

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería
Ingeniería Mecatrónica



Megaproyecto “DANI”:
Diseño e implementación de un Dragón Animatrónico Interactivo
con fines de promover los programas tecnológicos de la
Universidad del Valle de Guatemala

Trabajo de graduación en modalidad de Megaproyecto presentado por:

Aarón Ernesto Girón González

Samuel Alejandro Maldonado López

Kuk Ho Chung

para optar al grado académico de Licenciados en Ingeniería Mecatrónica

Guatemala,
2017

Megaproyecto “DANI”:
Diseño e implementación de un Dragón Animatrónico Interactivo
con fines de promover los programas tecnológicos de la
Universidad del Valle de Guatemala

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería
Ingeniería Mecatrónica



Megaproyecto “DANI”:
Diseño e implementación de un Dragón Animatrónico Interactivo
con fines de promover los programas tecnológicos de la
Universidad del Valle de Guatemala

Trabajo de graduación en modalidad de Megaproyecto presentado por:

Aarón Ernesto Girón González

Samuel Alejandro Maldonado López

Kuk Ho Chung

para optar al grado académico de Licenciados en Ingeniería Mecatrónica

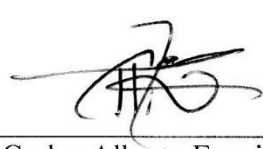
Guatemala,
2017

Asesor:

(f) 
MSc. Miguel Enrique Zea Arenales

Vo. Bo.:

(f) 
MSc. Miguel Enrique Zea Arenales

(f) 
MSc. Carlos Alberto Esquit Hernández

Fecha de aprobación: Guatemala, 28 de noviembre de 2017

ÍNDICE

| | Página |
|---|--------|
| LISTA DE CUADROS | xii |
| LISTA DE FIGURAS | xiii |
| RESUMEN | xvii |
| I. INTRODUCCIÓN | 1 |
| II. OBJETIVOS | 3 |
| A. Objetivo general | 3 |
| B. Objetivo específico | 3 |
| III. JUSTIFICACIÓN | 4 |
| IV. MARCO TEÓRICO | 5 |
| A. Procesamiento de señales (audio y video) | 5 |
| 1. Lenguajes de programación | 5 |
| a. Plataforma de programación | 5 |
| 2. Sistemas de reconocimiento de voz | 6 |
| a. API (Application Programming Interface) | 6 |
| 3. Sistemas de reconocimiento facial | 12 |
| a. OpenCV | 12 |
| b. Visual Servoing | 13 |
| c. Triángulos de cámara | 19 |
| d. Mini cámara – OOSXXX Mini Security Camera | 24 |
| e. Microsoft LifeCam Cinema | 24 |
| 4. Comunicación serial (señales de audio y video) | 25 |
| a. Comunicación serial Python-Arduino | 25 |
| 5. Librerías / Add-Ons de Python | 25 |
| a. pySerial | 25 |
| b. sys | 26 |
| 6. Machine / Deep Learning | 26 |
| a. Machine Learning | 26 |

| | | |
|----|--|----|
| b. | Deep Learning..... | 28 |
| B. | Diseño mecánico de brazos y gestos faciales de DANI..... | 29 |
| 1. | Anatomía de un brazo humano | 29 |
| a. | Tipos de anatomía | 29 |
| b. | Planos de anatomía | 30 |
| c. | Articulaciones | 30 |
| 2. | Punto de vista biológico de la cabeza humana..... | 34 |
| a. | Definición y características..... | 34 |
| 3. | Diseño mecánico para animatrónicos | 35 |
| a. | Introducción | 35 |
| b. | Modos de operación de un animatrónico | 38 |
| 4. | Plataforma CAD..... | 38 |
| a. | Gráficas de ingeniería | 38 |
| b. | Autodesk Inventor 2017..... | 39 |
| c. | ANSYS 17.2 | 40 |
| 5. | Análisis de materiales | 41 |
| a. | Introducción | 41 |
| b. | Resistencia y rigidez del material | 41 |
| c. | Tipos de materiales | 42 |
| d. | Propiedades mecánicas de los materiales..... | 43 |
| 6. | Análisis estructural..... | 45 |
| a. | Esfuerzos..... | 45 |
| b. | Elementos de esfuerzos..... | 45 |
| c. | Esfuerzos principales | 47 |
| d. | Esfuerzo cortante máximo | 48 |
| e. | Tensión y compresión directas..... | 48 |
| f. | Cortante directo..... | 49 |
| g. | Cortante torsional, para secciones circulares y no circulares..... | 49 |
| h. | Esfuerzos cortantes verticales en vigas | 52 |

| | | |
|-----|--|----|
| i. | Flexión | 53 |
| j. | Concentración de esfuerzos | 55 |
| 7. | Diseño de mecanismos..... | 55 |
| a. | Introducción | 55 |
| b. | Movilidad o grados de libertad | 56 |
| c. | Eslabones, juntas y cadenas cinemáticas | 56 |
| 8. | Métodos de unión..... | 58 |
| a. | Tipos de métodos de unión | 58 |
| 9. | Servomotores Dynamixel..... | 58 |
| a. | Características importantes | 59 |
| b. | Dynamixel AX-12A..... | 59 |
| c. | Dynamixel XL-320 | 60 |
| 10. | Métodos de manufactura de bajo costo..... | 61 |
| a. | Impresión 3D | 61 |
| b. | Corte láser | 62 |
| V. | ANTECEDENTES | 63 |
| A. | De funcionamiento repetitivo | 64 |
| B. | Teleoperados..... | 65 |
| VI. | METODOLOGÍA..... | 66 |
| A. | Procesamiento de audio y video..... | 66 |
| 1. | Entorno de programación..... | 66 |
| 2. | Sistema de reconocimiento de rostro | 67 |
| 3. | Sistema de reconocimiento de voz..... | 69 |
| B. | Diseño e implementación mecánica de brazo y garras de DANI..... | 70 |
| 1. | Iteraciones de los diseños del brazo | 70 |
| 2. | Análisis de esfuerzos..... | 72 |
| 3. | Análisis de tensión para los actuadores de la garra..... | 72 |
| 4. | Análisis de parámetros de los actuadores principales | 72 |
| C. | Diseño e implementación mecánica de gestos faciales de DANI..... | 73 |

| | |
|---|-----|
| VII. RESULTADOS | 74 |
| A. Componente de procesamiento de audio y video..... | 74 |
| 1. Trade Study para selección de sistemas de reconocimiento | 74 |
| a. Trade Study del sistema de reconocimiento de voz..... | 74 |
| 2. Estadísticas..... | 75 |
| a. Estadística de prueba de aciertos del Google Speech API..... | 75 |
| b. Estadísticas probabilísticas de aciertos del Google Speech API..... | 75 |
| 3. Diagramas de flujo..... | 76 |
| a. Diagrama de sistema de reconocimiento de voz..... | 76 |
| b. Diagrama de sistema de reconocimiento de rostro | 77 |
| c. Algoritmo de compensación de sistema de reconocimientos (Off-sets)..... | 78 |
| B. Pruebas de detección de rostro..... | 79 |
| C. Resultado de brazos mecánicos de DANI..... | 82 |
| 1. Iteraciones de los diseños del brazo | 82 |
| 2. Dimensiones más relevantes | 84 |
| 3. Puntos críticos para análisis de esfuerzos | 85 |
| 4. Diagrama de corte y momento en voladizo..... | 86 |
| 5. Análisis de esfuerzos..... | 87 |
| a. Punto crítico 1 | 87 |
| b. Punto crítico 2..... | 89 |
| c. Punto crítico 3..... | 91 |
| d. Punto crítico 4..... | 93 |
| e. Punto crítico 5..... | 95 |
| f. Punto crítico 6..... | 98 |
| 6. Tornillos entre servomotor AX-12A superior con el accesorio F2222..... | 100 |
| a. Comprobación de punto crítico 4..... | 101 |
| 7. Análisis de esfuerzos para el resto de piezas críticas..... | 102 |
| 8. Análisis de tensión para los hilos de las garras | 102 |
| 9. Estudio de carga teórica vs carga experimental del servomotor AX-12A | 103 |

| | | |
|-------|--|-----|
| 10. | Estudio de ángulo teórico vs ángulo logrado para servomotor AX-12A | 103 |
| 11. | Tiempo que dura el AX-12A a un ángulo definido..... | 104 |
| VIII. | Análisis de resultados | 106 |
| A. | Componente de procesamiento de audio y video..... | 106 |
| 1. | Análisis de Trade Study del sistema de reconocimiento de voz | 106 |
| B. | Análisis de estadísticas | 108 |
| 1. | Estadística de prueba de aciertos del Google Speech API..... | 108 |
| 2. | Estadísticas probabilísticas de aciertos del Google Speech API..... | 109 |
| C. | Descripción analítica de diagramas de flujos..... | 110 |
| 1. | Diagrama de sistema de reconocimiento de voz..... | 110 |
| 2. | Diagrama de sistema de reconocimiento de rostro | 110 |
| 3. | Algoritmo de compensación del sistema de reconocimiento (Off-sets) | 111 |
| D. | Análisis de prueba de detección de rostro..... | 111 |
| E. | Análisis de resultado de brazos mecánicos | 113 |
| 1. | Análisis de esfuerzos..... | 113 |
| 2. | Análisis de tensión para los actuadores de la garra..... | 114 |
| 3. | Análisis de parámetros de los actuadores principales | 115 |
| IX. | Análisis de resultados de mecanismos para emular gestos faciales..... | 116 |
| A. | Resultados de ojos de DANI..... | 116 |
| B. | Resultados boca DANI | 124 |
| C. | Resultados cejas | 129 |
| D. | Resultados orejas | 130 |
| E. | Resultados estructura DANI | 131 |
| X. | Conclusiones..... | 134 |
| A. | Conclusiones de procesamiento de audio y video..... | 134 |
| B. | Conclusiones de diseño de brazos mecánicos | 134 |
| C. | Conclusiones de diseño de gestos faciales | 135 |
| XI. | Recomendaciones | 136 |
| A. | Recomendaciones para procesamiento de audio y video | 136 |

| | | |
|-------|--|-----|
| B. | Recomendaciones para diseño de brazos mecánicos..... | 136 |
| C. | Recomendaciones para diseño de gestos faciales | 137 |
| XII. | Bibliografía..... | 138 |
| XIII. | Anexos | 143 |
| A. | Anexo para procesamiento de audio y video | 143 |
| B. | Anexo para diseño de brazos mecánicos..... | 148 |
| C. | Planos mecanismos gestos faciales..... | 164 |
| XIV. | Glosario | 173 |

LISTA DE CUADROS

| | Página |
|--|--------|
| 1. Velocidad de obturación clasificados según la aplicación | 20 |
| 2. Configuración de aperturas con respectivos tiempos de obturador | 22 |
| 3. Modelos mecánicos de juntas con el movimiento que permiten y su representación gráfica | 32 |
| 4. Propiedades mecánicas del MDF | 43 |
| 5. Propiedades físico-mecánicas del MDF | 44 |
| 6. Propiedades físicas del PLA | 44 |
| 7. Propiedades mecánicas del PLA | 44 |
| 8. Tipos de juntas con sus grados de libertad | 57 |
| 9. Métodos de unión dependiendo del material | 58 |
| 10. Especificaciones técnicas del servomotor Dynamixel AX-12A | 60 |
| 11. Especificaciones técnicas del servomotor Dynamixel XL-320 | 61 |
| 12. Características mecánicas del PLA | 61 |
| 13. Propiedades mecánicas del MDF | 62 |
| 14. Trade study realizado del sistema de reconocimiento de voz | 74 |
| 15. Gráfica que muestra los aciertos correctos e incorrectos de Google Speech API | 75 |
| 16. Gráfica que muestra los aciertos probabilísticos de Google Speech API | 75 |
| 17. Resultados del resto de piezas críticas | 102 |
| 18. Resultados de tensión para los hilos de la garras y torque necesario para moverlos | 102 |
| 19. Carga teórica vs carga experimental del AX-12A superior | 103 |
| 20. Ángulo teórico vs ángulo experimental del AX-12A superior | 103 |
| 21. Tiempo que se mantuvo encendido el servo AX-12A antes de apagarse por sobrecarga | 104 |
| 22. Valor de las dimensiones de la cabeza | 119 |
| 23. Masas y peso total de la carga del ojo | 121 |
| 24. Ángulo de giro de la boca | 126 |
| 25. Valores para análisis del torque | 128 |

LISTA DE FIGURAS

| | Página |
|--|--------|
| 1. Organización de la arquitectura de Jasper | 9 |
| 2. “Visual servoing” basado en posición | 14 |
| 3. “Visual servoing” basado en imagen | 15 |
| 4. Red de poses relativas para “Visual servoing” basado en posición | 16 |
| 5. Dos vistas del cuadro de objeto de la meta | 17 |
| 6. Velocidad de obturación lenta | 21 |
| 7. Velocidad de obturación alta | 21 |
| 8. Apertura ancha de f/2-f-stop bajo con profundidad de campo angosto | 22 |
| 9. Apertura ancha de f/16-f-stop alta con profundidad de campo ancho | 23 |
| 10. Comparaciones de las velocidades ISO | 23 |
| 11. OOSXX Mini Security Camera 600 TVL | 24 |
| 12. Cámara Microsoft LifeCam Cinema | 25 |
| 13. Proceso de machine learning | 27 |
| 14. Planos de anatomía | 30 |
| 15. Movimientos en las extremidades de un ser humano | 30 |
| 16. Ejemplos de articulación fibrosa | 31 |
| 17. Ejemplo de articulación cartilaginosa | 31 |
| 18. Ejemplo de articulación sinovial | 32 |
| 19. Junta de bisagra | 32 |
| 20. Junta condílea | 32 |
| 21. Junta plana | 33 |
| 22. Junta de pivote | 33 |
| 23. Junta de silla de montar | 33 |
| 24. Junta esferoidea | 33 |
| 25. Partes de la cabeza | 34 |
| 26. Geometría básica del modelo de paralelogramo para una extremidad superior | 37 |
| 27. Modos de operación de un animatrónico | 38 |
| 28. Esfuerzo-deformación material dúctil | 41 |
| 29. Material frágil | 41 |
| 30. Proceso de obtención de PLA | 43 |
| 31. Unidades de medición para los esfuerzos | 45 |
| 32. Representación gráfica de los elementos de esfuerzos | 46 |
| 33. Planteamiento gráfico para ubicar los esfuerzos principales | 47 |
| 34. Ecuaciones de los esfuerzos principales, sus ángulos y su representación gráfica | 47 |
| 35. Ecuaciones de los esfuerzos cortantes máximos, sus ángulos y su representación gráfica | 48 |
| 36. Representación gráfica de la distribución de esfuerzos para un eje circular bajo torsión | 50 |
| 37. Representación gráfica de la distribución de esfuerzos para un eje tubular bajo torsión.. | 50 |

| | |
|--|-----|
| 38. Criterios para el valor de K y Q según la forma de la sección transversal | 51 |
| 39. Diagrama de fuerza cortante | 52 |
| 40. Ecuaciones para esfuerzo cortante de perfiles comunes | 52 |
| 41. Diagrama de fuerza cortante vertical, momento flexionante, pendiente y deflexión | 54 |
| 42. Expresiones para calcular la deflexión de una viga | 54 |
| 43. Junta universal, 2 GDL | 56 |
| 44. Eslabones de diferente orden | 56 |
| 45. Servomotor Dynamixel AX-12A | 59 |
| 46. Servomotor Dynamixel XL-320 | 60 |
| 47. Estructura de cabeza del animatrónico “NAVI” de Disney | 63 |
| 48. Cabeza de “DANI” | 73 |
| 49. Diagrama de flujo del sistema de reconocimiento de voz por medio de Google Speech API | 76 |
| 50. Diagrama de flujo del sistema de reconocimiento de rostro empleando OpenCV | 77 |
| 51. Diagrama de flujo de algoritmo de compensaciones del rostro | 78 |
| 52. Prueba de detección inicial con cuadro de detección recto | 79 |
| 53. Prueba de detección flexible al cambio de posición del rostro | 79 |
| 54. Prueba de detección inicial con cuadro de detección recto | 80 |
| 55. Prueba de detección con cuadro de detección en posición alterna | 80 |
| 56. Prueba de detección inicial con cuadro de detección recto | 81 |
| 57. Prueba de detección con cuadro de detección en posición alterna | 81 |
| 58. Primeras tres iteraciones del diseño del brazo mecánico | 82 |
| 59. Últimas dos iteraciones del diseño del brazo mecánico | 83 |
| 60. Dimensiones más relevantes del brazo mecánico | 84 |
| 61. Ubicación de los puntos críticos del brazo mecánico | 85 |
| 62. Diagrama de corte y momento para el brazo completamente en voladizo | 86 |
| 63. Resultados obtenidos del diagrama de corte y momento | 87 |
| 64. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 1 | 87 |
| 65. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 1 | 88 |
| 66. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 2 | 89 |
| 67. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 2 | 90 |
| 68. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 3 | 91 |
| 69. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 3 | 92 |
| 70. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 4 | 93 |
| 71. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 4 | 94 |
| 72. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 5 | 95 |
| 73. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 5 | 96 |
| 74. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 6 | 98 |
| 75. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 6 | 99 |
| 76. Representación gráfica del elemento de esfuerzos en el tornillo | 100 |
| 77. Representación gráfica del elemento de esfuerzos en el punto crítico 4 | 101 |
| 78. Porcentaje de error del torque y ángulo del servo AX-12A superior | 104 |

| | |
|---|-----|
| 79. Tiempo que se mantuvo encendido el AX-12A antes de apagarse a cierto ángulo | 105 |
| 80. Mecanismo de ojos | 116 |
| 81. Mecanismo de cuatro barras analizado | 117 |
| 82. Mecanismo de cuatro barras, movimiento horizontal y vertical del ojo | 117 |
| 83. Dimensiones de la cabeza, vista frontal | 118 |
| 84. Dimensiones de la cabeza, vista lateral | 118 |
| 85. Mecanismo de ojos diseñado para “DANI” | 119 |
| 86. Ensamble del ojo junto al párpado | 120 |
| 87. Análisis de flexión en el perno | 121 |
| 88. Deformación del perno | 122 |
| 89. Ensamble de junta universal con el ojo que sostiene la cámara | 122 |
| 90. Ensamble de componente interno del ojo | 123 |
| 91. Mecanismo de boca, primera iteración | 124 |
| 92. Mecanismo de la boca de “DANI” | 125 |
| 93. Ángulo de apertura y distancia máxima de la boca | 125 |
| 94. Boca de “DANI” | 126 |
| 95. Ángulo de giro vs ángulo de apertura | 127 |
| 96. Ángulo de giro vs apertura de la boca | 127 |
| 97. Mecanismo de cejas, primera iteración | 129 |
| 98. Mecanismo de las cejas | 130 |
| 99. Mecanismo de orejas | 131 |
| 100. Aplicación de cargas | 132 |
| 101. Garganta de soldadura, y carga a tensión y cortante | 132 |
| 102. Momentos en la estructura | 133 |
| 103. Vista general de OOSXXX Mini Security Camera 600TVL | 143 |
| 104. Dimensionamiento de OOSXXX Mini Security Camera | 143 |
| 105. Especificación de conexión eléctrica de OOSXXX Mini Security Camera | 144 |
| 106. Ubicación de la cámara © OOSXXX para el reconocimiento de rostro | 144 |
| 107. Ubicación de la cámara (A) OOSXXX, (B) Vista aérea de los ojos | 145 |
| 108. Cámara OOSXXX sujeto al mecanismo interno del ojo | 145 |
| 109. Cámara OOSXXX sujeta al mecanismo, vista lateral | 146 |
| 110. Cámara OOSXXX integrada dentro del ojo | 146 |
| 111. Cámara OOSXXX vista externa frontal | 147 |
| 112. Especificaciones de Microsoft LifeCam Cinema | 147 |
| 113. Fotografía del brazo armado | 148 |
| 114. Fotografía del setup para las pruebas de los dedos | 148 |
| 115. Explosión del brazo completo | 149 |
| 116. Dimensiones de todo el brazo armado | 150 |
| 117. Dimensiones de todo el brazo armado desde otra vista | 151 |
| 118. Pieza F2222, estándar (no se indican dimensiones) | 152 |
| 119. Accesorio para el servomotor AX-12A | 153 |
| 120. Servomotor AX-12A, estándar (no se indican dimensiones) | 154 |

| | |
|--|-----|
| 121. Pieza F333, estándar (no se indican dimensiones) | 155 |
| 122. Pieza acople hembra, estándar (no se indican dimensiones) | 156 |
| 123. Pieza de acople macho, estándar (no se indican dimensiones) | 157 |
| 124. Accesorio del servo XL-320 | 158 |
| 125. Servomotor XL-320 | 159 |
| 126. Pieza codo-muñeca | 160 |
| 127. Pieza separador | 161 |
| 128. Pieza separador 2 | 162 |
| 129. Estructura de los dedos | 163 |

RESUMEN

El proyecto “DANI” es un animatrónico que imita a un dragón de aspecto infantil, con dos modos de operación: en el primer modo, es capaz de interactuar de forma auditiva y visual con un usuario; y en el segundo, implementa el sistema de teleoperación. Todo esto con la finalidad de que el robot sea capaz de dar la bienvenida a las personas. Se plantea que las partes a movilizar sean las orejas, cejas, ojos, boca, brazos, garras, y cola.

Para la elaboración de los mecanismos de la cabeza, se tuvo como primer requerimiento el espacio de trabajo, ya que debían adaptarse a la cabeza de “DANI”, la cual fue elaborada por artistas guatemaltecos. Los mecanismos se elaboraron con materiales de fácil mecanizado, de bajo peso y fácil acceso. Los mecanismos se lograron adaptar a las condiciones de la cabeza y así lograr el objetivo de imitar los gestos faciales.

Para lograr la interacción de forma audiovisual, se desarrolló un algoritmo para interpretación de señales provenientes de dispositivos de audio y video. El sistema de reconocimiento de voz implementa el Google Speech API, que es seleccionado en base al método de Pugh con trade study. El sistema provee un rendimiento constante en reconocer los comandos de nivel de complejidad que involucra una mezcla de 3 términos con acierto de 70% a 80%, mientras que el nivel de complejidad con combinación de 4 términos o mayor provee aciertos de 33%. El sistema de reconocimiento de rostro hace el uso de OpenCV para distinguir el rostro del usuario, y algoritmo del sistema para calibración durante la operación del reconocimiento utiliza las compensaciones u “Off-sets”. Las cámaras Microsoft LifeCam Cinema y OOSXXX Mini Security Camera son implementadas para el sistema, con diferentes calidades que se observan en las Figuras 54 al 57 con la plataforma de Python. La prueba con la plataforma de Matlab se observa en las Figuras 52 y 53.

