la medición de flujo óptico entre dos imágenes consecutivas, lo que permite realizar el control sin conocimiento a priori del objeto.

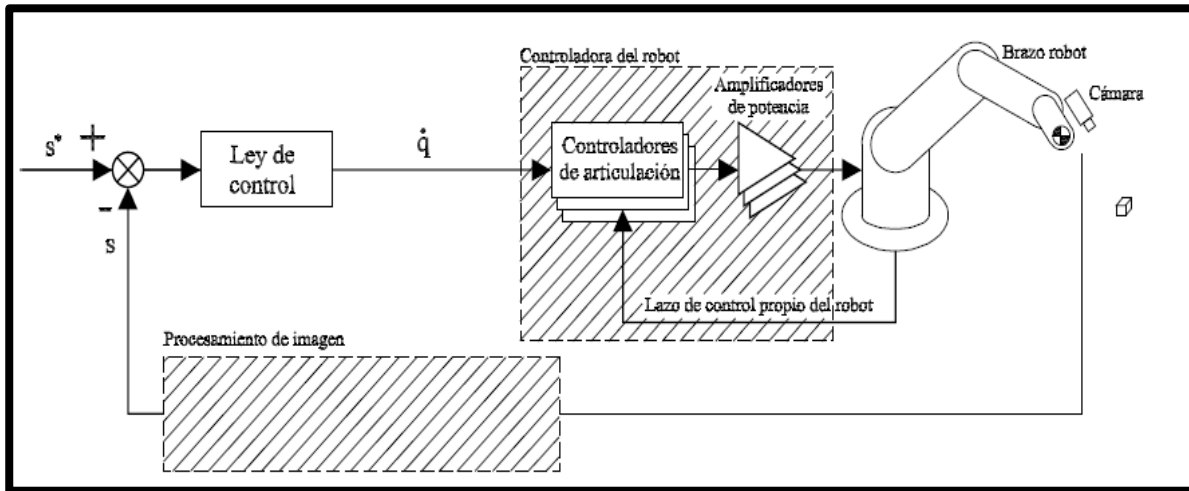


Figura 11. Control Visual Cinemático

Fuente: (García Gómez, 2010)

1.11.7. Control Visual Dinámico

Considerando la dinámica del robot, es posible proyectar otro tipo de controladores que calculan directamente la acción de control de las articulaciones del robot. Este tipo de control se define como Control Visual Dinámico en contrapunto con el Control Visual Cinemático (García Gómez, 2010).

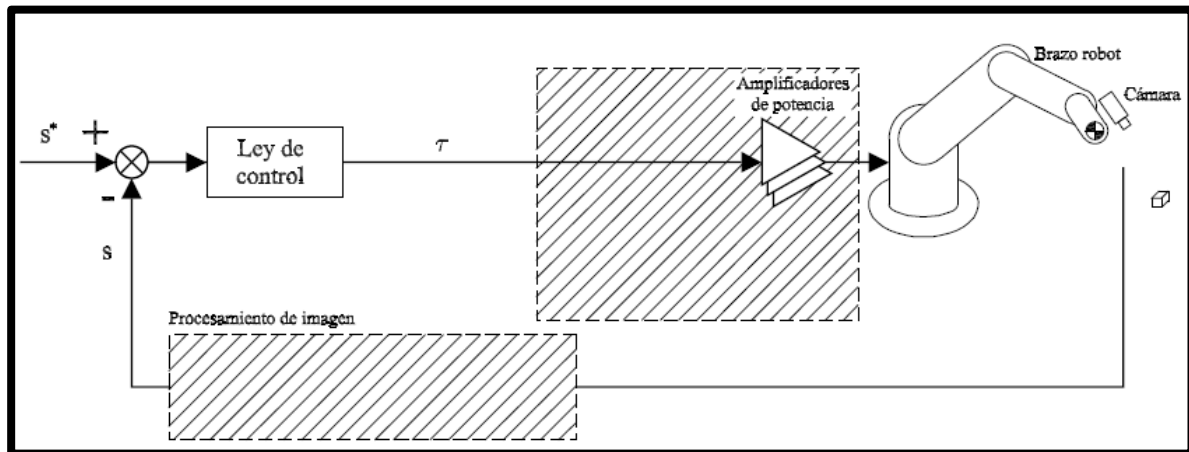


Figura 12. Control Visual Dinámico

Fuente: (García Gómez, 2010)

1.12. Detección facial usando Haar Cascades

Es un método efectivo utilizar las características de los clasificadores en cascada para la detección de objetos. Esta técnica es basada en el aprendizaje de máquina donde la función de Haar Cascades es la que entrena a una gran cantidad de imágenes entre positivas y negativas por ejemplo para localizar rostros, el programa debe tener una gran cantidad de imágenes con caras y sin caras para de esta forma poder entrenar al clasificador, de esta forma se extrae las características de la misma. Para esto, se utilizan Haar Cascades que se muestran a continuación en la figura 13 (Aprender Python, 2017).

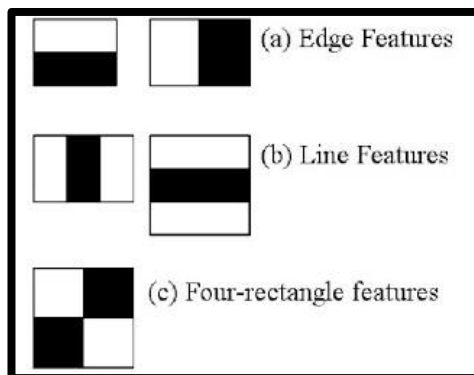


Figura 13. Imagen

Fuente: (Aprender Python, 2017)

En la Figura 14 se calcula las características se desea detectar en este caso los ojos para lo cual se resta el valor obtenido de la suma de los píxeles rectángulo blanco de la suma de los píxeles rectángulo negro.

Para calcular todas las características posibles se deben tomar en cuenta todos los tamaños y ubicaciones que pueda tomar el objeto a detectar. Para cada cálculo de la función, se debe encontrar la suma de los píxeles en virtud de rectángulos blancos y negros.

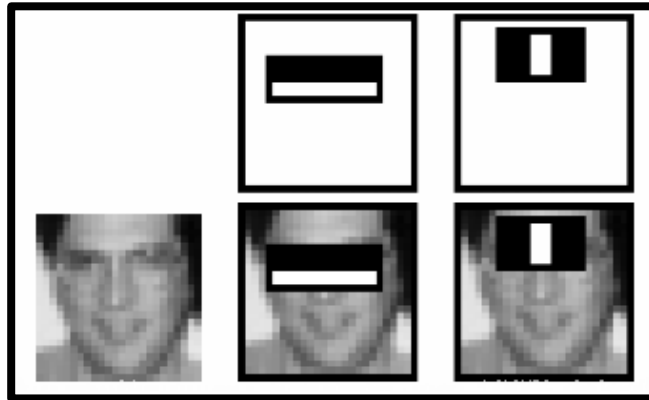


Figura 14. Características del rostro

Fuente: (Aprender Python, 2017)

Dos buenas características son mostradas por la fila superior. La primera característica seleccionada parece centrarse en la propiedad de que la zona de los ojos es a menudo más oscura que la zona de la nariz y las mejillas (Aprender Python, 2017).

La segunda característica seleccionada se basa en la propiedad de que los ojos son más oscuros que el puente de la nariz. Pero las mismas ventanas aplicando en las mejillas o en cualquier otro lugar son irrelevante.

Para seleccionar las mejores características, se aplican todas las funciones en las imágenes de entrenamiento. Por ejemplo, en la detección de la cara para obtener cada característica, se debe encontrar el mejor umbral para poder determinar en una imagen si existe un rostro o no. Para reducir los errores en la clasificación, se debe seleccionar las características con tasa de error mínimo, es decir las características que mejor clasifican la cara.

El clasificador final es una suma de los clasificadores débiles; se llaman débiles ya que individualmente no pueden clasificar a la imagen, pero en conjunto constituyen un fuerte clasificador. En la configuración final se tiene alrededor de 6000 funciones.

Las 6000 funciones se las aplica en diferentes etapas; normalmente primeras etapas contendrán un número con menos de características. Si una imagen no pasa la primera etapa, ya no continua con el reconocimiento de las demás características y se la considera como negativa.

1.13. OpenCV

Es una biblioteca libre dedicada a visión artificial que fue creada por Intel y varios programadores a nivel mundial, que ha permitido el uso de muchas aplicaciones.

Lo más importante es que ha crecido a tal grado, que ahora se cuenta con esta librería en casi todas las plataformas de desarrollo posibles, además contiene más de quinientas funciones que contienen procesos de visión artificial, visión estéreo y robótica, entre otros (Diaz, 2015).

1.13.1. Detección de Haar-cascada en OpenCV

OpenCV permite crear clasificadores para cualquier objeto como un carro, un avión, etc.; también contiene muchos clasificadores para detectar: cara, ojos, sonrisa, etc. Además, para agilizar el proceso de detección posee muestras positivas y negativas, en su biblioteca cuenta con un fichero .xml que contiene la información del clasificador con imágenes entrenadas para su respectiva detección (Marin, 2017).

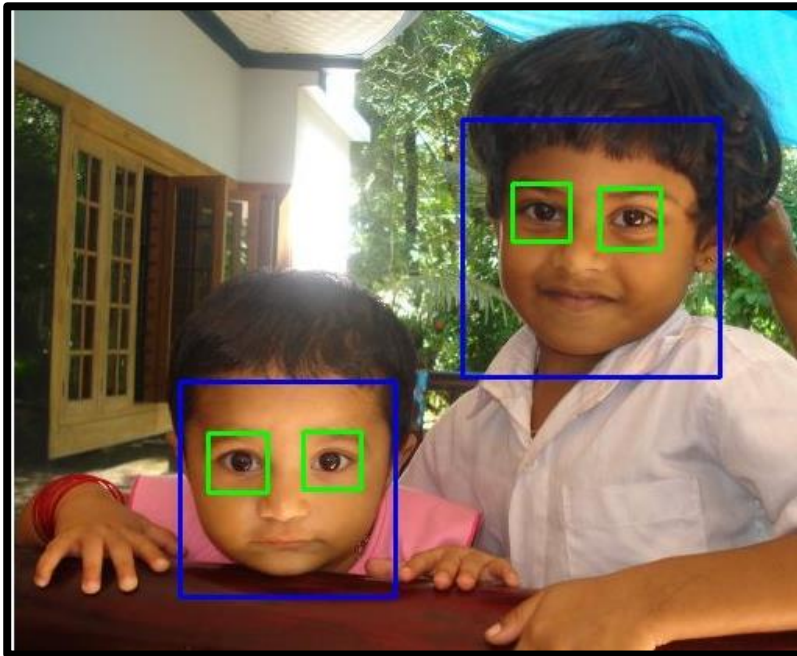


Figura 15. Detección de características de rostro en OpenCV

Fuente: (Marin, 2017)

1.14. Sistemas de reconocimiento de voz

En los últimos tiempos se ha creado una nueva interfaz humano-máquina que permiten combinar tecnologías para poder tener información por medio de la voz. Esta tecnología se basa principalmente en reconocer y capturar la voz, para que una persona controle distintos dispositivos, herramientas electrónicas e informáticos como control telefónico al banco, realizar compras o enviar correo por voz (Larios A. A., 1999).

1.14.1. Reconocimiento de voz

El reconocimiento de voz es la acción de convertir por medio de un computador una señal acústica de voz a texto.

1.14.2. Arquitectura de un sistema de reconocimiento de voz

Para poder comprender como funciona el sistema de reconocimiento de voz es necesario saber cuáles son sus componentes: el extractor de características y el clasificador. Cuando se recibe la señal de voz, el reconocedor actúa y da como resultado la palabra reconocida (Salazar, 2015). Después hay un procesamiento del lenguaje natural, una representación semántica y finalmente se realiza una acción comunicativa. La arquitectura para los sistemas de reconocimiento de voz se muestra a continuación (Aguirre Sánchez, 2015).

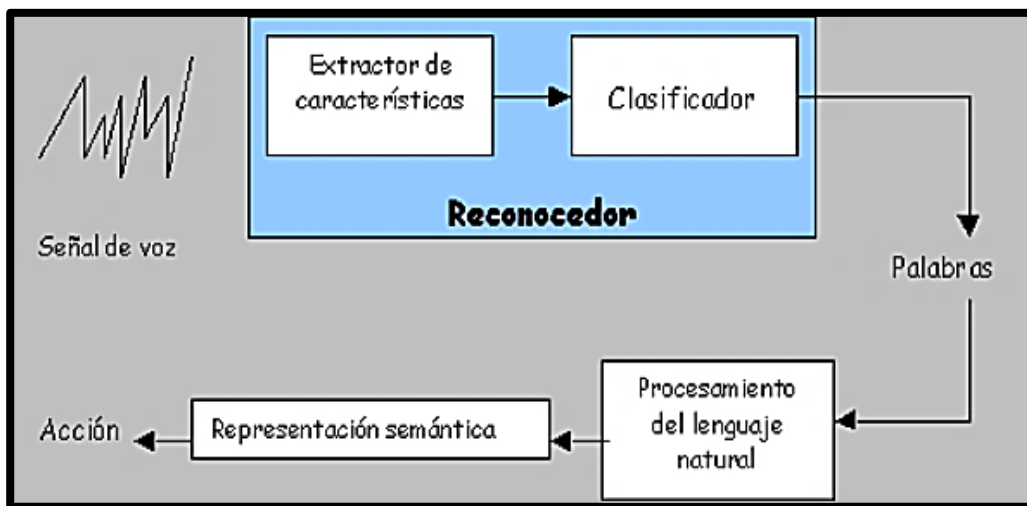


Figura 16. Arquitectura de un sistema de reconocimiento de voz

Fuente: (Aguirre Sánchez, 2015)

1.14.3. Extracción de características

Para la extracción de las características se realizan los siguientes pasos:

1. La señal obtenida se divide en segmentos.

2. Para obtener las características acústicas más significativas del segmento se aplican técnicas para procesar las señales requeridas.
3. Se realiza una nueva entrada de voz.

1.14.4. Clasificador probabilístico

Los pasos a realizar en este módulo son los siguientes:

1. Se crea un modelo probabilístico basado en redes neuronales como modelos ocultos de Markov, etc (Larios A. , 2013).
2. Las probabilidades que se realizaron nos ayudan a encontrar la secuencia de segmentos.

1.14.5. Tipos de sistemas de reconocimiento de voz que existen

La principal meta para desarrollar aplicaciones usando el reconocimiento de voz es crear sistemas que puedan identificar voces. Esto es difícil ya que aún no se cuenta con un sistema que resuelva las dificultades existentes debido a la variabilidad de las características de la señal acústica.

Así, los sistemas reconocimiento viene dado por:

Usuarios/locutores

Sus características son definidas por la variabilidad de los sonidos pronunciados por diversas personas.

Sistema dependiente del locutor

Funciona para un sólo hablante. Se puede desarrollar de manera más fácil y más precisa, pero no es flexible.

Sistema independiente del locutor

Sistema adaptable al hablante es difícil de desarrollar, no tiene tanta precisión, pero es más flexible.

1.14.6. Estilo del habla

Sus características son definidas por los diversos estilos de habla, como son habla continua y discreta. El habla continua funciona sobre un lenguaje en el que las palabras están conectadas, es decir, no están separadas por pausas (Inteligencia Artificial, 2014). Cuando se habla de forma seguida es más compleja de tratar por los diversos efectos; ya que es difícil saber dónde inician y terminan las palabras. La formación de los fonemas se complica porque se generan los fonemas uno a lado del otro, y también se ven afectados cada una de las palabras, al comienzo y al final. La frecuencia del habla también afecta el reconocimiento del habla (Berent & Vaknin, 2017).

La forma discreta funciona con palabras simples, en la que es necesario una pausa entre cada una de las palabras. Esta es la manera más simple de llevarlo a cabo porque los puntos de finalización son más fáciles de encontrar y no tiene problemas con la forma de pronunciar. Con este sistema se identifica con más exactitud las palabras.

1.14.7. Tamaño del vocabulario

Sus características son definidas frente al vocabulario que puede reconocer. Aquí es importante el tamaño ya que el sistema de identificación de voz afecta a su dificultad para lo que se necesita un proceso de precisión del sistema.

En algunos casos el sistema de identificación necesita unas pocas palabras como solo cifras y en otros requieren más palabras como un diccionario. Dado que no existe definiciones determinadas se utiliza la siguiente clasificación:

- Vocabulario pequeño - decenas de palabras
- Vocabulario mediano - cientos de palabras
- Vocabulario grande - miles de palabras
- Vocabulario muy grande - decenas de miles de palabras, entre otros (Muñoz, 2015).

1.14.8. Condiciones de trabajo

Sus características son definidas frente a las condiciones de entorno. Uno de ellos el ruido puede degradar bastante el nivel de desempeño del reconocedor, y éste es producido por: el ruido ambiental, suspiros, música, etc. Otros factores son el ruido producido por el locutor, la calidad del micrófono, entre otros.

1.14.9. Aplicaciones

Las características principales que debe tener el sistema de reconocimiento de voz son la capacidad de reconocer palabras o comandos de la aplicación de buena forma,

mantener un buen nivel de servicio frente a cambios de usuario, canal de comunicación, ruidos, etc., también aprobar el rechazo de palabras que no formen el vocabulario de reconocimiento y trabajar en tiempo real. (Lleida, 2000)

Las aplicaciones más relevantes de los sistemas de reconocimiento de voz que actualmente se pueden encontrar en etapa de experimentación o de explotación comercial son:

- Sistemas de control por voz para minusvalía
- Sistemas de control por voz para navegadores web y computadores
- Sistemas de acceso a bases de datos
- Sistemas de venta por catálogo
- Aplicaciones aeroespaciales
- Servicios comunicativos de telefonía sin operadora
- Validación de ventas con tarjeta de crédito
- Máquina de escribir que actúe por dictado
- Domótica
- Autoaprendizaje de idiomas

Entre las aplicaciones más actuales de los sistemas de reconocimiento de voz como interfaz entre individuo y máquina esta la ayuda a discapacitados físicamente. Mediante comandos de voz se pueden controlar muchas de las funciones y actividades cotidianas (Mayor, 2012). Ejemplos en etapa de experimentación son la silla de ruedas controlada

por voz, camas hospitalarias, control oral de dispositivos electrónicos y la activación por voz de aparatos y sistemas domésticos (Olan Robles, 2008).

1.15. Sistema de pesaje

Permite determinar resultados de pesaje dentro de la industria para dar precisión a los datos de almacenamiento de un producto.

1.15.1. Pesaje estático

El peso es la fuerza que tiene la gravedad sobre un cuerpo, lo principal que se mide es la variación de un cuerpo en su etapa elástica de deformación, al cual se añade un elemento que cambia sus propiedades eléctricas cuando este se deforma. Los sensores de carga más usadas son básculas (AREC, 2017).

El diseño de los depósitos de pesaje contempla también un ejemplo de la utilización de estos elementos denominados celdas de carga, las cuales están constituidas por sensores que emiten una señal eléctrica ante variaciones de peso; así la señal se acondiciona y trasmite hacia un controlador eléctrico, el cual permite visualizar una lectura precisa del peso de la materia prima. En la parte electrónica se construye registros y funciones para complementar el sistema (Nicola, Repositorio Digital Escuela Politécnica Nacional, 2009).

CAPÍTULO II

2. DISEÑO, SELECCIÓN E IMPLEMENTACIÓN DE LOS COMPONENTES DEL SISTEMA DE ALIMENTACIÓN

En este capítulo se describe la selección de los componentes y diseño estructural del sistema para alimentar a personas afectadas con osteoartritis. Para elegir los elementos de forma más adecuada se utiliza la escala de evaluación propuesta en el Manual para la Medición de la Eficiencia de Manejo de un Sistema de Arturo Izurieta.

Tabla 1

Escala de puntuación

Puntuación	% del Óptimo	Significado
0	< 35	Insatisfactorio
1	35 - 50	Poco satisfactorio
2	51 – 75	Medianamente satisfactorio
3	76 – 90	Satisfactorio
4	91 - 100	Muy satisfactorio

Fuente: (Izurieta, 1997)

2.1. Criterios técnicos

Es preciso determinar los criterios técnicos de implementación que se utilizan para determinar un buen espacio de funcionamiento del sistema. Por lo tanto, este proyecto describe criterios fundamentales como:

- Seleccionar y ubicar los diferentes elementos que conformarán el sistema de alimentación.
- Definir correctamente y seleccionar un mecanismo de traslación adecuado para el sistema.
- Identificar y utilizar un software para la programación del control.

2.2. Parámetros de diseño

Debido a que el proyecto debe realizar un buen proceso de alimentación para el usuario, el sistema cumplirá con las siguientes etapas:

- Activación y desactivación del sistema.
- Manipulación y transporte de comida.
- Determinación del área de trabajo, con el espacio necesario para la ubicación de los utensilios de comida, el mecanismo de manipulación y otros elementos.
- Mecanismo de regulación de altura de la estructura de soporte.
- Identificación del usuario para su alimentación.
- Control de pesaje.

Según las etapas dadas anteriormente, se estima:

- Activar y desactivar el sistema en dos modos semiautomático y automático.
- Usar el prototipo de brazo robótico para la manipulación y transporte de los alimentos hacia el usuario.
- Calcular el área de alimentación para diseñar y construir una estructura de soporte, para todo el sistema.
- Diseñar un mecanismo para regular la altura de la estructura de soporte.
- Utilizar el reconocimiento de la boca, para garantizar que el transporte de comida llegue al usuario.
- Realizar un sistema para el control de peso que permita identificar si el recipiente está lleno o vacío.

2.3. Diseño y selección de componentes

Después de analizar los parámetros de diseño, se establece la selección de los elementos del sistema de alimentación nombrados a continuación:

- Prototipo de brazo robótico, que soporte un peso mínimo de 150 gr, con un alcance mínimo de 320mm, 5 grados de libertad y una pinza para manipular objetos.
- Utensilios de alimentación para el usuario.
- Un sistema de activación por pulsadores, que actúe como auxiliar en el proceso de alimentación.
- Un sistema que permita realizar un control por voz para dar más comodidad a las personas.
- Un sistema de visión artificial, que permita determinar el área específica del reconocimiento de la boca del usuario.
- Una cámara con autoenfoque.
- Un sistema de pesaje para determinar si el brazo robótico continúa con el proceso de alimentación.
- Una mesa regulable en altura que contenga todos los elementos del sistema de alimentación.

2.3.1. Descripción de los componentes

Después de determinar los parámetros que debe tener el sistema de alimentación, a continuación, se realiza la selección de componentes en la que se detalla sus características.

2.3.1.1. Prototipo del brazo robótico TINKERKIT BRACCIO ROBOT

Se requiere un Brazo Robótico que permita alimentar a una persona, el mismo que debe estar diseñado para tener una buena movilidad y que su efector final tenga una carga máxima de más de 200g.

Opción 1 Prototipo del brazo robótico TINKERKIT BRACCIO ROBOT

Tabla 2

Características Físicas

Características	
Capacidad de carga	300 g
Alcance máximo	520 mm
Estabilidad	Su material es de plástico por lo que es más liviano su movilización
Torque	12,2 kg-cm
Servo shield	Braccio shield
Precio	400 dólares

Fuente: (Arduino, 2017)



Figura 17. Partes del Brazo Robótico

Fuente: (Arduino, 2016)

Opción 2: Prototipo del brazo robótico de 6 DOF

Tabla 3

Especificaciones técnicas del brazo robótico

Características	
Capacidad de carga	150 g
Alcance máximo	460 mm
Estabilidad	Su material es de aluminio, no es estable por el peso del material
Torque	8 kg-cm
Servo shield	Ninguno
Precio	200 dólares

Fuente: (Hobbytronics, 2017)

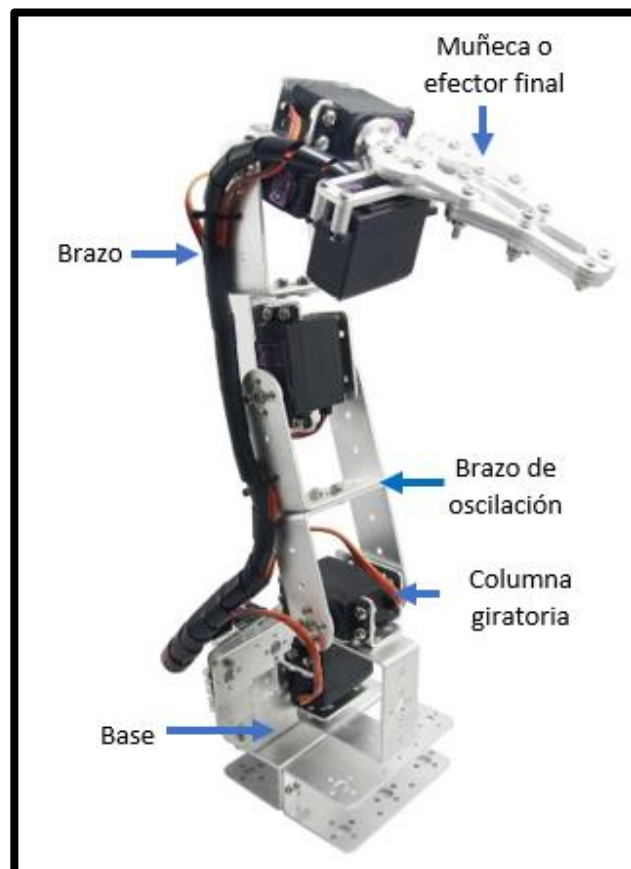


Figura 18. Partes del Brazo robótico de 6 DOF
Fuente: (Hobbytronics, 2017)

Selección del mejor kit de Brazo Robótico

Tabla 4

Comparativa de las opciones para elegir el mejor Brazo Robótico

Parámetros	Opción 1	Opción 2
Capacidad de carga	3	1
Alcance máximo	3	1
Flexibilidad	4	1
Torque	3	1
Servo shield	4	1
Precio	1	3
Suma	18	8

En base a la tabla 4 la opción que cumple con los requerimientos que necesita el proyecto es la Opción 1 el brazo robótico TINKERKIT BRACCIO ROBOT por su capacidad de carga, alcance máximo, flexibilidad, y el torque lo que hacen que esta alternativa sea la mejor.