Además, se describe el proceso de diseño mecánico de los brazos y garras del megaproyecto “DANI”, responsable de brindar las extremidades superiores a un animatrónico con materiales fáciles de maquinar y de bajo costo. Para ello se diseñó una estructura que se adaptara al diseño de un muñeco realizado por un artista nacional. Se consideraron cuatro grados de libertad para su movimiento con los servomotores Dynamixel AX-12A para dos grados en los hombros y los Dynamixel XL-320 para un grado en el codo y otro en la apertura de la garra. Se obtuvieron cinco iteraciones para llegar al diseño final en donde el peso total fue de 64.45 N (657.13g), habiendo seis puntos críticos obtenidos a partir del diagrama de corte y momento del modelo de una viga empotrada para el peor caso que fue el brazo completamente en voladizo.

El torque necesario para levantar el brazo fue de 1.375 N*m lo que representó aproximadamente el 92% de la capacidad de los servomotores. Esto al mismo tiempo provocó que los servomotores no soportaran más de ocho segundos en cualquier posición mayor o igual a 70° medidos desde el brazo en posición vertical hacia abajo. A partir de lo anterior, fue posible diseñar un brazo mecánico basado en el diseño de un artista nacional, pero se recomienda utilizar servomotores de torque más alto para evitar problemas de sobrecarga.

I. INTRODUCCIÓN

El trabajo de graduación que se presenta a continuación forma parte del megaproyecto titulado “Diseño e implementación de un Dragón Animatrónico Interactivo con fines de promover los programas tecnológicos de la Universidad del Valle de Guatemala DANI”, con el objetivo principal de diseñar e implementar un animatrónico interactivo con el fin de promover los programas STEM ofrecidos por la Universidad. Dada la diversidad de opciones que existen de plataformas del entorno de programación, y diseños mecánicos según sea el enfoque de la base de desarrollo del animatrónico, el megaproyecto DANI se conforma de seis módulos esenciales.

Cada uno de los seis módulos domina áreas específicas del funcionamiento del animatrónico, en el cual el robot tendrá un diseño adaptativo según sea el entorno de operación y las características de comunicación que se establece con el usuario presente en la perspectiva del animatrónico. Los modelos de prueba diseñados para estos módulos fueron para escenarios o ambientes robustos con fuertes condiciones de perturbaciones o sobrecargas para validar el funcionamiento sólido del rendimiento, la robustez de los diseños y algoritmos.

El módulo de desarrollo e implementación de algoritmos de interacción entre un humano y un animatrónico se basó en el procesamiento de audio y video para el desarrollo de un programa que interprete señales provenientes de dispositivos de audio y video. El estudio por medio de trade study con el método de Pugh, para la selección de la mejor opción para el sistema de reconocimiento de voz se logró implementar con funcionamiento con el usuario presente. Además, se desarrolló el programa empleando recursos de software existentes, con la implementación de OpenCV, para el reconocimiento facial por medio de la cámara OOSXXX y prueba con LifeCam Cinema en el entorno de programación de Python.

Para el diseño e implementación del movimiento mecánico de cejas, ojos, boca y orejas, los cuales deben ser capaces de emular los gestos faciales, se realizaron iteraciones previas para mejorar la movilidad de los mismos y llegar a los diseños finales, los cuales se acoplaron a la estructura del dragón, elaborada por artistas nacionales. Asimismo, se fabricó con materiales de fácil mecanizado, como también de fácil acceso. Entre ellos está el MDF y el PLA, los cuales se manufacturaron en la cortadora láser e impresora 3D de la Universidad del Valle de Guatemala. Para la movilidad realística de los mecanismos se utilizaron componentes que ofrecieran varios grados de libertad para que el movimiento fuese más suave y además cumpliera con otro requerimiento el cual fue de ensamblaje. Por último, a los mecanismos se les realizó un análisis superficial de esfuerzos debido a su bajo tamaño y peso con las condiciones que presentan en operación de que no existan cargas representativas que generen una elevación en los esfuerzos, asegurando que los mecanismos funcionan correctamente.

Ahora bien, en la parte mecánica de los brazos se pretendió mover los dos brazos robóticos con servomotores para dotarlo de cuatro grados de libertad: hombro (dos grados), codo (un grado) y garra (un grado) sin la habilidad de sostener objetos en el manipulador. Por lo tanto, la posición final de los brazos fue completamente horizontal tanto al frente como hacia los lados sin poder girar hacia atrás. Todo el mecanismo se adaptó a un muñeco elaborado por un artista nacional, por lo que del mismo se obtuvieron las dimensiones generales del brazo. Luego se diseñó en Autodesk Inventor un modelo en 3D del brazo para realizar un análisis de esfuerzos, determinar los puntos críticos y determinar el torque necesario en los motores para levantarlo. Al final, se logró realizar el brazo robótico adaptado al modelo estético del artista, pero se recomienda cambiar los motores seleccionados por otros más adecuados.

II. OBJETIVOS

A. Objetivo general

Diseñar e implementar un animatrónico interactivo con el fin de promover los programas STEM ofrecidos por la Universidad.

B. Objetivo específico

- Desarrollar un programa que interprete señales provenientes de dispositivos de audio y video.
- Diseñar e implementar el mecanismo que imite el movimiento de brazos y garras.
- Diseñar e implementar mecanismos para ojos, boca, orejas y cejas que emulen gestos faciales.

III. JUSTIFICACIÓN

Un animatrónico es un robot que emula el comportamiento de un ser viviente, sea este una persona o un animal, pudiendo ser ficticio o no. Y por tal motivo, se suele utilizar este tipo de robots en ambientes recreativos como parques de diversiones. Ahora bien, específicamente en Guatemala, este tipo de robots no se suelen realizar mientras que en el extranjero sí.

Entonces, los motivos por los cuales se realizó este animatrónico es para demostrar la posibilidad de diseñar, fabricar e implementar estos robots, de bajo costo, hecho por estudiantes del país en colaboración con artistas nacionales, para los parques de atracción de Guatemala; y, además, para demostrar los programas que ofrece la Universidad del Valle de Guatemala en el área de la tecnología.

Los dispositivos electrónicos son cada vez más intuitivos al momento de conectarse y compartir con las personas para involucrarse con el entorno, tal como los robots. El animatrónico, tiene la capacidad de poder comunicarse e interactuar con una persona pudiendo reconocer sus intenciones, de hecho, es aquí en donde el sistema de reconocimiento de voz y de rostro entra en juego. Todo dispositivo de audio o video tiene que estar acompañado de un programa encargado de procesar e interpretar la información recibida. Esto sirve para poder interpretar gestos humanos de parte de los usuarios y entablar una charla entre el animatrónico y el usuario. Asimismo, para que el animatrónico responda a gestos o palabras se utilizará un recurso existente de reconocimiento de voz debido a que es de código abierto, con actualizaciones constantes, adecuado y de fácil uso.

Uno de los procesos de mayor importancia en el diseño de un animatrónico es el diseño de los mecanismos que mejor repliquen los movimientos de un ser vivo. Entre estos movimientos están el de los brazos y las garras. Además, se debe elegir en base a criterios de diseño mecánico los materiales que mejor soporten los esfuerzos a los cuales estarán sometidos.

El diseño de brazos y garras, es importante porque sirve para llevar a cabo la comunicación no verbal, proceso en el cual no se usan las palabras, sino sólo movimientos corporales, por ejemplo, algo tan cotidiano como el saludo no se podría realizar sin los brazos ni las manos. Es por eso que el diseño de mecanismos debe emular el aspecto antropomórfico de un brazo completo, ya que debe parecer lo más humano posible y tener una buena interacción con los usuarios.

IV. MARCO TEÓRICO

A. Procesamiento de señales (audio y video)

1. Lenguajes de programación

a. Plataforma de programación

1) Python. Python es un lenguaje de programación de propósito general usualmente aplicado en el rol de guion o codificación. Es comúnmente definido como lenguaje de guion orientado a objetos, que define de tal forma que dirige rol de guion con mezcla de soporte sobre la programación orientado a objetos. Entonces, Python se define como el lenguaje de programación de propósito general que combina paradigmas de orientado a objetos, funcional y de desarrollo. (Python)

La referencia de lenguaje de Python describe la sintaxis y la semántica del lenguaje de Python, y librerías estándar de Python describe la librería estándar que se distribuye con la distribución de Python. La librería contiene módulos pre-integrados codificados en C que provee el acceso a la funcionalidad del sistema tales como las entradas y salidas que son inaccesibles por programadores de Python, y módulos que resuelven soluciones estandarizados para errores que genera en el ambiente de programación. Algunos módulos son diseñados de forma explícita para mejorar la portabilidad de programas de Python abstrayendo de específicos de las plataformas en APIs neutrales de plataforma. (Python)

2) Matlab. Matlab es la plataforma optimizado para soluciones de ingeniería y problemas científicas. La plataforma con implementación del lenguaje basado de matrices con el fin de expresar matemática computacional lo más natural posible. Esto analizar y diseñar los sistemas tales como sistemas de seguridad de los automóviles, vehículos espaciales interplanetario, procesamientos de señales y de imágenes, visión de la computadora, comunicaciones, finanzas computacionales, diseño de controles, y robótica como uno de los ejemplos más representativos de la implementación de Matlab. (MathWorks)

La plataforma tiene interfaz gráfica para facilitar análisis visual sobre datos visualizados, como puntos específicos o regiones de la gráfica. Posee librería predeterminada extensa de herramientas para proveer inmediatamente algoritmos matemáticos según sea la operación deseada. Entonces, la plataforma provee lenguaje de alto nivel para operaciones computacionales de ingeniería y científico, con herramientas adicionales que son posibles de obtener de rango amplio de aplicaciones de ingeniería y científica. La interfaz es posible enlazar con interfaces C/C++, Java, .NET, Python, SQL, Hadoop y Microsoft Excel. (MathWorks)

3) Arduino IDE. El Arduino IDE o Arduino Software, IDE que significa Entorno de Desarrollo Integrado es la plataforma que contiene un editor de texto para desarrollo del código, área de mensajes, consola de texto, barra de herramientas con botones para funciones comunes y una serie de menús. Esto conecta a los dispositivos de Arduino y Genuino para proceso de carga del programa y comunicación entre ellos. (Arduino)

2. Sistemas de reconocimiento de voz

a. API (Application Programming Interface)

1) Google Cloud Speech API. Google Speech API, en donde el API es Interfaz de Programación de Aplicación, es de la tecnología de reconocimiento de voz de parte de Google que se integra en la parte de Cloud Speech API. Este API forma entonces la parte de las aplicaciones de los desarrolladores, y permite al usuario enviar el audio captado en presente y recibir una transcripción del servicio. (Google)

El proceso de la conversión del audio captado al texto es realizado por medio de los modelos de redes neurales dentro del entorno de API. El API logra reconocer diferentes idiomas, y esto para que usuarios de diferentes regiones puedan utilizar en la parte del desarrollo de un sistema de aplicación. El desarrollo de la aplicación puede ser orientado para transcribir en un texto de parte del usuario quién está dictando por medio de un micrófono, o tener control por medio de comandos de voz. (Google)

El tema de Machine Learning se encuentra dentro de Google Cloud Speech API, el cual aplica el algoritmo de redes neuronales de aprendizaje profunda avanzada sobre el audio generado por el usuario para ofrecer alta precisión de parte de reconocimiento de voz. La fuente del audio puede ser en vivo, es decir captado por micrófono por el usuario y enviado en momento presente; o a partir de un archivo de audio pregrabado. Entonces, en caso de que el usuario esté hablando sobre el micrófono, el API devuelve los resultados de reconocimiento en vivo. El ruido del ambiente es manejado por un proceso que realiza al estar procesando el audio, captando partes relevantes del audio y cancelando el ruido grabado. (Google)

2) IBM Watson API: Speech to Text. IBM Speech to Text es el servicio que provee un Interfaz de Programación de Aplicación (API) para integrar la capacidad de transcribir el audio de una voz según las necesidades de la aplicación. Esto aprovecha la inteligencia de la máquina para analizar la información proveída por parte del audio, teniendo en consideración composición entre la gramática y la estructura del idioma dependiendo de la región. Esto tiene la característica de devolver respuesta y actualizar está respuesta activamente por medio de una transcripción hasta que el API haya detectado una pausa larga de reconocimiento de voz. (IBM)

La capacidad de reconocimiento de voz de IBM transcribe entonces de varios lenguajes formatos de los audios en un texto con baja latencia. Para mayoría de los lenguajes, el API que conecta el servicio esté reconocimiento de voz, soporta de 2 tasas de muestreo, y sea de banda ancha o corta. El API consiste de grupos de llamadas en los cuales las llamadas son las funciones que realizan trabajos específicos. Entonces, de las llamadas existen los modelos, WebSockets, Sessionless, Sessions, Asynchronous, Custom Models, Custom Corpora, y Custom Words. (IBM)

De este servicio de API, Websocket, HTTP e interfaces HTTP Asíncrono comparte características similares para realizar el proceso de transcribir el audio de la voz al texto. Entonces tal como dispositivos electrónicos pueden tratar sus características con entradas y salidas, estas interfaces es posible representar de esta forma también. (IBM)

Las entradas de las interfaces caracteriza por idiomas, donde soporta Francés, Japonés, Mandarín, Chino, Árabe estándar moderna, Español, e Inglés de Estados Unidos o Reino Unido; para mayoría de estos idiomas, soporta la banda ancha al cual el audio es muestreado a una tasa mínimo menor de 16kHz, e igual a banda corta al cual el audio es muestreado a una tasa mínimo menor de 8kHz; soportan diferentes tipos de formatos de Audio como FLAC, PCM, WAV, WebM o archivo de audio básico; y transmisión de audio puede pasar una gran cantidad de datos de audio o de un sólo envío, teniendo en fin dato completo en una sola vez. (IBM)

De las salidas de interfaz ofrece filtro de profanidad, resultados con opciones de alternativas para ajustar mejor al escenario, y reconocimiento inteligente conocido como “Smart Formatting” para convertir los datos reconocidos según sea el formato de números en formas más convencionales fáciles de entender. (IBM)

3) PocketSphinx. PocketSphinx es una librería y un motor de reconocimiento de voz que utiliza pocos recursos de espacio con especial enfoque sobre los dispositivos móviles. Sin embargo, este funciona de forma equivalente en equipos de escritorio. De hecho, entre las aplicaciones que tiene esta herramienta es el control por voz en computadoras de escritorio, dispositivos automovilísticos, casas inteligentes, transcripción de voz, traducción de voz, búsqueda por voz y aprendizaje de lenguajes. (CMUSphinx)

Esta herramienta depende de otra librería llamada Sphinxbase que provee la funcionalidad básica del reconocimiento. Para utilizar PocketSphinx es necesario instalar tanto PocketSphinx propiamente como Sphinxbase en equipos basados en Linux, MacOS, Android, iPhone o Windows. Su funcionamiento está basado en tres ideas principales, que son: (CMUSphinx)

- Es mucho más probable que se mantenga estable tanto en términos de fuentes como de compatibilidad binaria debido al uso de tipos abstractos.
- Tiene la opción de reconocer nuevas entradas para que no ocurran problemas al tener varios decodificadores en el mismo proceso.
- Ha permitido una drástica reducción en el tamaño del código y una considerable, pero modesta reducción en el consumo de memoria.

Para el uso avanzado del API se toman en cuenta las siguientes consideraciones: (CMUSphinx)

- Se provee de un iterado, para analizar segmentos de palabras con el fin de, como su nombre lo indica, iterar sobre secuencias de palabras. Dicho iterado es de tipo abstracto junto con varios accesorios que permiten obtener momentos en el tiempo, puntuaciones y probabilidades posteriores para cada palabra.
- De ser necesario, se puede acceder a la “rejilla”.
- Tiene métodos para acceder a la pronunciación.
- Se pueden configurar varias búsquedas y navegar entre ellas en tiempo real.

Específicamente para búsquedas, el programador puede configurar varios parámetros de búsqueda con diferentes gramáticas y modelos de lenguaje, así como intercambiarlos en tiempo real para brindar una experiencia más interactiva al usuario. Y entre los modos de búsqueda están: (CMUSphinx)

- keyword - permite buscar eficientemente una palabra clave en el dictado ignorando el resto. También permite definir un nivel de detección.
- grammar - reconoce el dictado de acuerdo a la gramática JSGF (Java Speech Grammar Format), que es una representación textual de la gramática para el uso de tecnologías de reconocimiento de voz. A diferencia de la metodología con keyword, aquí no se ignoran las palabras que no están en la gramática y, además, trata de reconocerlas.
- ngram/lm - reconoce dictado natural con un modelo de lenguaje.
- allphone - reconoce fonemas con un modelo de lenguaje fonético.

4) Snowboy. Snowboy caracteriza por un motor de detección de palabra clave, o conocido como “Hotword”, que es altamente personalizable. Esto es embebido en tiempo real y está siempre escuchando, aun cuando no hay conexión de línea, y es compatible con plataformas como Raspberry Pi, Ubuntu basado de Linux, Max OS X. (Snowboy)

El Hotword, reconocido como palabra de activación, es el término o frase clave que funciona como una señal para activar dichos eventos o acciones y es revisado constantemente según la plataforma que esté siendo

utilizado. Así los ejemplos como “Alexa” de Amazon Echo, “Ok Google” de los dispositivos que utilizan servicios de Google como base, y “Oye Siri” de iPhones son clasificados como Hotword. Estos son formas de iniciar un interfaz de interacción vocal entre el usuario y el dispositivo; dando la opción de realizar acciones de dar comandos y control sobre dichos eventos. (Snowboy)

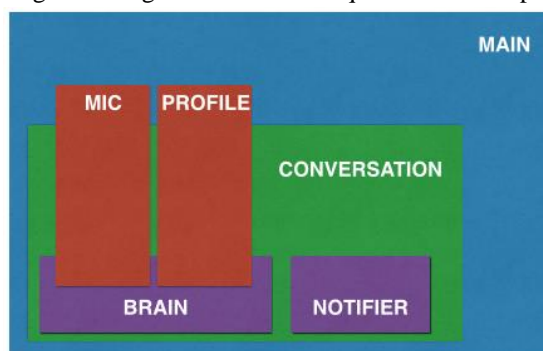
Además de tener la base de reconocimiento de voz, el API no requiere altas demandas de recursos de computación y ancho de banda. Los recursos que demanda en la plataforma de Raspberry Pi, con modelo Single-Core 700MHz ARMv6, es menor a 10% sobre CPU. Este aspecto viene en consideración como parte de la ventaja debido a la existencia de la solución de Reconocimiento de Voz Automático para realizar el trabajo de reconocer la palabra de activación. La desventaja que presenta en contra de Snowboy es que el proceso requiere recursos de forma contraria a diferencia de Snowboy. (Snowboy)

En otro mano, dado los Reconocimientos de Voz Automático basan en soluciones de nube, no provee la seguridad sobre la privacidad del usuario; al cual el Snowboy provee la seguridad tomando la ventaja de que el proceso de mejora de detección de palabra de activación se realiza por medio de aprendizaje profunda realizada en línea, pero para implementación requiere baja de archivo e implementar el algoritmo de Snowboy; así eliminando la necesidad de estar conectado a la red. (Snowboy)

5) Jasper. Jasper es diseñado específicamente para la plataforma de Raspberry Pi Modelo B. Para el funcionamiento completo del módulo de Jasper requiere dispositivos adicionales tales como el adaptador de WiFi y micrófono con conexión USB. Adicionalmente requiere una tarjeta SD de 4GB de memoria, cable Ethernet o un módulo de WiFi si llegará a poder expandir la cantidad de puertos USBs, cable micro-USB, cargador de pared con USB, y bocinas que podrá funcionar por medio de enchufe de audio de Raspberry Pi. (Shubhro Saha)

La arquitectura de Software de Jasper es organizada en diferentes niveles de los componentes de la Figura 1. La arquitectura conforma de micrófono, perfil, cerebro, notificador, conversación, y main.

Figura 1. Organización de la arquitectura de Jasper



(Shubhro Saha)

El algoritmo de Jasper en entorno de Python (`jasper.py`) es el que dirige todas las funcionalidades del mismo. Esto crea perfil, el micrófono e instancias de conversaciones. La instancia de la conversación es alimentada al micrófono y el perfil, el cual comporta como la entrada, desde el cual genera el notificador y el cerebro. El cerebro recibe el micrófono y el perfil que deriva originalmente del principal y carga todos los componentes interactivos en la memoria. El cerebro es esencialmente la interfaz entre los módulos hechos por los desarrolladores y estructural principal. Cada módulo necesita la implementación de `isValid()` y `handle()`, y definición de lista de `WORDS = [...]`. (Shubhro Saha)

La configuración inicial de Jasper es en ensamblaje del dispositivo en donde ingresa la tarjeta SD con alimentaciones al Raspberry Pi, y periféricos de entrada y salida que son el micrófono y la bocina. El cable Ethernet es utilizado para acceder al Raspberry Pi desde una computadora remota en el proceso de instalación del programa. Luego de la instalación, es posible utilizar la conexión a la red inalámbrica, el cual depende de la preferencia del usuario. (Shubhro Saha)

Para el uso del Jasper, requiere generar un perfil y selección de un motor. La generación de perfil basa en un archivo de configuración con la necesidad generar reportes con precisión tales como las condiciones del clima del área, envió de mensajes de texto, y más funciones que trae por defecto el Jasper. Los motores en este contexto refieren al motor STT y TTS. STT es “Speech-to-Text” el cual Jasper utiliza. Básicamente el motor STT es un fragmento de código que toma el audio grabado y convierte en un texto escrito. La precisión de detección varía según la calidad del micrófono o una configuración incorrecta de motor STT. Por lo tanto, la selección de un motor STT para funcionamiento de Jasper es crucial. Además, Jasper propone ser modular, por lo cual ofrece variedad de motores STTs en los cuales son PocketSphinx, Google STT, AT&T STT, Wit.ai STT, y Julius. (Shubhro Saha)

Pocketsphinx en contexto de integración o como parte de Jasper, se ve igual como un decodificador de voz de fuente abierta por parte de CMU Sphinx Project; y esto funciona de fluido debido a la naturaleza del diseño del algoritmo en sistemas embebidos. Google STT es el sistema que ofrece de parte de Google para convertir la voz en un texto, y es el exactamente el mismo motor que realiza operaciones de reconocimiento al decir “Ok, Google” de los teléfonos con sistema operativo Android. AT&T STT es el decodificador de voz de la compañía de telecomunicaciones AT&T, y al igual que Google realiza la decodificación en línea implicando la conexión de red obligatoria. Wit.ai STT basa en servicio de nube de wit.ai y utiliza el método de fuente de multitud de personas para entrenar el algoritmo de sistema de reconocimiento, y requiere conexión internet constantemente. Julius es un motor de reconocimiento de voz de fuente abierta de alto rendimiento y no requiere una conexión de inter activa. Al hacer el uso de Julius en especial como el motor STT de parte de Jasper, es necesario entrenar por cuenta del usuario el propio modelo acústico. (Shubhro Saha)

Todos estos motores STTs son posibles de selección con Jasper. Al contrario, a excepción de Pocketsphinx y Julius, los demás motores STT transfieren los datos de micrófonos, es decir audio de voces grabados, a la red. Entonces, si el usuario no desea que escuche a la voz del sí mismo no se recomienda utilizar motores STTs de Google, Wit.ai, y AT&T. Estas actividades de los motores STTs se conoce como audición pasiva, y de la perspectiva de rendimiento y de privacidad, se recomienda el uso de Pocketsphinx o Julius. A parte de los motores STT, los motores TTS son también usado por Jasper y realiza la operación contraria de un motor STT. El motor TTS toma el texto escrito y transforma en una voz y Jasper soporta diversidad de motores TTS que difiere por voz, tonalidad y de algunas características de voz alterables. (Shubhro Saha)

El eSpeak es un sintetizador de voz de fuente abierta compacta y la síntesis de voz es realizado sin línea. El Festival hace el uso del sistema de Festival Speech Synthesis y realiza la sintetización de la voz sin línea o sin necesidad de la conexión a la red. El Flite utiliza CMU Flite (festival-lite), que es un motor de baja peso y sintetización rápida que tiene diseño enfocado primariamente para las máquinas embebidas de pequeñas escalas. SVOX Pico TTS es el motor TTS utilizado en la versión de Android 1.6 “Donut” y es una aplicación de pequeña de fuente abierta y funciona sin la necesidad de conexión a la red. Google TTS, a diferencia de SVOX Pico TTS, es API que se encuentra implementado en los dispositivos de Android más recientes, y la operación de síntesis es procesada en el servidor de Google implicando conexión a la red. Ivona TTS utiliza el servicio de la nube de Amazon llamado Ivona Speech Cloud que se encuentra implementado en Kindle Fire. Mary TTS es el sistema TTS de fuente abierta desarrollado en base a Java y requiere configurar propio servidor de MaryTTS para que Jasper pueda hacer el uso de esto. Mac OS X TTS únicamente funciona siempre y cuando utiliza Jasper en sistema de operativo Mac. Al igual que el caso anterior de motor STT, si desea utilizar un motor TTS con seguridad sobre la privacidad, evita utilizar servicios con conexiones a la red obligatoria. (Shubhro Saha)

Jasper varía según la necesidad del usuario y existe API de desarrollador. El API de desarrollador es posible utilizar para crear módulos propios que hace el uso de control por medio de la voz. El módulo estándar es en que la interacción es activa y los usuarios inician la comunicación con Jasper y da comandos para ejecutar de los módulos que han creado o predeterminado. El módulo de notificación es pasivo en donde el Jasper realiza monitoreo discreto de un flujo de informaciones, tal como revisión de interrupciones por correos electrónicos o actividades de redes sociales y hablar cuando haya necesidad de reportar. (Shubhro Saha)

3. Sistemas de reconocimiento facial

a. **OpenCV**. Es la librería de OpenCV (Open Source Computer Vision Library) para brindar una infraestructura común para el desarrollo de aplicaciones de visión por computadora y software de machine learning. Su principal enfoque es el desempeño en tiempo real y eficiencia computacional. Tiene interfaces de Python, C, C++ y puede ser instalado en Android, iOS, Linux, MacOS y Windows. Esta librería tiene más de 2500 algoritmos, algunos de ellos optimizados para: (OpenCV)

- Detección de rostros.
- Reconocimiento de rostros.
- Identificación de objetos.
- Clasificación de acciones humanas en video.
- Seguimiento de movimientos de cámara.
- Seguimiento de objetos en movimiento.
- Extracción de modelos 3D de un objeto.

El OpenCV posee la estructura modular, el cual la librería o el paquete se desarrolla encima de diversas librerías estáticas o compartidas. Los módulos del OpenCV son core, imgproc, video, calib3d, features2d, objdetect, highgui, y gpu. Módulos adicionales son de soporte, es decir “wrappers” para conversiones de lenguajes u otros objetos de comunicación, FLANN, y entre otros más. (OpenCV)

El “core” es el módulo compacto para definición de estructura de datos básicos que incluyen arreglos multi-dimensionales densos matemáticos y funciones básicas usadas por demás módulos. El “imgproc” es el módulo de procesamiento de imágenes que incluyen filtros lineales y no lineales, transformaciones geométricas de las imágenes para cambios de propiedades de resolución y perspectiva. El “video” dedica a análisis del video que incluyen aproximación del movimiento, sustracción del fondo, algoritmos de seguimiento de objetos. El “calib3d” es paquete de algoritmos geométricos de vistas múltiples, con calibración de cámara estéreo y singular, aproximación de la pose de objeto, algoritmos de correspondencia estéreo, y reconstrucciones de elementos de tres dimensiones. El “features2d” son detectores y descriptores. El “objdetect” contiene clases predefinidas de detección de objetos e instancias. El “highgui” es la interfaz intuitiva para captura del video, codificadores del video e imágenes con capacidades simples de interfaz del usuario. Por último, el “gpu” son algoritmos acelerados por GPU desde diferentes módulos de OpenCV. Todo esto completa lo que es la librería OpenCV. (OpenCV)

El OpenCV entonces provee funciones para desarrollo de aplicaciones de visión por computadora. Las funciones de OpenCV implementados forma parte del desarrollo para la detección de rostros. La detección de rostros cumple con requisitos en que pueda distinguir el rostro de la persona dado un espacio permitido.

La restricción espacial se debe a prevención de detección de múltiples rostros en un espacio abierto y enfocarse a un usuario relevante en evento de comunicación. Las funciones básicas implementadas son las siguientes. (OpenCV)

- CascadeClassifier(.xml)
- VideoCapture()
- cvtColor()
- imshow()
- waitKey()

Cada una de las funciones basa en obtener datos de parte de dispositivo de captura de imagen, parámetros de rostro, y representación visual de la perspectiva de la cámara. CascadeClassifier () es una clase dedicado para detección del objeto y carga un classifier a partir de un archivo de parámetros de detección. El archivo de parámetros de detección para los rostros utilizó archivos de extensión .xml, el cual contiene configuraciones necesarias para distinguir el rostro. La función VideoCapture () es la clase para las capturas tomadas por medio del dispositivo de cámara, y es implementado para especificar por medio de identificador numeral cuál dispositivo de captura será utilizado para la operación. (OpenCV)

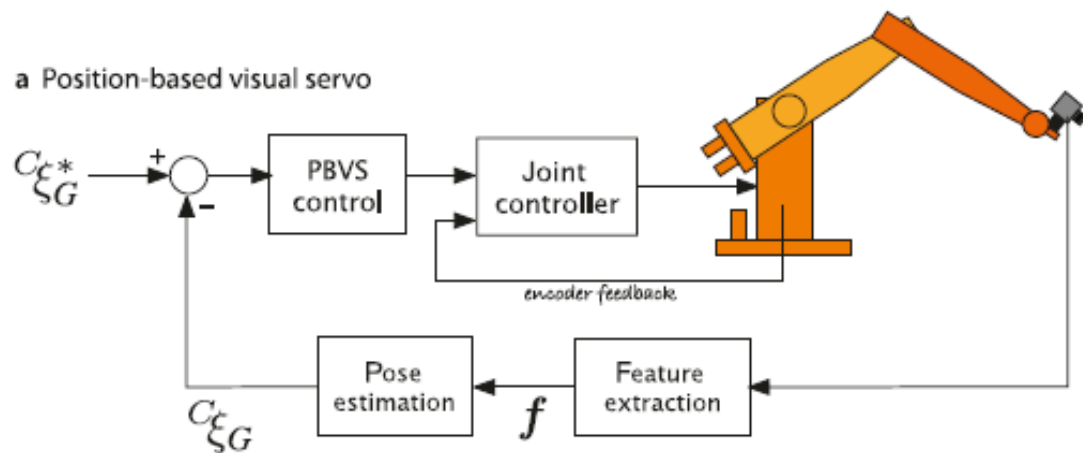
La función cvtColor() convierte la imagen, en este caso son las capturas del video, de un espacio de color a otro. Esta función para propósito de detección de rostro realiza la conversión de RGB a GRAY, donde realizan transformaciones dado el espacio RGB por medio de operaciones de agregar o eliminar canal Alpha, inversión del orden del canal, y conversión desde o hacia color RGB de 16 Bits. La función imshow() despliega la imagen en una ventana específica; y es implementado con el único fin de poder visualizar la perspectiva del cámara y el funcionamiento correcto de detección del rostro. Al igual que el propósito de imshow(), la función waitKey() espera el evento de la activación de teclado para generar un nuevo evento o acción. (OpenCV)

b. **Visual Servoing.** El sistema de reconocimiento Facial conlleva la idea general del sistema de reconocimiento por medio de la visión por defecto y la herramienta que establece la implementación de este sistema es el OpenCV. Al contrario, el reconocimiento de Facial únicamente distingue el rostro y esto mismo es el resultado. El resultado presenta las coordenadas respectivas sobre un plano 2D, y esto es el parámetro por controlar. Es decir, los ojos del animatrónico debe poder seguir el rostro del usuario cuando encuentra en un rango efectivo. El controlador entonces es el que se encarga de enviar valores requeridos a los servos motores para traducirlo luego en un movimiento de los ojos. El término para describir el sistema de utilización de los resultados de reconocimiento de rostro enviando a los servomotores y convirtiendo en movimientos de los ojos es “Visual Servoing”. (Bruno Siciliano, 2010)

El término “Visual Servoing” describe el control basado en realimentación de las mediciones visuales. El “Visual Servoing” basa su funcionamiento según en base de la posición o de la imagen. La visión permite al sistema robótico a obtener informaciones cualitativas y geométricas del entorno de operación para aplicaciones de estrategias del movimiento y control. La información obtenida por medio de visión es implementada a diferentes niveles de sistemas de control dependiendo de la estrategia de la aplicación. (Bruno Siciliano, 2010)

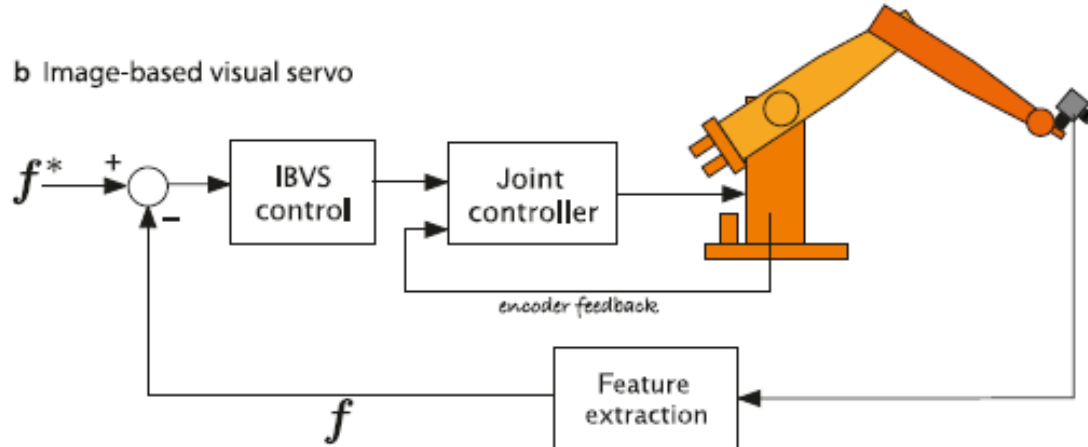
La característica clave de “Visual Servoing”, a diferencia de control de fuerza y de movimiento, las variables de control no son medidos directamente por el sensor; sino son obtenidos a partir de mediciones cuantificadas por medio de operaciones basadas en procesamientos de imágenes y de visión de computadora. Los esquemas de control basados en la visión son divididos en dos categorías en los cuales realizan la operación de “Visual Servoing” en espacio de operación o basado en posiciones, y los que realizan en espacio de imágenes o basado en las imágenes. Cada categoría basa en diferentes recursos para realizar operación, en donde basado en posiciones hace el uso de mediciones visuales para reconstrucción de pose relativa del objeto con respecto al robot o de forma viceversa. “Visual Servoing” basado en imágenes operan sobre los parámetros característicos de la imagen del objeto entre la pose actual y deseada. Existe un esquema en cual combina ambas características de las categorías refiriendo como el “Visual Servoing” híbrido. (Bruno Siciliano, 2010)

Figura 2. “Visual Servoing” basado en posición



(Bruno Siciliano, 2011)

Figura 3. “Visual Servoing” basado en imagen



(Bruno Siciliano, 2011)

Un aspecto adicional por control basado en visión es el tipo de cámara. La cámara puede que tenga parámetros de funciones en color o monocromo, de la resolución a cuál opera, longitudes focales variables o fijas, y tecnologías CCD o CMOS. Las configuraciones de la cámara varían según la forma de implementación. El sistema visual pueda que consista en una única cámara, o dos o más cámaras según sea la necesidad, con el objetivo de poder distinguir la profundidad de campo y reconstruir el área de captura con más detalle. Al contrario, el sistema visual se distingue por el posicionamiento de la cámara. Es decir, el posicionamiento de la cámara define la configuración a cuál realizará el trabajo de captura. (Bruno Siciliano, 2011)

Las configuraciones dependiendo del posicionamiento de la cámara se ajustan como una configuración fija o móvil. La configuración fija se conoce como configuración “eye-to-hand”, el cual el sistema visual observa los objetos a cuál será manipulado por medio de pose fijo en respecto al marco de base del manipulador. La ventaja es en que el campo de visión de la cámara no realiza cambio durante la ejecución del trabajo dando el resultado de mediciones precisas constante; aunque para operaciones de ensamblajes no será recomendado debido a la dificultad de evitar que el manipulador no obstruya la visión sobre los objetos. (Bruno Siciliano, 2011)

Al contrario, la configuración “eye-in-hand”, de la configuración móvil, la cámara se posiciona encima del manipulador y es posible que sea montado antes de o después del codo. La cámara puede observar el efector final por una pose favorable sin alguna obstrucción causados por el brazo manipulador. Entonces, por el ajuste de la configuración de la posición de la cámara normalmente observa únicamente el objeto. Por lo cual, los campos de visión de la cámara realizan cambios significativos durante el movimiento y esto produce alta variabilidad en precisión de las mediciones. Aunque esto se favorece cuanto el efector final es cercano

al objeto, en donde la precisión es relativamente constante y usualmente mayor que la configuración “eye-to-hand” con la ventaja en que la obstrucción es virtualmente ausente. (Bruno Siciliano, 2011)

En sistemas de “Visual Servoing” basado en posición tiene la estimación de pose de la meta o posición final deseada con respecto a la cámara. La estimación de la pose se requiere además el conocimiento de la geometría del objeto de la meta, los parámetros intrínsecos de la cámara, y características de las imágenes observadas. Se especifica la pose relativa deseada en respecto a la meta y determina el movimiento requerida para mover la cámara desde la pose inicial a la meta. La pose actual de la meta es desconocida por momento. Estas variables se pueden ver de la siguiente forma. (Bruno Siciliano, 2011)

- ${}^G\xi_G$ – Pose de la meta con respecto a la cámara
- ${}^{C^*}\xi_G$ – Pose relativa deseada con respecto a la meta
- ξ_Δ – Movimiento
- ξ_C – Pose inicial
- ξ_C^* – Pose inicial relativa deseada
- ξ_G – Pose de la meta

Entonces, el lazo de red de pose es,

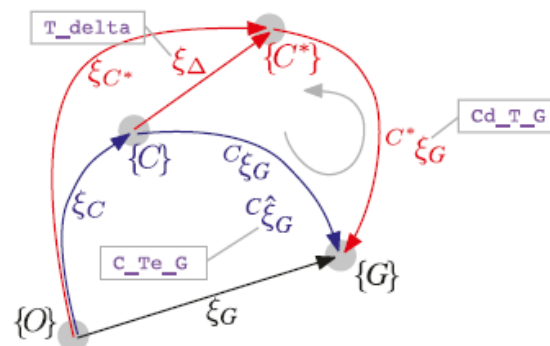
$$\xi_\Delta \oplus {}^{C^*}\xi_G = {}^C\hat{\xi}_G \quad (1)$$

, en donde ${}^C\hat{\xi}_G$ es la pose estimado de la meta relativa a la cámara. Al reorganizar esto genera la siguiente pose. (Bruno Siciliano, 2011)

$$\xi_\Delta = {}^C\xi_G \ominus {}^{C^*}\xi_G \quad (2)$$

, en el cual esto describe el movimiento de la cámara requerido a cumplir para obtener la pose relativa deseada.

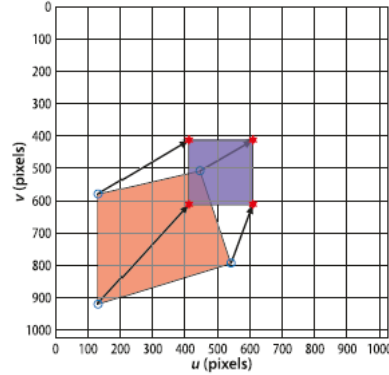
Figura 4. Red de poses relativos para “Visual Servoing” basado en posición



(Bruno Siciliano, 2011)

El caso del sistema “Visual Servoing” basado en imagen difiere fundamentalmente del sistema basado en posición por omisión de la estimación de la pose relativa de la meta. La pose relativa e implícita en los valores de las características de la imagen. (Bruno Siciliano, 2011)

Figura 5. Dos vistas del cuadrado de objeto de la meta



(Bruno Siciliano, 2011)

La Figura 5 demuestra dos vistas del cuadrado del objeto de la meta, y la vista desde pose de la cámara inicial se muestra en el color rojo y es claro que la cámara ve a la meta de forma oblicua. La vista deseada se muestra en azul en donde la cámara es lejos de la meta y los ejes ópticos es normal al plano de la meta. (Bruno Siciliano, 2011)

El problema de control se puede expresar en términos de las coordenadas de la imagen. El trabajo es realizar el movimiento de los puntos característicos indicados por los marcadores circulares grises vacíos sobre los puntos indicados por los marcados por las estrellas rojas. Los puntos pueda que sigan las rutas rectas marcadas por la flecha, pero no es obligatorio. Trasladando los puntos característicos en la imagen cambian la pose de la cámara de forma implícita, en el cual cambia entonces la necesidad de realizar la estimación de la pose a un problema de control de los puntos en la imagen. (Bruno Siciliano, 2011)

La proyección de perspectiva en forma funcional se expresa de la siguiente forma:

$$\mathbf{p} = \mathcal{P}(\mathbf{P}, \mathbf{K}, \xi_C) \quad (3)$$

, la derivada respecto al tiempo se ve de la siguiente forma:

$$\dot{\mathbf{p}} = \mathbf{J}_p(\mathbf{P}, \mathbf{K}, \xi_C) \boldsymbol{\nu} \quad (4)$$

, en donde $\boldsymbol{\nu} = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z, \mathbf{w}_x, \mathbf{w}_y, \mathbf{w}_z) \in \mathbb{R}^6$ es la velocidad de la cámara, y la velocidad espacial. \mathbf{J}_p es un objeto parecido al jacobiano, pero debido a la derivada con respecto a pose $\xi \in \mathbf{SE}(3)$, en vez de que sea un vector, técnicamente es conocido como la matriz de interacción. En el mundo de “Visual Servoing”,

la matriz de interacción se conoce más como el Jacobiano de la imagen o matriz de sensibilidad característica. Al considerar la cámara moviendo con la velocidad del cuerpo $\mathbf{v} = (\mathbf{v}, \mathbf{w})$ en marco mundial y el punto mundial P con coordenadas relativas de cámara $\mathbf{P} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$; la velocidad del punto relativo al marco de la cámara se expresa de la siguiente forma: (Bruno Siciliano, 2011)

$$\dot{\mathbf{P}} = -\boldsymbol{\omega} \times \mathbf{P} - \mathbf{v} \quad (5)$$

, el cual se puede expresar en forma escalar de la siguiente manera:

$$\begin{aligned} \dot{X} &= Y\omega_z Z - \omega_y - v_x \\ \dot{Y} &= Z\omega_x - X\omega_z - v_y \\ \dot{Z} &= X\omega_y - Y\omega_x - v_z \end{aligned} \quad (6)$$

La proyección de perspectiva para coordenadas del plano de la imagen normalizada y la derivada temporal utilizando la regla de cociente es de la siguiente forma (Bruno Siciliano, 2011):

$$\mathbf{x} = \frac{x}{z}, \mathbf{y} = \frac{y}{z} \quad \dot{x} = \frac{\dot{x}z - x\dot{z}}{z^2}, \dot{y} = \frac{\dot{y}z - y\dot{z}}{z^2} \quad (7)$$

Realizando la sustitución con ecuaciones de la forma escalar, $\mathbf{X} = \mathbf{xZ}$, y $\mathbf{Y} = \mathbf{yZ}$, es posible expresar en forma matricial de la siguiente manera (Bruno Siciliano, 2011):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{1}{z} & 0 & \frac{x}{z} & xy & -(1+x^2) \\ 0 & -\frac{1}{z} & \frac{y}{z} & 1+y^2 & -xy \end{pmatrix} \begin{pmatrix} y \\ -x \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (8)$$

, en el cual relaciona la velocidad espacial de la cámara a la velocidad característica en coordenadas de la imagen normalizada. Las coordenadas del plano de imagen normalizada son relacionadas a las coordenadas pixeles. (Bruno Siciliano, 2011)

$$u = \frac{f}{\rho_u} x + u_0, v = \frac{f}{\rho_v} y + v_0 \quad (9)$$

Al reorganizar la ecuación previa se obtiene las siguientes expresiones.

$$x = \frac{\rho_u}{f} \bar{u}, y = \frac{\rho_v}{f} \bar{v} \quad (10)$$

, en donde $\bar{u} = u - u_0$ y $\bar{v} = v - v_0$ son coordenadas pixeles relativos al punto principal. La derivada temporal es de la siguiente forma (Bruno Siciliano, 2011):

$$\dot{x} = \frac{\rho_u}{f} \dot{\bar{u}}, \dot{y} = \frac{\rho_v}{f} \dot{\bar{v}} \quad (11)$$

, y sustituyendo éstas en la forma matricial previa, se obtiene la siguiente matriz.