2.3.1.2. Utensilios de comida

El sistema para alimentación contará con tres platos de comida para el usuario y un vaso el mismo que será levantado por el brazo robótico. Por lo que quedan definidos las siguientes dimensiones para los utensilios a usar mostrados en la tabla 5.

Tabla 5

Utensilios a ser usados

Utensilio	Dimensiones		
	Parámetros	Máximos	Mínimos
Plato	Diámetro	154 mm	100 mm
	Altura	60 mm	40 mm
Cuchara	Capacidad	0,5 onzas (15g-15 ml)	0,35 onzas (10g-10 ml)
Vaso	Capacidad	4 onzas (114g-114ml)	3,5 onzas (100g-100ml)
	Altura	50 mm	40 mm
	Diámetro	60 mm	45 mm

2.3.1.3. Pulsadores

Para el control semiautomático se hace necesario utilizar un actuador amigable con el usuario y de fácil activación, como un pulsador interactivo.

Dado que el usuario presenta debilidad en sus extremidades superiores los pulsadores deben ser adecuados para poderlos presionar con los pies. En la mayoría de los automóviles los pedales de aceleración y embrague tienen una medida de 50mm x 70 mm los mismos que son diseñados de forma ergonómica para que el conductor se sienta cómodo y no tenga problemas a la hora de manejar; basado en estas dimensiones se utiliza 5 pulsadores de 60mm de diámetro los mismos que darán a la persona mayor confort y facilidad al momento de seleccionar el plato que desea comer.



Figura 19. Pulsadores grandes interactivos

Fuente: (AliExpress, 2012)

La tabla 6 muestra las características que posee el pulsador seleccionado:

Tabla 6

Características de los pulsadores

Característica	Descripción
Material	Policarbonato
Diámetro	60mm
Número de terminales	3
Voltaje	12V

Fuente: (AliExpress, 2012)

2.3.1.4. Computador

Para el proyecto se necesita un ordenador en el cual se pueda ejecutar de manera rápida y eficaz los paquetes: OpenCV y Dlib. También se requiere que se pueda comunicar con otra placa.

Opción 1: Raspberry Pi 3

La Raspberry Pi 3 es un computador de placa reducida. Una de las grandes ventajas es su uso libre tanto a nivel educativo como particular (Ramírez, 2016).

Su sistema operativo ofrece una versión adaptada de Debian, conocida como Raspbian, pero también permite otros sistemas operativos, tales como una versión de Windows 10. Entre sus características se destacan un mejor rendimiento, pues está conformado por un procesador de cuatro núcleos Cortex-A53, de 64 bits con una velocidad de reloj de 1.2 GHz. (Motyka, 2016)



Figura 20. Raspberry Pi 3

Fuente: (Motyka, 2016)

Tabla 7

Características de la Raspberry Pi

Característica	Descripción
Procesador	1,2 GHz de 64 bits de cuatro núcleos CPU ARMv8
Arquitectura	64-Bit
RAM	1 GB LPDDR2
Almacenamiento	MicroSD

Fuente: (Motyka, 2016)

Opción 2: Beaglebone

BEAGLEBONE es un computador Linux de tamaño pequeño el cual se puede conectar a Internet. Presenta una potencia de procesamiento para el análisis en tiempo real proporcionada por un procesador AM335x 720MHz ARM, BEAGLEBONE al igual que un ordenador normal puede conectarse con otras placas como Arduino para aumentar la funcionalidad (González S. , 2016).

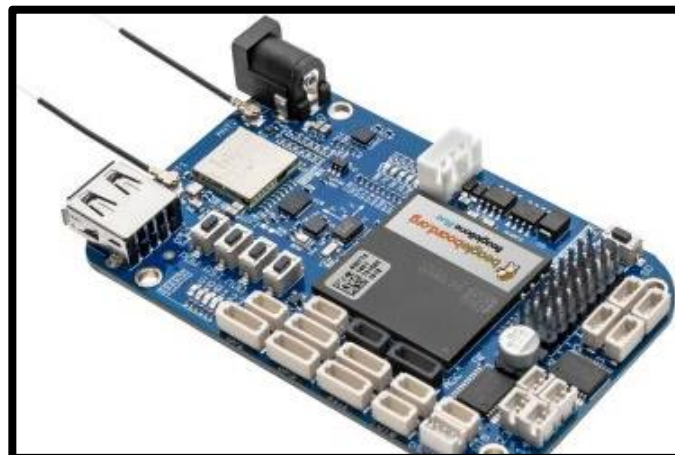


Figura 21. Beaglebone

Fuente: (González S. , 2016)

Tabla 8

Características de la Beaglebone

Característica	Descripción
Procesador	1 GHz TI Sitara AM3359 ARM Cortex A8
Arquitectura	32-Bit
RAM	512 DDR3L
Almacenamiento	MicroSD

Fuente: (González S. , 2016)

Selección de la mejor opción para el Computador

Tabla 9

Comparativa de las opciones para elegir el computador

Parámetros	Opción 1	Opción 2
Procesador	3	2
Arquitectura	3	2
RAM	4	1
Almacenamiento	2	2
Suma	12	7

En base a la tabla 9 la opción que cumple con los requerimientos que necesita el proyecto es la Opción 1 la cámara **Raspberry Pi 3** por su procesador, arquitectura y ram lo que hacen que esta alternativa sea la mejor.

2.3.1.5. Pantalla

Para una mejor interacción con el usuario y programador se necesita el uso de una pantalla que sea compatible con la tarjeta Raspberry Pi3.

Opción 1: HDMI LCD

Tabla 10

HDMI LCD

Parámetros	Características
Adaptabilidad	Necesita una carcasa
Compacto	Frágil
Fácil reparación	Elementos difíciles de encontrar
HDMI	Si

Fuente: (AliExpress, 2012)



Figura 22. HDMI LCD de 7 pulgadas (B)

Fuente: (AliExpress, 2012)

Opción 2: Pantalla de 7 pulgadas

Tabla 11

Pantalla de 7 pulgadas

Parámetros	Características
Adaptabilidad	Se puede colocar en una mesa o pared
Compacto	Resistente
Fácil reparación	Si existen los elementos
HDMI	Si

Fuente: (Amazon, 2017)



Figura 23. Pantalla de 7 pulgadas
Fuente: (Amazon, 2017)

Selección de la mejor opción para la pantalla

Tabla 12

Comparativa de las opciones para elegir la pantalla

Parámetros	Opción 1	Opción 2
Adaptabilidad	1	4
Compacto	2	3
Fácil reparación	1	4
HDMI	4	4
Suma	6	15

En base a la tabla 12 la opción que cumple con los requerimientos que necesita el proyecto es la Opción 1, la **Pantalla de 7 pulgadas** por su adaptabilidad, ser más compacta, fácil de reparar, lo que hace que esta alternativa sea la mejor.

2.3.1.6. Arduino

Arduino es una plataforma electrónica de código abierto, su software y hardware son fáciles de usar por su sencillez, además de su buena compatibilidad (Riquelme García &

Zamora Pedreño, 2014). Para el proyecto se requiere una placa Arduino la misma que ayudará con el control de los 6 servos del brazo robótico, 5 pines digitales de los pulsadores, 3 pines análogos para las celdas de carga y 2 puertos seriales para la comunicación del Arduino con la Raspberry Pi 3 y con la aplicación móvil.

Opción 1: Arduino Uno

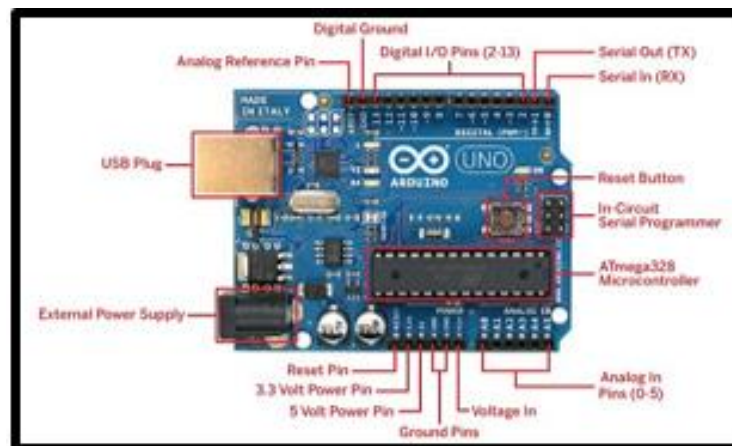


Figura 24. Arduino Uno
Fuente: (Arduino, 2014)

Tabla 13

Características del Arduino Uno

Tipo de microcontrolador	ATmega 328
Voltaje de entrada	7-12V
Número de pines digitales I/O	14 (6 Salidas PWM)
Número de entradas análogas	6
Capacidad de almacenamiento	32k
Velocidad de reloj	16MHz

Fuente: (Arduino, 2014)

En base a la tabla 15 la opción que cumple con los requerimientos que necesita el proyecto es la Opción 1 **Arduino Mega** por su número de pines digitales, puertos serie lo que hacen que esta alternativa sea la más adecuada.

2.3.1.7. Sensor de peso

En este proyecto se requieren 3 sensores de peso los mismos que deben medir el peso de los platos para saber si hay comida, entonces el brazo tome el alimento y si no la hay no tome el alimento. Los sensores deben ser estables, soportar un peso aproximado de más de 1 kg por plato, ser fáciles de implementar.

Opción 1: Celda de carga en viga

Tabla 16

Celda de carga en viga

Parámetros	Características
Estabilidad	80%
Fácil Implementación	70%
Temperatura	Adaptable a cualquier temperatura
Precio	\$10

Fuente: (Corona Leonel, 2014)



Figura 26. Celda de carga de 5 kg

Fuente: (Corona Leonel, 2014)

Opción 2: Flexi Force y FSR

Son similares a las celdas de carga, pero su diferencia es que su resistencia no varía ante una deformación. Para obtener un cambio en la resistencia eléctrica es necesario aplicar una fuerza sobre el extremo circular del sensor. El área de detección es considerada en un solo punto, por tanto, para obtener mediciones adecuadas es necesario que la fuerza se distribuya de manera uniforme sobre el área en donde actúa.

Tabla 17

Flexi Force y FSR

Parámetros	Características
Estabilidad	50%
Fácil implementación	55%
Temperatura	Mantiene una temperatura regular para trabajar
Precio	\$6

Fuente: (A-Electronics, 2017)

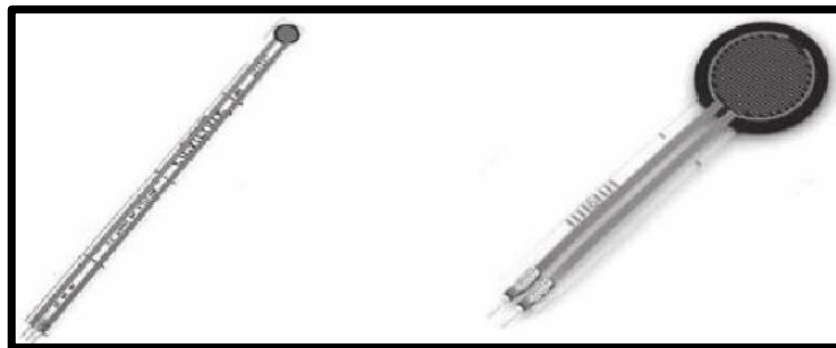


Figura 27. Flexi Force y FSR

Fuente: (A-Electronics, 2017)

Selección de la mejor opción para el sensor de peso

Tabla 18

Comparativa de las opciones para elegir el sensor de peso

Parámetros	Opción 1	Opción 2
Estabilidad	3	2
Fácil Implementación	3	2
Temperatura	4	1
Precio	2	3
Suma	12	8

En base a la tabla 18 la opción que cumple con los requerimientos que necesita el proyecto es la Opción 1 la **Celda de carga en viga** por su estabilidad, fácil implementación, temperatura lo que hacen que esta alternativa sea la mejor.

2.3.1.8. Cámara

Para el reconocimiento y seguimiento de la boca del usuario se requiere una cámara de alta de alta resolución, de bajo peso para que el brazo robótico no tenga problemas al momento de movilizarse, ser compatible con la Rasperry Pi 3.

Opción 1: Cámara web FACECAM 1000X 720p HD GENIUS



Figura 28. FACECAM 1000X
(Electronicabf, 2017)

Tabla 19

Especificaciones técnicas cámara FACECAM 1000X

Especificaciones técnicas	
Resolución	Imágenes: 1MP, 1280x720, 640x480 píxeles Video: CIF / VGA, 720p HD con hasta 30 cuadros por segundo
Sensor de imagen	CMOS HD 720p píxeles
Interfaz	USB 2.0
Tamaño	60x20x22 mm
Peso	50 g

Fuente: (Electronicabf, 2017)

Opción 2: Cámara web FACECAM 315 GENIUS



Figura 29. FACECAM 315

Fuente: (Estimamos, 2012)

Tabla 20

Especificaciones técnicas cámara FACECAM 315

Especificaciones técnicas	
Resolución	Imágenes: 0,3MP, 640x480 píxeles Video: CIF / VGA
Sensor de imagen	CMOS
Interfaz	USB 1.1
Tamaño	56x50x35 mm
Peso	62g

Fuente: (Estimamos, 2012)

Selección de la mejor opción para la Cámara

Tabla 21

Comparativa de las opciones para elegir la cámara

Parámetros	Opción 1	Opción 2
Peso	3	2
Tamaño	3	1
Resolución	4	1
Sensor	3	1
Interfaz	3	2
Precio	2	3
Suma	18	10

En base a la tabla 21 la opción que cumple con los requerimientos del sistema es la Opción 1 la cámara **FACECAM 1000X** por su peso, tamaño, resolución, y el sensor que hacen que esta alternativa sea la mejor.

2.4. Diseño de la estructura mecánica

Seleccionados ya los elementos que conforman el sistema para alimentación, se toma en cuenta las dimensiones de la tabla 2 para determinar las dimensiones de la estructura mecánica de soporte, y la ubicación de sus elementos. En la tabla 22, se identifican los elementos que forman parte del área máxima de alimentación.

Tabla 22

Elementos del área máxima de alimentación

Número	Elemento
1	Prototipo de brazo robótico 6DOF
2	Área de trabajo o zona de alimentación
3	Alcance máximo del brazo robótico

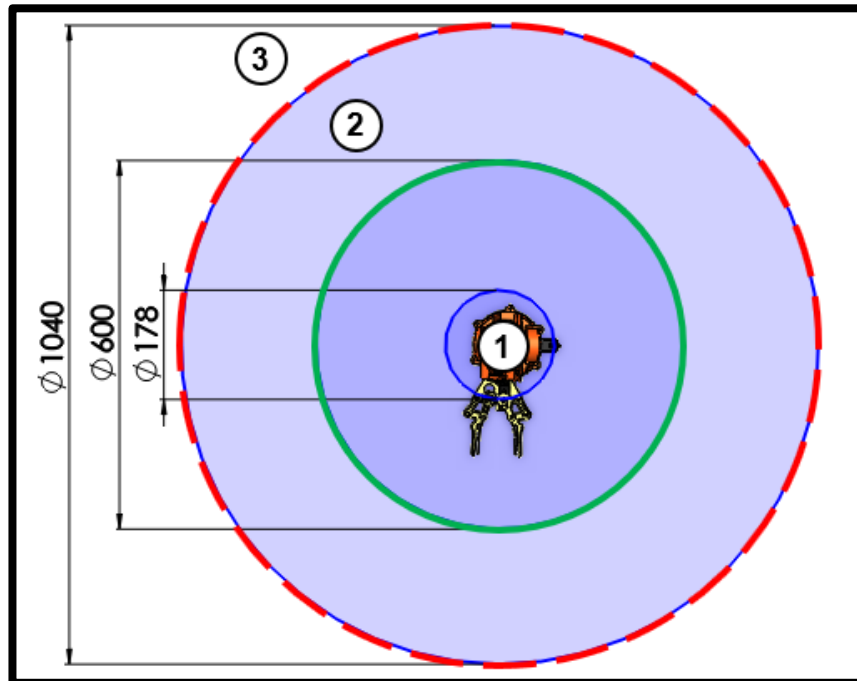


Figura 30. Área máxima de alimentación

El área máxima de alimentación es el espacio en donde el usuario puede ubicarse a degustar sus alimentos tiene un valor de 1040mm^2 , esto determinado por el alcance máximo del brazo robótico.

2.4.1. Diseño de la estructura mecánica de soporte

Para el diseño de la estructura mecánica de soporte se ha seleccionado una mesa de altura ajustable para dar mejor comodidad al usuario.

Los parámetros de terminología del dimensionamiento de la estructura mecánica de soporte siguen como referencia la norma NTE INEN 1641:1988, la misma que determina la terminología de dimensionamiento de una mesa común como muestra la Figura 31.

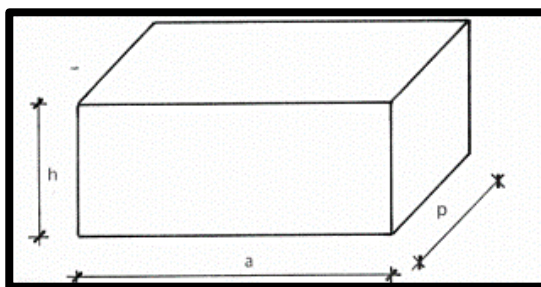


Figura 31. Terminología de dimensionamiento de una mesa
Fuente: (NTE INEN 1641, 2013)

En donde:

a = Ancho (respecto al usuario)

P = Profundidad (respecto al usuario)

h = Altura (respecto al usuario)

2.4.1.1. Dimensionamiento del ancho y profundidad de la superficie de la mesa, y dimensionamiento de ubicación de los utensilios de comida

La NORMA INEN 1641 determina valores nominales del espacio libre para el alojamiento de las piernas como muestra la figura 32, así el ancho de la superficie de una mesa tiene un valor mínimo de 580 mm, cabe recalcar que, por la implementación de otros accesorios en este caso se determinará un valor de 810mm. Además, la profundidad a nivel de los pies es determinada por un valor mínimo de 600 mm, y en este caso por la implementación de una plataforma de apoyo para los pies del paciente su valor será de 670mm.

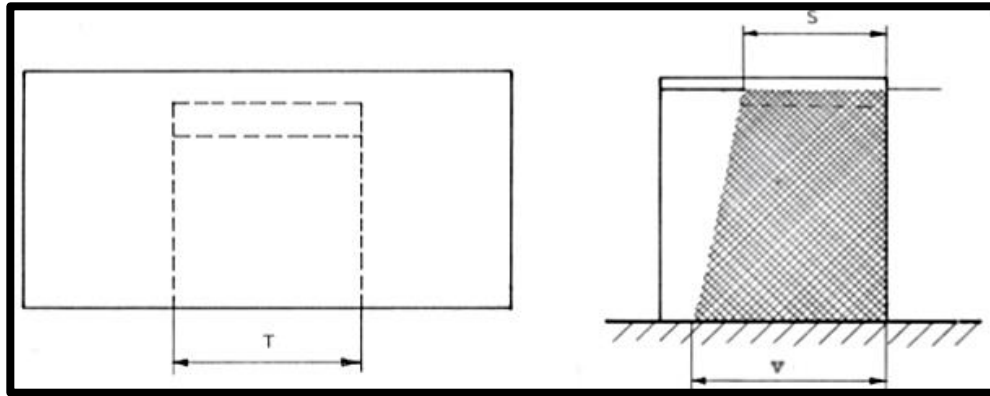


Figura 32. Espacio libre para el alojamiento de las piernas según la norma NTE INEN 1641:1988.

Fuente: (NTE INEN 1641, 2013)

En donde:

Ancho libre $T = 580$ mm mínimo

Profundidad a nivel de las rodillas $S = 450$ mm mínimo

Profundidad a nivel de los pies $V = 600$ mm mínimo

Así, se puede definir las dimensiones de la mesa la misma, que queda determinada por un ancho de 810 mm y una profundidad de 670mm, como muestra la figura 33.

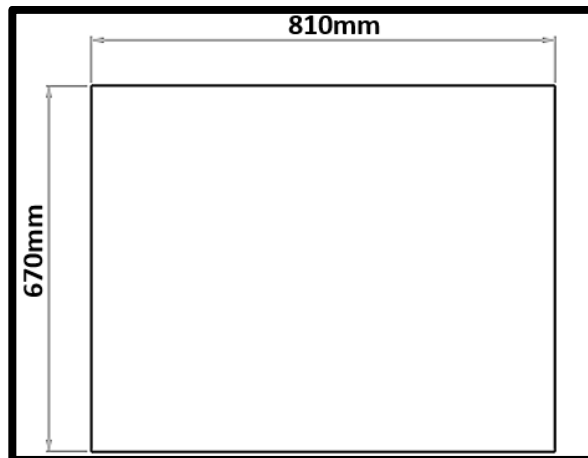


Figura 33. Dimensiones del ancho y profundidad de la superficie de la mesa

Para calcular el radio de ubicación exacta de los utensilios de comida se utiliza el teorema de Pitágoras, en donde el triángulo rectángulo está formado por el brazo de oscilación y la unión del brazo con la muñeca o efector final.

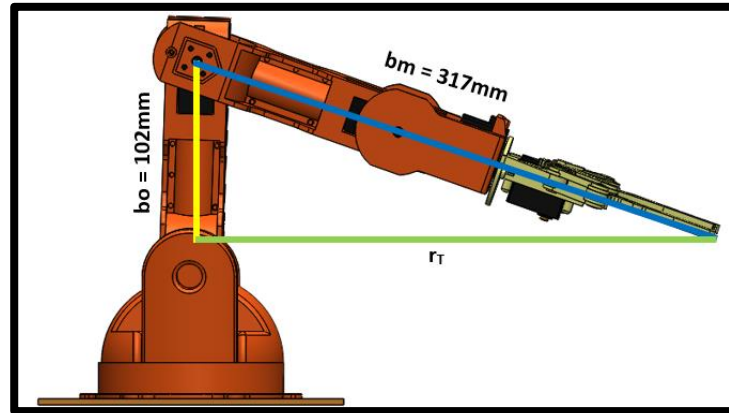


Figura 34. Triángulo rectángulo con medidas reales para el cálculo del radio de ubicación de los utensilios de comida con respecto al centro del brazo robótico

En donde,

bo = Medida del brazo de oscilación

bm = Medida del brazo + Medida de la muñeca o efector final

r_T = Medida del radio de ubicación de los utensilios de comida

Por lo tanto:

$$bm^2 = bo^2 + r_T^2 \text{ (Ec. 0.1)}$$

$$r_T = \sqrt{bm^2 - bo^2}$$

$$r_T = \sqrt{(317)^2 - (102)^2} \text{ mm}$$

$$r_T = 300\text{mm}$$

El radio total óptimo de ubicación de los utensilios de comida es 300 mm con respecto al punto inicial de ubicación del brazo robótico, como son en total tres platos de comida, para la ubicación de cada plato es necesario el cálculo de la longitud de separación entre ellos, es por eso que se utiliza la fórmula de cuerda asumiendo un ángulo de 36° como muestra la figura 35.

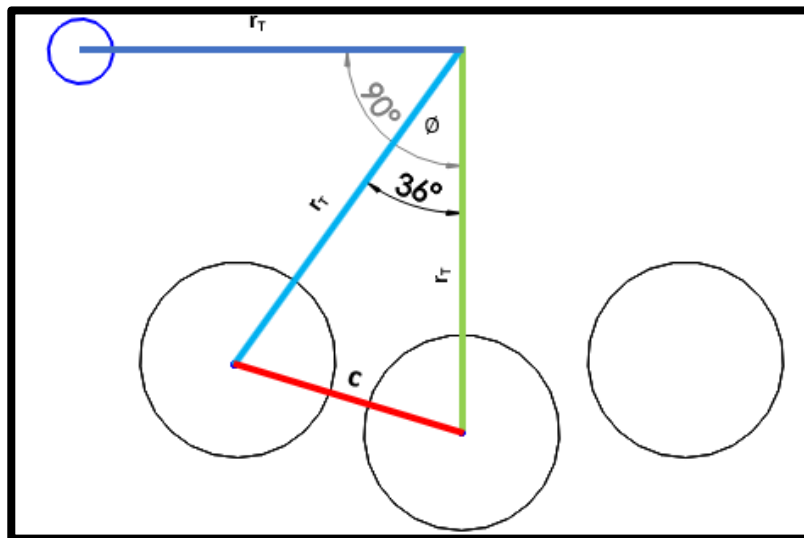


Figura 35. Dimensionamiento de ubicación de los utensilios de comida

En donde,

ϕ = Ángulo formado entre radios

c = Longitud de separación entre cada plato

Así:

$$c = 2r_T \sin \frac{\phi}{2}$$

$$c = \left[2(300) \sin \left(\frac{36^\circ}{2} \right) \right] \text{mm}$$

$$c = 185 \text{mm}$$

La separación entre plato y plato queda delimitada por un valor de 185mm y un ángulo de 36°, y el vaso de bebida a una distancia de 300mm con ángulo de 90° con respecto al segundo plato de comida.

2.4.1.2. Dimensionamiento de la altura de la mesa

Previo el dimensionamiento de la altura de la mesa, es necesario seleccionar el espesor de la superficie de la misma, cuyo valor debe ser de por lo, menos 20mm, establecido por la norma NTE INEN 1641, 2013 como muestra la tabla 24, en este caso se selecciona un valor de 15mm, y previo un análisis de esfuerzos se verifica si este espesor es el adecuado para soportar las cargas mostradas en la tabla 23.