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{pmatrix} = \begin{pmatrix} -\frac{f}{\rho_u Z} & 0 & \frac{\bar{u}}{Z} & \frac{\rho_v \bar{u} \bar{v}}{f} & -\frac{f^2 + \rho_u^2 \bar{u}^2}{\rho_u f} & \frac{\rho_v \bar{v}}{\rho_u} \\ 0 & -\frac{f}{\rho_v Z} & \frac{\bar{v}}{Z} & \frac{f^2 + \rho_v^2 \bar{v}^2}{\rho_v f} & -\frac{\rho_u \bar{u} \bar{v}}{f} & -\frac{\rho_u \bar{u}}{\rho_v} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (12)$$

, y para casos típicos en donde $\rho_u = \rho_v = \rho$, es posible expresar la longitud focal en pixeles $f' = f/\rho$ y expresar la matriz en términos de coordenadas pixeles con respecto al punto principal $\dot{p} = J_p(p, Z)v$. (Bruno Siciliano, 2011)

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{pmatrix} = \underbrace{\begin{pmatrix} -\frac{f'}{Z} & 0 & \frac{\bar{u}}{Z} & \frac{\bar{u} \bar{v}}{f'} & -\frac{f'^2 + \bar{u}^2}{f'} & \bar{v} \\ 0 & -\frac{f'}{Z} & \frac{\bar{v}}{Z} & \frac{f'^2 + \bar{v}^2}{f'} & -\frac{\bar{u} \bar{v}}{f'} & -\bar{u} \end{pmatrix}}_{J_p(p, Z)} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (13)$$

c. Triángulos de Cámara. La cámara es el factor esencial en donde es la visión que obtiene y captura el resultado de forma de imagen. Estas imágenes proveen datos al algoritmo de reconocimiento de rostro así poder distinguir si el rostro del usuario está presente en el rango efectivo del entorno o no. Entonces, las calidades de las imágenes son importantes a tomar en consideración para mejor resultado de reconocimiento de rostro. (Cambride in Colour)

La exposición de una imagen tomado por una cámara determina que tan brillo u oscuro será la imagen. La calidad esto se determinar por medio de tres configuraciones principales de la cámara, en los cuales son apertura, ISO y la velocidad de obturación. Estas tres configuraciones forman lo que es el triángulo de cámara. Lograr la exposición correcta al capturar la imagen es posible expresar como colección de la lluvia en un cubo. (Cambride in Colour)

Aunque la tasa de la lluvia no es controlable, se encuentran 3 factores bajo control, en los cuales son el ancho del cubo, la duración mientras que el cubo este presente durante la lluvia, y la cantidad de la lluvia deseada a recolectar. Solo es importante asegurar no recolectar demasiado poco, similar a baja exposición; pero tampoco recolectar mucho más de la cantidad deseada a recolectar, similar a alta exposición. El clave para lograr la cantidad correcta de recolección, dando analogía a la exposición de la captura de imagen de la cámara, depende de la combinación respectiva según sea el entorno de ancho, tiempo y la cantidad. Por ejemplo, para la misma cantidad de la lluvia, es posible recolectar la lluvia a un tiempo menor si utiliza un

cubo de mayor ancho; y de forma alternativa, para misma duración del cubo puesto en la lluvia, un cubo mucho más angosto provee la misma calidad siempre y cuando este recolectando menor cantidad de lluvia. Estos factores, ancho, tiempo y cantidad son análogos a la apertura, velocidad de obturación e ISO. Asimismo, como la tasa de la lluvia no es controlable, la luz natural no es controlable. (Cambride in Colour)

Entonces, cada una de las configuraciones de control expone de forma diferente. La apertura controla el área sobre el cual la luz puede entrar a la cámara. La velocidad de obturación controla la duración de la exposición. La velocidad ISO controla la sensibilidad del sensor del cámara dada la presente cantidad de la luz. (Cambride in Colour)

El obturador de la cámara determina cuando el sensor de la cámara estará abierto o cerrado sobre la luz de entrada desde los lentes de la cámara. La velocidad de obturación refiere porque tanto tiempo la luz es permitida entrar a la cámara. El tiempo de exposición es conceptos similares, en donde alta velocidad de obturación significa menor tiempo de exposición. La influencia de la velocidad de obturación sobre la exposición correlaciona exactamente 1:1 con la cantidad de la luz que entra a la cámara. (Cambride in Colour)

Cuadro 1. Velocidad de obturación clasificado según la aplicación

| Velocidad de obturación [Segundos] | Ejemplo |
|------------------------------------|---|
| 1 – 30+ | Fotos con baja-luz sobre el trípode o noche en especial |
| 2 – ½ | Para mayor profundidad de campo de imágenes de campo |
| ½ - 1/30 | Para agregar movimiento borroso al fondo del sujeto en movimiento / imágenes cuidadosamente capturados con estabilización |
| 1/50 – 1/100 | Capturas de imágenes de móviles sin zoom sustancial |
| 1/250 – 1/500 | Para capturar movimiento del sujeto en acción / capturas de imágenes de móviles con zoom sustancial |
| 1/1000 – 1/4000 | Para captura de movimiento del sujeto extremadamente rápidos |

(Cambride in Colour)

Figura 6. Velocidad de obturación lenta

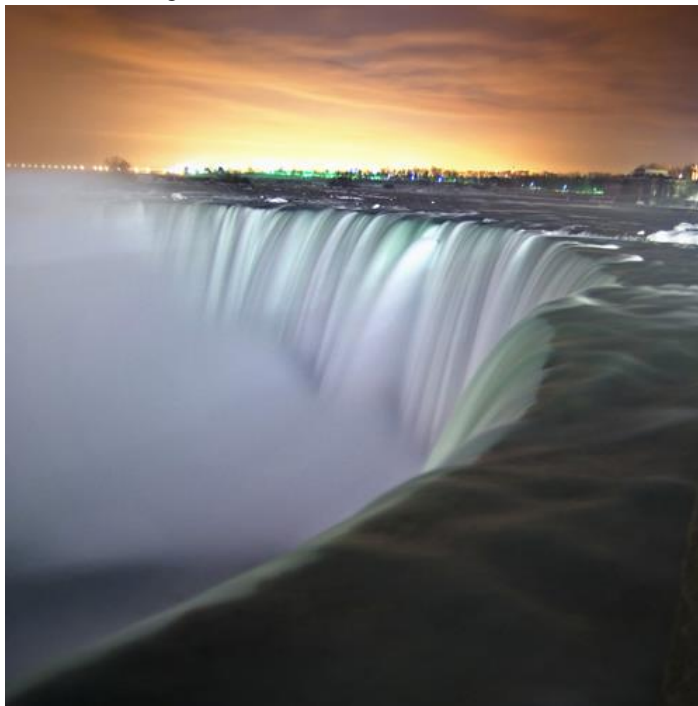


Figura 7. Velocidad de obturación alta



(Cambride in Colour)

La configuración de apertura controla el área el cual la luz pueda pasar a través de los lentes de la cámara. Esta configuración se especifica en términos del valor de “f-stop”, en donde el área permisible de la luz pueda pasar aumenta si disminuye el valor de “f-stop”. Entonces, en términos numéricos, siempre el valor de “f-stop” se hace mitad, el área de la colección de la luz permitida se cuadruplica. Existen opciones

estándares de los números de “f-stop” en cualquier dispositivo de cámara y estos números varían según los lentes de las cámaras. (Cambridge in Colour)

Cuadro 2. Configuración de aperturas con respectivos tiempos de obturador

| Configuración de apertura | Luz relativa | Velocidad de obturador [Segundos] |
|---------------------------|--------------|-----------------------------------|
| f/22 | 1X | 16 |
| f/16 | 2X | 8 |
| f/11 | 4X | 4 |
| f/8.0 | 8X | 2 |
| f/5.6 | 16X | 1 |
| f/4.0 | 32X | ½ |
| f/2.8 | 64X | ¼ |
| f/2.0 | 128X | 1/8 |
| f/1.4 | 256X | 1/15 |

(Cambridge in Colour)

Con amplia variedad de lentes, la capacidad de colección de la luz es afectada por la eficiencia de transmisión, aunque es un factor despreciable a diferencia de la apertura. La eficiencia de transmisión no es controlable de por usuario del cámara y las diferencias entre mismas eficiencias son normalmente más previstos en rangos de enfoques extremos. (Cambridge in Colour)

Por ejemplo, los lentes IS 24-105 mm f/4L recolecta aproximadamente de 10% a 40% menos luz a configuración de f/4, a diferencia de lentes 24-70 mm f/2.8L a f/4; en donde depende de la longitud focal. Entonces, la configuración de apertura de la cámara es el que determina la profundidad de campo de la imagen, en donde a valores menores de f-stop correlaciona con la profundidad de campo más angosto. La profundidad de campo se refiere al rango de la distancia sobre los objetos a los cuales aparecen en un enfoque más fino. (Cambridge in Colour)

Figura 8. Apertura ancha de f/2.0 – número de f-stop bajo con profundidad de campo angosto



(Cambridge in Colour)

Figura 9. Apertura angosta de f/16 – número de f-stop alto con profundidad de campo ancho



La velocidad ISO determina que tan sensible sea la cámara en contra de luz de entrada. Similar a la velocidad de obturación, correlaciona 1:1 con el aumento de exposición o de forma viceversa. Debido a que mayores de velocidades ISO generan imágenes con más ruidos, se desea velocidad ISO baja. Entonces, la velocidad ISO es aumentado desde el valor mínimo si la apertura y la velocidad de obturación no es posible obtener. Las velocidades ISO comunes son 100, 200, 400, y 800. Cámaras compactas ofrecen velocidades de ISO en rango de 50 a 200 para producción de imágenes de bajo ruido; mientras que las cámaras SLR es posible manejar desde 50 hasta 800. (Cambride in Colour)

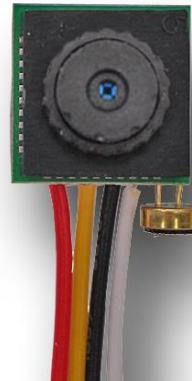
Figura 10. Comparaciones de las velocidades ISO



d. Mini cámara – OOSXXX Mini Security Camera. La cámara implementada para cumplir con obtención de las imágenes y transmitir éstas imágenes para procesarlos; mientras cumple el requerimiento mecánico sobre el espacio y peso, es el OOSXXX Mini Security Camera. La cámara provee tamaño compacto con la facilidad de integrar a la estructura mecánica móvil y resolución de imágenes aceptables para procesarlos y distinguir el rostro del usuario, el cual no se distingue fácilmente en tiempo de operación de animatrónico.

La cámara posee el sensor de imagen $\frac{1}{4}$ CMOS de resolución de 1280 x 900 y resolución horizontal de 600TVL; lente de 3.6mm a 75 grados de visión; iluminación mínima de 0.5 Lux; ración S/N mayor de 48dB; gama característica de 0.45; frecuencia de actualización de 60 f/s; obturador automático de 1/50 a 1/100,000s; y temperatura de operación de -40 a 85 grados Celsius. El dispositivo especifica con consumo de potencia de 5.5V a 12V DC con 80mA \pm 10mA; equilibrio de blanco automático con salida del video de 1vp-p 75 Ω ; sistema de sincronización interno; y dimensión de 12mm x 12mm x 8mm pesando 41 gramos. Adicionalmente trae una salida de Audio de un solo carril.

Figura 11. OOSXXX Mini Security Camera 600TVL



(OOSXXX)

e. Microsoft LifeCam Cinema. La cámara de implementación inicial y de prueba es el Microsoft LifeCam Cinema. El dispositivo posee dimensiones relativamente mayores al de OOSXXX Mini Security Camera con largo de 55.9 milímetros, ancho de 46.0 milímetros, altura de 40.0 milímetros, y peso de 95.3 gramos. Debido al dimensionamiento mayor, la implementación basó en propósitos de pruebas de sistema de reconocimiento de rostro. La interfaz de conexión de la cámara es por medio de USB 2.0 de alta velocidad. (Microsoft)

La característica de la cámara basa en posesión del sensor de tecnología CMOS y trabaja con resolución de 1280 x 720 pixeles en video, mientras que con imagen la resolución sube hasta calidad de 2880 x 1620 pixeles interpolado. La velocidad de grabación tiene capacidad hasta 30 marcos por segundo; con campo de visión de 73 grados diagonalmente. La característica de imagen basa en ajustes o calibraciones de calidad automática según el entorno y enfoque automático a partir de 6 pulgadas de distancia. Adicionalmente posee características de audio que es el micrófono unidireccional digital integrada con respuesta de frecuencia de 200Hz a 8000Hz con margen de 4 dB. (Microsoft)

Figura 12. Cámara Microsoft LifeCam Cinema



4. Comunicación serial (señales de audio y video)

a. Comunicación serial Python-Arduino. Este tipo de comunicación es bastante simple debido a la implementación de instrucciones de lectura y escritura por parte de Arduino y Python. Sin embargo, para este último, a pesar de que la escritura y lectura es como la escritura de un archivo, siempre es necesaria la instalación de una librería y en este caso es necesaria la librería llamada pySerial. (Python)

5. Librerías / Add-Ons de Python

a. pySerial. Este es un módulo de licencia abierta que abarca el acceso al puerto serial por parte de Python en sistemas Windows, Linux, OSX, BSD e IronPython. Posee las siguientes características:

- Es compatible con la librería io.
- Los archivos en el paquete son basados completamente en Python.
- Es la misma interfaz basada en clases para todas las plataformas.
- Se puede ingresar a los ajustes del puerto a través de las propiedades de Python.
- Puede funcionar con o sin recibir timeout.
- Archivo tipo API con instrucciones de lectura y escritura simples.

b. sys. Esta librería permite acceder a información de intérprete de Python. Con ella se pueden editar variables utilizadas por el intérprete, o bien, funciones que interactúan exclusivamente con el intérprete. (Python)

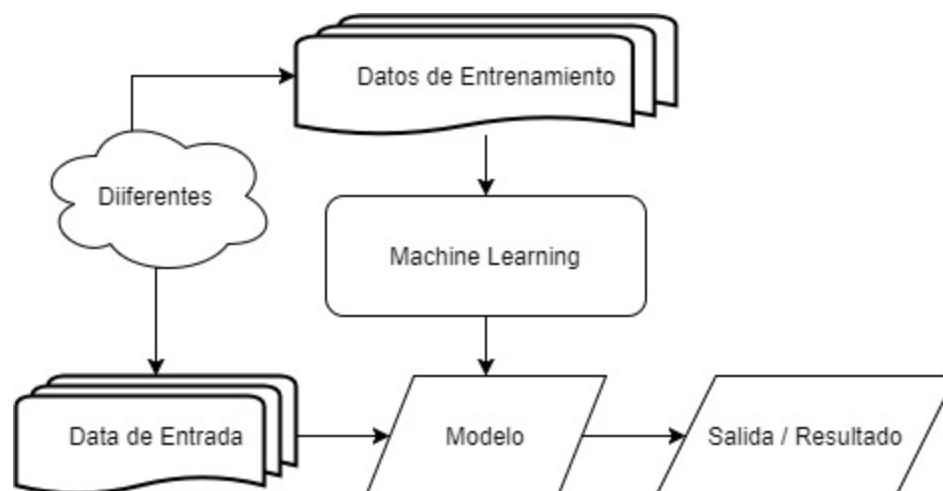
6. Machine / Deep Learning. En los desarrollos de los algoritmos en los cuales requieren mejoras tras el uso o que tiene que mejorar conforme la cantidad de datos, se encuentra el tema de Machine Learning y Deep Learning. En general, el Machine Learning, Deep Learning, e inteligencia artificial están encadenados de tal forma que Deep Learning es similar al Machine Learning, y Machine Learning es similar a la inteligencia artificial. (Phil Kim, 2017)

La inteligencia artificial es un término bastante común y puede implicar varios tipos de cosas. Es posible que indique cualquier forma de tecnología que incluye algunos aspectos inteligentes, en vez de especificar un campo de tecnología. Al contrario, Machine Learning refiere a un campo en específico, y utiliza el término Machine Learning para indicar a un grupo específico tecnológico de inteligencia artificial. Esto mismo posee variedad de tecnologías y una de estas tecnologías es Deep Learning. (Phil Kim, 2017)

Deep Learning, como parte de Machine Learning es muy importante y es debido a que se relaciona de forma extensa la inteligencia artificial, Machine Learning, y Deep Learning. El tema de Deep Learning ha resuelto problemas que reto a la inteligencia artificial, y el rendimiento de esto es excepcional en variedad de campo. Aunque tiene limitaciones empezando a partir de conceptos fundamentales que hereda desde Machine Learning. Como el tipo de Machine Learning, Deep Learning no se puede evitar los problemas que poseen Machine Learning. (Phil Kim, 2017)

a. Machine Learning. Machine Learning es una técnica de modelación que involucra el dato. Es decir, es la técnica el cual resuelve el modelo a partir del dato. El dato en este contexto se refiere literalmente la información tales como las señales de audio, e imagen y entre otros más. Entonces el modelo es el producto o el resultado final de Machine Learning. El nombre tal como refleja la técnica que analiza el dato y busca o encuentra un modelo por sí mismo en vez de dependerse de una persona. Se refiere a “Learning” o aprendizaje debido a que el proceso reemplaza el entrenamiento con los datos para resolver el problema de encontrar un modelo. Entonces el dato que Machine Learning utiliza en proceso de modelación se llama entrenar los datos. (Phil Kim, 2017)

Figura 13. El proceso de Machine Learning



(Phil Kim, 2017)

Entonces, Machine Learning usualmente se refiere a los sistemas que realizar trabajos u operaciones asociados con inteligencia artificial. Tales operaciones involucran reconocimiento, diagnósticos, planificación, control robótico, y predicciones. Los cambios pueden ser las mejoras sobre los sistemas ya existentes o síntesis de los nuevos de sistemas. En términos de ingeniería, Machine Learning muestra dichos detalles importantes. (Nils J. Nilsson, 2005)

Estos detalles son como algunos trabajos u operaciones que no pueden ser definidos de forma correcta a excepción del ejemplo. Es decir, será posible de especificar pares de entradas o salidas, pero con una relación concisa entre las entradas y salidas deseadas. Entonces sería bueno que las máquinas puedan ajustar las estructuras internas de los mismos para producir salidas correctas para cantidad de muestras de entradas y, por consiguiente, restringir apropiadamente las funciones de entradas o salidas a una relación próxima de forma implícitas en los ejemplos. (Nils J. Nilsson, 2005)

La posibilidad de que la relación escondida entre la gran cantidad de datos son relaciones y correlaciones importantes. Entonces, los métodos de Machine Learning puede que sea el método para aplicar con el fin de extraer estas relaciones que se conoce como minería de datos. Los ambientes o los entornos cambian con tiempo, entonces máquinas que puedan adaptarse al entorno variable ofrecerán la ventaja de reducir la necesidad de rediseñar constantemente. Los métodos de Machine Learning podrán ser implementados para mejora en vivo mientras la operación sobre los diseños de máquinas ya existentes. La cantidad del conocimiento disponible de ciertos trabajos u operaciones podrá ser muy grandes para codificación explícita por las personas, entonces las máquinas que aprenden este conocimiento de forma gradual podrán capturar más de estas informaciones que estar escribiéndose. Además, existe flujo constante de nuevo eventos en el entorno, entonces reanudando las repeticiones de los diseños de sistemas de inteligencia artificial para

conformar nuevos conocimientos o informaciones es impráctico. Por lo tanto, los métodos de Machine Learning podrán ser de una guía para seguir trazos de este flujo de datos constantes. (Nils J. Nilsson, 2005)

Las operaciones en Machine Learning se convergen desde varias fuentes, y diferentes tradiciones proveen diferentes métodos que ahora son asimilados en una disciplina única. De las disciplinas que han contribuido en tema de Machine Learning son las estadísticas, modelo cerebral, teoría de control adaptativo, modelos psicológicos, inteligencia artificial y modelos de evolución. En especial, la teoría de control adaptativo basa a partir de los estudios de problemas de controlar el proceso teniendo parámetros desconocidos que deben ser estimados durante la operación. A veces los parámetros cambian durante la operación, y el proceso de control debe seguir estos cambios, entonces algunos aspectos de controlar el robot basado en entradas de los sensores representan instancias de esta categoría del problema. (Nils J. Nilsson, 2005)

b. Deep Learning. Deep Learning permite los modelos computacionales que son compuestos de múltiples capas de procesamientos para aprender la representación de los datos con múltiples niveles de abstracción. Estos métodos han mejorado drásticamente el estado de arte de sistemas de reconocimiento de voz, reconocimiento de objetos visuales, detección de objetos en general. Este método de aprendizaje descubre la estructura intrincada en un conjunto de datos grandes por medio de la aplicación de algoritmo de retro-propagación para indicar como la máquina debe cambiar los parámetros internos que son utilizados para computar la representación de cada capa individual a partir de la representación de la capa previa. Redes convolucionales profundas trajeron las mejoras en procesamiento de imágenes, de video, de voz y de audio, mientras que las redes recurrentes mejoraron los datos secuenciales tales como el texto y dialecto. (Ian Goodfellow, 2016)

Los sistemas de Machine Learning son utilizados para identificar los objetos en las imágenes, transcribir la señal de la voz en texto, o seleccionar resultados de la búsqueda relevantes. Estas aplicaciones, con el aumento, hacen el uso de la clase de la técnica que se conoce como el Deep Learning. Las técnicas convencionales de Machine Learning eran limitadas por las habilidades mismas para procesar datos naturales en sus formas puras. Por largo tiempo el sistema de Machine Learning ha requerido un proceso de ingeniería riguroso para el diseño de extractores característicos que transforman los datos puros, tales como los valores de los píxeles de la imagen, en una representación interna apropiada o un vector característico de cual el subsistema de aprendizaje, tal el clasificador, pueda detectar o clasificar los patrones de la entrada. (Yann LeCun, 2015)

El aprendizaje de representación es el conjunto de métodos que permite a la máquina sea alimentado con los datos puros y descubrir automáticamente las representaciones requeridas para la detección o clasificación. Los métodos de Deep Learning son métodos de aprendizaje de representaciones con múltiples niveles de representación obtenidos por composición de módulos no lineales simples que transforman la representación de un nivel, empezando con la entrada pura, en una representación a un nivel más alto con el nivel de abstracción mayor. Con la suficiente composición de tales transformaciones, funciones o

características de complejidad mayor. Para operaciones de clasificación, capas de mayor nivel de representación amplifican los aspectos de la entrada que son importantes para eliminar variaciones irrelevantes. (Yann LeCun, 2015)

Una imagen, por ejemplo, proviene en una forma de arreglo de valores de los píxeles, y las características aprendidas en la primera capa de la representación típicamente representan la presencia o ausencia de los vértices en orientaciones particulares y ubicaciones en la imagen. La segunda capa típicamente detecta los sujetos por medio de identificación de arreglos de los vértices, independientemente de las pequeñas variaciones de las ubicaciones de los vértices. La tercera capa puede ensamblar los sujetos en combinaciones más grandes que corresponden a partes de los objetos familiares, las capas subsecuentes detectarán los objetos como combinaciones de estas partes. El aspecto de Deep Learning basa en que estas capas características no son diseñadas por los ingenieros, sino son aprendidos a partir de los datos usando proceso de aprendizaje de propósito general. (Yann LeCun, 2015)

B. Diseño mecánico de brazos y gestos faciales de DANI

1. Anatomía de un brazo humano

a. Tipos de anatomía

1) Anatomía clínica. Es la que se basa en sistemas, regiones o ambos, con el fin de solucionar problemas o evaluaciones clínicas, o bien, obtener información anatómica a partir de observaciones clínicas.

2) Anatomía regional. Se enfoca en las divisiones, partes o regiones del cuerpo para obtener las relaciones estructurales entre los diversos sistemas, tales como, sistema muscular, nervioso, arterial, etc.

3) Anatomía sistemática. Aquí el enfoque es organizar el análisis del cuerpo a partir de sistemas orgánicos resaltando los aspectos generales de cada uno.