Tabla 23

Valores de masa a levantar

Descripción	Cantidad	Masa g
Platos	3	466,5
Vaso	1	39,63
Brazo robótico	1	592,26
Monitor 7 pulgadas LCD	1	379,00
Porta cuchara y cuchara	-	85,77
Otros	-	543,00
Masa de los brazos de la persona sobre la mesa -		16417,96

Para verificar si el espesor de la superficie de madera a usar es el correcto se toma como referencia la tapa de la mesa, la misma que por medio de la teoría de la deformación de Von Mises manifiesta que un material comienza a ceder en una ubicación cuando la tensión de Von Mises es igual al límite de tensión (Fanghanel Córdova, 2015). En la mayoría de los casos, el límite elástico se utiliza como el límite de tensión (SolidWorks, 2018).

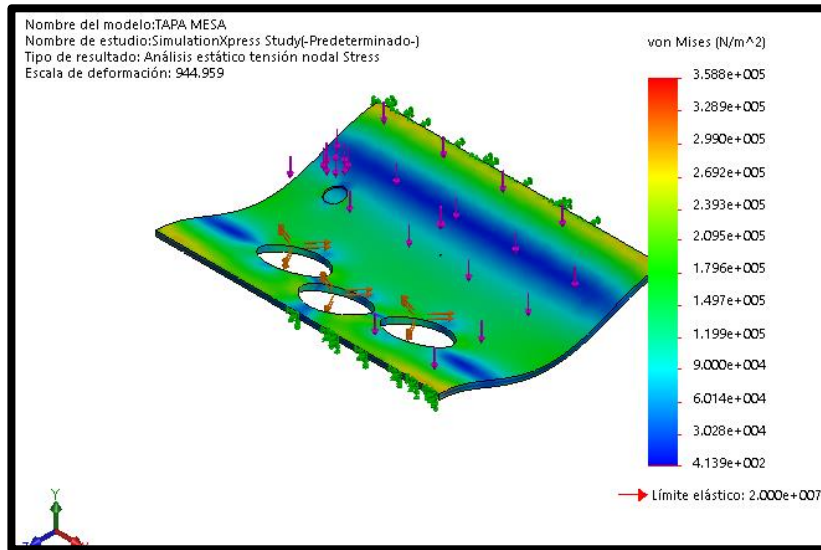


Figura 36. Análisis de esfuerzo Von Mises

Es así que la mayor tensión de Von Mises se encuentra donde se recibe todo el peso de la superficie con un valor de 0,358MPa. La tensión máxima a la tracción de diseño del material a usar es de 0,4MPa(madera), es así que se cumple:

$$\sigma' < \sigma_d$$

$$0,385Mpa < 0,4Mpa$$

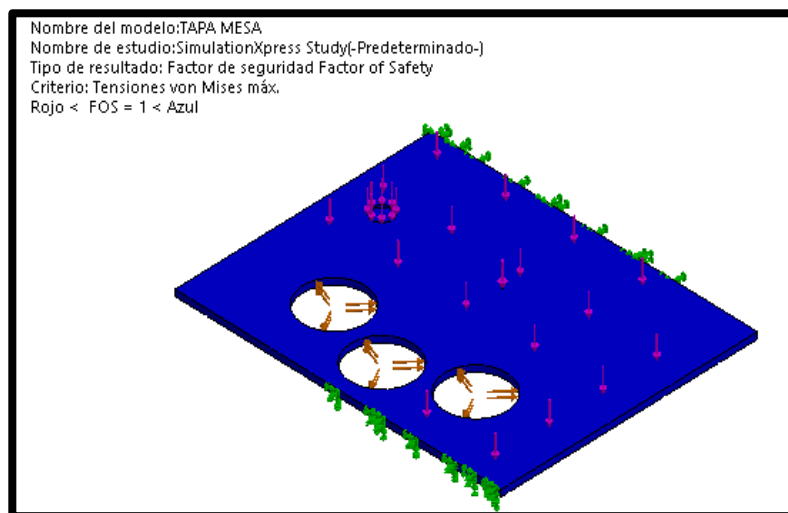


Figura 37. Factor de seguridad de la superficie

El factor de seguridad estático mínimo para los ensayos en la madera es de 1, en la figura 37 se observa que el valor cumple el requerimiento de diseño para evitar la falla estructural con un valor de 1.

Existen parámetros para determinar la altura de la mesa uno de ellos, la profundidad de los platos de comida mostrada en la Tabla 5, con un valor 50mm, el espesor de la superficie de la mesa 15mm, y el espacio en donde se ubicarán los sensores de peso.

Entonces:

$$h = (50 + 15 + 10)mm = 75mm$$

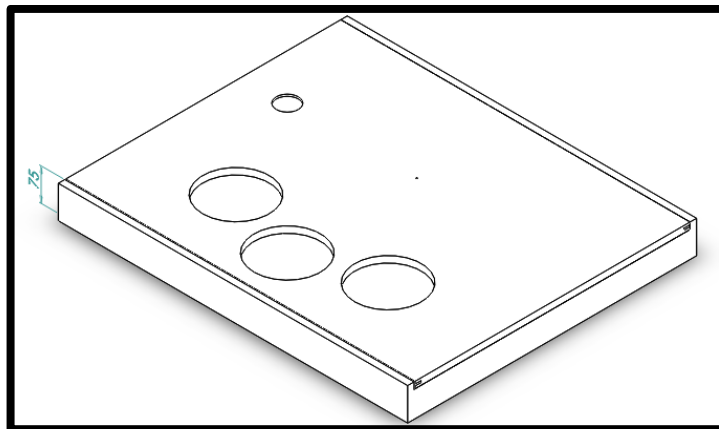


Figura 38. Altura de la superficie de la mesa

2.4.1.3. Determinación del intervalo de altura regulable de la mesa

Para la determinación del intervalo de altura regulable se utiliza la NORMA INEN 1641. Por lo que se hace necesario sumar a la altura poplítea en posición sentado y la distancia del espacio desde la silla hasta el codo, esto para dar mejor ergonomía y comodidad al usuario, en la figura 39 y en la tabla 24 se pueden apreciar estas partes y sus dimensiones.

Así:

$$h_{\text{codo}} = A + E$$

En donde:

Dimensión del espacio desde la silla hasta el codo $E = 260\text{mm}$

Altura poplíteica en posición sentado $A = 450\text{mm}$

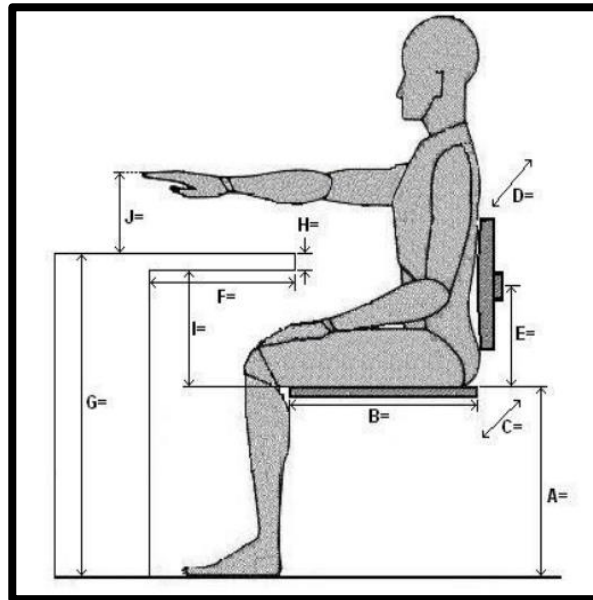


Figura 39. Puesto de trabajo sentado

Fuente: (Jaravito, 2009)

Tabla 24

Datos para el dimensionamiento de la superficie de la altura de la mesa

Cota	Referencia	Medida (mm)
A	Altura del asiento	470
B	Profundidad del asiento	400
C	Ancho del asiento	400
D	Ancho del respaldo	480
E	Altura del soporte lumbar	180
F	Espacio para las piernas	400
G	Altura de la superficie de trabajo	650
H	Grosor de la superficie de trabajo	20
I	Espacio para los muslos	260
J	Altura máxima para controles de uso frecuente	80
K	Altura total del asiento	800
L	Ancho de la superficie de trabajo	400

Fuente: (NTE INEN 1641, 2013)

La altura mínima a la que se debe fijar la superficie de trabajo h_1 viene determinado entonces por la altura h_{codo} , en donde:

$$h_1 = (470 + 180)\text{mm} = 650\text{mm}$$

La norma INEN 1641:2014 establece un valor de altura mínima de 650mm para mesas plenamente regulables, así para este caso se establecerá una altura de 800mm hasta máximo 920mm, para la comodidad de cada usuario.

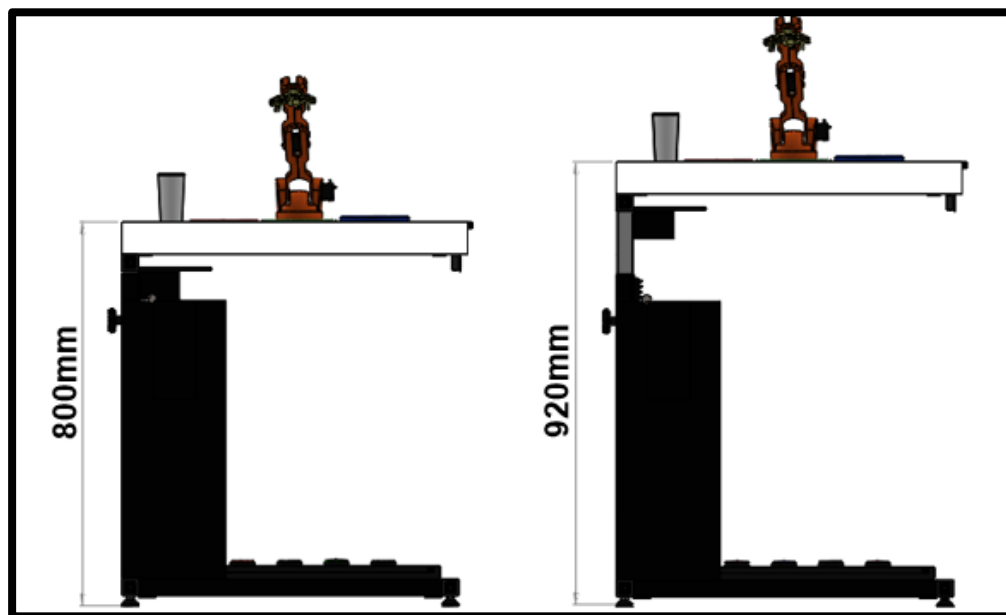


Figura 40. Intervalo de altura de la mesa

2.4.1.4. Mecanismo de regulación de altura

Para el mecanismo de regulación del intervalo de altura se usará la transmisión de movimiento circular a lineal, con los elementos piñón y cremallera, para reducir de costos y mejorar la comodidad del usuario.

Diseño del piñón y cremallera

En el mercado existe gran variedad de piñones con cremalleras estándar, en mayor parte con ángulo de presión de 20° y elaborados de nylon, para aplicaciones de choque uniforme, operación segura, baja fricción, resistencia a la corrosión y buenas propiedades en cuanto a desgaste (Pantoja & Villota, 2013).

Es indispensable también que los elementos piñón y cremallera cumplan con ciertos requisitos de diseño para evitar posibles fallos durante el movimiento del mecanismo, como el choque de la punta del diente del piñón y el chaflán o raíz del diente de la cremallera. Así es recomendable definir un número mínimo de dientes del piñón. (Mott, 1995)

Tabla 25

Número de dientes en el piñón para asegurar que no exista interferencia

Para un piñón que se enlaza con una cremallera		Para un piñón de profundidad total a 20° que se enlaza con un engrane	
Forma de diente	Número mínimo de dientes	Número de dientes del piñón	Número de dientes en los engranes
Envolvente, profundidad total, 141/2°	32	17	1309
		16	101
		15	45
Envolvente, profundidad total, 20°	18	14	26
		13	16
Envolvente, profundidad total, 25°	12		

Fuente: (Mott, 1995)

Por lo tanto, con base a la tabla 25 para el sistema envolvente de profundidad total a 20°, se utiliza no menos de 18 dientes para asegurar que no haya interferencia.

Tabla 26

Piñones cilíndricos con ángulo de presión de 20°

Z	Mod. 3				Mod. 3,5				Mod. 4				Mod. 5				Mod. 6			
	d _e	d _p	d _m	D ₁	d _e	d _p	d _m	D ₁	d _e	d _p	d _m	D ₁	d _e	d _p	d _m	D ₁	d _e	d _p	d _m	D ₁
12	42	36	27	12	49	42	35	12	56	48	35	14	70	60	45	20	84	72	54	20
13	45	39	30	12	52.5	45.5	40	12	60	52	40	14	75	65	50	20	90	78	60	20
14	48	42	35	12	56	49	45	12	64	56	45	14	80	70	55	20				
15	51	45	35	12	59.5	52.5	45	12	68	60	45	14	85	75	60	20	102	90	70	20
16	54	48	38	14	63	56	50	14	72	64	50	16	90	80	65	20	108	95	75	20
17	57	51	42	14	66.5	59.5	50	14	76	68	50	16	95	85	70	20				
18	60	54	45	14	70	63	50	14	80	72	50	16	100	90	70	20	120	108	80	20
19	63	57	45	14	73.5	66.5	60	14	84	76	60	16	105	95	70	20				
20	66	60	45	14	77	70	60	14	88	80	60	16	110	100	80	20	132	120	90	20
21	69	63	45	16	80.5	73.5	70	16	92	84	70	16	115	105	80	20				
22	72	66	50	16	84	77	70	16	96	88	70	16	120	110	80	20				
23	75	69	50	16	87.5	80.5	75	16	100	92	75	20	125	115	90	20				
24	78	72	50	16	91	84	75	16	104	96	75	20	130	120	90	20	156	144	100	25
25	81	75	50	16	94.5	87.5	75	16	108	100	75	20	135	125	90	20	162	150	110	25
26	84	78	50	18	98	91	75	18	112	104	75	20	140	130	100	20				

Fuente: (Roydisa, 2012)

Para la selección de piñón también se determina un espacio mínimo para su ubicación con un valor de 92mm, así se selecciona un piñón de 21 dientes como muestra la tabla 26, además de considerar como material el nylon, para especificar que la tensión por flexión permisible sea mayor al esfuerzo calculado por flexión, el mismo que por medio de los siguientes cálculos permitirán su aprobación definitiva en la implementación dentro de la estructura de la mesa regulable, también como dato adicional se tiene la longitud de la cremallera de 120mm que es determinada por el intervalo de altura regulable.

Nomenclatura de las variables:

$D_{\text{Referencia}}$ = Espacio mínimo de referencia para ubicación del piñón

l = Longitud de la cremallera rectangular

z_p = Número de dientes seleccionado del piñón

m = Módulo del piñón o cremallera

D_p = Diámetro primitivo del piñón

p_d = Paso diametral del piñón o cremallera

n_p = Número de RPM de piñón

$n^\circ\text{pasos}$ = Número de pasos efectuados

$n^\circ\text{vueltas}_p$ = Número de vueltas del piñón

t = Tiempo empleado en el número de vueltas del piñón

n_p = Revoluciones por minuto del piñón

V_{ac} = Velocidad de avance de la cremallera en mm/s

a = Cabeza

b = Raíz

c = Espaciamiento

D_{RP} = Diámetro de la raíz

D_O = Diámetro exterior

h_t = Profundidad total

t = Espesor del diente

Datos:

material \rightarrow nylon

$D_{Referencia} = 92\text{mm}$

$l = 120\text{mm}$

$z_p = 21$

Geometría del piñón y cremallera

Cálculo aproximado del módulo del piñón y cremallera:

$$m = \frac{D_{Referencia}}{z} = \frac{92}{21} \text{mm} = 4\text{mm}$$

Cálculo del diámetro de paso:

$$D_p = m \times z = (4 \times 21)\text{mm} = 84\text{mm}$$

Cálculo del paso diametral:

$$p_d = \pi m = \pi \times 4 = 12,57\text{mm}$$

Determinación del número de RPM del piñón:

$$l = p_d \times n^\circ\text{pasos}$$

$$n^\circ\text{pasos} = \frac{l}{p_d} = \frac{120\text{mm}}{12,57\text{mm}} = 9,56 \text{ pasos}$$

$$n^\circ\text{pasos} = z_p \times n^\circ\text{vueltas}_p$$

$$n^\circ\text{vueltas}_p = \frac{n^\circ\text{pasos}}{z_p} = \frac{9,56}{21} = 0,45 \text{ Vueltas}$$

$$n_p = \frac{n^\circ\text{vueltas}_p}{t} = \frac{0,47\text{Vueltas}}{1,8\text{s}} \times \frac{2\pi\text{REV}}{1\text{Vuelta}} \times \frac{60\text{s}}{1\text{MIN}} = 95\text{RPM}$$

Velocidad de avance de la cremallera en mm/s:

$$V_{ac} = p_d z_p n_p = 12,57\text{mm} \times 21 \times 1,59\text{REVS} = 419\text{mm/s}$$

Cabeza, raíz y espaciamento:

Tabla 27

Fórmulas de características de dientes de los engranes

Característica	Símbolo	Sistema envolvente de profundidad total		
		Paso grueso ($p_d < 20$)	Paso fino ($p_d \geq 20$)	Sistema de módulo métrico
Cabeza	a	$\frac{1}{p_d}$	$\frac{1}{p_d}$	1,00m
Raíz	b	$\frac{1,25}{p_d}$	$\frac{1,200}{p_d + 0.002}$	1,25m
Espaciamento	c	$\frac{0,25}{p_d}$	$\frac{0,200}{p_d + 0.002}$	0,25m

Fuente: (Mott, 1995)

$$a = 1,00\text{m} = 1,00 \times 4 = 4\text{mm}$$

$$b = 1,25m = 1,25 \times 4 = 5\text{mm}$$

$$a = 0,25m = 0,25 \times 4 = 1\text{mm}$$

Diámetro exterior:

$$D_O = D_p + 2a = 84 + 2 \times 4 = 92\text{mm}$$

Diámetro de la raíz:

$$D_{RP} = D_p - 2b = 84 - 2 \times 5 = 74\text{mm}$$

Profundidad total:

$$h_t = a + b = 4 + 5 = 9\text{mm}$$

Profundidad de trabajo:

$$h_k = a + a = 1 + 1 = 2\text{mm}$$

Espesor del diente:

$$t = \frac{p_d}{2} = \frac{12,57}{2} = 6,285\text{mm}$$

La carga crítica en un diente se va a generar cuando la carga está en el punto más alto de contacto en un solo diente por lo que la fuerza tangencial a usar es el valor de la masa que va a ser levantada, la misma que generará un momento de tensión o flexión máximo en la base de los dientes del piñón y cremallera, para este caso mediante el uso de un software de diseño se estima un valor aproximado de la masa total.

Tabla 28

Valores de masa a levantar

Descripción	Cantidad	Masa g
Tapa superior mesa	1	3763,26
Tapa posterior mesa	1	5859,00
Guías mesa	2	521,48
Platos	3	1399,5
Vaso	1	39,63

CONTINÚA 

Brazo robótico	1	592,26
Otros	-	1000,00
Perfil parte superior para la mesa -	1	1775,59
Perfil parte superior para la mesa		1586,19
Perfil parte superior vertical para la mesa	1	1688,73
Platina triangular de soporte	2	1120,98
Platina vertical de soporte	1	338,68
Tapas perfil	3	54,03
Masa de los brazos de la persona sobre la mesa	-	16417,96
Total, masa		36157,29g = 36,16 Kg

Fuerza tangencial

$$W_t = \text{masa} \times g$$

$$W_t = \text{Fuerza tangencial vertical teórica. [N]}$$

$$\text{masa} = \text{Masa a elevar o trasladar. [kg]}$$

$$W_t = 36,16 \text{ Kg} \times 9,81 \frac{\text{m}}{\text{s}^2} = 354,73 \text{ N}$$

Cálculo de la tensión o esfuerzo flexionante en los dientes del piñón y cremallera

$$\sigma_t = \frac{W_t}{F_m J} \frac{K_a K_s K_m K_B}{K_v}$$

Donde:

K_a = factor de aplicación para la resistencia a la flexión

K_s = factor de tamaño para la resistencia a la flexión

K_m = factor de distribución de carga para la resistencia a la flexión

K_B = factor de espesor de la corona

K_v = factor de dinámica para la resistencia a la flexión

Factor aplicación, K_a

Para el factor aplicación K_a se utiliza la tabla 29 como referencia.

Tabla 29

Factores de aplicación sugeridos, K_a

Fuente de poder	Máquina que es impulsada			
	Uniforme	Choque ligero	Choque moderado	Choque pesado
Uniforme	1,00	1,25	1,50	1,75
Choque ligero	1,20	1,40	1,75	2,25
Choque moderado	1,30	1,70	2,00	2,75

Fuente: (Mott, 1995)

La fuente de poder a utilizar es solo la fuerza de la persona encargada de regular la variación de altura, la misma será constante, por lo que se usará un factor de aplicación uniforme por lo tanto $K_a=1$.

Factor de tamaño, K_s

El módulo métrico es menor a 5 lo que determina un factor de tamaño de 1.

Tabla 30

Factores de tamaño que se sugieren, K_s

Paso diametral, P_d	Módulo métrico, m	Factor de tamaño, K_s
≥ 5	≤ 5	1,00
4	6	1,05
3	8	1,15
2	12	1,25
1,25	20	1,40

Fuente: (Mott, 1995)

Factor de distribución de carga, K_m

Debido a varios factores que provocan una desalineación en el diente del piñón es necesario determinar un factor de carga específico. El valor del factor K_m está determinado por la ecuación:

$$K_m = 1,0 + C_{pf} + C_{ma}$$

Donde:

C_{pf} = Factor de proporción del piñón(Figura 42)

C_{ma} = Factor por alineamiento de engranado(Figura 42)

La figura 42 muestra el debido análisis para determinar C_{pf}

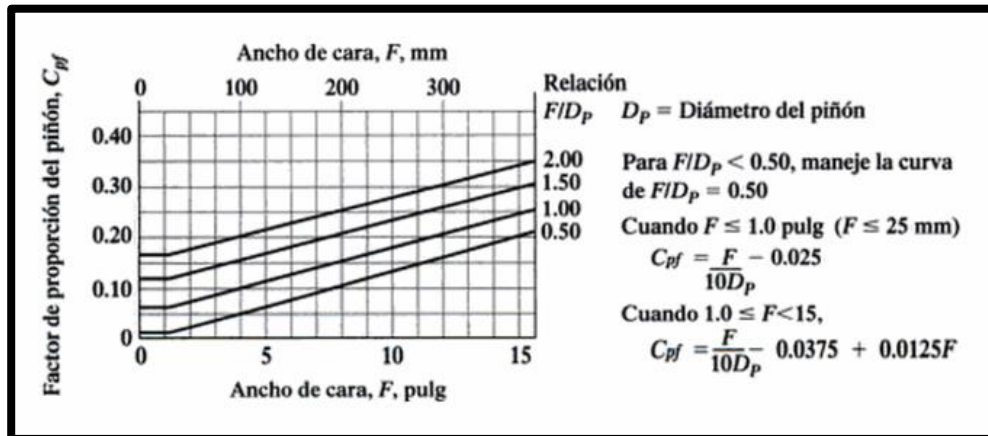


Figura 41. Factor de proporción del piñón, C_{pf}

Fuente: (Mott, 1995)

El ancho de cara es determinado por la ecuación:

$$F = 12 \times m = 12 \times 4 = 48\text{mm} \approx 50\text{mm}$$

Como $25\text{mm} < F < 360\text{mm}$

Entonces:

$$C_{pf} = \frac{F}{10D_p} - 0,0375 + 0,0125F = \frac{50}{10(84)} - 0,0375 + 0,0125(50) = 0,65$$

La figura 43 muestra el debido análisis para determinar C_{ma}

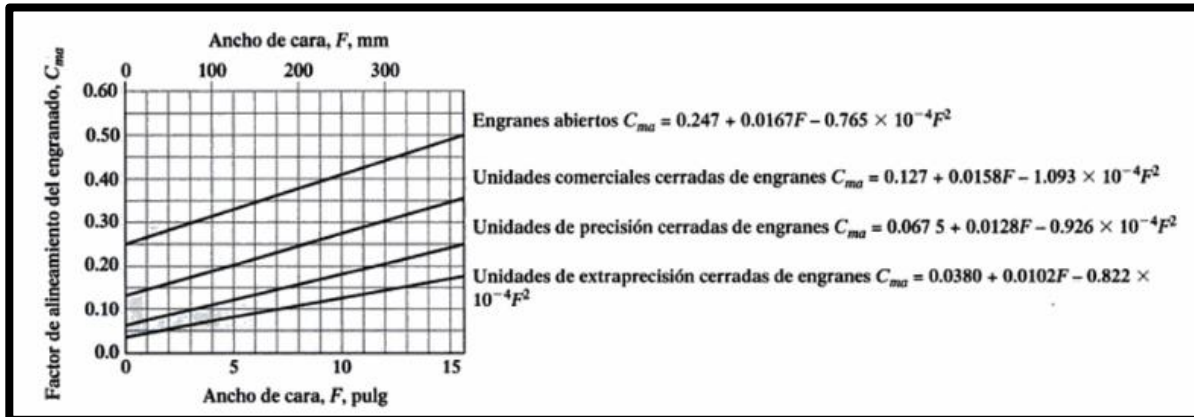


Figura 42. Factor de alineamiento del engranado, C_{ma}

Fuente: (Mott, 1995)

Para una unidad de precisión cerrada se tiene:

$$C_{ma} = 0,0675 + 0,0128F - 0,926 \times 10^{-4}F^2 = 0,476$$

$$K_m = 1,126$$

Factor de espesor de la corona, K_B

Las siguientes ecuaciones sirven para determinar una relación de apoyo, para estimar el grado de influencia del espesor de la corona, ya que si es muy delgada puede deformarse.

$$m_B = \frac{t_R}{h_t}$$

t_R = espesor de la corona

h_t = profundidad total de los dientes del engrane

$$t_R = 11$$

$$m_B = \frac{11}{9} = 1,2$$

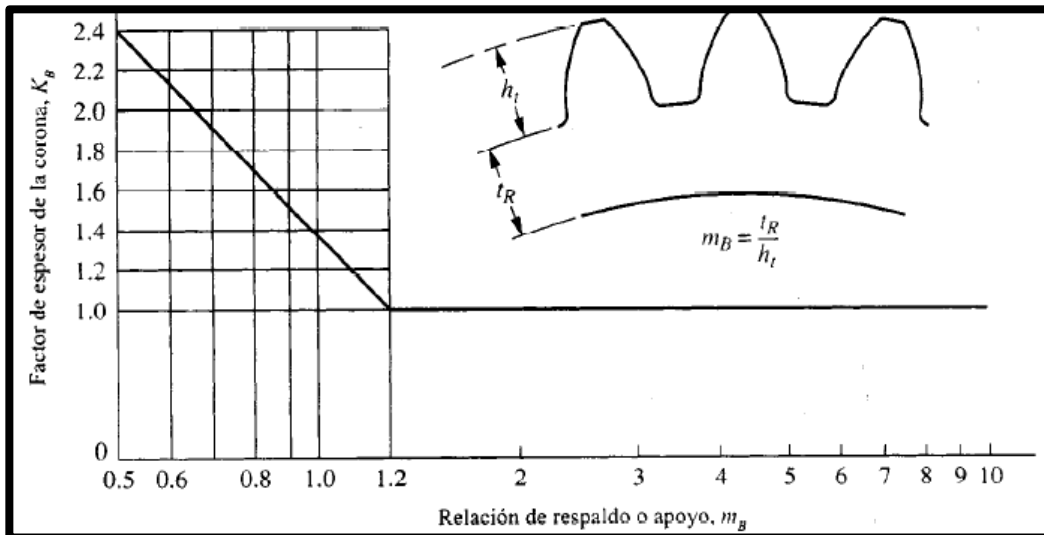


Figura 43. Factor de espesor de la corona, K_B

Fuente: (Mott, 1995)

El valor de $m_B > 1,2$, lo que garantiza la resistencia y rigidez de la corona, además de proveer un valor de $K_B = 1,0$.

Factor de dinámica, K_V

Se hace necesario calcular la velocidad de línea de paso:

$$v_1 = \frac{n_p D_p \pi}{60000} = \frac{94 \times 84 \times \pi}{60000} = 0,41 \frac{\text{m}}{\text{s}}$$

La figura 45 muestra el valor de K_V dado por el valor de La velocidad lineal calculada con valor de 0,41 m/s.

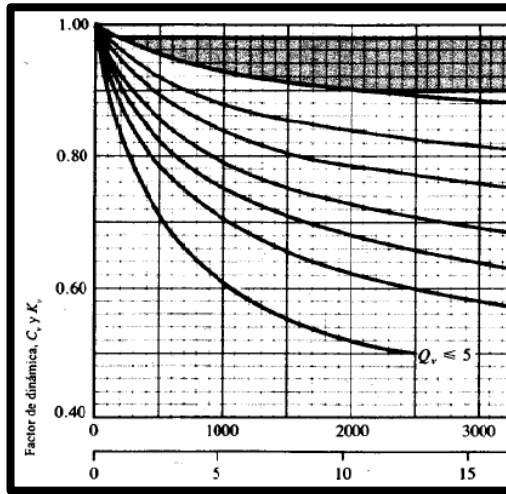


Figura 44. Factor de dinámica, C_V y K_V

Fuente: (Mott, 1995)

$$C_V = K_V = \frac{50}{50 + \sqrt{200v_1}} = \frac{50}{50 + \sqrt{200(0,41)}} = 0,85$$

Factor de geometría, J

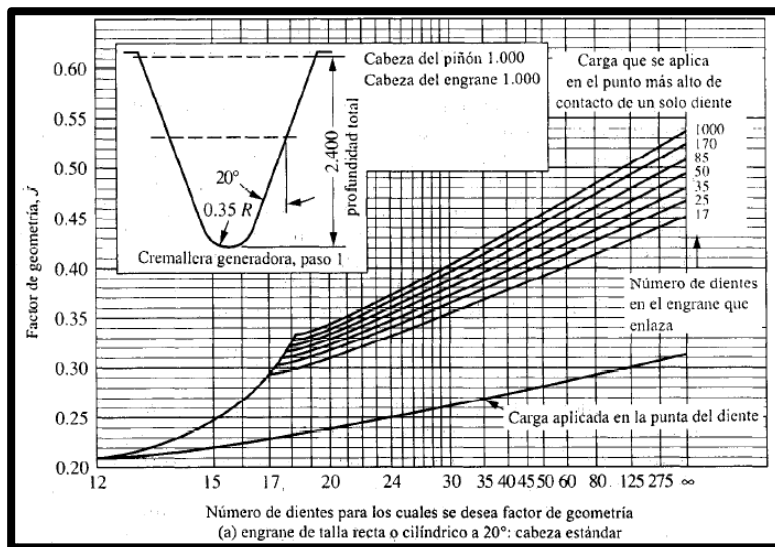


Figura 45. Factor de geometría J

Fuente: (Mott, 1995)

Para el piñón el factor de geometría es $J=0,33$ y de la cremallera el factor de geometría es $J=0,35$

Esfuerzo por flexión permisible, S_{at}

La tabla 31 muestra el esfuerzo por flexión permisible aproximado en dientes de engranes de plástico.

Tabla 31

Esfuerzo por flexión permisible aproximado

Material	Tensión por flexión permisible aproximado (Ksi, MPa)	
	Sin relleno	Relleno con vidrio
ABS	3 (21)	6 (41)
Acetal	5 (34)	7 (48)
Nylon	6 (41)	12 (83)
Polycarbonato	6 (41)	9 (62)
Polyester	3,5 (24)	8 (55)
Poliuretano	2,5 (17)	-

Fuente: (Mott, 1995)

El esfuerzo por tensión en el piñón es:

$$\sigma_{tp} = \frac{354,73}{50 \times 4 \times 0,33} \frac{1 \times 1 \times 1,126 \times 1}{0,85} = 7,12 \text{ MPa}$$

Identificando el valor de Y_N , para un valor estandarizado de 20000 horas de vida de diseño se tiene:

$$N_{cp} = 60 \ln_p q = 60 \times 20000 \times 94 \times 1 = 1,13 \times 10^8 \text{ ciclos}$$

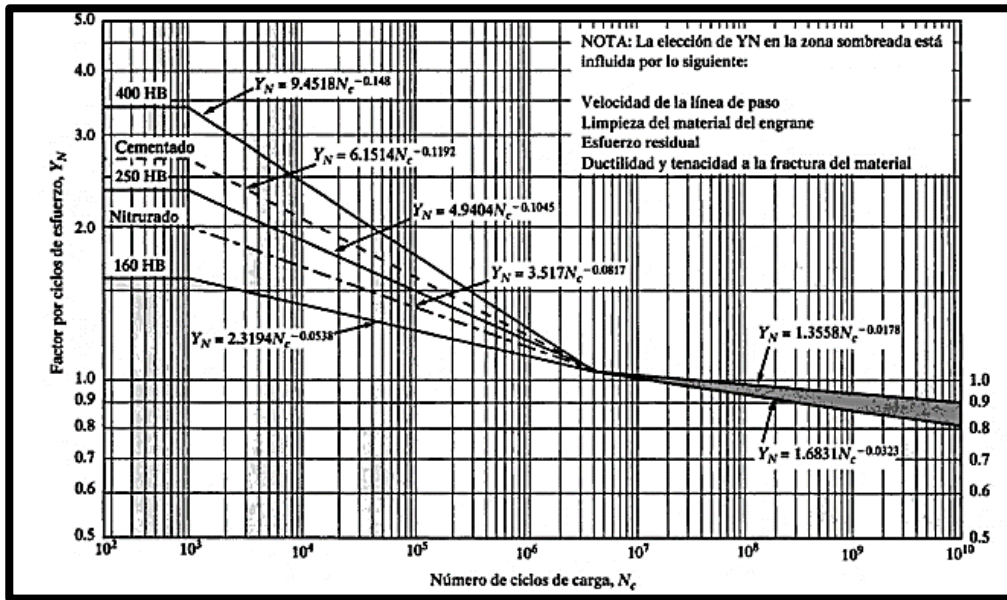


Figura 46. Factor de resistencia flexionante por ciclos de esfuerzo Y_N
Fuente: (Mott, 1995)

$$Y_N = 1,3558 N_{cp}^{-0,0323} = 1,3558 \times (1,13 \times 10^8)^{-0,0323} = 0,74$$

Factor de confiabilidad, K_R

Tabla 32

Factor de confiabilidad, K_R

Confiabilidad	K_R
0,90, una falla en 10	0,85
0,99, una falla en 100	1,00
0,999, una falla en 1000	1,25
0,9999, una falla en 10 000	1,50

Fuente: (Mott, 1995)

Asumiendo una confiabilidad de 99,9% el valor de $K_R = 1,25$ y un factor de seguridad

SF = 1,5 se tiene:

$$\frac{K_R(SF)}{Y_{Np}} S_t = \frac{1,25 \times 1,5}{0,74} (7,12) = 18,04 \text{ Mpa}$$

$$18,04 \text{ MPa} < S_{at}$$

$$18,04 \text{ MPa} < 41 \text{ Mpa}$$

Los factores en la ecuación de esfuerzo por tensión son los mismos para la cremallera con excepción del valor J de geometría entonces el esfuerzo en la cremallera puede calcularse como:

$$\sigma_{tc} = \sigma_{tp} \left(\frac{J_p}{J_c} \right) = 7,12 \times \left(\frac{0,33}{0,35} \right) = 6,71 \text{ MPa}$$

$$N_{cc} = 60 \times 20000 \times 76 \times 1 = 9,1 \times 10^7 \text{ ciclos}$$

Según la figura 47:

$$Y_N = 1.3558 N_{cp}^{-0,0323} = 1,3558 \times (9,1 \times 10^8)^{-0,0323} = 0,75$$

$$K_R = 1,00$$

$$SF = 1,5$$

$$\frac{K_R(SF)}{Y_{Nc}} S_t = \frac{1,25 \times 1,5}{0,75} (6,71 \text{ MPa}) = 16,78 \text{ MPa}$$

$$16,78 \text{ MPa} < S_{at}$$

$$16,8 \text{ MPa} < 41 \text{ MPa}$$

El material adecuado para el piñón y cremallera, que servirán para el mecanismo de regulación de una altura con un intervalo de 120mm y una masa total para ser levantada de 36,16kg es el nylon, pues el esfuerzo de tensión que se espera debido a la flexión en el piñón (7,18Mpa) es un valor menor al esfuerzo de flexión permisible (41Mpa). Y para la cremallera el esfuerzo de tensión que se espera debido a la flexión en el piñón es de (16,8Mpa) que es un valor menor al esfuerzo de flexión permisible con un valor de (41Mpa).

CAPÍTULO III

3. DESARROLLO DEL ALGORITMO DEL SISTEMA DE ALIMENTACIÓN

El sistema de alimentación cuenta con dos modos de activación, un modo manual y otro automático, modos que actúan independientemente mediante una aplicación en el celular, que ofrece comodidad para el usuario por su fácil manejo.

El sistema de pesaje inicia con el pesaje de los platos o seteo, si la selección es el modo manual el usuario puede elegir cualquiera de los tres platos de comida o vaso con los pulsadores ubicados en la parte inferior de la mesa, la selección puede variar entre sopa, arroz postre o bebida. Al elegir un plato de comida, el programa compara si contiene comida o se encuentra vacío por medio del sistema de pesaje, si hay la existencia de comida, el brazo toma el alimento con movimientos pre programados y se lo entrega al usuario.

Si la selección es el modo automático, el usuario puede elegir los platos, con la misma aplicación en el celular por comandos de voz, al igual que el modo manual, el programa realiza la comparación de pesos en los platos y el brazo toma la comida, pero se diferencia del modo manual en la entrega de la comida al usuario, pues actúa un algoritmo de visión artificial que detecta y realiza el seguimiento de la boca del usuario, mediante una cámara web colocada en la pinza del brazo robótico, para después detenerse por un leve momento hasta que el usuario deguste los alimentos.

Para el desarrollo del algoritmo del sistema de alimentación se utilizará como referencia los ejes de ubicación para el usuario como muestra la siguiente figura.

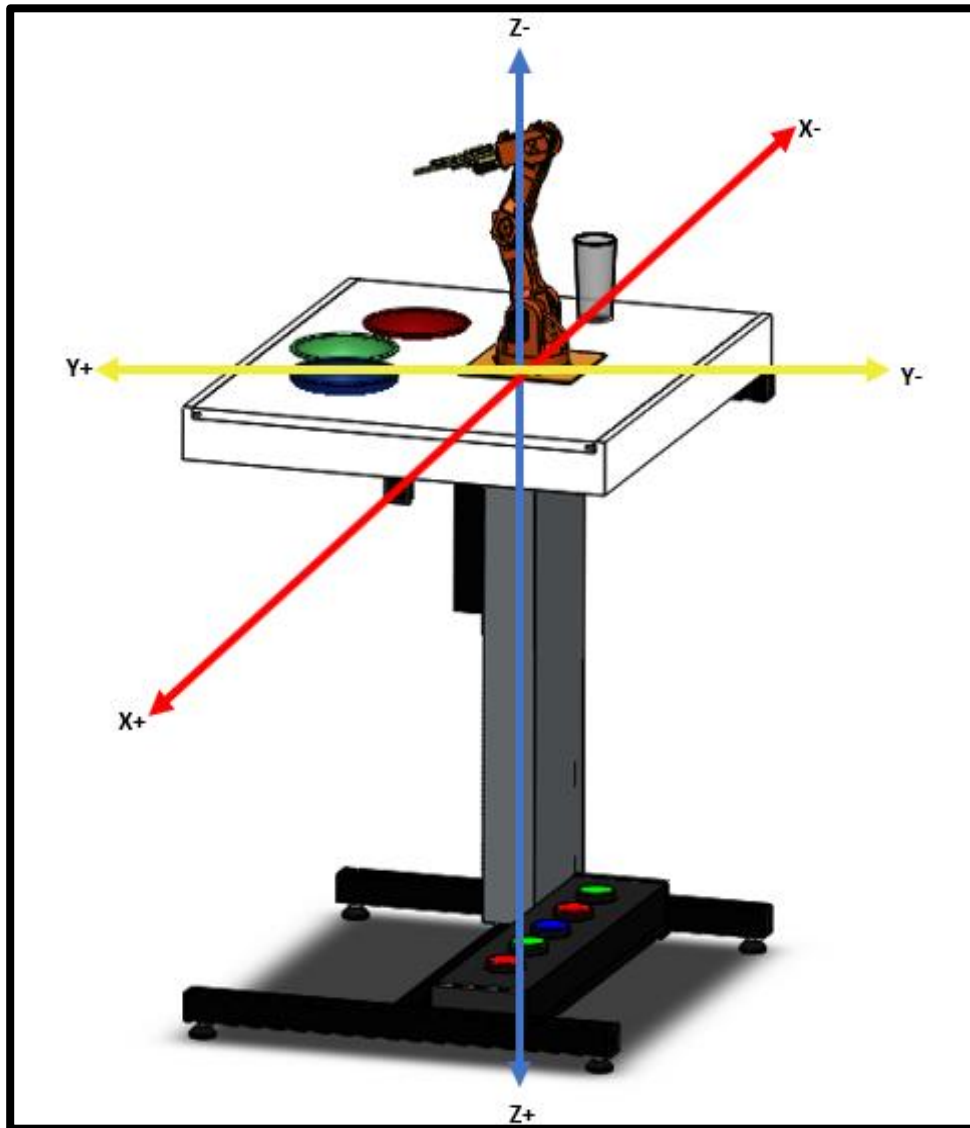


Figura 47. Ejes de ubicación

A continuación, el diagrama de flujo resume las etapas del proyecto.

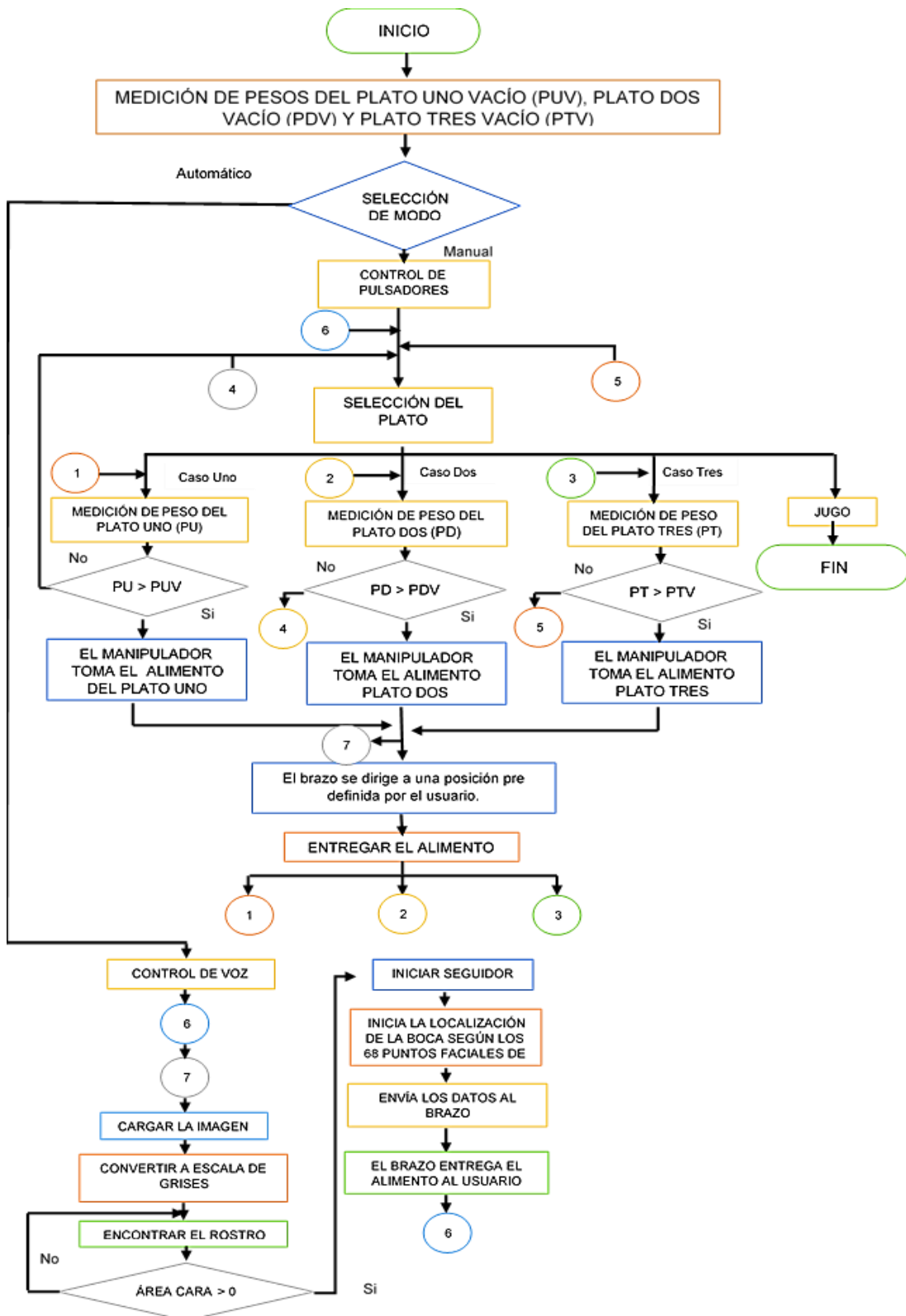


Figura 48. Diagrama de flujo del algoritmo

3.1. Algoritmo de detección y seguimiento de la boca en Python

Para lograr que el brazo robótico se posicione según la boca del usuario y la siga a una posición adecuada y conforme para el usuario, es necesario realizar una detección del rostro, por medio de un clasificador en cascada y un detector de rostros.

3.1.1. Bibliotecas

Para realizar el algoritmo se requiere de las siguientes bibliotecas:

NUMPY es un paquete de procesamiento numérico de Python y en este caso es necesario para el correcto funcionamiento de Open CV.

Open CV es un paquete de visión artificial de código abierto. Esta librería está basada en C y C++, compatible tanto Linux, como Windows (Aprendiendo Arduino, 2016). Tiene múltiples aplicaciones y varias de ellas en tiempo real usando procesadores con varios núcleos.

DLIB es una biblioteca multi-plataforma basada en lenguaje de C++. Este paquete es un conjunto de componentes de software independientes, contienen algoritmos de análisis de imágenes, compresión, obtención de partes relevantes entre otros.

IMUTILS tiene varias funciones de confort para realizar procesamiento de imágenes básicas y realizar una clasificación de contornos, detección de bordes entre otras; haciendo más fácil el uso de OpenCV (Python.org, 2017).

3.1.2. Clasificador en cascada

En la detección del rostro se usa la función Haar Cascade presente en la biblioteca de OpenCV que utiliza clasificadores pre-entrenados para la identificación de rostros. Todas las caras humanas comparten algunas propiedades similares. Estas semejanzas pueden ser emparejadas usando las siguientes características:

- La región del ojo es más oscura que las mejillas superiores.
- La región del puente de la nariz es más brillante que los ojos.
- La ubicación en que se encuentran los ojos, la boca y el puente de la nariz.
- Valor de gradientes orientados a las intensidades de los píxeles (Barba Guamán, 2015).

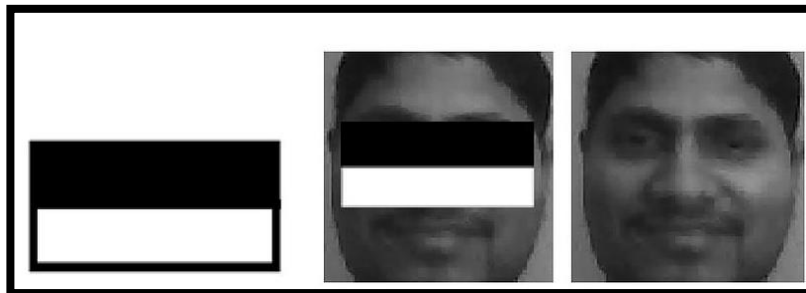


Figura 49. Región del ojo es más oscuro que las mejillas

Fuente: (Aprender Python, 2017)

3.1.3. Algoritmo de detección del rostro

Para el uso del algoritmo se carga el archivo “xml” en este caso el clasificador **frontal face** de la dirección `FaceCascade = cv2.CascadeClassifier ('/home/ pi/opencv-3.1.0/ data/ haarcascades/ haarcascade_frontalface_default.xml')`

Luego se carga la entrada de video, y es transformada al modo escala de grises, para un mejor procesamiento de la imagen, a continuación, la detección del rostro es realizada por medio de la función detectMultiScale, la misma que con una copia en escala de grises encuentra todos los rostros en cada trama.



Figura 50. Detección del rostro con Haar Cascade

3.1.3. Algoritmo de seguimiento del rostro

En los videos de tiempo real, no es necesario aplicar la detección de rostros en cada trama. En su lugar, se puede utilizar algoritmos de seguimiento pues resulta más sencillo enfocar el rostro durante el video. Cuando una persona realiza un movimiento el algoritmo de detección tiene que volver a cero para poder detectar la nueva posición del rostro; por lo que se necesita un algoritmo de seguimiento para complementar el algoritmo de detección. Con esto no sólo se mejora la velocidad de seguimiento mediante la eliminación de la necesidad de volver a detectar la cara en cada cuadro, también el sistema se vuelve más robusto, ya que después de detectar el rostro conserva las

características detectadas y se vuelve más resistentes a la rotación y cambios fotométricos (Wikipedia, 2017).

En este proyecto es necesario el seguimiento de un solo rostro (el del usuario) ya que si se realiza el seguimiento de más rostros puede generar errores al momento de enviar los datos al Arduino, lo cual provocaría errores en el movimiento del brazo.

Dlib permite seguir la posición de un objeto a medida que se mueve cuadro a cuadro en una secuencia de video. Este seguidor de correlación posee un cuadro delimitador del objeto que desea seguir en el marco de video actual. Los seguidores de correlación como su nombre lo indica funciona mediante la correlación de un conjunto de píxeles de un cuadro a otro (Martínez Andrade, 2013).

Para el seguimiento del rostro se utiliza el seguidor correlación Dlib, el mismo que extrae las características del rostro detectado, para posteriormente realizar el seguimiento en los cuadros posteriores.

```
import dlib
#Crear seguidor de correlacion
tracker = dlib.correlation_tracker()
#variable utilizada para el seguidor
trackingFace = 0
```

Figura 51. Biblioteca Dlib

Se dibuja un rectángulo en el área donde se ha detectado el rostro para lo cual primero se inicializan las variables requeridas en cero.

```
maxArea = 0
x = 0
y = 0
w = 0
h = 0
```

Figura 52. Inicialización de las variables

Donde:

- maxArea es el área del rectángulo donde se encuentra el rostro
- x es la coordenada inicial del eje horizontal
- y es la coordenada inicial del eje vertical
- w es la coordenada final del eje horizontal
- h es la coordenada final del eje vertical

El seguidor Dlib acepta solo números enteros por lo cual se transforman las variables x, y, w y h a enteros para iniciar el seguidor.

```
for (_x, _y, _w, _h) in faces:
    if _w*_h > maxArea:
        x = int(_x)
        y = int(_y)
        w = int(_w)
        h = int(_h)
        maxArea = w*h
```

Figura 53. Transformación de variables

Para saber si **Haar cascade** ha detectado uno más rostros es necesario el cálculo del área del rectángulo de los mismos, si es mayor a cero el detector funciona e inicia el seguidor. Fig.56.

```

#si no esta siguiendo nada el seguidor busca encontrar la region a seguir
if not trackingFace:

    #si encuentra una o mas caras inicia el rastreador
    if maxArea > 0 :

        #Inicializa el rastreador
        tracker.start_track(ImagenOrig,
                            dlib.rectangle( x,
                                            Y,
                                            x+w,
                                            y+h))

```

Figura 54. Inicio del seguidor

Donde:

start_track: Inicia el seguimiento

ImageOrig: Es la ventana en la cual se va a realizar el seguimiento

Dlib rectangle: es el cuadro donde está la región que se va a seguir.