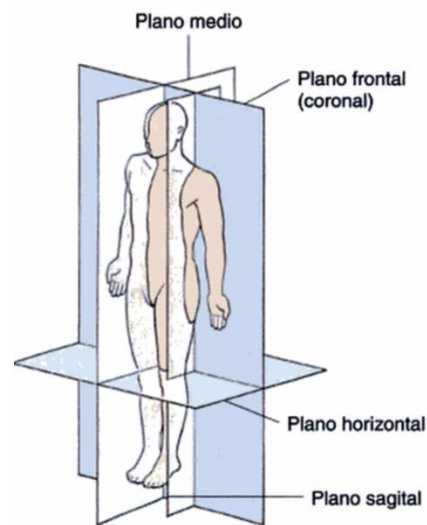
b. Planos de anatomía. Se dividen en cuatro con el fin de realizar “cortes” al cuerpo humano y así poder hacer descripciones anatómicas. Estos planos son:

1) Plano medio (sagital medio). Atraviesa verticalmente y longitudinalmente el cuerpo a través del centro produciendo la mitad izquierda y derecha.

2) Planos frontales (coronales). Son planos verticales y perpendiculares al plano medio que dividen al cuerpo en parte dorsal (posterior) y frontal (anterior).

3) Planos horizontales (transversales). Son planos perpendiculares al plano medio y los planos frontales que dividen el cuerpo en parte superior e inferior. En este caso, al poder existir varios planos de este tipo, es adecuado mencionar una referencia para indicar el nivel en el que se encuentra, por ejemplo, un plano horizontal al nivel de la cintura (Moore, 2002).

Figura 14. Planos de anatomía.

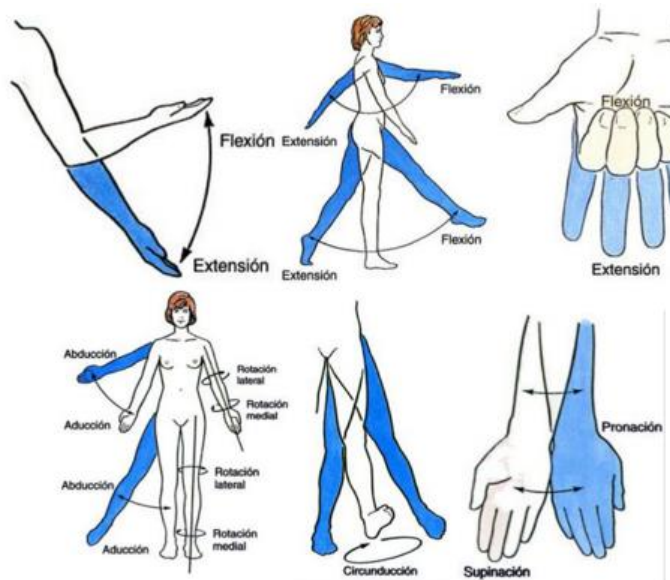


(Moore, 2002)

c. Articulaciones. Son las uniones entre dos o más partes rígidas del cuerpo, pudiendo tener varias funciones y formas. Unas de ellas tienen movimiento libre, leve o nulo. Estas se clasifican así:

1) Los movimientos que permiten. A continuación, se presenta gráficamente los movimientos en las extremidades que permiten las articulaciones en el cuerpo humano (Moore, 2002)

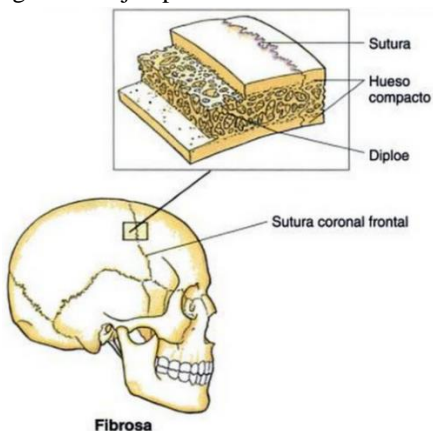
Figura 15. Movimientos en las extremidades de un ser humano.



(Moore, 2002)

2) **Fibrosas.** Permite un movimiento parcial-leve. Un ejemplo son las suturas del cráneo que unen los huesos planos vecinos.

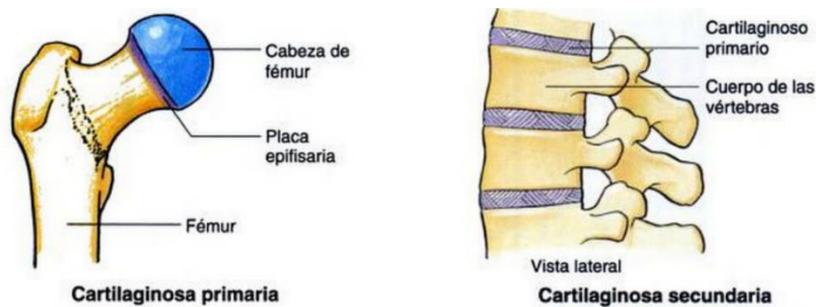
Figura 16. Ejemplo de articulación fibrosa.



(Moore, 2002)

3) **Cartilagosas.** Permite movimientos parciales-libres. En esta categoría caen las articulaciones de la cabeza del fémur y las articulaciones entre vértebras.

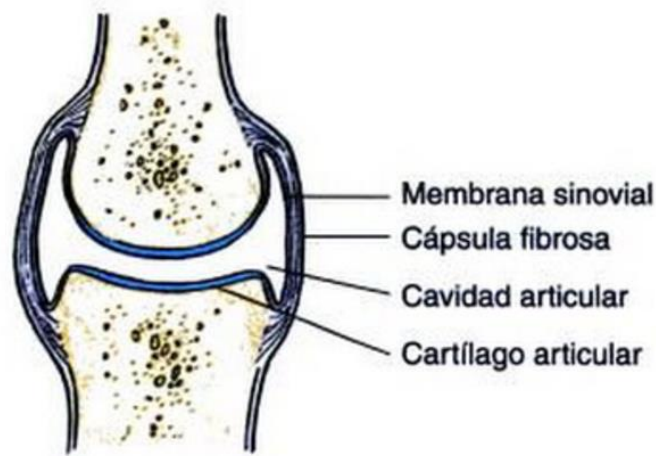
Figura 17. Ejemplo de articulación cartilaginosa.



(Moore, 2002)

4) **Sinoviales.** Son las más comunes y permiten movimientos libres. Estas se pueden encontrar en los codos, rodillas y dedos. Hay varios tipos de dicha articulación, las cuales se pueden observar a continuación (Moore, 2002):

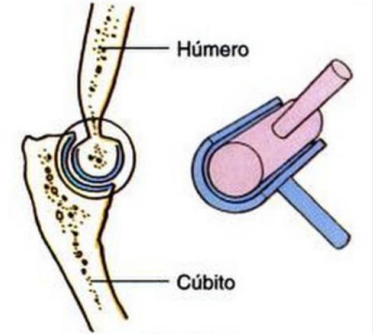
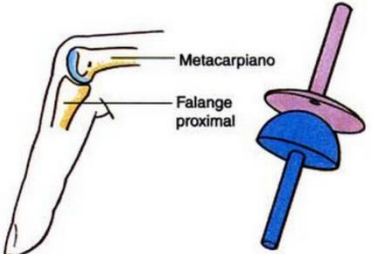
Figura 18. Ejemplo de articulación sinovial.



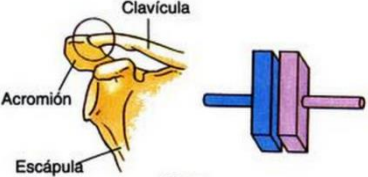
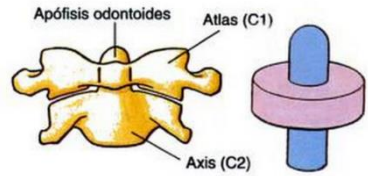
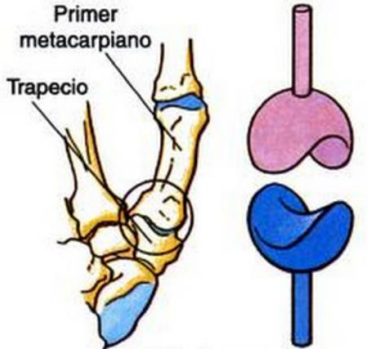
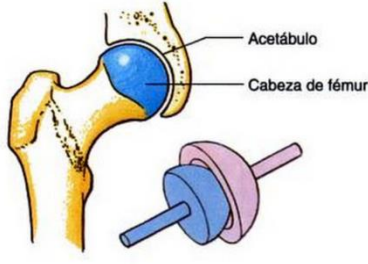
Articulación sinovial

(Moore, 2002)

Cuadro 3. Modelos mecánicos de juntas con el movimiento que permiten y su representación gráfica

| Tipo | Movimiento que permite | Representación gráfica |
|-------------------|--|---|
| En bisagra | Flexión y extensión. | <p data-bbox="1143 1045 1430 1073">Figura 19. Junta de bisagra</p>  <p data-bbox="1127 1430 1268 1457">(Moore, 2002)</p> |
| Condílea | Abducción, aducción, circunducción, flexión y extensión. | <p data-bbox="1166 1486 1442 1514">Figura 20. Junta condílea</p>  <p data-bbox="1127 1808 1268 1835">(Moore, 2002)</p> |

Continuación Cuadro 3

| Tipo | Movimiento que permite | Representación gráfica |
|---------------------------|---|--|
| Plana | Traslación y deslizamiento. | <p data-bbox="1166 233 1404 264">Figura 21. Junta plana</p>  <p data-bbox="1127 453 1268 485">(Moore, 2002)</p> |
| En pivote | Rotación. | <p data-bbox="1143 512 1425 543">Figura 22. Junta de pivote</p>  <p data-bbox="1143 732 1284 764">(Moore, 2002)</p> |
| En silla de montar | Abducción, aducción, extensión y flexión. | <p data-bbox="1136 810 1471 842">Figura 23. Junta de silla de montar</p>  <p data-bbox="1127 1224 1268 1255">(Moore, 2002)</p> |
| Esferoideas | Flexión, extensión, abducción, aducción, rotación lateral, rotación medial y circunducción. | <p data-bbox="1154 1289 1450 1320">Figura 24. Junta esferoidea</p>  <p data-bbox="1127 1619 1268 1650">(Moore, 2002)</p> |

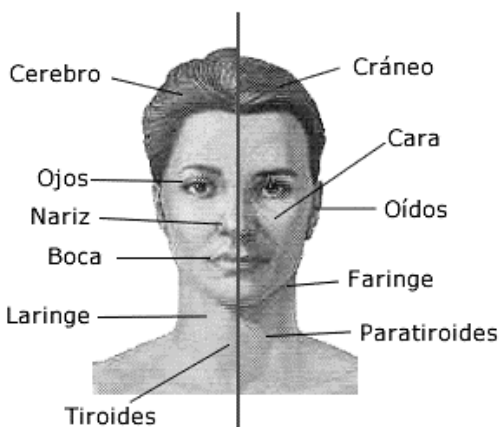
(Moore, 2002)

2. Punto de vista biológico de la cabeza humana

a. Definición y características

1) Partes de una cabeza. La cabeza comprende dos partes, el cráneo y la cara. El cráneo está situado en la parte superior y posterior, y su estructura ósea hace que sea rígida y encierre todos los órganos de la cabeza. En la cara se encuentran otros componentes que son los ojos, párpados, cejas, oídos, boca y nariz.

Figura 25. Partes de la cabeza



La anatomía del ojo es una estructura constituida de múltiples capas con una cornea que difracta la luz, un cristalino que la enfoca y una retina rica en fotorreceptores. Cada globo ocular está ubicado en una cavidad ósea protectora en forma de taza. Los músculos esqueléticos que corren desde la parte de atrás del ojo hasta los huesos de la órbita mueven el globo ocular en distintas direcciones. (Starr, Taggart, 2009, págs. 586-591)

Los párpados, las pestañas y lágrimas ayudan a proteger los delicados tejidos del ojo. En la capa media del ojo se encuentra el iris muscular, el cual a través de la pupila entra la luz y dicho iris ajusta el diámetro de la pupila en respuesta a las condiciones de luz. (Starr, Taggart, 2009, págs. 586-591)

Los oídos, permite oír, lo cual es la percepción del sonido, la cual es una forma de energía mecánica. Un sonido surge cuando la vibración de un objeto causa variaciones en la presión del aire, el agua o algún otro medio. El oído está compuesto por cartílagos, los cuales en el ámbito animal les permite moverlas como medio de protección, contra insectos. (Starr, Taggart, 2009, págs. 586-591)

2) Mecanismo de enfoque de los ojos. Con el acomodo visual, la forma o la posición de una lente se ajusta de tal manera que los rayos de luz que inciden en el ojo se dirigen hacia la retina. Dentro del ojo se encuentra el músculo ciliar el cual permite ajustar la forma del lente del ojo para así poder tener la visión de objetos cercanos o lejanos. (Starr, Taggart, 2009, págs. 586-591)

3) Máxima apertura de la boca por medio de la mandíbula. Estudios realizados a 300 pacientes, siendo 195 del sexo masculino y 105 del sexo femenino, donde el rango de edad va de 17 a 60 años. Se utilizó el paquímetro de Lyman modificado, llamado calibrador “F”, se obtuvo que el promedio de apertura de boca en hombre es de 45.12 mm, mientras que en mujeres es de 42.88 mm. (Fogaca, Carlos, 1998)

4) Fuerza de mordida de la boca. La fuerza de mordida es un componente de la función masticatoria, es un indicador de su estado funcional y se ha definido como la máxima fuerza generada entre los dientes maxilares y mandibulares. La generación de la fuerza de mordida depende de la acción, volumen y coordinación de los músculos masticatorios y de los mecanismos de articulación temporomandibular. (ADM, 2011)

El rango de valores de fuerza mandibular en hombres adultos sanos se ha reportado un valor promedio de 727 N (74.15 Kgf), en niños 425 N (43.35 Kgf) y en mujeres 370 N (37.74 Kgf). (ADM, 2011)

3. Diseño mecánico para animatrónicos

a. Introducción. Uno de los elementos principales que entra en juego cuando se implementa un animatrónico es su diseño mecánico. Ya sea que esté cubierto por elementos estéticos que imiten la piel del ser vivo que se está imitando o que los elementos mecánicos y electrónicos estén expuestos, siempre será necesario establecer las dimensiones, las posiciones, los grados de libertad, los métodos de unión, las velocidades de las partes móviles, los métodos de ventilación (de ser necesario), materiales para las piezas, el peso de las mismas, las propiedades mecánicas que se requieren en determinadas partes del robot, los esfuerzos en los puntos críticos, los actuadores, los sensores, etc. Todo esto con el objetivo de desarrollar un sistema que integre la mecánica, la electrónica y la informática de forma eficiente (Siciliano, 2010).

Ahora bien, hablando específicamente de los brazos de un animatrónico antropomorfo, en realidad se está hablando de la categoría de los robots “manipuladores”. Estos consisten en una secuencia de eslabones o cuerpos rígidos interconectados por articulaciones. La estructura fundamental de un manipulador es la “cadena cinemática abierta” que básicamente significa una serie de eslabones conectados en uno de sus dos extremos. Mientras que un manipulador con una “cadena cinemática cerrada” se identifica por una secuencia de eslabones que forma un lazo cerrado, físicamente hablando.

Generalmente, un manipulador está conformado por un brazo que asegura movilidad, una muñeca para ayudar en la orientación y un efector final que ejecuta el objetivo del robot. Las articulaciones entre dos cuerpos rígidos se pueden realizar con juntas revolutas o prismáticas. Cualquiera de las dos otorga un grado de libertad a la movilidad del robot, siendo la diferencia entre ellas la forma del grado de libertad: angular y lineal respectivamente. Las juntas revolutas suelen preferirse sobre las prismáticas porque ocupan menos espacio y son más “confiables”, además de que directamente en ellas puede ubicarse un actuador giratorio como un motor. Por otro lado, las juntas prismáticas brindan movimiento traslacional relativo entre dos cuerpos rígidos.

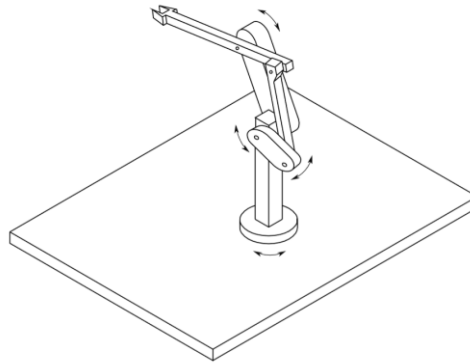
Los grados de libertad deben ser distribuidos de forma correcta a través de la estructura mecánica para poder realizar una tarea con la cantidad suficiente de ellos. Generalmente, para robots que manipulan objetos se requieren de 6 grados de libertad, mientras que para un animatrónico cuyo objetivo no es manipular objetos sino imitar el movimiento real de un ser vivo, la cantidad de grados de libertad puede reducirse o aumentar según la complejidad del mismo.

Al espacio en el que el efector final se puede mover se le llama “espacio de trabajo”. Dicho espacio puede tener diferentes tamaños, formas y volúmenes que son dependientes de los límites de las articulaciones utilizadas y de la estructura del manipulador. Este manipulador puede tener diferentes clasificaciones dependiendo del tipo y secuencia de los grados de libertad del brazo mecánico. Las clasificaciones son: cartesiano, cilíndrico, SCARA, antropomórfico y esférico.

La geometría antropomórfica cuenta con tres juntas de revolución en donde la primera es perpendicular a los ejes de rotación de las otras dos. Esto es, a grandes rasgos, similar a la geometría de un brazo humano. De hecho, es por ello que a la segunda junta se le suele llamar “hombro” y a la tercera “codo”. Y debido a que todas las juntas son de revolución usualmente se utiliza motores eléctricos como actuadores en las articulaciones.

Si bien la geometría antropomórfica de los brazos de un animatrónico es una cadena cinemática abierta, esta tendrá que cerrarse en alguna región cuando se requiera levantar mucho peso. Incluso un animatrónico que no suele levantar objetos pesados, puede tener este problema debido a las dimensiones o al peso de los brazos. Y es que, al aumentar la rigidez de la estructura mecánica, se garantiza no solo una mejor resistencia estructural sino mejor precisión en los movimientos. Para este caso una geometría de paralelogramo entre el hombro y el codo suele recomendarse, tal como en la Figura 26. (Siciliano, 2010)

Figura 26. Geometría básica del modelo de paralelogramo para una extremidad superior



Para posicionar la “muñeca” del brazo robótico es necesario orientar el efector final del manipulador en un espacio tridimensional. Lo que se traduce en 3 grados de libertad (de revolución) más, solo para el efector final. Y como la muñeca es la parte ubicada al extremo del brazo robótico, esta tiene que ocupar poco espacio, lo que complica el diseño mecánico. Para este caso se necesitaría una junta esférica porque sus 3 ejes se interceptan en un mismo punto.

A pesar de que un manipulador robótico es muy versátil, no se debe asumir que se puede utilizar para cualquier tarea. Por lo tanto, la principal condición es la aplicación para la que se quiere utilizar porque esta brinda los límites máximos de carga, dimensiones, formas, velocidades, desempeño, etc. (Siciliano, 2010).

En el caso de un brazo que se mueve solamente en el hombro, alrededor de un eje y en el mismo plano, es un cuerpo rígido que se puede modelar como un péndulo simple. Ya que este considera una varilla cuya longitud no cambia y cuya masa es despreciable, ninguna fuerza de fricción en el pivote y una masa puntual en el extremo de la varilla. Además, para encontrar el torque necesario para levantar el brazo es necesario recurrir, también, a la ecuación de la conservación de la energía de un péndulo. Las ecuaciones de ambos modelos son, respectivamente (Normani, 2017):

$$\tau = I * \alpha \quad (1)$$

donde,

τ = Torque en el motor

I = Inercia de una varilla de masa despreciable con una masa en su extremo

α = Aceleración angular del péndulo

$$e_p + e_c = \text{constante} \quad (2)$$

$$mgh + \frac{1}{2} * I * w^2 = \text{constante} \quad (3)$$

donde,

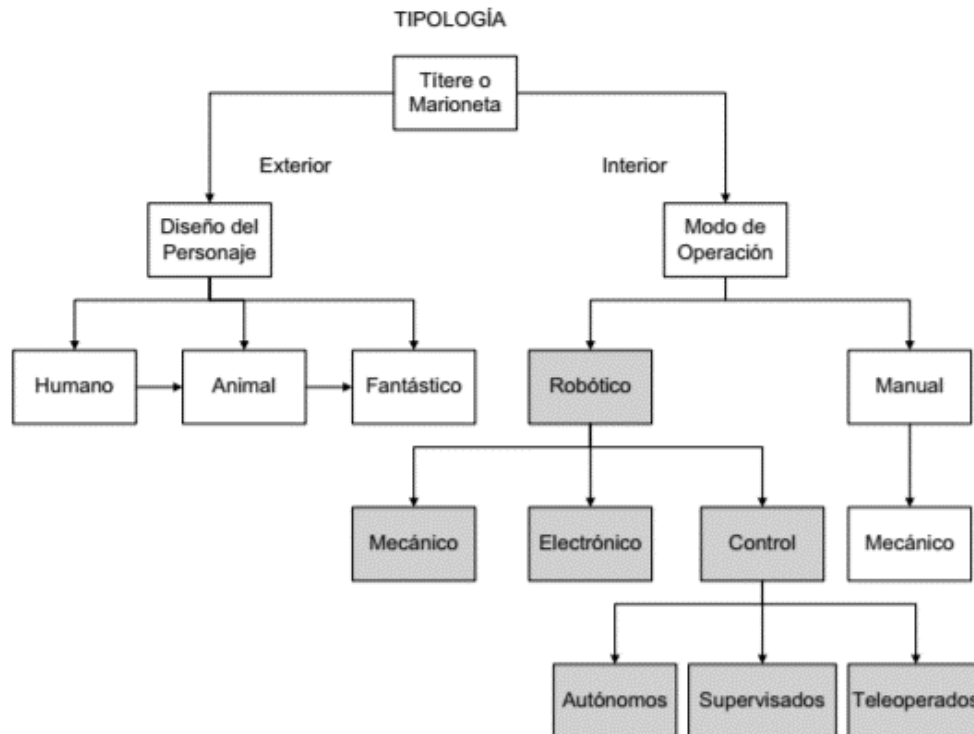
m = masa del péndulo

g = gravedad

h = Altura entre la posición inicial y actual de la masa en el extremo de la varilla
 I = Inercia de una varilla de masa depreciable con una masa en su extremo
 w = velocidad angular del péndulo

b. Modos de operación de un animatrónico. Los animatrónicos tienen distintos modos de operación los cuales se presentan a continuación:

Figura 27. Modos de operación de un animatrónico



4. Plataforma CAD

a. Gráficas de Ingeniería. Por fines de manufactura y construcción, se requiere una representación ilustrativa del diseño. Esta representación ilustrativa muestran los detalles importantes del diseño que se debe respetar y tomarlo en consideración al ingresar al proceso de manufactura y construcción. Para llevar a cabo dicha tarea es necesario recurrir a las Gráficas de Ingeniería, conocido principalmente como Dibujo Técnico. Las Gráficas en Ingeniería es la técnica para realizar el proceso de crear una representación con precisión del diseño con el fin de manufacturar y construir (Shih, 2016).