Se establece una variable (trackingFace) para saber si el seguidor está siguiendo una cara en la cual será igual a 1. Después se actualiza el seguidor.

```

        trackingFace = 1

    if trackingFace:

        #actualizacion de la informacion y del seguimiento
        trackingQuality = tracker.update( ImagenOrig )

```

Figura 55. Actualización del seguidor

Donde:

Tracker.update es el comando que actualiza el seguidor en la ventana ImagenOrig

Posteriormente a la actualización del seguidor y dependiendo de la calidad de actualización (el nivel de confianza del seguidor para determinar si se sigue realizando

un seguimiento de la misma cara) se dibuja un rectángulo alrededor de la región indicada por el seguidor.

```
#si el seguimiento es aceptable determina la
#posicion del seguidor de la region y dibuja el
#rectangulo
if trackingQuality >= 8.75:
    tracked_position = tracker.get_position()

    t_x = int(tracked_position.left())
    t_y = int(tracked_position.top())
    t_w = int(tracked_position.width())
    t_h = int(tracked_position.height())

    cv2.rectangle(resultImage, (t_x, t_y),
                    (t_x + t_w , t_y + t_h),
                    rectangleColor ,2)
```

Figura 56. La calidad de seguidor (trackingQuality)

La calidad de seguidor (trackingQuality) se ha sincronizado en valores mayores a 8.75 con los cuales se obtiene un margen de error menores al 6%. Si estos valores son muy bajos se pueden tener más falsos positivos y si son valores muy altos se puede perder información del seguimiento.

Si se desea la detección y seguimiento de múltiples caras se debe hacer las siguientes modificaciones dentro del bucle principal:

Actualizar todos los seguidores de correlación y eliminar todos los seguidores que no se consideran fiables (por ejemplo, los que tengan demasiado movimiento)

Cada 10 fotogramas, se debe hacer lo siguiente:

- Usar el detector de rostros en el actual cuadro para encontrar todas las caras presentes en la pantalla.

- Para cada rostro encontrado se debe comprobar si existe un seguidor el cual tiene que estar en el punto central de la cara detectada y también dentro de la región del seguidor. Si no existe un seguidor, significa que se ha detectado una nueva cara y se tiene que empezar un nuevo seguidor a este rostro.

Utilizar la información de los rostros detectados (incluyendo las nuevas caras detectadas en el paso anterior) para dibujar los rectángulos delimitadores los cuales contienen las características de los rostros que se detectaron.

3.1.4. Seguimiento de la boca

La necesidad del usuario radica en posicionarse en un lugar confortable y preciso al momento de ingerir sus alimentos, para ello, también se ha implementado la detección y seguimiento de la boca. Se puede usar directamente un clasificador para la boca, pero debido a su alta tasa de error y muestra de muchos falsos positivos, primero se encuentra la posición de la cara para después encontrar la boca.

En la detección de la cara se usa un detector de Dlib, `shape_predictor_68_face_landmarks`, que es un modelo preformado de detección de puntos de referencia por medio del alineamiento de la cara en un milisegundo, con 68 puntos faciales, figura 59, que definen las regiones de la cara, como las esquinas de la boca, cejas, ojos, nariz y boca, utilizando partes de una matriz sencilla en Python. (CodesofInterest, 2017)

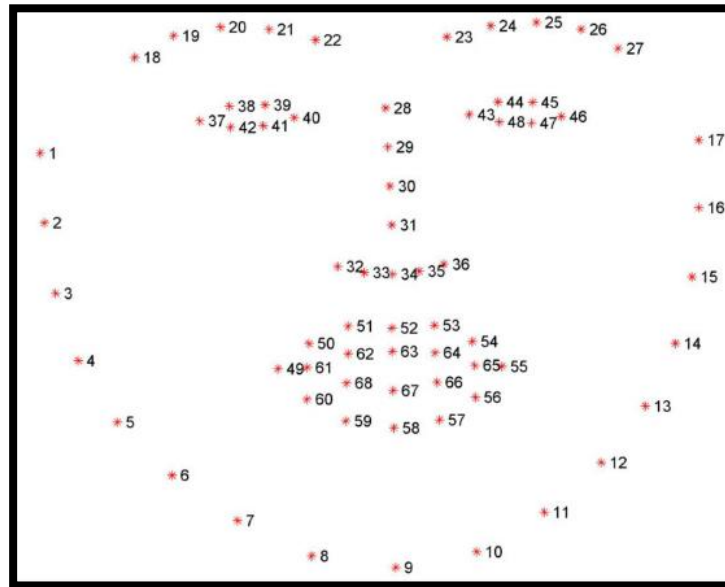


Figura 57. 68 puntos de coordenadas faciales
Fuente: (Marin, 2017)

El alineamiento facial es realizado por un conjunto de árboles de regresión, el cual consiste en tomar una imagen en forma de matriz numpy y la devuelve en una matriz de 68x2, cada fila está conformada con las coordenadas x, y de un punto de característica común en la imagen de entrada, es decir, contiene coordenadas para una característica facial particular, por ejemplo, la fila 49 contiene la coordenada del inicio de la boca. Por lo tanto, las regiones de la cara están divididas con los siguientes puntos de referencia faciales:

- La boca [49, 68].
- La ceja derecha [18, 22].
- La ceja izquierda [23, 27].
- El ojo derecho [37, 42].
- El ojo izquierdo [43, 48].

- La nariz [28, 36].
- La mandíbula [1, 17].

Teniendo en cuenta los puntos anteriores, se utiliza los de la boca, para la detección y seguimiento de la misma. Para acceder al algoritmo de seguimiento de los puntos faciales de la boca se introduce la dirección en la que se encuentra. Fig.60.

```
# crear los puntos faciales
predictor = dlib.shape_predictor
              ('/home/pi/Downloads/pi-drowsiness-detection/shape_predictor_68_face_landmarks.dat')
```

Figura 58. Dirección del algoritmo de seguimiento con los 68 puntos faciales

A continuación, se inicializa los índices mStart y mEnd, con los cuales se van a extraer las regiones de la boca del conjunto de los rasgos faciales:

```
(mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
```

Figura 59. Índices para los puntos de referencia faciales de la boca

Determina los puntos de referencia facial para la región de la boca, luego los convierte a coordenadas de referencia (x,y) a una matriz NumpyPy.

```
shape = predictor(resultImage, rect)
shape = face_utils.shape_to_np(shape)
```

Figura 60. Extracción de las regiones de la boca

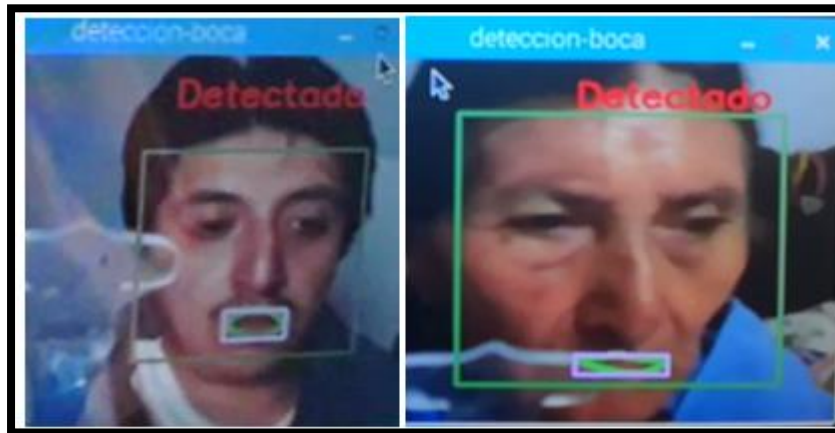


Figura 61. Detección de la boca con los 68 puntos faciales

Se determina los puntos de referencias faciales para cada región de la cara. Después se transforma los 68 puntos a una matriz NumPy.

Con la función **boundingRect**, figura 64, se obtienen las dimensiones del rectángulo de la matriz donde se encuentra la boca. Después se procede a calcular el área y los puntos medios de la boca, para proceder al envío de datos para que el manipulador siga la boca del usuario.

```
(xb, yb, wb, hb) = cv2.boundingRect(np.array([shape[mStart:mEnd]]))
Areab = wb*hb
pxm=xb+wb/2
pym=yb+hb/2
```

Figura 62. Cálculo de las dimensiones de la región de la boca

3.2. Implementación de los sensores y su programación

3.2.1. Acondicionamiento de los sensores

El sistema de peso para los platos, consta de 3 galgas extensiométricas las cuales están acondicionadas cada una a un amplificador AD620 el cual aumenta la diferencia

de potencial con una ganancia de 100. Las señales obtenidas son conectadas a las entradas analógicas A0, A1 y A2 de los sensores y las envía al Arduino; en donde se recibe la señal análoga y compara el peso de un plato verificando si está vacío o si está lleno.

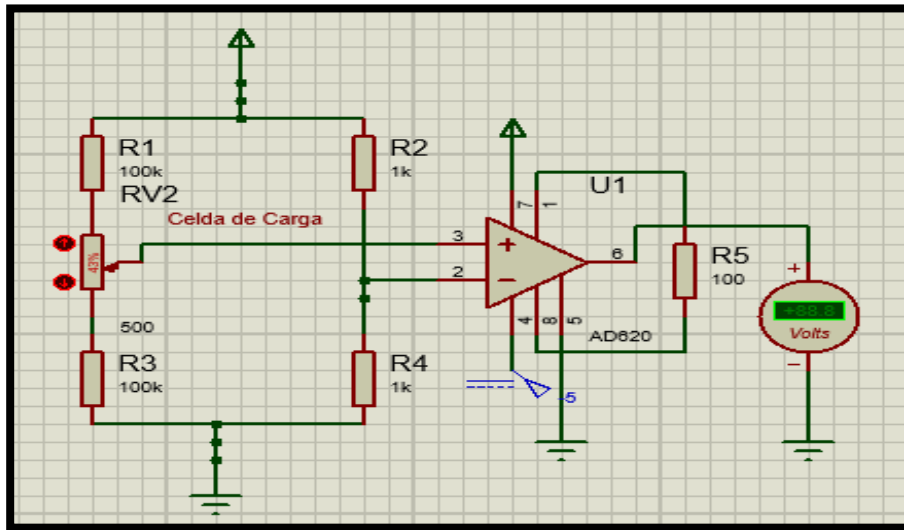


Figura 63. Circuito de las celdas de carga



Figura 64. Galgas extensiométricas

3.2.3. Funciones de mediciones de peso

Se declara las variables `xs`, `xs1` y `xs2` en las que se lee las señales analógicas provenientes de los pines 0, 1 y 2.

```
xs = analogRead(0);  
xs1 = analogRead(1);  
xs2 = analogRead(2);
```

Figura 65. Lectura de sensores

Se crean tres funciones: **peso**, **pesouno** y **pesodos** en las que cada una lee la señal enviada por cada sensor. Después se realiza una lectura de 100 señales y se obtiene un promedio de los valores leídos por cada sensor dado que los sensores no son muy precisos y tienen una pequeña variación en sus mediciones. El promedio de los valores leídos por los sensores ayuda obtener mediciones más exactas.

```
xs = analogRead(0);  
for(int i = 0; i < 100; i ++)  
{  
    xs=xs+analogRead(0);  
    delay (7);  
}  
xs=xs/100;
```

Figura 66. Promedio de voltaje leído de la Celda de Carga

El Arduino Mega no posee la capacidad para trabajar con señales analógicas, de tal manera que necesita transformar las señales analógicas en señales digitales.

El valor de 0 voltios en la señal analógica, en la señal digital es 0 mientras que el valor de 5V analógico en señal digital es 1024 (Picuino, 2017). Por lo cual para todo valor analógico intermedio se expresa con un valor entre 0 y 1024.

Entonces la conversión para el voltaje en mV es:

$$V_{out} = V_{ADC} * \frac{5000}{1024}$$

```
v=1000*5*xs/1024;//voltaje promedio mv
```

Figura 67. Convertidor ADC

3.3. Programación para el control del brazo robótico

El kit de brazo robótico cuenta con una shield, en donde se conecta los servomotores del manipulador. Esta shield permite hacer una lectura más rápida de los pines con lo cual se ofrece mayor flexibilidad en el control de los servomotores. Para generar los movimientos del brazo robótico se lo hace a través de la biblioteca **Braccio**. Es así que primero se incluye esta biblioteca conjuntamente con la de **Servo** en el encabezado del programa.

```
#include <Braccio.h>  
#include <Servo.h>
```

Figura 68. Biblioteca para el control del brazo robótico

Después se declara los servos con el nombre de las partes del manipulador.

```
Servo base;  
Servo shoulder;  
Servo elbow;  
Servo wrist_rot;  
Servo wrist_ver;  
Servo gripper;
```

Figura 69. Declaración de los servos

Para inicializar la biblioteca se lo hace a través del comando **Braccio.begin()** en el **void setup ()** del programa. Ahora ya se pueden enviar las posiciones que requiere el brazo robótico.

```
Braccio.ServoMovement (step delay, M1, M2, M3, M4, M5, M6);  
  
Braccio.ServoMovement(20, 90, 140, 40, 0, 90, 50);
```

Figura 70. Comando para mover los servos

Donde, los ángulos máximos y mínimos vienen proporcionados por la biblioteca Braccio es así que:

- Step delay es el tiempo de espera en milisegundos entre movimientos de cada servo
- M1 genera el movimiento de 0 a 180 del servo base
- M2 genera el movimiento de 15 a 165 del servo shoulder
- M3 genera el movimiento de 0 a 180 del servo elbow
- M4 genera el movimiento de 0 a 180 del servo wrist_rot
- M5 genera el movimiento de 0 a 180 del servo wrist_ver
- M6 genera el movimiento de 10 a 73 del servo gripper

3.3.1. Funciones manual y automática

La función manual es la que controla el brazo robótico a través de los pulsadores y la automática controla al manipulador por medio de comandos por voz. Para activar una de estas dos opciones la aplicación “**SISTEMA DE ALIMENTACIÓN ARKAZ 2.0**”, compatible con el Sistema Operativo Android, comunica el celular del usuario con el Arduino por medio del módulo bluetooth HC – 05 que se encuentra conectado en el puerto serial 3 del Arduino Mega (pines 14 y 15).

La función manual tiene 5 pulsadores: siendo para la sopa (pin 53), el arroz (pin 48), el postre (pin 34), la bebida (52) y un paro general de finalización de cada proceso (pin 50). Cuando se presiona los pulsadores de sopa, arroz o postre indistintamente, el algoritmo llama a la función **cuchara** en la que el brazo robótico toma la cuchara del porta utensilio y se coloca en la posición de **home**, mientras el sensor mide el peso del plato seleccionado con las funciones de medición peso.



Figura 71. Función Home

Se establece un lazo do while donde el brazo recoge la comida del plato seleccionado y la entrega al usuario hasta que el valor de la medición que se hizo al principio del sistema y la última se igualen, es ahí en donde se interrumpe el lazo y el brazo deja de tomar el alimento para después con la función **devC** el brazo devuelva a la cuchara al porta utensilio, regresando a la posición de **home** y a la espera de la siguiente orden del usuario.

```
    if(pulsador_sopa== HIGH)
    {
        cuchara();

    do
    {
        v1=pesouno (xsl);
        v15=v1+25;
        if (v1>v15)
        {
            aux7=1;
            platoDos();
        }
        if( digitalRead(pare))
        {
            break;
            devC();
            home();
            delay(15);
        }
    }while (v1>v15) ;
```

Figura 72. Programación manual para la sopa

Para los platos de postre y arroz el brazo robótico tomará 5 cucharadas de cada área del plato, para tomar la mayor cantidad de comida del plato.

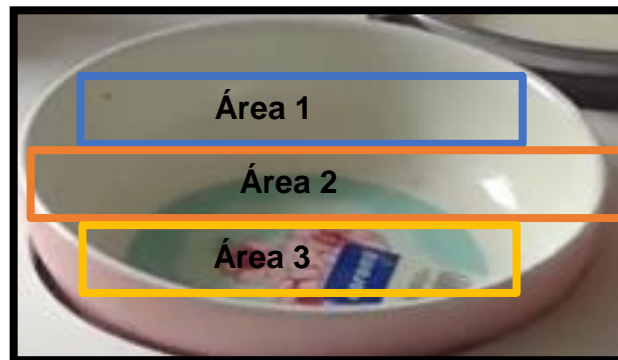


Figura 73. Áreas del plato

La función automática funciona de forma similar a la **manual** con la diferencia que es activada por voz en el celular y después de recoger la comida del plato se proceda a llamar a la función **comunicación** y el brazo puede encontrar y seguir la boca del usuario.

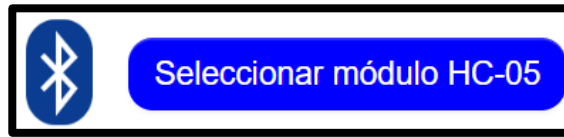
3.3.2. Aplicación “SISTEMA DE ALIMENTACIÓN ARKAZ 2.0”

La aplicación se desarrollada en APP Inventor, la misma que es compatible con el Sistema Operativo Android, y sirve para la activación manual y automática del sistema de alimentación.

Interfaz gráfica y botones de activación

La interfaz gráfica sirve para que el usuario pueda interactuar con el sistema de alimentación, para ello se crea un Screen1 o pantalla que contiene los elementos:

Botón **Seleccionar módulo HC-05**, por medio de este botón se selecciona el módulo bluetooth con el que se va a emparejar el dispositivo móvil.



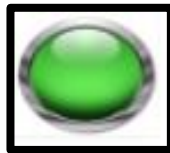
Botón modo de activación **Manual**, se selecciona la activación por pulsadores del sistema de alimentación.



Botón modo de activación **Automático**, se selecciona la activación por voz del sistema de alimentación.



Botón **Reset**, resetea los modos de activación permitiendo activarlos nuevamente.



Botón **Voz**, permite que el usuario por medio de comandos de voz active el brazo robótico para la alimentación del usuario.



Botón **Repetir comando**, ayuda a que el usuario verifique si el comando de voz fue el correcto.



Botón **CERRAR APP**, finaliza y cierra la aplicación del sistema de alimentación.



Generación de los bloques de programación

El siguiente bloque, fig. 76, abre la pantalla inicial y por medio de la función text to speech el usuario puede interactuar con la aplicación por medio de un mensaje de bienvenida establecido en la programación, si el bluetooth del dispositivo está apagado da un mensaje para encenderlo.

```
when Screen1.Initialize
do
  set DisposiciónVertical1.Visible to true
  set DisposiciónTabular1.Visible to true
  set DisposiciónTabular2.Visible to true
  call TextToSpeech1.Speak
  message "BIENVENIDO AL SISTEMA DE ALIMENTACION ARKAZ 2.0"
  if not BluetoothClient1.Enabled
  then
    call TextToSpeech1.Speak
    message "SALIR DE LA APLICACIÓN Y ACTIVAR SU BLUETOOTH"
    set Lp_Seleccionar.Text to "ACTIVAR BLUETOOTH"
```

Figura 74. Bloque de verificación de activación del bluetooth del dispositivo móvil

Por medio del botón **Seleccionar módulo HC-05**, se muestran los dispositivos bluetooth disponibles para la vinculación.

```

when Lp_Seleccionar .BeforePicking
do
  call TextToSpeech1 .Speak
  message " SELECCIONAR MÓDULO "
  set Lp_Seleccionar . Elements to BluetoothClient1 . AddressesAndNames
  set BLUETOOTH_DESACTIVADO . Text to " Seleccionar módulo "

```

Figura 75. Bloque de presentación de los dispositivos bluetooth disponibles
 Aquí, se selecciona el módulo bluetooth HC-05, y también se habilitan los botones manual y automático de la aplicación y si no existe vinculación de dispositivos repite un mensaje de error.

```

when Lp_Seleccionar .AfterPicking
do
  set Lp_Seleccionar . Selection to call BluetoothClient1 .Connect
  address Lp_Seleccionar . Selection
  set BLUETOOTH_DESACTIVADO . Text to " Activar modo "
  call TextToSpeech1 .Speak
  message " ACTIVAR MODO MANUAL O AUTOMÁTICO "
  set MANUAL . Enabled to true
  set AUTOMÁTICO . Enabled to true
  set Botón_manual . Enabled to false
  set Boton_voz . Enabled to false
  set Salir . Enabled to true
  set Lp_Seleccionar . Text to Lp_Seleccionar . Selection
  if not BluetoothClient1 . IsConnected
  then
    call TextToSpeech1 .Speak
    message " INTENTAR OTRA VEZ "
    set BLUETOOTH_DESACTIVADO . Text to " Emparejar "

```

Figura 76. Bloque de selección del módulo bluetooth a utilizar

Activado el botón **Manual**, el dispositivo móvil repite un mensaje de confirmación, y envía un carácter para el posterior accionamiento de los pulsadores.Fig.79.

```
when MANUAL .Click
do
  call TextToSpeech1 .Speak
  message "MODO MANUAL ACTIVADO"
  call BluetoothClient1 .SendText
  text "m"
  set Botón_manual .Enabled to false
  set MANUAL .Text to "Activado"
  set AUTOMÁTICO .Enabled to true
```

Figura 77. Bloque de activación del botón manual

Activado el botón **Automático**, el dispositivo móvil repite un mensaje de confirmación y envía un carácter para el posterior accionamiento por comandos de voz.Fig.80.

```
when AUTOMÁTICO .Click
do
  call TextToSpeech1 .Speak
  message "MODO AUTOMÁTICO ACTIVADO"
  call BluetoothClient1 .SendText
  text "a"
  set MANUAL .Enabled to false
  set MANUAL .Text to "Desactivado"
  set Boton_voz .Enabled to true
  set AUTOMÁTICO .Text to "Activado"
  set Botón_manual .Enabled to true
```

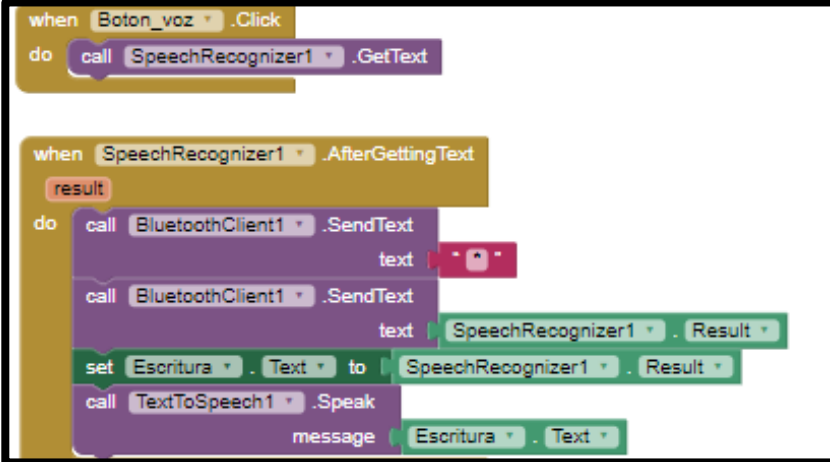
Figura 78. Bloque de activación automático

El siguiente bloque, muestra el reseteo de los modos de activación manual y automática, por medio del botón **Reset**.

```
when Botón_manual .Click
do
  set MANUAL .Enabled to true
  set MANUAL .Text to "MANUAL"
  set AUTOMÁTICO .Text to "AUTOMÁTICO"
```

Figura 79. Bloque de reseteo

Por medio del botón **Voz** se habilita el envío de comandos por voz, y la función speech recognizer permite el reconocimiento de los comandos para su posterior envío.



```
when Boton_voz .Click
do call SpeechRecognizer1 .GetText

when SpeechRecognizer1 .AfterGettingText
result
do call BluetoothClient1 .SendText
   text " "
   call BluetoothClient1 .SendText
      text SpeechRecognizer1 . Result
   set Escritura . Text to SpeechRecognizer1 . Result
   call TextToSpeech1 .Speak
      message Escritura . Text
```

Figura 80. Bloque que muestra el envío de comandos por voz

Los comandos a usar son:

- **Sopa:** Con este comando el brazo robótico se dirige al plato de sopa que se encuentra en el medio
- **Segunda:** Mediante este comando el manipulador robótico se dirige al plato fuerte que es el plato del extremo derecho.
- **Fruta:** Mediante este comando el robot se dirige al plato del extremo izquierdo.
- **Agua:** Con este comando el brazo toma el vaso y se lo entrega al usuario para que pueda beber y regresa al brazo a su lugar

Cuando el botón **Salir** es accionado, este genera la finalización de la aplicación, y el dispositivo móvil da un mensaje de confirmación.Fig.83.



Figura 81. Bloque para cerrar la aplicación

3.3.3. Funciones cuchara, devC

Son funciones pre programadas para recoger la cuchara del porta utensilio cuando el usuario ha seleccionado un plato y para dejarla en el porta utensilio después que haya terminado de comer el usuario.



Figura 82. Función cuchara

3.3.4. Funciones platoUno, platoDos, post y vaso

Son funciones pre programadas para tomar el alimento del plato seleccionado por el usuario. La función vaso entrega la bebida al usuario.



Figura 83. Activación de la función vaso

3.3.5. Función ik_arm

Para realizar el correcto posicionamiento de la muñeca o efector final del brazo robótico es necesario obtener su cinemática inversa, para lo cual se usa el método de transformaciones geométricas con el fin determinar el ángulo al que debe girar cada uno de los servomotores.

En este proyecto para determinar los ángulos de cada articulación los mismos que permitan al manipulador robótico tomar la posición requerida según la coordenada se debe realizar los cálculos primero sobre el plano YZ del brazo, de esta forma los cálculos se realizaran en el plano para mayor facilidad y no en la forma espacial de los ejes XYZ.

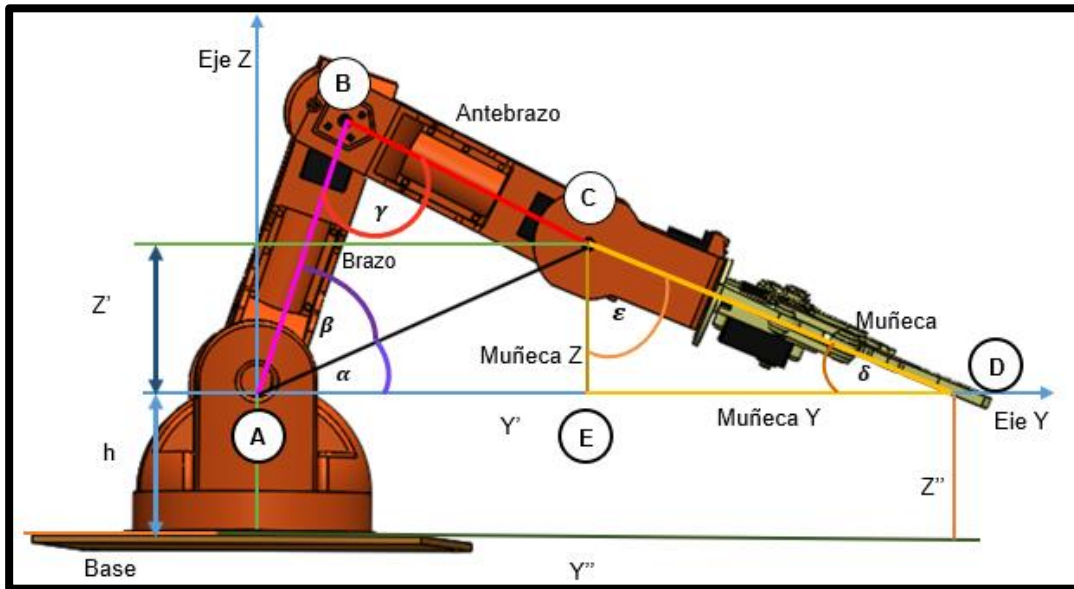


Figura 84. Componentes geométricos de la cinemática inversa

Para facilitar el cálculo al brazo se lo ha dividido en tres vectores (Figura 86): **Brazo**, **Antebrazo** y **Muñeca**. Al realizar el análisis se procede a calcular el ángulo δ el mismo que es impuesto por el usuario, este ángulo es en el que se apoya el brazo en la coordenada espacial establecida y servirá de base para calcular el resto de ángulos. A continuación, se procede a descomponer el vector **Muñeca**:

$$\text{Muñeca Y} = \text{Muñeca} * \cos(\delta)$$

$$\text{Muñeca Z} = \text{Muñeca} * \sin(\delta)$$

Primero se debe calcular Y' y Z' para después encontrar la diagonal AC y el ángulo α :

$$Y' = Y'' - \text{Muñeca Y}$$

$$Z' = \text{Muñeca Z} + Z''$$

$$AC = \sqrt{(Y')^2 + (Z')^2}$$

$$\text{tang}(\alpha) = \frac{Z'}{Y'}$$

$$\alpha = \text{tang}^{-1}\left(\frac{Z'}{Y'}\right)$$

Mediante la ley de cosenos se calcula los ángulos β y γ :

$$\cos(\beta) = \frac{\text{Brazo}^2 - \text{Antebrazo}^2 + AC^2}{2 * \text{Brazo} * AC}$$

$$\cos(\gamma) = \frac{\text{Brazo}^2 + \text{Antebrazo}^2 - AC^2}{2 * \text{Brazo} * \text{Antebrazo}}$$

Después de encontrar β y γ ya se puede determinar a qué ángulos pueden girar el antebrazo:

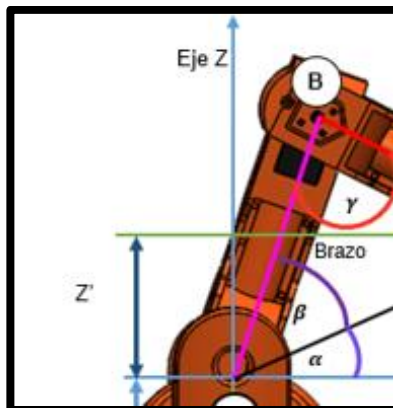


Figura 85. Angulo del brazo y antebrazo

$$\text{AnguloBrazo} = 90^\circ + \beta + \alpha$$

$$\text{AnguloAntebrazo} = \gamma$$

Para el cálculo de ε se localiza a los triángulos ABC, CDE y ACE y después mediante ángulos complementarios se determina ε :

$$\varepsilon = \varepsilon' + \varepsilon'' + \varepsilon'''$$

En el triángulo ABC se obtiene la siguiente ecuación

$$\varepsilon' = 180^\circ - \beta - \gamma$$

En el triángulo ACE se tiene la siguiente ecuación

$$\varepsilon'' = 180 - 90^\circ - \alpha$$

$$\varepsilon'' = 90^\circ - \alpha$$

En el triángulo CDE se tiene la siguiente ecuación

$$\varepsilon''' = 180 - 90^\circ - \delta$$

$$\varepsilon''' = 90^\circ - \delta$$

Para el cálculo de ε se localiza a los triángulos ABC, CDE y ACE y después mediante la fórmula de los ángulos complementarios se determina ε :

$$\varepsilon = 180^\circ - \beta - \gamma + 90^\circ - \alpha + 90^\circ - \delta$$

Para determinar el ángulo de giro de la base con el cual el manipulador se podrá ya finalmente ubicar en el plano X, Y, Z, se debe conocer el ángulo entre X y Y para obtener su posición espacial (Cruz, 2007)

$$\text{Angulo de giro} = \tan^{-1}\left(\frac{x}{y}\right)$$

3.3.6. Función para la comunicación

Esta función es parte importante del modo automático ya que es la encargada de establecer la comunicación entre Arduino y Python para que el brazo robótico pueda seguir a la boca y entregar el alimento al usuario. Después que el manipulador ha tomado el alimento del plato; el Arduino envía un carácter para activar la detección y seguimiento de la boca en Python y este a la vez empieza a enviar caracteres dependiendo de la posición de la persona.

Posiciones que toma la boca en la imagen

Para conocer la posición de la boca se inicia con la captura de la imagen original, como es necesario un mayor aceleramiento en el procesamiento de la imagen se hace una copia de la original y se redimensiona con valores de 320x240 pixeles.

En la figura 88 se observa la imagen de referencia con los valores asumidos, sus respectivos ejes de localización de la boca, y su división en 4 cuadrantes.

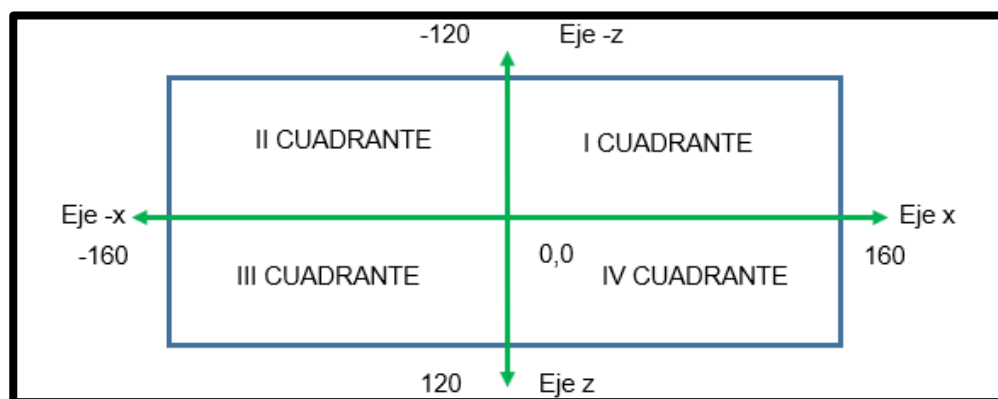


Figura 86. Dimensiones de la imagen

Básicamente existe un área de delimitación de la boca, si el usuario se mueve fuera del área del centro de la imagen como muestra la figura 89, el brazo robótico empezara el seguimiento hasta ubicarse dentro del centro del área, es decir si el usuario está ubicado en un sector con coordenadas P (60,-100), por medio de la comunicación serial ya establecida, se enviará los caracteres de ubicación respectivos para posicionar al brazo robótico al centro de la imagen.

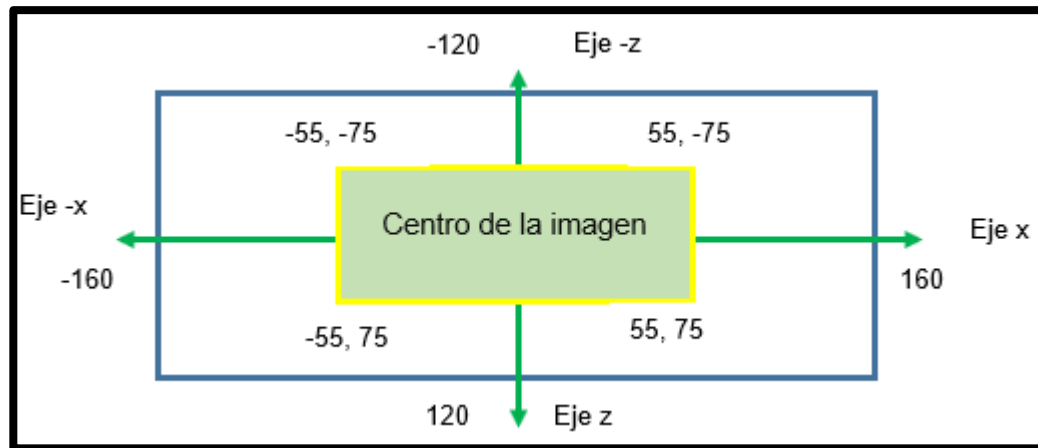


Figura 87. Área del centro de la imagen

Para el movimiento en el eje y, se toma la longitud de la horizontal del rostro (t_w), el brazo robótico se acercará al usuario hasta que la dimensión sea mayor a 188, es aquí en donde termina el seguimiento y la comunicación por el envío de un carácter al Arduino, se inhibe el movimiento del brazo robótico por un tiempo de masticación de por lo menos 30 segundos por bocado, hasta que el usuario deguste una cucharada de alimento, como muestra la figura 90.



Figura 88. Dimensión de la horizontal del rostro

3.3.7. Calibración del sistema para el modo automático

Para el reconocimiento de la boca es necesario que la cámara pueda enfocar de manera correcta la cara del usuario. Para realizar un buen enfoque se debe tener en cuenta el tamaño del sensor, la distancia focal y el ángulo de visión de la cámara (Bustamante, 2016).

Para obtener la expresión matemática que relaciona el ángulo de visión con la distancia focal y el tamaño del sensor es necesario tomar en cuenta el objeto de mayor tamaño H , generando una imagen de tamaño h , para una lente de distancia focal f . Fig.91.

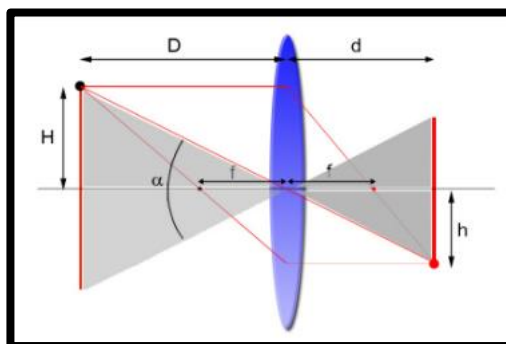


Figura 89. Diagrama del ángulo de visión, la distancia focal y el tamaño del sensor

Fuente: (Foto igual, 2017)

De la figura 91 se puede deducir el ángulo de visión (α):

$$\tan \frac{\alpha}{2} = \frac{H}{D}$$

$$\alpha = 2 \arctan \frac{H}{D}$$

Dado que los triángulos de la figura 91 son semejantes, el ángulo (α) es el mismo en los dos triángulos :

$$\alpha = 2 \arctan \frac{h}{d}$$

Para facilitar los cálculos se toma en cuenta la siguiente figura para determinar de manera más sencilla el ángulo en función de la distancia focal:

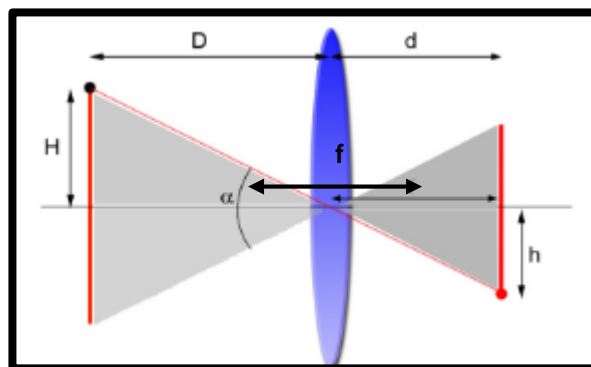


Figura 90. Gráfica del ángulo en función de la distancia focal

Fuente: (Foto igual, 2017)

La ecuación del ángulo de visión se da en función de la distancia focal f y el tamaño total del sensor S . Es así que el tamaño del sensor es $S = 2h$.

$$\alpha = 2\arctan \frac{S}{2f}$$

Para distancias de enfoque pequeñas (en este caso el rostro del usuario), la fórmula es:

$$\alpha = 2\arctan \frac{S}{2f(m + 1)}$$

Donde m es el factor de magnificación.

En este proyecto se cuenta con un sensor de 36 mm de tamaño y una distancia focal de 50 mm para una distancia de enfoque de 55 cm. El factor de magnificación es:

$$m = \frac{f}{D - f} = \frac{50}{550 - 50} = 0,1$$

El ángulo de visión es:

$$\alpha = 2\arctan \frac{S}{2f(m + 1)} = \alpha = 2\arctan \frac{36}{2(50)(0,1 + 1)} = 36^\circ$$

Para obtener el valor de H se despeja de la fórmula del ángulo visual (α):

$$\tan \frac{\alpha}{2} = \frac{H}{D}$$

$$D * \tan \frac{\alpha}{2} = H$$

$$H = 550 * \tan \frac{36}{2} = 178,7\text{mm}$$

Tomando en cuenta el resultado de $H= 178,7\text{mm}$, se puede calibrar el valor de la coordenada en z al inicio del sistema, para las diferentes estaturas de los usuarios y con ello realizar un buen enfoque para poder detectar y realizar el seguimiento de la boca del usuario.

Tabla 33

Calibración de la coordenada z

Coordenada z	Estatura
40	1,80 m a 1,63 m
60	1,63 m a 1,46 m
80	1,46 m a 1,29 m

CAPÍTULO IV

4. IMPLEMENTACIÓN PRUEBAS Y RESULTADOS

4.1. Construcción e implementación de la estructura mecánica de soporte

La estructura mecánica de soporte cuenta con los elementos que se muestran en la figura 93.



Figura 91. Partes de la estructura mecánica de soporte

Tabla 34

Elementos de la estructura mecánica de soporte

Número	Elementos
1	Área de trabajo o zona de alimentación
2	Mecanismo de regulación
3	Base de accionamiento
4	Cables de alimentación y control

En el área de trabajo o zona de alimentación el usuario interactúa con el brazo robótico, el mecanismo de regulación permite al usuario regular la elevación dependiendo de la altura del usuario, la base de accionamiento está constituida por pulsadores que sirven para la activación manual del brazo robótico para la manipulación de alimentos de cada plato de comida o bebida.

4.1.1. Área de trabajo o zona de alimentación

El área de trabajo o zona de alimentación está constituida por el brazo robótico con la cámara de acople para la visualización de la boca del usuario, los elementos de control e interacción con el usuario para la activación del algoritmo de control, el porta cuchara que sirve de sujeción para la cuchara, la mesa de apoyo que posee una superficie deslizante y sirve de apoyo para todos los elementos eléctricos y de interacción con el usuario, los platos de comida para el arroz, sopa, postre y el vaso de bebida.



Figura 92. Área de alimentación

Tabla 35

Elementos de la estructura mecánica de soporte

Número	Elementos
1	Prototipo de brazo robótico 6DOF
2	Elementos de control e interacción
3	Porta cuchara
4	Mesa de apoyo
5	Platos de comida
6	Vaso de bebida

4.1.2. Mecanismo de regulación

El mecanismo de regulación varía de 800mm a 920mm, con intervalos de 50mm para dar mayor comodidad al usuario, está conformado por una perilla de sujeción que es usada después de ajustar la altura de la mesa con la palanca, a continuación, se muestran estos elementos.

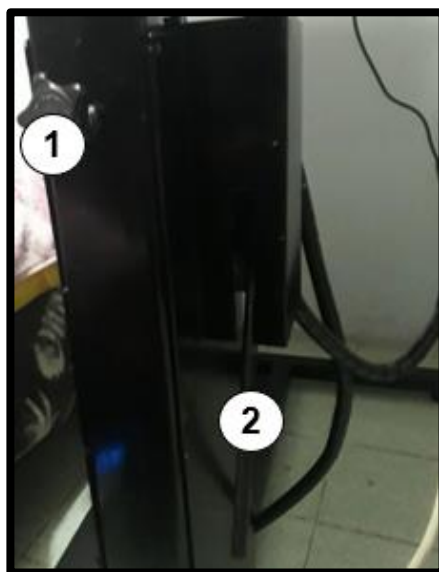


Figura 93. Mecanismo de regulación de altura

Tabla 36

Elementos de la estructura mecánica de soporte

NúmeroElementos	
1	Perilla de sujeción
2	Palanca de regulación

4.1.3. Base de accionamiento

La base de accionamiento funciona como apoyo para los pies del usuario y cuenta con la disposición de cinco pulsadores interactivos para cada plato de comida y bebida, además de la implementación de un paro de emergencia que al ser accionado regresa al brazo robótico a la posición de Home el cual espera una nueva orden. Este pulsador se usa si el usuario ya no desea más alimento o quiere cambiar de plato.



Figura 94. Base de accionamiento

Tabla 37

Elementos de la estructura mecánica de soporte

Número	Elementos
1	Pulsador arroz
2	Pulsador sopa
3	Pulsador postre
4	Pulsador bebida
5	Pulsador stop

4.2. Análisis, pruebas y resultados del sistema de alimentación

Luego de realizar la programación es necesario evaluar el desempeño del algoritmo del sistema de alimentación, para lo cual se realizaron varias pruebas con el fin de observar el comportamiento y resultado en cada una de ellas.

4.2.1. Análisis para el tiempo de alimentación del usuario

“La Mg. Eulalia Isabel Analuisa Jiménez, docente de la Carrera de Enfermería de la Universidad Técnica de Ambato, afirma que para una correcta alimentación es recomendable masticar los alimentos entre 30 y 50 veces por bocado en un minuto, sea para los alimentos sólidos o semisólidos, pues la correcta masticación, permite una mejor

trituration de los alimentos y con esto mejoran los procesos digestivos y de absorción de nutrientes. Además, la comida tiene que ser digerida en pequeños trozos en el caso de un adulto mayor, pues tienen más dificultad para la masticación por la falta de piezas dentales, enfermedades degenerativas, diabetes y otras enfermedades, es por ello que afirma que el tiempo de duración para ingerir la suficiente cantidad de alimento en una persona es aproximadamente de 30 minutos a una hora y en un adulto mayor de 30 minutos a una hora y media". (Analuisa Jiménez, 2018)

4.2.3. Porciones adecuadas para cada usuario

Un adulto requiere comer una taza de fruta de 125g, legumbres 100g, pasta, arroz, avena, fréjol, haba, lenteja 80g, carne cocida, pollo, pescado, pulpa de res o de cerdo, ternera, atún o sardina 150g, aceite, mantequilla, mayonesa, azúcar 40g y una bebida fría o caliente 120 ml, en total de alimentos sólidos 495 g. (Salud180, 2012)

Un adulto mayor con problemas degenerativos y en general con su metabolismo muy lento, requiere una mayor atención en su alimentación, es por ello que se sugiere alimentos blandos, fáciles de masticar, además que sus porciones sean pequeñas para mejorar su digestión y evitar la pesadez estomacal. (González A. L., 2015)

Básicamente la dieta constituye en comer una ración de fruta de 40g, legumbres 40g, pasta o arroz 40g, papas 40g, carnes, pescado, huevos pollo 100g, leche, yogurt 150ml, en total alimentos sólidos 260g y 100ml de leche o yogurt.

4.2.4. Pruebas de accionamiento y control por voz

Se realizaron las pruebas en tres usuarios, los dos primeros adultos de edades entre 20 a 25 años, y a un adulto mayor con osteoartritis de 86 años de edad. Los usuarios interactuaron correctamente con la aplicación del celular, “SISTEMA DE ALIMENTACIÓN ARKAZ 2.0”, ya sea con el botón Manual o Automático.

El botón Manual permitió accionar de manera favorable cada uno de los pulsadores para que el brazo robótico se moviera hacia cada plato de comida.

El botón Automático, permitió al usuario acceder satisfactoriamente al control por voz, con los comandos sopa, segunda, fruta, agua, para el posterior seguimiento de la boca con el brazo robótico.



Figura 95. Aplicación “SISTEMA DE ALIMENTACIÓN ARKAZ 2.0”

4.2.5. Pruebas iniciales en el sistema de pesaje

Para esta prueba se usaron 3 tipos diferentes de platos (porcelana, plástico y aluminio), los cuales se mantuvieron vacíos, para probar que el brazo robótico no toma el alimento de los platos que no tienen comida.



Figura 96. Pruebas de pesaje con platos de porcelana y aluminio

El sistema hizo una comparación del peso medido al principio y después de seleccionar el plato. En las tres pruebas efectivamente el manipulador no tomo el alimento del plato.

4.2.6. Pruebas de levantamiento de cantidades con el brazo robótico

La siguiente tabla muestra las cantidades mínimas y máximas que pueden ser depositadas en la cuchara y vaso para la posterior verificación del levantamiento con el brazo robótico.

Tabla 38

Peso de la cuchara vacía y con los diferentes alimentos

Cuchara	Capacidad(g)	Logra levantar
Vacía	15	Si
Sopa	17	Si
Arroz	21	Si
Postre (fruta picada)	22	Si

Tabla 39

Peso del vaso vacío y con diferentes pesos

Vaso	Capacidad(g)	Logra levantar
Vacío	30	Si
Cuarto	100	Si
Medio	200	Si peso recomendable
Tres cuartos	300	Si con dificultad
Lleno	330	No

En la tabla 38 se observa que el brazo robótico es capaz de levantar la cuchara vacía, con sopa, arroz y postre sin dificultad, por lo tanto, se puede afirmar que la cantidad mínima de levantamiento del brazo robótico queda definida con un valor de 15 g.

En la tabla 39 se observa que el manipulador puede levantar un vaso de hasta 200g sin dificultad, pero con un peso de 300g levanta el vaso con dificultad, por lo tanto, la cantidad máxima de levantamiento del brazo robótico recomendable queda definido con un valor de 200g.

4.2.7. Pruebas del modo manual y modo automático

En las siguientes pruebas se observa el comportamiento del sistema en modo manual y automático.

Prueba modo manual

En este modo se evalúan el tiempo de demora en alimentar al usuario, y el porcentaje de comida que se consume. En las tablas 40, 41, 42, 43 y 44, se tienen las siguientes variables vacío (esta variable representa al plato sin comida), plato con comida inicial (la variable representa al plato con la comida que el usuario va a consumir), plato con comida final (Esta variable representa al plato con comida sobrante después de alimentar al usuario), resultado (es la resta entre el plato inicial y plato final para saber cuánto ha consumido el usuario). La figura 99 muestra a los usuarios y su interacción con el brazo robótico en el modo manual.



Figura 97. Pruebas de modo manual a los usuarios

Usuario uno: modo manual

Tabla 40

Prueba del plato de sopa del usuario uno

	Plato sopa(g)	Tiempo o arroz(g (min))	Tiempo de duración(mi n)	Plato postre(g)	Tiempo de duración(mi n)	Jugo(ml)	Tiempo(mi n)	
Vacío	393	12	393	15,5	393	6,58	30	0,15
Plato con comida final	440		400		402		37	
Plato con la comida inicial	630		500		468		143	
Resultados	190		100		66		106	
Consumo total de alimentos sólidos (g)	462							
Porcentaje consumido en alimentos sólidos	90,7%							
Porcentaje en líquidos	80,1%							
Tiempo total de consumo de los alimentos(min)	34,23							

Usuario dos: modo manual

Tabla 41

Prueba del plato de sopa del usuario dos

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	15	393	13,3	393	8	30	0,15
Plato con comida final	440		414		408		38	
Plato con la comida inicial	650		528		470		139	
Resultados	210		114		62		101	
Consumo total de alimentos sólidos (g)	386							
Porcentaje consumido en alimentos sólidos	80,99%							
Porcentaje en líquidos	84%							
Tiempo total de consumo de los alimentos(min)	36,45							

Usuario tres: modo manual

Tabla 42

Prueba del plato de sopa del usuario tres

	Plato sopa(g)	Tiempo o (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	11	393	10	393	7,39	30	0,15
Plato con comida final	413		399		400		38	
Plato con la comida inicial	495		470		460		139	
Resultados	82		71		60		101	
Consumo total de alimentos sólidos (g)	213							
Porcentaje consumido en alimentos sólidos	90,88%							
Porcentaje en líquidos	84%							
Tiempo total de consumo de los alimentos(min)	28,54							

Usuario cuatro: modo manual

Tabla 43

Prueba del plato de sopa del usuario cuatro

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	15	393	17,54	393	7,58	30	0,15
Plato con comida final	442		410		400		40	
Plato con la comida inicial	645		520		475		149	
Resultados	203		110		75		109	
Consumo total de alimentos sólidos (g)	462							
Porcentaje consumido en alimentos sólidos	89,04%							
Porcentaje en líquidos	80,56%							
Tiempo total de consumo de los alimentos(min)	40,27							

Usuario cinco: modo manual

Tabla 44

Prueba del plato de sopa del usuario cinco

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	10,45	393	12	393	8,25	30	0,15
Plato con comida final	409		404		405		34	
Plato con la comida inicial	480		483		475		128	
Resultados	71		79		70		94	
Consumo total de alimentos sólidos (g)	172							
Porcentaje consumido en alimentos sólidos	86,57%							
Porcentaje en líquidos	81,61%							
Tiempo total de consumo de los alimentos(min)	31,25							

Tabla 45

Porcentajes y tiempo de consumo de cada usuario

	Porcentaje consumido en alimentos sólidos	Porcentaje en líquidos	Tiempo total (min)
Usuario Uno	90,7%	80,1%	34,23
Usuario Dos	80,99%	84%	36,45
Usuario Tres	90,88%	84%	28,54
Usuario Cuatro	89,04%	80,56%	40,27
Usuario Cinco	86,57%	81,61%	31,25
Promedio	87,64%	82,05%	34,148

En la tabla 45 se observa el tiempo total que cada usuario se demoró en consumir la sopa, el arroz, el postre y la bebida; los cuales se encuentran dentro del tiempo estimado de alimentación, según el análisis mostrado anteriormente con la entrevista realizada.