El Dibujo en Ingeniería, proceso tomado en Gráficas de Ingeniería, es el tipo de dibujo técnico utilizado para definir los requerimientos de elementos o artículos de ingeniería. Este artículo de ingeniería es

usualmente creado bajo convenciones estandarizadas por propiedades como la apariencia, la dimensión, la nomenclatura, y la esquemática. De esta clase de dibujo, requiere la habilidad de realizar un dibujo preliminar lo cual impide la comprensión de la geometría con ciertas habilidades de un diseñador.

El dibujo preliminar en un tiempo era un proceso demandante requiriendo ser hábil en diseño. Hoy en día, los procesos de dibujo preliminar son automatizados y acelerados por medio del uso de sistemas de Diseño Asistido por Computadora (CAD). El Diseño Asistido por Computadora (CAD) se trata de proceso de crear un diseño con la asistencia de las computadoras. La asistencia de computadora incluye análisis de datos de diseño y la creación de dibujos necesarios (Shih, 2016).

b. Autodesk Inventor 2017. Autodesk Inventor es un paquete de software de Diseño Asistido por Computadora desarrollado por parte de Autodesk, Inc. Esta es una herramienta que permite realizar diseños y de dibujo preliminar. Esto tiene una característica de poder transferir el archivo de diseño, sea en 2-dimensiones o 3-dimensiones, a otros programas de computadora para realizar análisis más profundo y simulaciones. Los modelos de computadora, es decir diseño creado por este software, puede utilizarse con equipos de manufactura; que pueden ser centros de maquinado o máquinas de prototipado rápida para manufacturar el producto (Shih, 2016).

El software conforma de varias aplicaciones de módulos de software, es decir elemento con cierta acción para realizar un proceso de diseño. Todos estos módulos de software comparten la base de dato común con el fin de integrar varios procesos de diseño necesario para crear el producto final. Procesos de diseño que se puede encontrar en Autodesk Inventor están la creación de modelos sólidos, dibujos de ingeniería y ensamblajes.

El componente principal de Autodesk Inventor, por lo que se caracteriza, en modelado sólido basa en la técnica de modelado paramétrico basado en características. Características en este contexto se refieren a los partes predefinidos o herramientas de construcción que son implementados para definición de parámetros elementales. Parámetros son definiciones geométricas del diseño que permite ser modificados durante el proceso de diseño tal como dimensión. Esto ofrece el control de la geometría del modelo por medio del uso de las variables del diseño. En fin, por procesos de revisión y de diseño automatizado con estas características paramétricas, permite el software para que tenga la tecnología avanzada de modelado del sólido como una herramienta eficiente del diseño.

El beneficio que ofrece el modelado sólido de forma paramétrica, es que empieza con modelo simple y conceptual con detalles al mínimo para poder realizar modificaciones, inspecciones, y pruebas preliminares con mucha más facilidad que luego se va aumentando de complejidad resultando en mejor diseño. Esto es para conformar con el detalle de la geometría ante el tamaño, o la escala real. Restricciones geométricas, dimensionales y relaciones paramétricas son implementables con propósito de ofrecer rigidez al diseño.

Datos del diseño previamente creados o existentes pueden ser reutilizados para creación de nuevos diseños. Además, posee la habilidad de actualizar el entorno del sistema cuando existe uno o más cambios de parámetros del dibujo, partes o ensambles para lograr diseño más complejo; y es posible explorar y evaluar diferentes variaciones y alternativas del diseño para un mejor diseño optimizado.

Paradigma de Ensamble Céntrica trata de permitir los usuarios o los diseñadores enfocarse en el diseño sin alguna dependencia sobre los parámetros o restricciones asociados. De esto, la función de ajuste basado en ensamble del Inventor permite determinar los tamaños y posiciones de las partes de forma automática mientras que el usuario tenga la opción de especificar las juntas o las uniones entre los componentes del diseño. Esta propiedad define que las relaciones entre las juntas o componentes directamente sin alguna orden de dependencia, y se conoce como enfoque de ensamble adaptativo directo (Shih, 2016).

c. ANSYS 17.2. ANSYS es software que implementa la teoría de elementos finitos en las simulaciones sobre la variedad de problemas. Estos problemas pueden ser desde problema estructural, mecánica, fluido, electromagnético y campo de acople. Las simulaciones realizadas para resolución de problemas utilizan métodos de elementos finitos para resolver ecuaciones dominantes. El propósito de simulación por medio de este software es para asesorar la eficiencia del diseño y ofrece mayor cantidad de herramientas dedicado para simulación (Lee, 2017).

El ANSYS Workbench es la interfaz principal para configurar, modelar y simular el diseño importado desde software CAD. El ANSYS es el software visto como Ingeniería Asistida por Computadora por la opción de que tiene la opción de realizar un diseño y al mismo tiempo realizar dicha simulación de interés sin la necesidad de exportar el archivo para utilizar en programas de las terceras.

En Workbench, ofrece variedad de entornos para poder someter el diseño a la simulación para dicho análisis necesario o de interés. Al especificar el entorno, ofrece opciones adicionales de acceder a Datos de Ingeniería en donde permite detallar los materiales a propiedades personalizados, e igual con las iteraciones a diferentes parámetros de las variables que el usuario configure. Modelo Geométrico es la sección que permite realizar el modelado del sólido. Así mismo permite cambiar la dimensión de la simulación; es decir, simulaciones especificados en 2-dimensiones o en 3-dimensiones.

La sección de la división geométrica en elementos finitos es donde define la utilidad principal del programa. El método de elementos finitos es implementado en esta sección para poder luego realizar la simulación con dichos parámetros configurados por el usuario y esto define el nivel de exactitud que tendrá al finalizar la simulación. Esta sección del entorno de configuración de simulación se conoce como Meshing. Luego, condiciones son impuestos para restringir y emular el ambiente de aplicación y en fin generan resultados, sea de esfuerzos, deformación, y factor de seguridad (Lee, 2017).

5. Análisis de materiales

a. **Introducción.** La selección de materiales para construir una maquina o elemento estructural es una de las decisiones más importantes que se deben tomar antes de establecer las dimensiones de la pieza. Seguido se deben estimar esfuerzos y deflexiones los cuales dependen de las propiedades de los materiales y así poder determinar que materiales son los adecuados a utilizar y que soporten las cargas a las cuales estarán sometidos. (Shigley, 2011)

b. **Resistencia y rigidez del material.** Los materiales seleccionados deben ser capaces de garantiza que no se deformen excesivamente y que no se fracturen. Un análisis para evaluar la deformación del material es por medio de la probeta de ensayo estándar que se usa para obtener una variedad de características y resistencias de los materiales. A partir de esto se obtienen resultados los cuales se grafican como un diagrama de esfuerzo-deformación unitaria el cual puede ser para un material dúctil o frágil, los dúctiles se deforman más que los frágiles. A continuación, se muestran los dos diagramas:

Figura 28. Esfuerzo-deformación material dúctil

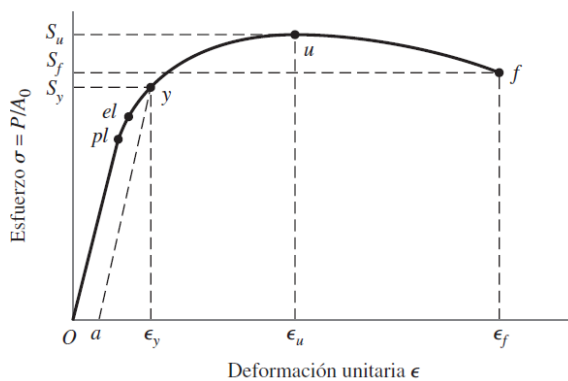
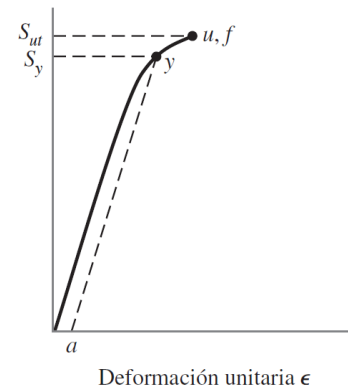


Figura 29. Material frágil



En la Figura 28 se puede observar que “pl” es el límite de proporcionalidad, punto en el cual se empieza a formar la curva. A este punto si se retira la carga no se observará ninguna deformación. Dicha parte lineal representa el módulo de elasticidad (E) el cual representa una medida de la rigidez de un material. El punto “el” es el límite elástico el cual si es sobrepasado el material sufrirá una deformación permanente al retirar la carga a la que esté sometida y antes de este punto es elástico es decir el material regresara a su forma natural a pesar de haber sufrido una cantidad límite de deformación. “Sy” es la resistencia a la fluencia el cual es el punto donde la deformación aumenta de forma representativamente y rápidamente sin que haya un cambio correspondiente en el esfuerzo. Por último, esta “Su” que es la resistencia ultima o de tensión que representa el esfuerzo máximo alcanzado en dicho diagrama. (Shigley, 2011)

c. Tipos de materiales

1) Tablero de fibra de densidad media (MDF). Este tipo de tablero está fabricado a partir de fibras de maderas fibras de madera, adhesivos, recubrimientos y aditivos. La composición típica suele ser un 80% de fibras, 10% de resinas sintéticas, 7% de agua y hasta un 1 % de parafinas. Lo que le aporta una mayor densidad de la que presentan aglomerados tradicionales o la madera contrachapada. Comúnmente se le llama madera MDF sin embargo esto no es exacto, ya que no estamos hablando de madera, sino de un producto derivado de ella. Durante el proceso de fabricación se pueden añadir determinados productos químicos con el fin de añadir características adicionales al MDF, como repeler el agua o evitar la aparición de hongos o moho. (Madera Santana, 2017)

Una de las fundamentales características del MDF es **la presión o densidad**, está nos diría el peso por metro cuadrado. Cuando hablamos de densidad media esta se sitúa entre 500 y 800 kg/m³, por encima de esta hablamos de alta densidad y por debajo de baja densidad. (Madera Santana, 2017)

Según su utilización se clasifican según la normativa (UNE-EN 622-5) (AITIM, 2015):

- MDF para utilización general en ambiente seco.
- MDF.H para utilización general en ambiente húmedo.
- MDF.LA estructurales para utilización en ambiente seco.
- MDF. HLS estructurales para utilización en ambiente húmedo.
- L-MDF ligeros para aplicaciones no estructurales para utilización general en ambiente seco.
- L-MDF.H ligeros para aplicaciones no estructurales utilizadas en ambiente húmedo.
- UL1-MDF ultraligeros para aplicaciones no estructurales para utilización general en ambiente seco.
- UL2-MDF ultraligeros para aplicaciones no estructurales para utilización general en ambiente seco.
- MDF.RWH para utilización como subcapas rígidas en muros y cubiertas.

2) Ácido Poliláctico (PLA). EL PLA debe su aparición al científico de la marca Dupont: Wallace Carothers. Consiguió en 1932 un polímero de bajo peso molecular. Fue patentado (conocido como proceso de Carothers) bajo la misma firma a mediados de los años cincuenta, tras optimizar el proceso. El PLA es un polímero biodegradable que podemos encontrar en el mercado de comercialización de plásticos desde 1990. Es uno de los plásticos en mayor estudio y desarrollo por sus múltiples aplicaciones en el mercado. (DIMA 3D, 2015)

Este interesante polímero tiene su origen en el almidón proveniente de vegetales como: maíz, yuca, trigo, remolacha o caña de azúcar, lo que facilita su fuente de extracción alejándola de los procesos de transformación de los hidrocarburos que caracterizan la obtención de otros polímeros. Además, es claro y brillante como el poliestireno (se utiliza para fabricar baterías y juguetes). Resistente a la humedad y a la grasa. Tiene características de barrera del sabor y del olor similares al plástico de polietileno tereftalato, usado para las bebidas no alcohólicas y para otros productos no alimenticios. (DIMA 3D, 2015)

Figura 30. Proceso de obtención del PLA



d. Propiedades Mecánicas de los Materiales

1) MDF. Se caracterizan por su uniformidad y homogeneidad en todo su espesor. No presentan problemas para su corte y se mecanizan y molduran con mucha facilidad. (AITIM, 2015)

Cuadro 4. Propiedades Mecánicas del MDF

| Propiedad | Tolerancia | Unidad | Valor |
|---|---------------|------------|---------------|
| Resistencia a la humedad | Media | % | 4-10 |
| Conductividad térmica | Mal Conductor | Kcal/mh*°C | 0.07-0.14 |
| Aislamiento acústico | Media | dbA | 25-30 |
| Reacción al fuego | N/A | M | M-4, M-1, M-2 |
| Comportamiento frente agente biológicos | Buena | N/A | N/A |

Además, se encuentran otras propiedades físico-mecánicas que se presentan a continuación (MASISA, 2017):

Cuadro 5. Propiedades físico-mecánicas del MDF

| Propiedades físico-mecánicas | | | | | | | | |
|------------------------------|--------|------------|-------|-------|-------|-------|-------|-------|
| Propiedad | Unidad | Tolerancia | | | | | | |
| Espesor | mm | ± 0,2 | 9 | 12 | 15 | 18 | 25 | 30 |
| Densidad | Kg/m3 | ±20 | 700 | 680 | 650 | 630 | 610 | 600 |
| Resistencia a la Tracción | N/mm2 | ±0,15 | 0,50 | 0,50 | 0,50 | 0,50 | 0,45 | 0,45 |
| Resistencia a la Flexión | N/mm2 | ±2 | 19 | 18 | 18 | 18 | 16 | 14 |
| Módulo de Elasticidad | N/mm2 | Mínimo | 2.400 | 2.300 | 2.100 | 2.100 | 2.000 | 2.000 |

2) PLA. A continuación, se presentan unas propiedades físicas del PLA en comparación con otros Plásticos (Villaverde, M, 2017):

Cuadro 6. Propiedades físicas del PLA

| Característica | PLA DEL CDP | PLA de Kuraray | Rayón | Lyocell | Poliester |
|---------------------------|-------------|----------------|---------|---------|-----------|
| Densidad (g/cm3) | 1.25 | 1.27 | 1.51 | 1.52 | 1.38 |
| Punto de fusión (°C) | 120-170 | 170 | Ninguno | Ninguno | 260 |
| Cuenta (dtex) | 1.7 (e.g.) | 1.7 | 1.7 | 1.7 | 1.7 |
| Tenacidad (cN/tex) | 50 | 25-45 | 20-25 | 40-45 | 35-65 |
| Alargamiento (%) | 35 | N/A | 18-22 | 14-16 | 15-40 |
| Recobro de la humedad (%) | 0.4 - 0.6 | 0.48 | 13 | 12 | 0.4 |

También se presentan propiedades mecánicas del PLA aproximadamente a temperatura ambiente (PROSPECTOR, 2017):

Cuadro 7. Propiedades mecánicas del PLA

| Prop. mecánicas | Valor Típico | Unidad | Método de Ensayo |
|--------------------------|-----------------|--------|------------------|
| Módulo de tensión | | | |
| 73°F | 293000 a 514000 | psi | ASTM D638 |
| 73°F | 45000 a 815000 | psi | ISO 527-2 |
| Tensión | | | |
| Punto de Fluencia, 73°F | 8840 a 9500 | psi | ASTM D638 |
| Punto de Fluencia, 73°F | 2250 a 10400 | psi | ISO 527-2 |
| Rotura, 73°F | 7080 a 8150 | psi | ASTM D638 |
| Rotura, 73°F | 2000 a 10200 | psi | ISO 527-2 |
| 73°F | 6930 a 10000 | psi | ASTM D638 |
| Elongación | | | |
| Punto de Fluencia, 73°F | 9,8 a 10 | % | ASTM D638 |
| Punto de Fluencia, 73°F | 1,0 a 8,5 | % | ISO 527-2 |
| Rotura, 73°F | 0,50 a 9,2 | % | ASTM D638 |
| Rotura, 73°F | 1,0 a 12 | % | ISO 527-2 |
| Módulo de Flexión | | | |
| 73°F | 347000 a 715000 | psi | ASTM D790 |
| 73°F | 44200 a 1,38E+6 | psi | ISO 178 |
| Resistencia a la flexión | | | |
| 73°F | 6950 a 16000 | psi | ASTM D790 |
| 73°F | 1310 a 16100 | psi | ISO 178 |

6. **Análisis estructural.** La idea de este análisis es poder determinar el punto crítico de una pieza o una estructura sometida a cargas externas con el fin de ubicar en dónde se encuentra el nivel máximo de esfuerzos y así establecer si ocurrirán fallas, deformaciones, etc. A continuación, se encuentran algunas suposiciones que se toman cuando se trabaja con esfuerzos en el análisis estructural (Mott, 2006):

1. La línea de acción de la carga debe pasar por el centroide de la sección transversal de la pieza (para esfuerzos normales).
2. La pieza debe tener sección transversal uniforme cerca de donde se calcule el esfuerzo (para esfuerzos normales).
3. Para piezas en compresión, la misma debe ser corta con el fin de que no suceda pandeo.
4. El material debe ser isotrópico y homogéneo.

a. **Esfuerzos.** Un esfuerzo es el resultado de aplicar una carga externa sobre una unidad de área de un material. En otras palabras, es la resistencia interna que presenta dicho material ante dicha carga. Las unidades de un esfuerzo, sin importar si es tensión directa, compresión directa, cortante directo, cortante torsional o flexión; siempre son fuerza por unidad de área. (Mott, 2006)

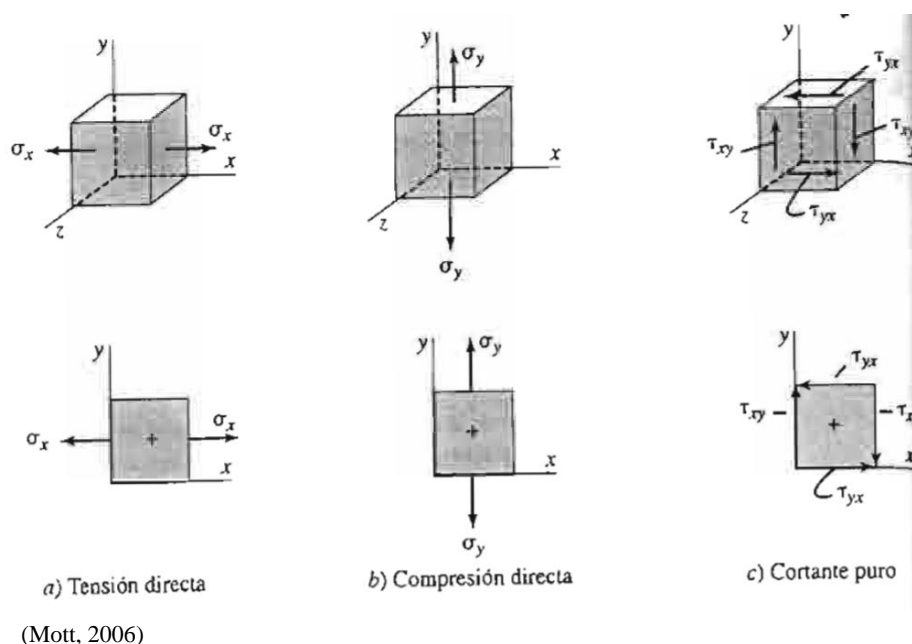
Figura 31. Unidades de medición para los esfuerzos.

| <i>Sistema Estadounidense Tradicional</i> | <i>Unidades SI métricas</i> |
|---|--|
| $\text{lb/pulg}^2 = \text{psi}$ | $\text{N/m}^2 = \text{pascal} = \text{Pa}$ |
| $\text{kips/pulg}^2 = \text{ksi}$ | $\text{N/mm}^2 = \text{megapascal} = 10^6 \text{ Pa} = \text{MPa}$ |
| Nota: $1.0 \text{ kip} = 1000 \text{ lb}$ | |
| $1.0 \text{ ksi} = 1000 \text{ psi}$ | |

(Mott, 2006)

b. **Elementos de esfuerzos.** Para representar los esfuerzos se visualiza un elemento de esfuerzos, el cual consta de un cubo infinitesimalmente pequeño y simple en la ubicación crítica en donde se desee realizar el análisis. En esta representación se indican los vectores propios de los esfuerzos que existen específicamente sobre el elemento en dicha ubicación. Además, es importante asignar un sistema de coordenadas x, y, z para indicar la orientación del elemento de esfuerzos. A continuación, se muestran tres ejemplos de elementos de esfuerzo (Mott, 2006):

Figura 32. Representación gráfica de los elementos de esfuerzos

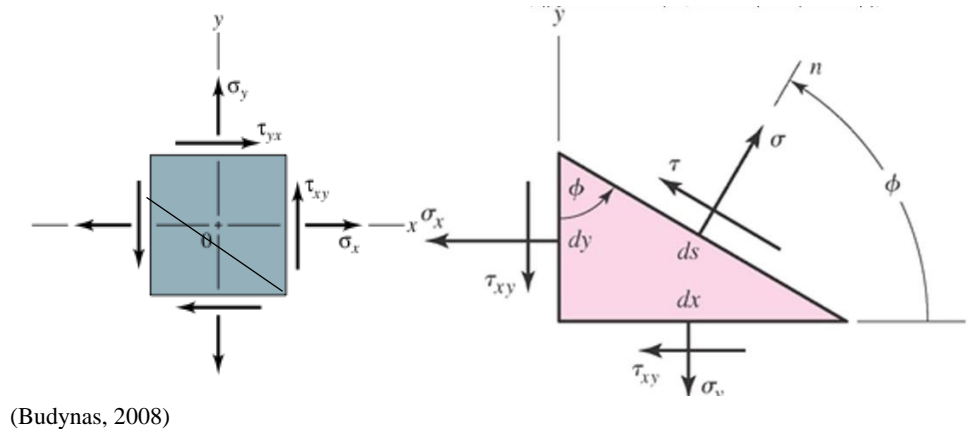


En la Figura 32 se puede observar que los elementos de esfuerzo se pueden realizar como un cubo tridimensional completo, o bien, un cuadrado bidimensional simplificado. De hecho, se recomienda primero realizar la forma cúbica para luego decidir el plano de interés que permita ver los esfuerzos y así realizar la forma bidimensional. En cada uno de los elementos de la figura actúa un esfuerzo diferente, por tal motivo cada uno tiene los vectores en una dirección distinta.

Para los esfuerzos de tensión y compresión directa los vectores actúan perpendicularmente sobre caras opuestas del elemento de esfuerzos. Para esfuerzos de cortante directo, un par de vectores actúa en dirección opuesta y paralelos a los planos que tienden a ser “cortados”, mientras que otro par de vectores actúa siempre con dirección opuesta sobre los planos restantes. Esta notación es importante, porque de existir solo un par de vectores, el elemento no estaría en equilibrio y por lo tanto tendería a girar. Además, los esfuerzos cortantes tienen una convención de signos: son positivos si tienden a girar al elemento de esfuerzos en sentido de las manecillas del reloj y son negativos si tienden a girarlo en la dirección contraria. Y finalmente, en la notación, los esfuerzos cortantes se indican con doble subíndice, de forma que en la figura los esfuerzos perpendiculares al eje x tienen el subíndice xy mientras que los esfuerzos perpendiculares al eje y tienen el subíndice yx . (Mott, 2006)

c. Esfuerzos principales. Cuando una pieza mecánica está sometida a combinaciones de esfuerzos normales y cortantes, se debe calcular el valor y estado de los esfuerzos reales en el componente. Esto se lleva a cabo calculando los esfuerzos principales y el esfuerzo cortante máximo en el punto crítico de la pieza. Primeramente, se realiza un corte bidimensional y un análisis vectorial del elemento de esfuerzos tal como se observa en la Figura 33 (Mott, 2006).