También se observa el promedio del porcentaje de comida en general mayor al 80%, con ello se verifica que el sistema cumple con un nivel manual óptimo en su uso.

Prueba modo automático

En las siguientes pruebas se observa el comportamiento del sistema en modo automático, aquí se evalúan el tiempo de demora del sistema en alimentar al usuario, el porcentaje de comida que consume el usuario, la detección y seguimiento de la boca.

En las tablas 46, 47, 48, 49 y 50, se tienen las siguientes variables vacío (esta variable representa al plato sin comida), plato con comida inicial (la variable representa al plato con la comida que el usuario va a consumir), plato con comida final (Esta variable representa al plato con comida sobrante después de alimentar al usuario), resultado (es la resta entre el plato inicial y plato final para saber cuánto ha consumido el usuario).

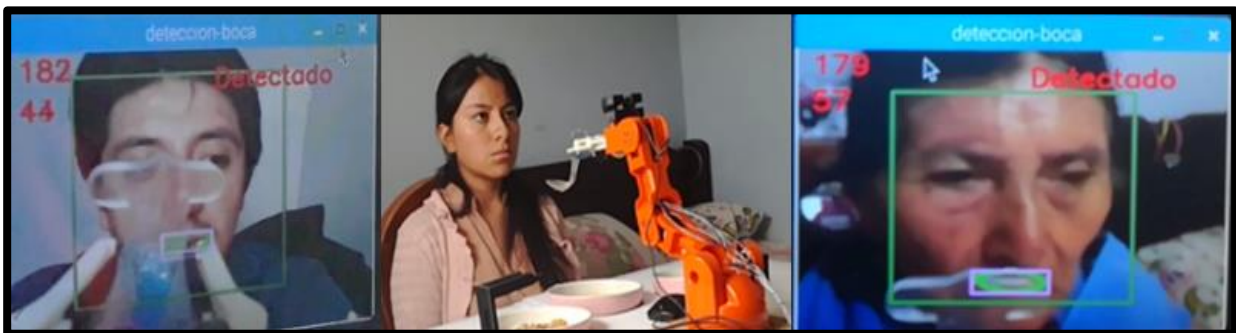


Figura 98. Pruebas en el modo automático a los usuarios

Usuario uno: modo automático

Tabla 46

Prueba del plato de sopa del usuario uno

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración (min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	16	393	15,3	393	7,22	30	0,15
Plato con comida final	428		410		405		38	
Plato con la comida inicial	633		496		479		145	

CONTINÚA →

Resultados	205	86	74	107
Consumo total de alimentos sólidos (g)	365			
Porcentaje consumido en alimentos sólidos	84,77%			
Porcentaje en líquidos	85,4%			
Porcentaje en líquidos				
Tiempo total de consumo de los alimentos(min)	38,67			
Detección y seguimiento de la boca	Si en todos los casos sin dificultad			

Usuario dos: modo automático

Tabla 47

Prueba del plato de sopa del usuario dos

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	13	393	16	393	8,5	30	0,15
Plato con comida final	430		407		406		38	
Plato con la comida inicial	620		496		461		133	
Resultados	190		89		55		95	
Consumo total de alimentos sólidos(g)	334							
Porcentaje consumido en alimentos sólidos	83,65%							
Porcentaje en líquidos	81,95%							
Tiempo total de consumo de los alimentos(min)	37,65							
Detección y seguimiento de la boca	Si en todos los casos sin dificultad							

Usuario tres: modo automático

Tabla 48

Prueba del plato de sopa del usuario dos

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	13,5	393	14,10	393	8	30	0,15
Plato con comida final	410		405		400		38	
Plato con la comida inicial	520		489		452		129	

CONTINÚA 

Resultados	110	84	52	91
Consumo total de alimentos sólidos(g)	246			
Porcentaje consumido en alimentos sólidos	87,82%			
Porcentaje en líquidos	86,61%			
Tiempo total de consumo de los alimentos(min)	37,75			
Detección y seguimiento de la boca	Si en todos los casos sin dificultad			

Usuario cuatro: modo automático

Tabla 49

Prueba del plato de sopa del usuario cuatro

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	18,1	393	16,4	393	8,5	30	0,15
Plato con comida final	420		414		409		42	
Plato con la comida inicial	640		502		485		139	
Resultados	205		86		74		95	
Consumo total de alimentos sólidos(g)	201							
Porcentaje consumido en alimentos sólidos	81,67%							
Porcentaje en líquidos	89,07%							
Tiempo total de consumo de los alimentos(min)	43,15							
Detección y seguimiento de la boca	Si en todos los casos sin dificultad							

Usuario cinco: modo automático

Tabla 50

Prueba del plato de sopa del usuario cinco

	Plato sopa(g)	Tiempo (min)	Plato arroz(g)	Tiempo de duración(min)	Plato postre(g)	Tiempo de duración(min)	Jugo(ml)	Tiempo(min)
Vacío	393	15	393	18	393	9,5	30	0,15
Plato con comida final	440		408		414		42	
Plato con la comida inicial	647		503		478		139	
Resultados	190		89		55		95	
Consumo total de alimentos sólidos(g)	195							

CONTINÚA 

Porcentaje consumido en alimentos sólidos	80,83%
Porcentaje en líquidos	81,50%
Tiempo total de consumo de los alimentos(min)	43,05
Detección y seguimiento de la boca	Si en todos los casos sin dificultad

Tabla 51

Porcentajes y tiempo de consumo de cada usuario

	Porcentaje consumido en alimentos sólidos	Porcentaje en líquidos	Tiempo total (min)
Usuario Uno	84,77%	85,4%	38,67
Usuario Dos	83,65%	81,95%	38,65
Usuario Tres	87,82%	86,61%	37,75
Usuario Cuatro	81,67%	89,07%	43,15
Usuario Cinco	80,83%	81,50%	43,05
Promedio	83,75%	84,91%	40,25

En la tabla 51 se observa el tiempo total que cada usuario se demoró en consumir la sopa, el arroz, el postre y el jugo; los cuales se encuentran dentro del tiempo estimado de alimentación. El tiempo es mayor al modo manual dado que el manipulador robótico realiza un desplazamiento en función de la referencia (boca del usuario); pero es más sencillo porque en el método manual el brazo se ubica en una posición predefinida mientras que el automático entrega el alimento en la posición que este el usuario sin que la persona tenga que moverse.

También se observa en la tabla 51 el porcentaje promedio de consumo de alimentos sólidos y líquidos con porcentajes de 83,75% y 84,91% respectivamente obteniendo así un sistema automático óptimo.

4.2.8. Comprobación de la hipótesis

Para la comprobación de la hipótesis se ha usado el método de distribución t de Student, debido a que el tamaño de la muestra es pequeño, el cual permite aceptar o rechazar la hipótesis planteada.

Existen 2 tipos de hipótesis:

H_0 : Hipótesis nula: El brazo robótico no permite la alimentación independiente de personas afectadas con osteoartritis.

H_1 : Hipótesis alternativa: El brazo robótico permite la alimentación independiente de personas afectadas con osteoartritis.

Nomenclatura de variables

x = Dato del porcentaje de consumo de alimento de las tablas

n = Número de datos

u = Valor de la hipótesis

\bar{x} = Promedio de los datos obtenidos de las tablas

t = Para el cálculo de la prueba estadística

s = Valor de la varianza

α = Nivel de significancia

$$\sum(x - \bar{x})^2 =$$

Sumatoria de cada dato (x) menos el promedio de los datos, elevado al cuadrado

Gl = Grados de libertad

Consumo de alimentos sólidos de las pruebas del modo manual

Datos:

$$n = 5$$

$$u = 80$$

$$\alpha = 0,01$$

Tabla 52

Porcentaje de consumo de alimentos sólidos

Porcentaje de consumo de alimentos sólidos (x)	$x - \bar{x}$	$(x - \bar{x})^2$
90,7%	5,86	34,28
80,99%	-3,86	14,86
90,88%	6,04	36,42
89,04%	4,20	17,60
86,57%	1,73	2,98
438,18	67,19	

El promedio de los datos (\bar{x}) está dado por la siguiente ecuación:

$$\bar{x} = \frac{\sum x}{n}$$

$$\bar{x} = \frac{438,18}{5} = 87,64$$

Para el cálculo de la varianza (s) se usa la siguiente ecuación:

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

$$s = \sqrt{\frac{67,19}{5-1}} = 4,10$$

Por lo tanto, la prueba estadística (t) es la siguiente:

$$t = \frac{\bar{x} - u}{\frac{s}{\sqrt{n}}}$$

$$t = \frac{\bar{x} - u}{\frac{s}{\sqrt{n}}} = \frac{87,64 - 80}{\frac{4,10}{\sqrt{5}}} = 4,17$$

$$Gl = n - 1 = 5 - 1 = 4$$

Con los valores de $\alpha=0,01$ y grados de libertad =5 ya se puede determinar si la hipótesis es aceptada o rechazada; para lo cual se verifica el que el valor de t sea mayor al de la tabla 56.

Como $t(4,17) > 3,365$ se rechaza la hipótesis nula y se acepta H_1 ; con lo que se concluye que el algoritmo de Control Visual Basado en Posición para comandar el movimiento del brazo robótico permite la alimentación independiente de personas afectadas en el modo manual con osteoartritis; ya que el manipulador robótico puede alimentar a una persona con un porcentaje mayor al 80% del contenido de alimentos sólidos de cada plato.

Consumo de alimentos líquidos de las pruebas del modo manual

Datos:

$$n = 5$$

$$u = 78$$

$$\alpha = 0.01$$

Tabla 53

Porcentaje de consumo de alimentos líquidos

Porcentaje de consumo alimentos líquidos (x)	$x - \bar{x}$	$(x - \bar{x})^2$
80,1%	-1,95	3,82
84%	1,95	3,79
84%	1,95	3,79
80,56%	-1,49	2,23
81,61%	-0,44	0,20
410,27	13,82	

El promedio de los datos (\bar{x}) está dado por :

$$\bar{x} = \frac{410,27}{5} = 82,054$$

Para el cálculo de la varianza (s):

$$s = \sqrt{\frac{13,82}{5 - 1}} = 1,86$$

Por lo tanto, la prueba estadística (t) es:

$$t = \frac{82,054 - 78}{\frac{1,86}{\sqrt{5}}} = 4,87$$

$$G1 = 5 - 1 = 4$$

Con los valores de $\alpha=0,01$ y grados de libertad $=5$ ya se puede determinar si la hipótesis es aceptada o rechazada; para lo cual se verifica el que el valor de t sea mayor al de la tabla 56.

Como $t(4,87) > 3,365$ se rechaza la hipótesis nula y se acepta H_1 ; con lo que se concluye que el algoritmo de Control Visual Basado en Posición para comandar el movimiento del brazo robótico permite la alimentación independiente de personas afectadas en el modo manual con osteoartritis; ya que el manipulador robótico puede alimentar a una persona con un porcentaje mayor al 78% del contenido de alimentos líquidos.

Consumo de alimentos sólidos de las pruebas del modo automático:

Datos:

$$n = 5$$

$$u = 80$$

$$\alpha = 0,01$$

Tabla 54

Porcentaje de consumo de alimentos sólidos

Porcentaje de consumo alimentos (x)	$dex - \bar{x}$	$(x - \bar{x})^2$
84,77%	0,79	0,62
83,65%	-0,33	0,11
87,82%	3,84	14,73
81,67%	-2,31	5,35
82%	-1,98	3,93
419,91	24,74	

El promedio de los datos (\bar{x}) está dado por:

$$\bar{x} = \frac{419,91}{5} = 83,98$$

Para el cálculo de la varianza (s):

$$s = \sqrt{\frac{30,7}{5-1}} = 2,42$$

Por lo tanto, la prueba estadística (t) es:

$$t = \frac{83,98 - 80}{\frac{2,49}{\sqrt{5}}} = 3,58$$

$$G l = 5 - 1 = 4$$

Con los valores de $\alpha=0,01$ y grados de libertad =5 ya se puede determinar si la hipótesis es aceptada o rechazada; para lo cual se verifica el que el valor de t sea mayor al de la tabla 56.

Como t (3,58) >3,365 se rechaza la hipótesis nula y se acepta H_1 ; con lo que se concluye que el algoritmo de Control Visual Basado en Posición para comandar el movimiento del brazo robótico permite la alimentación independiente de personas afectadas en el modo manual con osteoartritis; ya que el manipulador robótico puede alimentar a una persona con un porcentaje mayor al 78% del contenido de alimentos sólidos.

Consumo de alimentos líquidos de las pruebas del modo automático

Datos:

$$n = 5$$

$$u = 80$$

$$\alpha = 0,01$$

Tabla 55

Porcentaje de consumo de alimentos líquidos

Porcentaje de consumo alimentos (x)	$x - \bar{x}$	$(x - \bar{x})^2$
85,4%	0,49	0,24
81,95%	-2,96	8,74
86,61%	1,70	2,90
89,07%	4,16	17,34
81,50%	-3,41	11,60
424,53	40,83	

El promedio de los datos (\bar{x}) está dado por :

$$\bar{x} = \frac{424,43}{5} = 84,90$$

Para el cálculo de la varianza (s) se usa:

$$s = \sqrt{\frac{40,83}{5 - 1}} = 3,19$$

Por lo tanto, la prueba estadística (t) es la siguiente:

$$t = \frac{84,90 - 78}{\frac{3,19}{\sqrt{5}}} = 4,83$$

$$G I = 5 - 1 = 4$$

Con los valores de $\alpha=0,01$ y grados de libertad =5 ya se puede determinar si la hipótesis es aceptada o rechazada; para lo cual se verifica el que el valor de t sea mayor al de la tabla 56. Como $t(4,83) > 3,365$ se rechaza la hipótesis nula y se acepta H_1 ; con lo que se concluye que el algoritmo de Control Visual Basado en Posición para comandar el movimiento del brazo robótico permite la alimentación independiente de personas afectadas en el modo manual con osteoartritis; ya que el manipulador robótico puede alimentar a una persona con un porcentaje mayor al 80% del contenido de alimentos líquidos.

Tabla 56

Tabla de t

t	α	0,25	0,2	0,15	0,1	0,05	0,025	0,01	0,005
1		1,00	1,376	1,963	3,078	6,314	12,706	31,821	63,656
2		0,816	1,061	1,386	1,886	2,920	4,303	6,965	9,925
3		0,765	0,978	1,250	1,638	2,353	3,182	4,541	5,841
4		0,714	0,941	1,190	1,533	2,132	2,776	3,747	4,604
5		0,727	0,920	1,156	1,476	2,015	2,571	3,365	4,032

Elaborado por: (Wikilibros, 2015)

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

A continuación, se establecen las conclusiones y recomendaciones de la investigación, diseño e implementación de un algoritmo de control visual basado en posición para comandar el movimiento de un brazo robótico que permita la alimentación independiente de personas afectadas con osteoartritis, las mismas que respaldan y resumen el proyecto de investigación realizado.

5.1. Conclusiones

Debido a que la tensión de Von Mises (0,358MPa) es menor al límite de tensión (0,4MPa) y el factor de seguridad estático (1) es el mínimo requerido para la falla estructural, la madera, esta es aceptada como material a usar para la construcción de la mesa de la estructura mecánica.

El material adecuado para el piñón y cremallera, que servirán para el mecanismo de regulación de una altura con un intervalo de 120mm y una masa total para ser levantada de 36,16kg es el nylon, pues el esfuerzo de tensión que se espera debido a la flexión en el piñón (7,18Mpa) es un valor menor al esfuerzo de flexión permisible (41Mpa). Y para la cremallera el esfuerzo de tensión que se espera debido a la flexión en el piñón es de (16,8Mpa) que es un valor menor al esfuerzo de flexión permisible con un valor de (41Mpa).

Este proyecto se realizó en un software libre para reducir costos, dado que no tiene costo alguno a diferencia de los softwares con sistemas robóticos privados.

Python es un lenguaje de programación de propósito general, muy usado por su ser claridad y facilidad de código. También se destaca por compatibilidad con otros programas como Arduino. Pues su entorno electrónico es de código abierto en donde su software y hardware son fáciles de usar por su sencillez.

La plataforma App Inventor, permitió realizar la aplicación de interacción con el usuario, permitiendo el accionamiento manual y automático del sistema de alimentación.

La librería Dlib utilizada para implementar el algoritmo de seguimiento de la boca ayuda considerablemente ya que permite mantener un registro de la región detectada cuadro a cuadro, disminuir errores, mejorar la velocidad de seguimiento y con ello el sistema se vuelve más eficiente.

La librería Dlib contiene una base de datos de estructuras faciales que están divididas en 68 coordenadas, de esta manera se pueden extraer diferentes partes de la cara. Con esta base de datos es más sencillo y eficaz la detección de la boca.

Para poder instalar la biblioteca Dlib para Python correctamente en la Raspberry Pi, es necesario actualizar el sistema con el fin de recuperar memoria.

Numpy es una biblioteca optimizada para operaciones numéricas. Todas las estructuras de matriz de OpenCV se convierten en matrices de NumPy; para realizar operaciones en Numpy, y puedan ser combinadas con OpenCV.

Para reducir errores y tener mayor efectividad en el algoritmo de visión primero se hace la detección y seguimiento del rostro, y para así después localizar la boca dentro del área de la cara.

Este proyecto emplea el control servo visual para los movimientos del manipulador, pero al requerir movimientos rápidos es necesario tomar en cuenta la dinámica del robot, en donde es posible utilizar otro tipo de controladores (control visual dinámico, la fórmula de Newton-Euler) que calculan directamente la acción de control de las articulaciones del robot.

Con el uso del sensor shield se incrementa la eficiencia del brazo robótico, pues permite enviar o recibir desde el Arduino la señal únicamente y realizar la alimentación de todos los servomotores desde la conexión externa del sensor shield, ya que el Arduino Mega, no posee suficiente corriente para energizar los servomotores.

La cinemática inversa es un método para posicionar el efector final del brazo en el espacio (X, Y, y Z), mediante el cálculo de los ángulos de sus articulaciones. Con la cinemática inversa el manipulador puede llegar a un punto requerido como un sistema sincronizado, solo ingresando la posición en la de los ejes X, Y, y Z. Donde z es la altura, Y es la distancia desde el centro de la base hacia fuera de la misma, X es el movimiento

horizontal. El eje z sólo puede ser positivo debido a que la base del brazo robótico se encuentra en el eje x.

La librería Braccio, usada para el control de los servomotores, es fácil de usar al poder colocar la posición de los 6 servos en una línea de programación, con esto se mejora la velocidad de accionamiento de los servomotores, debido a la reducción de líneas de programación.

Con el sistema "Eyeinhand" se minimiza el tiempo de procesamiento necesario para extraer las informaciones visuales necesarias, en este caso la posición de la boca. Este sistema también permite ahorrar en costos ya que no se necesita de tanta inversión como otros sistemas de procesamiento de imágenes, siendo otra ventaja su costo.

5.2. Recomendaciones

Es muy importante colocar los platos vacíos al inicio del sistema para que el brazo robótico finalice el proceso de alimentación de cada plato de comida, pues se realiza una medición al peso de los platos, es decir, hay una comparación de los valores entre la medida inicial y última del plato para que el sistema asuma un plato vacío.

En el modo automático, para el reconocimiento de la boca es necesario que la cámara pueda enfocar de manera correcta la cara del usuario, para lo cual se calibra el sistema acorde a la altura de la boca del usuario, por medio de la regulación de la mesa a una

altura deseada, y calibración del eje Z según la estatura del usuario como se indica en la tabla 33.

Es recomendable cortar el alimento en trozos pequeños, para la correcta digestión del usuario y también la cuchara colocada en el brazo robótico recoga la comida sin inconvenientes.

No regar bebidas dentro de los orificios de los platos ya que se pueden dañar los elementos del sistema. Este proyecto es un prototipo de un sistema de alimentación.

Revisar que los pulsadores se encuentren conectados de forma correcta, para un buen accionamiento de los mismos.

Realizar un mantenimiento periódico de la parte mecánica, como ajustar los pernos del brazo robótico, de los sensores para que en el sistema no se produzcan errores.

Se recomienda no poner un contenido en el vaso con un peso mayor a 200 g ya que este valor es ideal para que el manipulador lo levante sin dificultad y no exponer tanta carga a los servomotores para alargar la vida de los mismos

Deslizar la parte superior de la mesa con precaución ya que se pueden desconectar los elementos electrónicos.

Verificar que los sensores estén conectados correctamente en el pin correspondiente del Arduino, para evitar errores en la medición de peso.

Verificar que cada servo esté conectado en forma correcta en la shield ya que puede provocar movimientos erróneos en el brazo y producir colisiones que dañarían la estructura del manipulador.

Revisar que la fuente de voltaje, la Raspberry Pi 3 y el brazo robótico estén energizados; ya que si uno no lo está puede producir error en el sistema de alimentación.

Si el usuario elige la parte automática se recomienda tener el bluetooth del dispositivo electrónico encendido durante todo el proceso de alimentación para no generar errores en la comunicación.

REFERENCIAS BIBLIOGRÁFICAS

ADINGOR. (24 de Noviembre de 2015). Recuperado el 17 de Abril de 2017, de [http://adingor.es/congresos/web/uploads/cio/cio2010/QUANTITATIVE_METHODOD S/1631-1640.pdf](http://adingor.es/congresos/web/uploads/cio/cio2010/QUANTITATIVE_METHODOD_S/1631-1640.pdf)

A-Electronics. (Marzo de 2017). Recuperado el 15 de Junio de 2017, de http://a-electronics.com.mx/metepec/index.php?id_product=32&controller=product

Aguilera Gómez, H. (25 de Marzo de 2015). Slideshare. Recuperado el 16 de Junio de 2017, de <https://es.slideshare.net/haivernicolasaguileragomez/implante-de-brazo-antropomrfico-neuronal-1002>

Aguirre Sánchez, C. A. (2015). Scribd. Recuperado el 23 de Abril de 2017, de <https://es.scribd.com/document/348473002/DISENO-DE-UN-SISTEMA-DOMOTICO-ACCIONADO-POR-VOZ-JARVIS-pdf>

AliExpress. (9 de Septiembre de 2012). Recuperado el 12 de Junio de 2017, de es.aliexpress.com/store/product/Aluminium-Robot-6-DOF-Arm-Clamp-Claw-Mount-Kit-Mechanical-Robotic-Arm-for-Arduino-Compatible/614216_32661204916.html?src=ibdm_d03p0558e02r02

Amazon. (20 de Septiembre de 2017). Recuperado el 13 de Junio de 2017, de https://es.aliexpress.com/store/product/7-TFT-LCD-Color-2-Video-Input-Car-Rear-View-Headrest-Monitor-DVD-VCR-Monitor-With/3198058_32844587020.html?spm=a219c.search0303.3.101.2e9f79f60nTjU

U&ws_ab_test=searchweb0_0,searchweb201602_0_10152_10151_10618_1005
9_

Analuisa Jiménez, E. I. (30 de Enero de 2018). Alimentación saludable. (A. Jiménez, Entrevistador)

Aprender Python. (10 de Enero de 2017). Recuperado el 22 de Abril de 2017, de <https://www.aprenderpython.net/deteccion-rostros-caras-ojos-haar-cascad/>

Aprender Python. (10 de Enero de 2018). Recuperado el 22 de Abril de 2017, de <https://www.aprenderpython.net/deteccion-rostros-caras-ojos-haar-cascad/>

Aprendiendo Arduino. (18 de Mayo de 2016). Recuperado el 3 de Octubre de 2017, de <https://aprendiendoarduino.wordpress.com/proyectos/>

Aranda Aboy, J. J. (2009). Slideplayer. Recuperado el 30 de Marzo de 2017, de <http://slideplayer.es/slide/2261835/>

Aranda Delgadillo, M., Medina Muñoz, L. A., Rodríguez Espinosa, I., & González López, S. (Marzo de 2017). Revista de prototipos tecnológicos. Recuperado el 31 de Marzo de 2017, de https://ecorfan.org/spain/researchjournals/Prototipos_Tecnologicos/vol3num7/Revista_de_Prototipos_Tecnologicos_V3_N7_2.pdf

Arduino. (24 de Diciembre de 2014). Recuperado el 14 de Junio de 2017, de Arduino: <http://arduino.cl/arduino-uno/>

Arduino. (24 de Diciembre de 2014). Recuperado el 18 de Junio de 2017, de Arduino:

<http://arduino.cl/arduino-mega/>

Arduino. (2016). Recuperado el 27 de Abril de 2017, de

<http://www.arduino.org/products/tinkerkit/arduino-tinkerkit-braccio>

Arduino. (13 de Marzo de 2017). Recuperado el 5 de Junio de 2017, de

<https://store.arduino.cc/usa/tinkerkit-braccio>

AREC. (2017). Recuperado el 26 de Abril de 2017, de [http://www.arec.com.co/pesaje-](http://www.arec.com.co/pesaje-estatico.html)

[estatico.html](http://www.arec.com.co/pesaje-estatico.html)

Artrosis. (30 de Enero de 2014). Recuperado el 22 de Marzo de 2017, de Artrosis:

<http://artrosisaldia.com/artrosis-de-mano/>

Avutem. (6 de Agosto de 2013). Recuperado el 23 de Mayo de 2017, de

<http://avutem.blogspot.com/>

Barba Guamán, L. R. (Julio de 2015). Academia. Recuperado el 12 de Octubre de 2017,

de

[http://www.academia.edu/25845666/Utilizaci%C3%B3n_de_m%C3%A9todos_de
_visi%C3%B3n_artificial_para_PC_como_apoyo_en_la_automoci%C3%B3n](http://www.academia.edu/25845666/Utilizaci%C3%B3n_de_m%C3%A9todos_de_visi%C3%B3n_artificial_para_PC_como_apoyo_en_la_automoci%C3%B3n)

Barrientos, P. P. (2007). Fundamentos de Robótica, Segunda Edición. Madrid: Mcgraw-

Hill.