Figura 33. Planteamiento gráfico para ubicar los esfuerzos principales

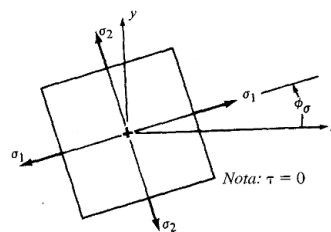


Luego, analizando equilibrio y diferenciando la expresión respecto al ángulo phi, se obtienen los dos esfuerzos principales, uno siendo el esfuerzo normal máximo σ_1 y el otro el esfuerzo normal mínimo σ_2 . Cabe mencionar que de este análisis se observa que el esfuerzo principal máximo siempre está a 90° del esfuerzo principal mínimo, y, además, el esfuerzo cortante es cero. A continuación, las expresiones para encontrar los esfuerzos principales, sus ángulos y la representación gráfica, respectivamente.

Figura 34. Ecuaciones de los esfuerzos principales, sus ángulos y su representación gráfica

$$\sigma_1, \sigma_2 = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (4)$$

$$\tan 2\phi_p = \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \quad (5)$$



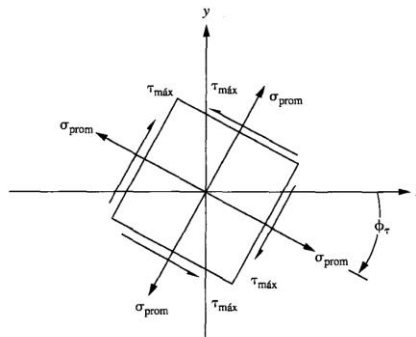
(Budynas, 2008)

d. Esfuerzo cortante máximo. Para este esfuerzo se vuelve a realizar análisis de equilibrio, pero respecto a TAU y luego se deriva respecto a phi. Lo que se obtiene son los dos ángulos para los cuales se obtiene el esfuerzo cortante máximo. Para este caso también existe un esfuerzo normal, siendo el promedio de los esfuerzos normales originales. A continuación, las expresiones para encontrar los dos esfuerzos cortantes máximos, sus ángulos y la representación gráfica, respectivamente. (Mott, 2006)

Figura 35. Ecuaciones de los esfuerzos cortantes máximos, sus ángulos y su representación gráfica

$$\tau_1, \tau_2 = \pm \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (6)$$

$$\tan 2\phi_s = -\frac{\sigma_x - \sigma_y}{2\tau_{xy}} \quad (7)$$



(Budynas, 2008)

e. Tensión y compresión directas. Este tipo de esfuerzos se representa con el símbolo SIGMA y se clasifica como esfuerzos normales porque actúan perpendicularmente sobre caras opuestas de una pieza o estructura. El esfuerzo de compresión tiende a aplastar dicha pieza y se suele identificar con signo negativo, mientras que el esfuerzo de tensión tiende a estirarlo y a identificarse con signo positivo. (Mott, 2006)

$$\sigma = \frac{\text{Fuerza}}{\text{Área}} = \frac{F}{A} \quad (8)$$

Además de calcular el esfuerzo, siempre es importante definir cuánto se deformará la pieza. Y para calcularlo, se utiliza la siguiente fórmula:

$$\delta = \frac{FL}{EA} \quad (9)$$

donde

δ = deformación total del miembro que soporta la carga axial

F = carga axial directa

L = longitud original total del miembro

$E = \text{módulo de elasticidad del material}$

$A = \text{área de la sección transversal del elemento}$

f. Cortante directo. Tal como su nombre lo indica, este tipo de esfuerzo tiende a cortar una pieza. Esto porque su forma de actuar es una fuerza aplicada hacia arriba sobre una cara mientras que otra fuerza de la misma magnitud, pero de dirección contraria es aplicada hacia abajo sobre la cara paralela opuesta y en el mismo punto de la otra fuerza, tal y como funciona una tijera. Este esfuerzo se representa con el símbolo TAU y se calcula bastante parecido a como se calculan los esfuerzos normales (Mott, 2006):

$$\tau = \frac{\text{Fuerza de corte}}{\text{Área al corte}} = \frac{F}{A_s} \quad (10)$$

g. Cortante torsional, para secciones circulares y no circulares. En este caso el corte sucede cuando una parte de una pieza gira en relación con otra. Esto ocurre cuando un momento de torsión o par de torsión se aplica sobre la pieza y tiende a deformarse por torcimiento. El caso más común en donde sucede este tipo de esfuerzo es en un eje al cual se le transmite potencia. Dado que la distribución de este esfuerzo no es uniforme en la sección transversal, tal como los esfuerzos normales y el cortante directo, la fórmula para calcularlo es diferente.

En un eje sólido o hueco y redondo que recibe un par de torsión, el esfuerzo cortante torsional máximo (y también la máxima deformación cortante unitaria) se encuentra en la superficie externa mientras que el mínimo se encuentra en su eje neutro. El esfuerzo cortante torsional máximo se calcula así (Mott, 2006):

$$\tau_{\text{máx}} = \frac{Tc}{J} \quad (11)$$

donde $c = \text{radio de la superficie externa del eje}$

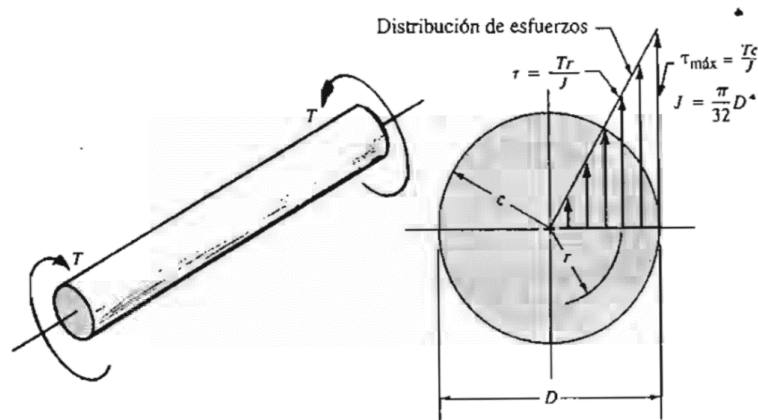
$J = \text{momento polar de inercia}$

Otra forma de calcular el esfuerzo cortante torsional, especialmente importante para el diseño, es utilizando el módulo de sección polar Z_p . Este método es importante para problemas de diseño porque con el módulo de sección polar se relaciona la geometría transversal de la pieza, tal como se observa en la ecuación a continuación.

$$Z_p = \frac{J}{c} \quad (12)$$

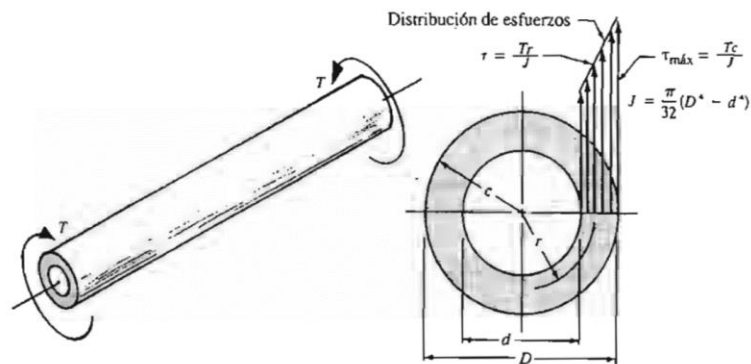
$$\tau_{\text{máx}} = \frac{T}{Z_p} \quad (13)$$

Figura 36. Representación gráfica de la distribución de esfuerzos para un eje circular bajo torsión



(Mott, 2006)

Figura 37. Representación gráfica de la distribución de esfuerzos para un eje tubular bajo torsión



(Mott, 2006)

Por otro lado, también pueden ocurrir deformaciones angulares debidas a la torsión. Esto en otras palabras, es cuando una sección transversal de un eje gira respecto a otras secciones transversales del mismo. Esta deformación se calcula así:

$$\theta = \frac{TL}{GJ} \quad (14)$$

donde θ = ángulo de torsión (radianes)

L = longitud del eje donde se calcula el ángulo de torsión

G = módulo de elasticidad del material del eje en cortante

Ahora bien, para torsión sobre elementos de sección transversal no circular el comportamiento de dichas secciones es muy diferente al de elementos con sección transversal circular. Sin embargo, los valores importantes siguen siendo el esfuerzo máximo y el ángulo total de deformación, que convenientemente,

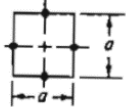

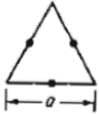

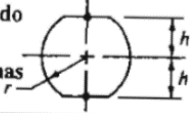
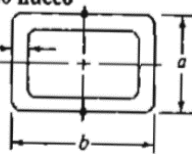

tienen fórmulas bastante parecidas. Es importante mencionar que en estas fórmulas se asume que los extremos del elemento tienen libertad de deformación. Esto significa que, si el extremo está soldado a otra pieza, los resultados cambian bastante.

$$\tau_{m\acute{a}x} = \frac{T}{Q} \quad (15)$$

$$\theta = \frac{TL}{GK} \quad (16)$$

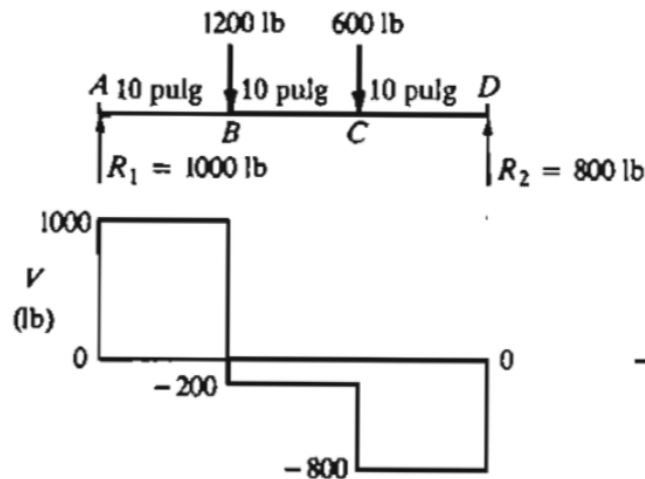
Los valores de K y Q son diferentes para cada tipo de sección transversal y se determinan así:

Figura 38. Criterios para el valor de K y Q según la forma de la sección transversal

| Forma de la sección transversal | $K =$ para usar en $\theta = TL/GK$ $Q =$ para usar en $\tau = T/Q$ | El punto negro (*) indica la ubicación de $\tau_{m\acute{a}x}$ | | | | | | | | | | | | | | | | | | | | | |
|---|---|--|-------|------|------|------|-----|-----|-----|-------|------|------|------|------|------|------|-------|------|------|------|------|------|------|
| Cuadrado  | $K = 0.141a^4$ $Q = 0.208a^3$ | $\tau_{m\acute{a}x}$ en el punto medio de cada lado | | | | | | | | | | | | | | | | | | | | | |
|  | $K = bh^3 \left[\frac{1}{3} - 0.21 \frac{h}{b} \left(1 - \frac{(h/b)^2}{12} \right) \right]$ $Q = \frac{bh^2}{[3 + 1.8(h/b)]}$ | (Aproximado; dentro de $\approx 5\%$) $\tau_{m\acute{a}x}$ en el punto medio de los lados largos | | | | | | | | | | | | | | | | | | | | | |
| Triángulo (equilátero)  | $K = 0.0217a^4$ $Q = 0.050a^3$ | | | | | | | | | | | | | | | | | | | | | | |
| Eje redondo con una cara plana  | $K = C_1 r^4$ $Q = C_2 r^3$ | <table border="1"> <thead> <tr> <th>h/r</th> <th>0</th> <th>0.2</th> <th>0.4</th> <th>0.6</th> <th>0.8</th> <th>1.0</th> </tr> </thead> <tbody> <tr> <td>C_1</td> <td>0.30</td> <td>0.51</td> <td>0.78</td> <td>1.06</td> <td>1.37</td> <td>1.57</td> </tr> <tr> <td>C_2</td> <td>0.35</td> <td>0.51</td> <td>0.70</td> <td>0.92</td> <td>1.18</td> <td>1.57</td> </tr> </tbody> </table> | h/r | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | C_1 | 0.30 | 0.51 | 0.78 | 1.06 | 1.37 | 1.57 | C_2 | 0.35 | 0.51 | 0.70 | 0.92 | 1.18 | 1.57 |
| h/r | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | | | | | | | | | | | | | | | | | |
| C_1 | 0.30 | 0.51 | 0.78 | 1.06 | 1.37 | 1.57 | | | | | | | | | | | | | | | | | |
| C_2 | 0.35 | 0.51 | 0.70 | 0.92 | 1.18 | 1.57 | | | | | | | | | | | | | | | | | |
| Eje redondo con dos caras planas  | $K = C_3 r^4$ $Q = C_4 r^3$ | <table border="1"> <thead> <tr> <th>h/r</th> <th>0.5</th> <th>0.6</th> <th>0.7</th> <th>0.8</th> <th>0.9</th> <th>1.0</th> </tr> </thead> <tbody> <tr> <td>C_3</td> <td>0.44</td> <td>0.67</td> <td>0.93</td> <td>1.19</td> <td>1.39</td> <td>1.57</td> </tr> <tr> <td>C_4</td> <td>0.47</td> <td>0.60</td> <td>0.81</td> <td>1.02</td> <td>1.25</td> <td>1.57</td> </tr> </tbody> </table> | h/r | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | C_3 | 0.44 | 0.67 | 0.93 | 1.19 | 1.39 | 1.57 | C_4 | 0.47 | 0.60 | 0.81 | 1.02 | 1.25 | 1.57 |
| h/r | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | | | | | | | | | | | | | | | | | |
| C_3 | 0.44 | 0.67 | 0.93 | 1.19 | 1.39 | 1.57 | | | | | | | | | | | | | | | | | |
| C_4 | 0.47 | 0.60 | 0.81 | 1.02 | 1.25 | 1.57 | | | | | | | | | | | | | | | | | |
| Rectángulo hueco (uniforme)  | $K = \frac{2t(a-2t)^2(b-2t)^2}{(a+b-2t)}$ $Q = 2t(a-2t)(b-2t)$ | Obtiene el esfuerzo promedio; buena aproximación al esfuerzo máximo si t es pequeño-tubo de pared delgada Los rincones interiores deben tener filetes generosos | | | | | | | | | | | | | | | | | | | | | |
| Tubo abierto Radio promedio (r)  t (uniforme) | $K = 2\pi r^3/3$ $Q = \frac{4r^2t^2}{(6\pi r + 1.8t)}$ | t debe ser pequeño-tubo de pared delgada | | | | | | | | | | | | | | | | | | | | | |

h. Esfuerzos cortantes verticales en vigas. Una viga es cualquier elemento que soporta cargas transversales a su eje. Estas cargas provocan la aparición de fuerzas de corte que se representan con la letra V. Lo que se acostumbra es definir la fuerza cortante a lo largo de la viga utilizando un diagrama de fuerza cortante como el que se observa a continuación. (Mott, 2006)

Figura 39. Diagrama de fuerza cortante



(Mott, 2006)

Para calcular el esfuerzo cortante para perfiles de uso frecuente se utilizan las siguientes expresiones:

Figura 40. Ecuaciones para esfuerzo cortante de perfiles comunes

$\tau_{\text{máx}}$ para
el rectángulo

$$\tau_{\text{máx}} = 3V/2A \text{ (exacta)}$$

Donde A = superficie transversal total de la viga

$\tau_{\text{máx}}$ para
el círculo

$$\tau_{\text{máx}} = 4V/3A \text{ (exacta)}$$

$\tau_{\text{máx}}$ para
viga I

$$\tau_{\text{máx}} \approx V/ih \text{ (aproximada: un 15% baja)}$$

donde t = espesor del alma

h = altura del alma (por ejemplo, una viga de patín ancho)

$\tau_{\text{máx}}$
para tubo de
pared delgada

$$\tau_{\text{máx}} \approx 2V/A \text{ (aproximada: un poco alta)}$$

En todos estos casos, el esfuerzo cortante se presenta en el eje neutro.

(Mott, 2006)

i. Flexión. Aquí se toma en cuenta que las cargas transversales a una viga provocan momentos de flexión, y, por ende, esfuerzos de flexión. Estos esfuerzos son esfuerzos normales, pero con la variación de que su magnitud cambia linealmente dentro del área transversal, empezando en cero desde el eje neutro hasta alcanzar su valor máximo en la superficie (Mott, 2006).

$$\sigma = \frac{Mc}{I} \quad (17)$$

donde

M = magnitud del momento de flexión en esa sección

I = momento de inercia del área transversal con respecto a su eje neutro

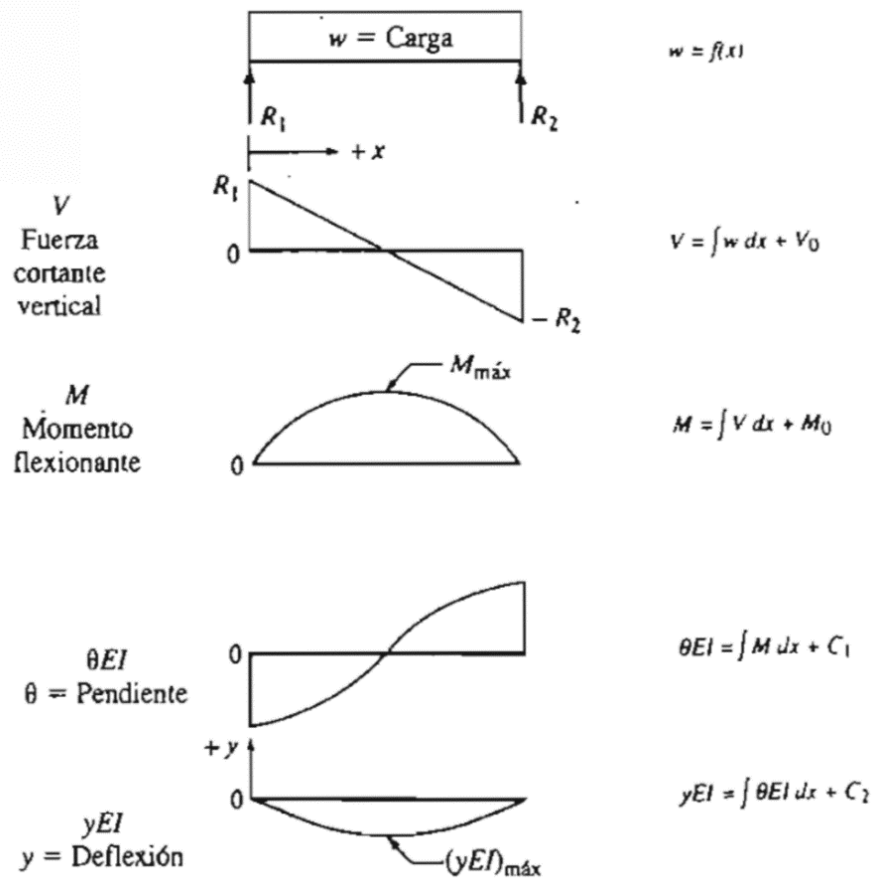
c = distancia del eje neutro a fibra más alejada, en la sección transversal de la viga

En este caso se hacen las siguientes suposiciones:

1. No hay cargas axiales.
2. El esfuerzo cortante debe ser cero o despreciable.
3. La viga tiene que estar en flexión pura.
4. El material de la viga debe comportarse según la Ley de Hooke.
5. No debe haber torsión aplicada a la viga.
6. El módulo de elasticidad debe ser igual en tensión y en compresión.
7. No hay falla por pandeo.
8. La viga es recta al principio y es de sección transversal constante.

Cuando se habla de flexión hay que tomar en cuenta un factor crítico: la deflexión de la viga. Esta ocurre cuando una viga inicialmente recta se deforma y termina con una forma ligeramente curva. Existen programas para computadora que modelan las vigas y permiten el cálculo de estas deflexiones, pero para interpretar sus resultados es importante entender cómo se determinan. Y una herramienta para llevar a cabo lo anteriormente mencionado son los diagramas de fuerza cortante vertical, momento de flexión, pendiente de la viga y curva de deflexión. Los cuales se desarrollan así:

Figura 41. Diagramas de fuerza cortante vertical, momento flexionante, pendiente y deflexión



(Mott, 2006)

Por eso para calcular la deflexión de una viga se pueden utilizar las siguientes relaciones:

Figura 42. Expresiones para calcular la deflexión de una viga

$$y = f(x)$$

$$\theta = \frac{dy}{dx}$$

$$\frac{M}{EI} = \frac{d^2y}{dx^2}$$

$$\frac{V}{EI} = \frac{d^3y}{dx^3}$$

$$\frac{w}{EI} = \frac{d^4y}{dx^4}$$

(Mott, 2006)

j. **Concentración de esfuerzos.** En la mayoría de los casos, las piezas tienen cambios de geometrías inherentes a lo largo de su estructura para cumplir alguna función en especial, por lo que no se puede asumir una sección transversal constante a lo largo de toda la pieza. Esto lo que provoca es la aplicación de un factor de concentración de esfuerzos que incrementa el esfuerzo calculado con las fórmulas anteriores para tomar en cuenta los efectos de dichos cambios geométricos. En general, estos factores se aplican así (Mott, 2006):

$$\sigma_{m\acute{a}x} = K_t \sigma_{nom} \quad (18)$$

$$\tau_{m\acute{a}x} = K_t \tau_{nom} \quad (19)$$

El factor K depende de la forma de la discontinuidad, del tipo de esfuerzo y de la geometría específica. Por eso, para calcularlo existen varias fuentes de referencia que se basan en gráficas producidas por evaluaciones de prueba y error para cada caso. Pero para saber cuándo aplicar los factores y cómo entenderlos, se toman las siguientes pautas:

1. Muchos factores de concentración de esfuerzos son valores empíricos que solo se basan en la forma de la carga y la geometría de un elemento.
2. El peor de los casos ocurre para áreas en tensión.
3. Para materiales dúctiles con cargas estáticas, se puede ignorar las concentraciones de esfuerzos.
4. Para materiales frágiles con cargas estáticas o de fatiga, es imperativo el uso de factores de concentración de esfuerzos.
5. Alteraciones en el material como muescas, ralladuras, corrosión, galvanoplastia y aspereza excesiva en la superficie pueden provocar concentraciones de esfuerzos.

7. Diseño de mecanismos

a. **Introducción.** En el estudio de los mecanismos se definen dos conceptos importantes que son:

- **Cinética:** Estudio de las fuerzas sobre sistemas en movimiento.
- **Cinemática:** Estudio del movimiento sin considerar las fuerzas.

El objetivo fundamental de la cinemática de mecanismos es crear (diseñar) los movimientos deseados de las partes mecánicas y luego calcular matemáticamente las posiciones, velocidades y aceleraciones que los movimientos crearán en las partes (Norton Robert, 2009).