Beagleboard. (7 de Noviembre de 2017). Recuperado el 19 de Mayo de 2017, de

<http://beagleboard.org/bone>

Berent, I., & Vaknin, V. (2017). Scribd. Recuperado el 25 de Abril de 2017, de <https://es.scribd.com/document/247114718/Incompleto-La-Dislexia-Afecta-el-Reconocimiento-del-Habla-pero-Puede-Dejar-Intacta-la-Competencia-Fonologica-doc>

Buenastareas. (15 de Diciembre de 2012). Recuperado el 25 de Mayo de 2017, de <http://www.buenastareas.com/ensayos/Cordenadas-Homogeneas/6874545.html>

Bustamante, M. (26 de Agosto de 2016). Los colores de la noche. Recuperado el 11 de Enero de 2018, de <http://loscoloresdelanoche.blogspot.com/2016/08/como-hacer-una-panoramica-nocturna.html>

Calero, W. (8 de Diciembre de 2010). Ingeniería de Software. Recuperado el 21 de Marzo de 2017, de <http://ingenieraupoliana.blogspot.com/2010/12/cliente-servidor.html>

Capek, K. (25 de Noviembre de 2010). Recuperado el 20 de Abril de 2017, de <http://luzrobotic2.blogspot.com/2010/>

Carreño, A. D. (Mayo de 2011). Brazos mecánicos. Recuperado el 5 de Mayo de 2017, de <http://brazosmecanicos.blogspot.com/>

Chávez, F. (2008). Recuperado el 30 de Marzo de 2017, de <https://es.scribd.com/document/314047772/Sistema-de-Control-de-Inventario>

CodesofInterest. (2 de Julio de 2017). Recuperado el 18 de Octubre de 2017, de <https://www.codesofinterest.com/2016/10/getting-dlib-face-landmark-detection.html>

Colle, R. (17 de Julio de 2014). Issuu. Recuperado el 20 de Mayo de 2017, de Issuu:
<https://issuu.com/raymondcolle/docs/librotcsc>

Conche, N. (26 de Enero de 2011). Blogspot. Recuperado el 22 de Mayo de 2017, de
<http://roboticafascinante.blogspot.com/2011/01/cinematica-de-un-robot-industrial.html>

Corona Leonel, A. G. (2014). Sensores y Actuadores. Grupo Editorial Patria.

Coursehero. (2015). Recuperado el 28 de Mayo de 2017, de
<https://www.coursehero.com/file/p27j91g/Vision-Development-Bundle-el-cual-incluye-los-tres-m%C3%B3dulos-necesarios-que-usted/>

Cruz, A. B. (2007). Fundamentos de robótica. Madrid: McGraw-Hill.

Diaz, A. (27 de Abril de 2015). Ironsistem Software. Recuperado el 22 de Abril de 2017,
de <http://ironsistem.com/tutoriales/python/deteccion-facial-en-python-y-opencv/>

Doutel, F. (20 de Agosto de 2016). Xataka. Recuperado el 13 de Mayo de 2017, de
<https://www.xataka.com/autor/fernando-doutel/rss2.xml>

Dueñas, A. (30 de Noviembre de 2011). Scribd. Recuperado el 27 de Marzo de 2017, de
<https://www.scribd.com/document/119877859/Parametros-de-DENAVIT-HARTENBERG>

Duran, D. (5 de Noviembre de 2013). Prezi. Recuperado el 12 de Abril de 2017, de
<https://prezi.com/zhhghnsrccdw/copy-of-vision-artificial/>

Electronicabf. (15 de Marzo de 2017). Recuperado el 16 de Junio de 2017, de <http://www.electronicabf.com/camaras-web/8482-32200223101-4710268246435.html>

Emaze. (16 de Julio de 2015). Recuperado el 13 de Abril de 2017, de <https://www.emaze.com/@ALWCLOLZ/Brazo-rob%EF%BF%BDticoUn#!>

Escalona, I. (30 de Septiembre de 2014). Gestiopolis. Recuperado el 16 de Abril de 2017, de Gestiopolis: <https://www.gestiopolis.com/transductores-y-sensores-en-la-automatizacion-industrial/>

Esparza, C. H., Carvajal, S. O., & García, V. H. (Diciembre de 2010). UTS. Recuperado el 29 de Marzo de 2017, de <http://www.uts.edu.co/portal/files/ARTICULORESET5.pdf>

Estimamos. (2012). Recuperado el 18 de Junio de 2017, de <http://estimamos.com/webcam-genius/genius-facecam-315/>

Etitudela. (27 de Mayo de 2010). Recuperado el 15 de Abril de 2017, de <http://etitudela.com/celula/downloads/visionartificial.pdf>

Fanghanel Córdova, H. (Junio de 2015). Scribd. Recuperado el 24 de Junio de 2017, de <https://es.scribd.com/document/355182684/Tesis-diseno-por-elementos-finitos>

Fisioterapia. (7 de Marzo de 2017). Recuperado el 22 de Marzo de 2017, de <https://html.rincondelvago.com/fisioterapia-de-lesiones-de-mano-y-dedo.html>

Fisioterapia ayllón. (21 de Octubre de 2012). Recuperado el 22 de Marzo de 2017, de <https://fisioterapiaayllon.wordpress.com/>

Flores Pérez, L. A. (6 de Enero de 2017). Slideshare. Recuperado el 12 de Abril de 2017, de <https://es.slideshare.net/AntonioFloresPerez/unidad-5-de-inteligencia-artificial-sistemas-computacionales>

Foto igual. (12 de Marzo de 2017). Recuperado el 22 de Enero de 2018, de <http://fotoigual.com/5-factor-magnificacion/>

García Gómez, G. J. (2010). Universidad de Alicante. Recuperado el 22 de Abril de 2017, de https://rua.ua.es/dspace/bitstream/10045/16063/1/Tesis_gabrieljesus_garcia.pdf

Garcia, D. (3 de Mayo de 2013). Scribd. Recuperado el 3 de Junio de 2017, de <https://www.scribd.com/document/142601525/Unidad-5-docx>

Gómez Molina, N. Y., & Ríos Velásquez, R. M. (2009). DocPlayer. Recuperado el 16 de Abril de 2017, de DocPlayer: <http://docplayer.es/7388736-Sistema-computacional-de-vision-artificial-para-el-diagnostico-de-patrones-de-marcha-humana-de-las-protesis-bk.html>

González Sánchez, M. E. (s.f.). Scribd. Recuperado el 19 de Abril de 2017, de <https://es.scribd.com/document/354469136/Analisis-biometrico-orejas-pdf>

González, A. L. (05 de Octubre de 2015). Perfil. Recuperado el 30 de Enero de 2018, de www.revistaperfil.com

González, S. (17 de Agosto de 2016). Recuperado el 11 de Junio de 2017, de <http://arduino-robotica-sotelo-gonzalez.blogspot.com/2016/08/>

González, V. (05 de Junio de 2005). Robot Industriales. Recuperado el 22 de Marzo de 2017, de http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm

González, V. (2014). Control y Robótica. Recuperado el 22 de Marzo de 2017, de Control y Robótica: http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm

Grupo de investigación NBIO. (Abril de 2012). Recuperado el 25 de Mayo de 2017, de <http://nbio.umh.es/files/2012/04/practica2.pdf>

Guitar, D. (Lunes de Junio de 2014). Estudios Técnicos. Recuperado el 15 de Abril de 2017, de Estudios Técnicos: <https://exploratecnica.blogspot.com/2016/05/robots-industriales.html>

Hobbytronics. (12 de Marzo de 2017). Recuperado el 7 de Junio de 2017, de <http://www.hobbytronics.co.za/p/1084/robotpark-6-dof-aluminum-robot-arm-kit-info>

Huang Lin, C. (2 de Noviembre de 2014). SlideShare. Recuperado el 15 de Mayo de 2017, de <http://es.slideshare.net/chaohl/proyecto-de-grado-disear-e-implementar-un-sistema-de-control-cinemtico-para-el-robot-industrial-motoman-k3s>

Institutos Nacionales de la Salud. (30 de Noviembre de 2014). Recuperado el 22 de Marzo de 2017, de Institutos Nacionales de la Salud: https://www.niams.nih.gov/Portal_en_espanol/Informacion_de_salud/Artritis/osteoartritis_ff_espanol.asp

Inteligencia Artificial. (Marzo de 2014). Recuperado el 25 de Abril de 2017, de <https://webinteligenci.weebly.com/reconocimiento-del-habla.html>

Izurieta, A. (1997). Manual para la medición de la Eficiencia de Manejo de un Sistema. En A. Izurieta, Manual para la medición de la Eficiencia de Manejo de un Sistema (pág. Costa Rica). IICA/CATIE.

Jaravito, J. (Febrero de 2009). Escuela Colombiana de Ingeniería Julio Garavito. Recuperado el 26 de Junio de 2017, de <http://copernico.escuelaing.edu.co/lpinilla/www/protocols/ERGO/DISENO%20DE%20PUESTO%20DE%20TRABAJO%202009-2.pdf>

Juste, A. (25 de Abril de 2014). Libre de lácteos. Recuperado el 5 de Febrero de 2018, de Libre de lácteos: <http://libredelacteos.com/salud/artritis-y-dieta-i-nutrientes-esenciales/>

Laguna, J. L. (29 de Agosto de 2006). Demayores.com. Recuperado el 25 de Marzo de 2017, de <http://www.demayores.com/2006/08/29/ecuador-el-60-de-los-mayores-sufren-artrosis/>

Larios, A. (24 de Septiembre de 2013). Universidad de las Américas Puebla. Recuperado el 23 de Abril de 2017, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/ahuactzin_l_a/capitulo1.pdf

Larios, A. A. (9 de Diciembre de 1999). UDLAP Bibliotecas. Recuperado el 23 de Abril de 2017, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/ahuactzin_l_a/

Lipiello, V. (2007). Position-Based Visual Servoing in Industrial.

Lleida, E. (10 de Octubre de 2000). Bonnng. Recuperado el 25 de Abril de 2017, de <http://physionet.cps.unizar.es/~eduardo/investigacion/voz/rahframe.html>

López Apostolovich, L. F. (Agosto de 2009). PUCP. Recuperado el 28 de Marzo de 2017, de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/377/LOPEZ_LUIS_MODELACION_BRAZO_ROBOTICO.pdf.txt?sequence=5

López García, C. (10 de Enero de 2012). Quiroconsulta. Recuperado el 26 de Marzo de 2017, de http://quiroconsulta.blogspot.com/2012/01/iv-dolencias-musicales-dolencias-mas_10.html

López Germán, A. (Septiembre de 2014). DocPlayer. Recuperado el 23 de Abril de 2017, de <http://docplayer.es/16924757-Trabajo-de-fin-de-grado.html>

López Puigdollers, D., Cabeza Laguna, R., & Villanueva Larre, A. (1 de Julio de 2016). Academica. Recuperado el 20 de Abril de 2017, de <http://academica->

e.unavarra.es/bitstream/handle/2454/21994/TFG_Dan_Lopez_Puigdollers.pdf?sequence=1&isAllowed=y

Lowren, P. (2009). Library. Recuperado el 25 de Marzo de 2017, de https://library.e.abb.com/public/ef9f52910446397dc12576880039d426/42-46%204M913_SPA_72dpi.pdf

Mancilla, G. (Marzo de 2016). Scribd. Recuperado el 17 de Octubre de 2017, de <https://es.scribd.com/document/348201642/Trabajo-Proceso-de-Metalizado-Robotizado>

Marin, C. (17 de Noviembre de 2017). Tutor de programación. Recuperado el 23 de Abril de 2017, de <http://acodigo.blogspot.com/2017/11/face-landmarks-detector-con-dlib-y.html>

Martínez Andrade, A. E. (2013). Scribd. Recuperado el 17 de Octubre de 2017, de <https://es.scribd.com/document/356788836/Streaming-para-tv-por-internet>

Martínez Prado, M. A. (Octubre de 2007). Scribd. Recuperado el 16 de Mayo de 2017, de <https://es.scribd.com/document/124359206/Sistema-de-Control-Para-Robots-Industriales>

Mason, M. P. (15 de Diciembre de 2009). Brainline. Recuperado el 24 de Mayo de 2017, de <http://www.brainline.org/content/2009/12/pregntele-al-experto-tecnologa-asistencial-para-personas-con-una-lesin-cerebral-traumtica.html>

Mayor, V. (16 de Enero de 2012). Martillo, Yunque y Estribo. Recuperado el 25 de Abril de 2017, de <https://martilloyunqueyestribo.wordpress.com/2012/01/16/reconocimiento-automatico-del-habla/>

Medina, E. (Enero de 2013). Scribd. Recuperado el 27 de Mayo de 2017, de <https://es.scribd.com/document/378445940/RobticaMdica-1>

Montealegre, J. (5 de Febrero de 2016). Issuu. Recuperado el 14 de Mayo de 2017, de <https://issuu.com/juanmontealegre/docs/inteligencia-artificial-investigaci>

Mott, R. (1995). Diseño de Elementos de Máquinas. México: Prentice Hall.

Motyka, J. (29 de Febrero de 2016). Computerhoy. Recuperado el 10 de Junio de 2017, de <http://computerhoy.com/noticias/hardware/raspberry-pi-3-wifi-bluetooth-ya-venta-41181>

Muñoz, J. M. (14 de Diciembre de 2015). Docslide. Recuperado el 20 de Abril de 2017, de <http://myslide.es/documents/antecedentes-reconocimiento-de-voz.html>

NewYork Presbyterian Hospital. (16 de Diciembre de 2016). Recuperado el 22 de Marzo de 2017, de <http://www.nyp.org/womens/library/85%257CP04195?l=es>

Nicola, A. (10 de Enero de 2009). Dokumen. Recuperado el 23 de Marzo de 2017, de <http://dokumen.tips/documents/celda-de-carga-56180e5957651.html>

Nicola, A. (Enero de 2009). Repositorio Digital Escuela Politécnica Nacional. Recuperado el 26 de Abril de 2017, de <http://bibdigital.epn.edu.ec/bitstream/15000/1122/1/CD-1965.pdf>

NTE INEN 1641. (5 de Enero de 2013). Recuperado el 23 de Junio de 2017, de <https://archive.org/details/ec.nte.1641.1988>

Olan Robles, J. (2008). Scribd. Recuperado el 22 de Abril de 2017, de <https://es.scribd.com/document/363797843/Aplicaciones-Del-RAH>

Osorio, A. (16 de Noviembre de 2014). El blog de Alcibied. Recuperado el 23 de Marzo de 2017, de El blog de Alcibied: <https://alcibiedosorio.wordpress.com/2014/11/16/enfermedades-de-los-huesos/>

Pantoja, M., & Villota, D. (Junio de 2013). Universidad Autónoma de Nariño. Recuperado el 28 de Junio de 2017, de http://docplayer.es/8730253-Engranajes-pantoja-mario-villota-danilo-fher_0011-hotmail-com-universidad-autonoma-de-narino.html

Pari Pinto, P. L. (25 de Junio de 2015). Muti Language Document. Recuperado el 15 de Abril de 2017, de Muti Language Document: <https://vdocuments.site/documents/pablo-lizardo-pari-pinto.html>

Pérez Rodríguez, A. U., Domínguez Sosa, G., & González Baños, E. (2014). Archivos de Medicina. Recuperado el 23 de Marzo de 2017, de <http://www.archivosdemedicina.com/medicina-de-familia/factores-de-riesgoextrnsecos-para-cadasen-un-hogar-para-adultosmayores-de-tabascomxico.php?aid=542>

Pertuz Ocampo, J. (2014). Scribd. Recuperado el 26 de Marzo de 2017, de <https://es.scribd.com/document/251203611/Curso-de-Robotica-Avanzada-2014>

Picuíno. (Diciembre de 2017). Recuperado el 25 de Octubre de 2017, de <https://www.picuíno.com/es/arduprog/control-pid-digital.html>

Portillo Vélez, R. (12 de Diciembre de 2013). Scribd. Recuperado el 30 de Marzo de 2017, de Scribd: <https://es.scribd.com/document/232243024/Diseno-y-construccion-de-un-brazo-robotico-de-dos-gradoss-de-libertad-RR>

Proyecto Inteligencia Artificial UMG. (2015). Recuperado el 22 de Mayo de 2017, de <http://umg2015.weebly.com/visioacuten-artificial.html>

Python.org. (20 de 12 de 2017). Recuperado el 10 de Octubre de 2017, de <https://pypi.python.org/pypi/imutils>

Ramírez Benavides, K. (12 de Abril de 2012). Universidad de Costa Rica. Recuperado el 15 de Mayo de 2017, de <http://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaDirectaRobot.ppt>

Ramírez Benavides, K. (12 de Abril de 2012). Universidad de Costa Rica. Recuperado el 18 de Mayo de 2017, de Universidad de Costa Rica: <http://www.kramirez.net/wp-content/uploads/2012/04/CinematicaInversaRobot.pdf>

Ramírez Benavides, K. (14 de Mayo de 2017). Universidad de Costa Rica. Recuperado el 13 de Mayo de 2017, de <http://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaRobot.pdf>

Ramírez, F. (2016). DocPlayer. Recuperado el 28 de Abril de 2017, de <http://docplayer.es/12009627-Boletin-tecnologico-arjac-raspberry-pi-una-computadora-en-la-palma-de-tu-mano-www-rcestrada-org.html>

Retro Informática. (21 de Junio de 2015). Recuperado el 10 de Abril de 2017, de <http://www.fib.upc.edu/retro-informatica/avui/assistencial.html>

Riquelme García, M., & Zamora Pedreño, R. (24 de Agosto de 2014). Scribd. Recuperado el 14 de Junio de 2017, de <https://es.scribd.com/doc/309633019/Router-Cnc>

Robotica. (2015). Recuperado el 22 de Marzo de 2017, de <http://robotic242004.blogspot.com/p/el-glosa.html>

Rodriguez, E. (2 de Septiembre de 2010). Blog de un universitario. Recuperado el 21 de Abril de 2017, de <http://elblogdeununiversitario.blogspot.ca/2010/09/estudio-del-desarrollo-de-la-ingenieria.html>

Romero Duarte, M. E. (12 de Abril de 2011). Wikispaces. Recuperado el 15 de Mayo de 2017, de <http://brazorboticodesoldadura.wikispaces.com/file/view/Marco+Teorico.pdf>

Romero Velásquez, J., Olgúin Díaz, E., & Morales Díaz, A. (2009). DocPlayer. Recuperado el 25 de Junio de 2017, de <http://docplayer.es/58459851-Modelo->

dinamico-de-un-robot-bipedo-planar-de-9-grados-de-libertad-en-fase-de-vuelo-y-fases-de-contacto.html

Romero, M. E. (12 de Abril de 2011). Docplayer. Recuperado el 30 de Mayo de 2017, de <http://docplayer.es/11605012-Diseno-de-celda-robotica-marca-west-arco-abb-para-proceso-de-soldadura-por-arco-por-medio-del-software-robotstudio-abb.html>

Roydisa. (18 de Noviembre de 2012). Recuperado el 24 de Junio de 2017, de <http://www.roydisa.es/wp-content/uploads/2012/12/engranajes.pdf>

Ruiz, R. (31 de Octubre de 2011). El blog de Ricardo Ruiz. Recuperado el 25 de Marzo de 2017, de <https://ricardoruizdeadana.blogspot.com.es/2011/10/diagnostico-de-la-artrosis.html>

Salazar, X. (11 de Junio de 2015). Prezi. Recuperado el 23 de Abril de 2017, de <https://prezi.com/w-t-7ebqnx4z/tecnologias-de-voz/>

Saldaña, E. (13 de Julio de 2013). Slideshare. Recuperado el 12 de Marzo de 2017, de Slideshare: <http://es.slideshare.net/ohckang/robotica-edinsoncs-ockangplcautomatas-cinematicaautomatizacion-y-robotica>

Saltarén Pazmiño, R. (15 de Octubre de 2015). SlideSearch. Recuperado el 10 de Junio de 2017, de <https://www.slidesearch.net/slide/practicas-de-robotica-utilizando-matlab-roque>

Salud180. (9 de Marzo de 2012). Recuperado el 30 de Enero de 2018, de <https://www.salud180.com/artritis>

Santiago, I. (28 de Noviembre de 2015). Academia.edu. Recuperado el 22 de Marzo de 2017, de http://www.academia.edu/15424261/brazos_robots

Scribd. (2017). Recuperado el 11 de Abril de 2017, de <https://es.scribd.com/document/305681127/Parte-de-Marco-Teorico>

Senmache Torres, J. C. (2008). Scribd. Recuperado el 22 de Marzo de 2017, de <https://es.scribd.com/doc/292994010/Brazo-Robotico-y-Banda-Transportadora>

Slideshare. (Octubre de 2010). Recuperado el 6 de Mayo de 2017, de <https://es.slideshare.net/mmastino/geometra-de-los-robots>

Sociedad Española de Reumatología. (17 de Diciembre de 2016). Recuperado el 24 de Marzo de 2017, de <http://www.ser.es/pacientes/enfermedades-reumaticas/artrosis/>

SolidWorks. (2018). Recuperado el 25 de Junio de 2017, de http://help.solidworks.com/2012/Spanish/SolidWorks/Cworks/prot_vonm.htm?format=P&value=

Tecnología y discapacidad. (31 de Marzo de 2009). Recuperado el 12 de Abril de 2017, de <http://tecnologiadispar.blogspot.com/>

Torbert, M. (16 de Noviembre de 2016). Elsumario.com. Recuperado el 22 de Marzo de 2017, de <http://elsumario.com/terapia-y-colageno-para-aliviar-tu-artrosis/>

United States Department of Labor. (5 de Julio de 2015). Recuperado el 22 de Marzo de 2017, de https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html#2

Vásquez, A. (24 de Septiembre de 2014). SlideShare. Recuperado el 15 de Abril de 2017, de <https://es.slideshare.net/leonleonc/cintica-de-un-robot>

Vida Abuelo. (13 de Enero de 2015). Recuperado el 25 de Marzo de 2017, de <http://vidaabuelo.com/dolor-lumbar-en-la-tercera-edad-parte-1/>

Vidal, C. P. (2008). Control Sensorial de Sistemas Robóticos. Valencia: Universidad Miguel Hernández de Elche.

Villa, J. (15 de Junio de 2013). PLC. Recuperado el 28 de Marzo de 2017, de <http://blog.espol.edu.ec/johnvilla/2013/06/15/tipos-de-brazos-roboticos/>

Villegas Cortez, J. (30 de Mayo de 2015). ResearchGate. Recuperado el 25 de Mayo de 2017, de https://www.researchgate.net/publication/277403965_Implementacion_de_reconocimiento_de_objetos_por_color_y_forma_en_un_robot_movil

Wikilibros. (16 de Septiembre de 2015). Recuperado el 3 de Junio de 2018, de [https://es.wikibooks.org/wiki/Administraci%C3%B3n_de_empresas/Estad%C3%A1stica_para_los_negocios/Prueba_de_hip%C3%B3tesis_\(estad%C3%ADstica\)/Herramientas/Tabla_de_la_distribuci%C3%B3n:_T](https://es.wikibooks.org/wiki/Administraci%C3%B3n_de_empresas/Estad%C3%A1stica_para_los_negocios/Prueba_de_hip%C3%B3tesis_(estad%C3%ADstica)/Herramientas/Tabla_de_la_distribuci%C3%B3n:_T)

Wikipedia. (28 de Enero de 2017). Recuperado el 15 de Octubre de 2017, de https://en.wikipedia.org/wiki/Viola-Jones_object_detection_framework

Wikipedia. (22 de Junio de 2017). Recuperado el 13 de Marzo de 2018, de <https://es.wikipedia.org/wiki/Malware>

Wikiuniversidad. (13 de Junio de 2017). Recuperado el 11 de Agosto de 2017, de https://es.wikiversity.org/wiki/Tecnolog%C3%ADas_Multimedia_e_Interacci%C3%B3n/Curso_2015-2016:_Caso_5

Wilmur, R. (19 de Junio de 2017). Cuidemonos unos a otros. Recuperado el 12 de Agosto de 2017, de <http://cuidemonosunosotros.blogspot.com/feeds/posts/default>

Yumpu. (30 de Marzo de 2017). Recuperado el 5 de Abril de 2017, de http://icaro.eii.us.es/descargas/tema_4_parte_2.pdf

Yumpu. (30 de Marzo de 2017). Recuperado el 7 de Mayo de 2017, de http://icaro.eii.us.es/descargas/Tema4_parte3.pdf

ANEXOS



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Se certifica que el siguiente trabajo fue desarrollado por los señores Katherine Estefhania Aroca Remache y Arturo Fabián Jiménez Bonilla bajo la supervisión.

Ing. Darío J. Mendoza Ch.

Director del proyecto

Ing. Vicente D. Hallo C.

Director de Carrera

Dr. Rodrigo. Vaca C.

Secretario Académico