- **Mecanismo:** Es un dispositivo que transforma el movimiento en un patrón deseable, y por lo general desarrolla fuerzas muy bajas y transmite poca potencia.

- Máquina: Contiene mecanismos que están diseñados para producir y transmitir fuerzas significativas.

b. Movilidad o Grados de Libertad. La movilidad de un mecanismo mecánico se puede clasificar de acuerdo con el número de grados de libertad que posee. Los grados de libertad (GDL) del sistema es igual al número de parámetros independientes que se requiere para definir de manera única su posición en el espacio en cualquier instante de tiempo. Los GDL se definen respecto a un marco de referencia seleccionado (Norton Robert, 2009).

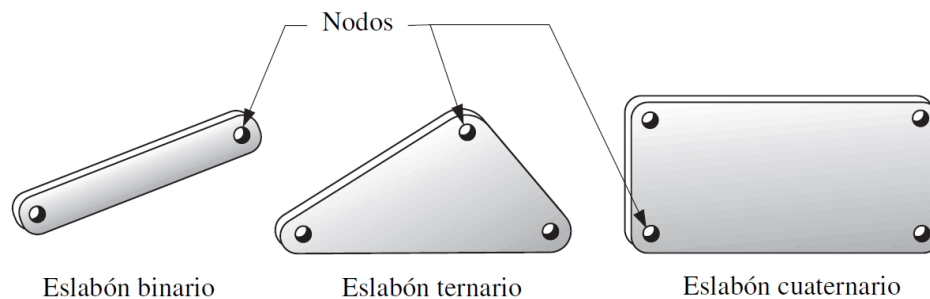
Figura 43. Junta Universal, 2 GDL



c. Eslabones, juntas y cadenas cinemáticas. A continuación, se presenta cierta información sobre la caracterización de los eslabones y juntas:

- Eslabón: Es un cuerpo rígido que posee por lo menos dos nodos que son puntos de unión con otros eslabones (Norton Robert, 2009). Se clasifican en: Eslabón binario, ternario y cuaternario.

Figura 44. Eslabones de diferente orden



- Junta: Es una conexión entre dos o más eslabones (en sus nodos), la cual permite algún movimiento, o movimiento potencial, entre los eslabones conectados. Las juntas también llamadas pares cinemáticos se pueden clasificar de la siguiente manera (Norton Robert, 2009):
 - Por el tipo de contacto entre los elementos, de línea, de punto o de superficie
 - Por el número de grados de libertad permitidos en la junta
 - Por el tipo de cierre físico de la junta: cerrada por fuerza o por forma
 - Por el número de eslabones unidos (orden de la junta)

Tipos de Juntas, con sus grados de libertad:

Cuadro 8. Tipos de juntas con sus grados de libertad

| Los seis pares inferiores | | |
|---------------------------|-----|----------|
| Nombre (símbolo) | GDL | Contiene |
| Revoluta (R) | 1 | R |
| Prismático (P) | 1 | P |
| Helicoidal (H) | 1 | RP |
| Cilíndrica (C) | 2 | RP |
| Esférica (S) | 3 | RRR |
| Plana (F) | 3 | RPP |

Cadena cinemática: Es un ensamble de eslabones y juntas interconectados de modo que produzcan un movimiento controlado en respuesta a un movimiento suministrado (Norton Robert, 2009).

Definición de ciertos eslabones de una cadena cinemática

- Manivela: Es un eslabón que realiza una revolución completa y está pivoteada a la bancada.
- Balancín: Es un eslabón que tiene rotación oscilatoria (de vaivén) y está pivotado a la bancada.
- Acoplador: Es un eslabón que tiene movimiento complejo y no está pivotado a la bancada
- Bancada: Se define como cualquier eslabón o eslabones que están fijos con respecto al marco de referencia.

Para calcular los grados de libertad o movilidad se emplea la ecuación modificada de Kutzbach de la ecuación de Gruebler de la siguiente forma:

$$M = 3(L - 1) - 2J_1 - J_2 \quad (11)$$

Donde:

M = grado de libertad o movilidad

L = número de eslabones

J_1 = número de juntas de 1 GDL (completas)

J_2 = número de juntas de 2 GDL (semi)

8. Métodos de unión

a. Tipos de métodos de unión. En la mayoría de los casos, las máquinas, las herramientas, útiles y mecanismos están compuestos por varias piezas unidas entre sí con distintos accesorios de unión para así realizar su función deseada. Los tipos de uniones pueden ser:

1) **Desmontables.** Permiten separar las piezas con facilidad, sin romper el medio de unión ni las propias piezas. Entre las más destacadas están los pernos con roscas, chavetas, lengüetas y pasadores (Hervas Villalba, 2011).

2) **Fijas o no desmontables.** Se realizan con piezas cuyo desmontaje no se prevé durante la vida útil de la máquina o estructura o, en otros casos, por seguridad o exigencia del diseño. Para la separación de las piezas necesitamos romper el elemento de unión o, en muchos casos, deteriorar alguna de las piezas. Destacan el roblonado, el remache tubular y los diferentes tipos de soldaduras, así como también la aplicación de adhesivos naturales o sintéticos. (Hervas Villalba, 2011)

A continuación, se muestra una tabla con las uniones más utilizadas según el material a utilizar:

Cuadro 9. Métodos de unión dependiendo de material

| Material | Unión desmontable | Unión fija |
|-----------|--|---|
| Metales | <ul style="list-style-type: none"> • Elementos roscados • Pasadores • Chavetas • Ejes estriados • Guías | <ul style="list-style-type: none"> • Remaches y roblones • Piezas ajustadas a presión • Soldadura • Adhesivos |
| Plásticos | <ul style="list-style-type: none"> • Elementos roscados. | <ul style="list-style-type: none"> • Adhesivos |
| Madera | <ul style="list-style-type: none"> • Elementos roscados | <ul style="list-style-type: none"> • Clavo • Adhesivos |
| Textiles | <ul style="list-style-type: none"> • Botón y ojal • Cremallera • Tiras de velero • Corchetes | <ul style="list-style-type: none"> • Costura • Adhesivos |
| Cerámicos | <ul style="list-style-type: none"> • Elementos roscados | <ul style="list-style-type: none"> • Adhesivos |
| Pétreos | | <ul style="list-style-type: none"> • Adhesivos • Cementos |

9. Servomotores Dynamixel

Dynamixel es una marca dedicada al desarrollo de servomotores inteligentes específicamente para aplicaciones robóticas o estructuras mecánicas. Por ello, hoy en día es el referente principal de actuadores

para robots en donde se necesita un alto desempeño. Las características más importantes de sus productos son (Robotis, 2017):

a. Características importantes

- Estructura de conexión simple: Se pueden interconectar entre sí por una configuración Daisy Chain.
- Configuración de torque: En aplicaciones donde se requiera levantar carga se pueden configurar 1023 niveles de torque según sea el requerimiento.
- Retroalimentación: Del servomotor se puede obtener información tal como su posición, velocidad, carga y temperatura interna.
- Actuación a través de una red: A cada servomotor se le asigna un ID diferente para ser controlado con un paquete de comunicación determinado por el protocolo del actuador. Según el modelo del mismo, puede soportar redes TTL, RS485, CAN, entre otros.
- LED de estado: Es una retroalimentación visual para verificar el estado del servomotor.
- Alarma: Para casos en donde el voltaje de alimentación, el torque o la temperatura, entre otros, sobrepasan un límite establecido, el actuador enciende el LED de estado o brinda otra forma de retroalimentación de la situación.
- Estructura externa para ensamblaje: El encapsulado y los accesorios están diseñados para ensamblajes modulares entre varios servomotores.
- Consumo de baja corriente/alto voltaje: Con esta combinación de factores se obtiene una alta eficiencia y estabilidad del sistema robótico.

b. Dynamixel AX-12A

Figura 45. Servomotor Dynamixel AX-12A



(RobotShop, 2017)

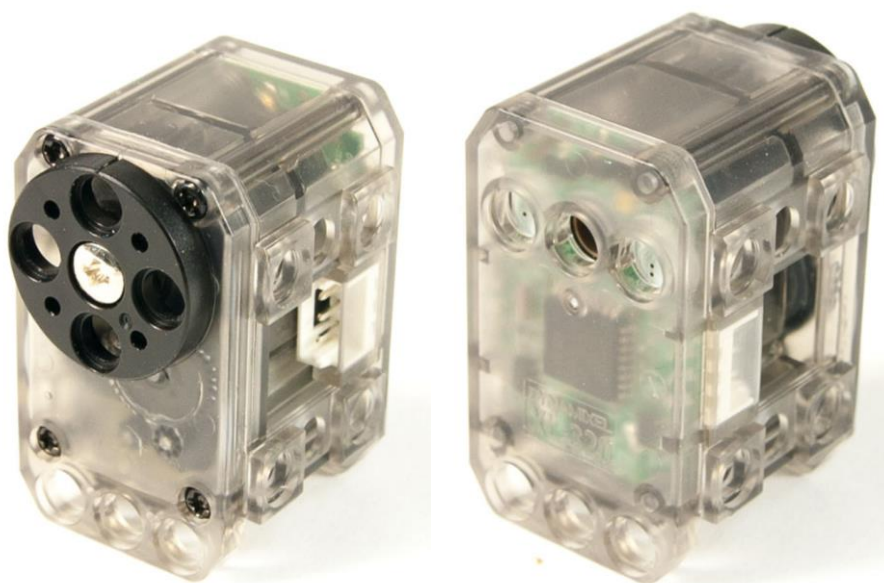
Cuadro 10. Especificaciones técnicas del servomotor Dynamixel AX-12A

| ESPECIFICACIÓN | VALOR |
|-------------------------------|---|
| Dimensiones | 32mm * 50mm * 40mm |
| Peso | 54.6g |
| Voltaje de alimentación | 9 ~ 12V (Recomendado: 11.1V) |
| Velocidad de comunicación | 7343bps – 1 Mbps |
| Resolución | 0.29° |
| Relación de reductor | 254:1 |
| Torque | 1.5N.m (12.0V, 1.5A) |
| Velocidad sin carga | 59rpm (12V) |
| Ángulo de giro | 0° a 300° |
| Temperatura de funcionamiento | -5°C ~ +70°C |
| Señal de comando | Paquete digital |
| Tipo de protocolo | Half Duplex Asynchronous Serial Communication (8 bits, 1 stop, sin paridad) |
| Conexión | TTL Level Multi Drop (conector tipo Daisy Chain) |
| Id's disponibles | 254 (0-253) |
| Retroalimentación | Posición, Temperatura, Carga, Voltaje de alimentación, etc. |
| Material | Plástico de ingeniería |

(Robotis, 2010)

c. Dynamixel XL-320

Figura 46. Servomotor Dynamixel XL-320



(Trossen Robotics, 2017)

Cuadro 11. Especificaciones técnicas del servomotor Dynamixel XL-320

| ESPECIFICACIÓN | VALOR |
|-------------------------------|---|
| Dimensiones | 24mm *36mm * 27mm |
| Peso | 16.7g |
| Voltaje de alimentación | 6 ~ 8.4V (Recomendado: 7.4V) |
| Velocidad de comunicación | 7343bps – 1 Mbps |
| Resolución | 0.29° |
| Relación de reductor | 238:1 |
| Torque | 0.39 N.m (7.4V, 1.1A) |
| Velocidad sin carga | 114 rpm (7.4V, 0.18A) |
| Ángulo de giro | 0° a 300° |
| Temperatura de funcionamiento | -5°C ~ +70°C |
| Señal de comando | Paquete digital |
| Tipo de protocolo | Half Duplex Asynchronous Serial Communication (8 bits, 1 stop, sin paridad) |
| Conexión | TTL Level Multi Drop (conector tipo Daisy Chain) |
| Id's disponibles | 253 (0-252) |
| Retroalimentación | Posición, Temperatura, Carga, Voltaje de alimentación, etc. |
| Material | Plástico de ingeniería |

(Robotis, 2010)

10. Métodos de manufactura de bajo costo

a. **Impresión 3D.** Esta es una tecnología de manufactura aditiva que utiliza como materia prima diferentes tipos de materiales, tales como: PVA, PLA, HIPS, ABS, fibra de carbono, fibra de madera, PC, nylon, etc. Uno de los más utilizados es el PLA (Ácido Poliláctico) debido a su facilidad de impresión (Castro, 2017). Entre sus características mecánicas se encuentran:

Cuadro 12. Características mecánicas del PLA

| Propiedades | Cantidad | Observaciones |
|---------------------------------|------------------------|---|
| Densidad | 1.27 g/cm ³ | |
| Dureza Shore | Grado 63.5 | Dureza de plásticos |
| Módulo de elasticidad | 3.53 GPa | Permite utilizarlo en aplicaciones estructurales a pequeña escala. Esto porque es menos elástico y menos propenso a sufrir deformaciones debido a cargas. |
| Deformación a la rotura | 6% | |
| Módulo de flexión | 4 GPa | |
| Resistencia a la flexión | 80 MPa | |
| Módulo de corte | 2.4 GPa | |
| Relación fuerza-peso | 40 kN-m/kg | |
| Resistencia a la tracción (UTS) | 50 MPa | |
| Resistencia a la fluencia | 15.5 MPa | A 22.7°C y con la norma de evaluación ISO 527-2 (Prospector, 2017) |

(HXX, 2017)

b. Corte láser. Las máquinas de corte láser y grabado CO2 son capaces de trabajar con varios materiales, como, por ejemplo: mármol, cerámica, plástico (sin PVC), madera, acrílico, piel, caucho, vidrio, etc. (Pérez, 2017). Uno de los materiales que también se puede utilizar es el MDF (Fibra de Densidad Media), teniendo las siguientes propiedades mecánicas:

Cuadro 13. Propiedades mecánicas del MDF

| Propiedades | Cantidad |
|---------------------------|-----------------------------|
| Densidad | 730 a 800 kg/m ³ |
| Resistencia a la flexión | 40 N/mm ² |
| Resistencia a la tracción | 1 N/mm ² |
| Módulo de elasticidad | 2500 N/mm ² |

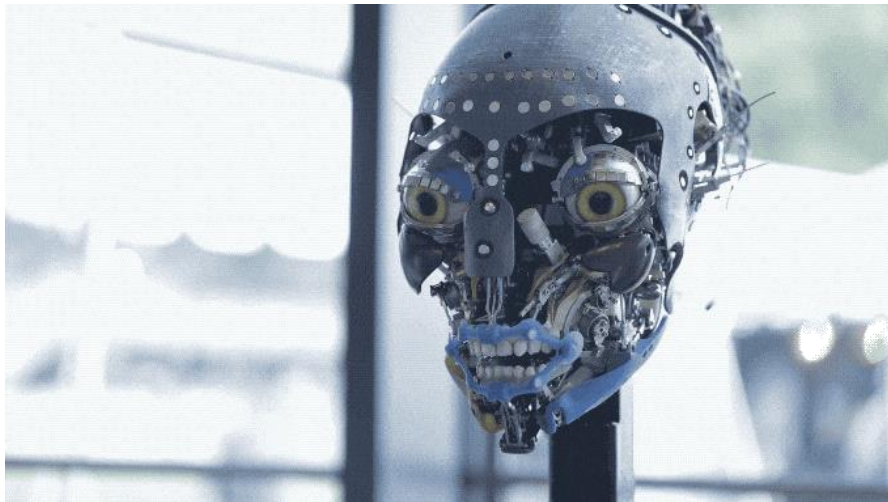
(Masisa, 2017)

V. ANTECEDENTES

En parques de diversiones, comúnmente, se encuentran los animatrónicos, los cuales dan la bienvenida a los visitantes del parque o emulan un personaje de tal forma que el usuario pueda entretenerse. El campo de aplicación de un animatrónico es amplio de tal manera que se encuentra en lugares de eventos tales como conciertos o lugares festivos, siempre y cuando el entorno esté basado en una cantidad masiva de personas. Los animatrónicos son los medios de entretenimiento que trae los personajes ficticios ilustrados en una hoja o diseñado digitalmente al mundo real. Siendo una de las formas más intuitivas de acercar la ficción a la realidad ante la vista de las personas, el costo para realizar esto es considerable.

Existen algunas industrias, compañías u organizaciones que desarrollan propios animatrónicos en los parques de diversión representativamente encuentra el Animatrónico más avanzado del mundo que es el Navi de la película “Avatar”, desarrollado por Disney. Garner Holt Productions, Inc., es la compañía enfocada en manufactura de animatrónicos más grande del mundo y posee variedad de opciones de personajes ficticias o animales, aunque el rango de costo es sumamente amplio y elevado al mismo tiempo.

Figura. 47. Estructura de cabeza de animatrónico “Navi” de Disney



Actualmente no existen muchos diseños abiertos de los animatrónicos, el cual se debe a su nivel de complejidad que implica en el proceso de desarrollo. Entonces debido a la complejidad del nivel de desarrollo de esto, hace que los desarrolladores patenten o guarden las ideas implementadas en el sistema del robot con el fin de proteger los derechos del diseño o tener como una técnica exclusiva de parte de estos desarrolladores. El alto de costo de animatrónicos generalmente es por esta razón, en que no existen muchas compañías u organizaciones quienes desarrollan los animatrónicos presentes en la comunidad pública. Además, la tecnología que involucra en parte de desarrollo hace elevar el costo de estos robots.

Adicionalmente, mayoría de los animatrónicos que existen, a excepción de Navi de Disney, actúan o emulan en rutinas predeterminadas. Es decir, el animatrónico realiza acciones y reproduce el movimiento característico de los personajes ficticios o imitados, pero no tiene la capacidad de interactuar.

Esta es la categoría de los animatrónicos más avanzados desde el punto de vista del procesamiento de información. Esto porque casi no requieren intervención de un usuario. Son capaces de tomar sus propias decisiones a partir de percepciones, planificaciones, actuaciones y modelaje del entorno para alcanzar su objetivo. A diferencia de los robots tele operados, estos sí pueden estar por sí solos en áreas poco estructuradas (Baturone, 2001).

Actualmente lo que está en auge es el desarrollo de inteligencia artificial. La cual se basa en modelos geométricos del entorno para aplicar métodos simbólicos de tratamiento de la información. Sin embargo, por las tareas que pueden realizar, se necesita mucha capacidad de procesamiento para que interprete la incertidumbre del entorno con el fin de obtener resultados en tiempo real. Una forma de resolver este problema es aumentar la cantidad de información que se tiene del entorno, pero siempre va a estar limitada por el sistema sensorial y la arquitectura del sistema de control y de información. A continuación, algunos ejemplos de estos robots autónomos (Lazalde, 2017):

- Pepper de SoftBank Robotics
- Nexi del MIT
- ASIMO de Honda
- BigDog y Atlas de Boston Dynamics
- X-47B de Northrop Grumman

Por lo tanto, la parte de procesamientos de audio y video que enfoca en desarrollo de sistema de reconocimiento de voz y de rostro ofrezca la habilidad de poder comunicar e interactuar con las personas con bajo costo de desarrollo, con alto nivel de rendimiento en precisión para comunicación fluida con el sujeto.

A. De funcionamiento repetitivo

Estas máquinas casi no toman información del entorno, sino que solamente cumplen tareas que no varían y que son predecibles. Por este motivo sus aplicaciones se ven principalmente en las cadenas de producción industrial. Y gracias a que el trabajo es repetitivo y se está hablando de una máquina, los robots de este tipo suelen ser muy precisos, rápidos y productivos ya que no se cansan (Baturone, 2001). A continuación, algunos ejemplos de robots de funcionamiento repetitivo (Knight, 2017):

- Baxter de Rethink Robotics
- Famulus de KUKA Roboter GmbH
- IRB 120 de ABB
- Cobra Scara de Adept

B. Teleoperados

Aquí todos los movimientos son realizados por seres humanos quienes evitan la tarea de dotar de planificación, manipulación compleja y percepción de los alrededores a los robots. Por tal motivo, el usuario que se identifica como operador está “conectado” al robot por medio de un lazo cerrado de control. Ya para las versiones más avanzadas, es posible obtener feedback del robot si éste tiene sensores que perciban alteraciones en su entorno, ya sea de sonido, fuerza, distancia, movimiento, etc. También se utilizan controladores automáticos para el movimiento de manos y brazos antropomórficos pudiendo, el operador, mover un objeto a escala del manipulador para controlarlo (Baturone, 2001).

Entre sus aplicaciones están los trabajos difíciles de automatizar, tareas en entornos no estructurados como un lugar de mantenimiento de torres eléctricas o en el área constructora, acceso remoto en áreas peligrosas o contaminadas, etc.

Sin embargo, el problema que surge con este tipo de animatrónicos es que la computadora y el hombre tienen diferentes capacidades de procesamiento numérico y de precisión lo que afecta la coordinación y el acoplamiento entre ambos. El retraso de información, para ciertas aplicaciones, representa un requisito indispensable por lo que un tiempo muy prolongado de retraso en la transmisión afecta el sistema de control y define la interfaz entre la persona y la computadora como uno de los puntos más críticos. Entonces, tomando en cuenta dicho problema, actualmente se reduce la acción operador permitiendo solamente intervenir al momento de tomar decisiones a partir de la experiencia, habilidad e información de los sensores. A continuación, algunos ejemplos de estos robots (Knight, 2017):

- Sojourner de la NASA
- Rocky 7 de la NASA
- iPEK Rovver 125 de Asthead Technology

VI. METODOLOGÍA

A. Procesamiento de audio y video

Con el fin de desarrollo organizado manteniendo efectividad en cada sistema de reconocimiento, el proceso de desarrollo consistió en tres bloques generales. Cada bloque provee una serie de procesos del desarrollo, con el uso de modelo de prueba o el método de estudio, para validación sólida de la selección y aplicación que forma parte del sistema de reconocimiento.

1. **Entorno de programación.** Los desarrollos de los sistemas de reconocimientos, de rostro y de voz, realizan a partir de la selección del entorno de programación. Se seleccionaron Python y de Arduino (C++) para desarrollo principal de la estructura del programa. Cada entorno por la compatibilidad extensa facilita el trabajo de entablar la comunicación con motivos de realimentación de datos necesarios a procesar. La plataforma de programación Matlab forma parte adicional del entorno de programación para realizar pruebas de algoritmos que requieren potencia en enfoque de operaciones matemáticas.

Entonces Python es implementado para realizar operaciones en base de las computadoras o máquinas principales para operaciones numéricas y de segmentación para identificación de códigos para ejecución de eventos sea respuesta por audio o por voz, o de combinación. El protocolo serial es realizada para establecer comunicación con la plataforma de Arduino. El algoritmo de reconocimiento de voz y de rostro, al igual que la comunicación serial, son desarrollados a partir de la librería con funciones dedicadas a realizar operaciones necesarias.

Inicialmente se lleva a cabo el desarrollo del sistema de reconocimiento de voz y de rostro por separado. Cada algoritmo de sistema de reconocimiento conforme el desarrollo, crean versiones actualizadas al cumplir con operaciones requeridas como parte del funcionamiento esencial. Entonces, por ejemplo, al lograr distinguir el rostro del usuario; se realiza una actualización a la versión con una clonación en un local, y pasa a realizar el algoritmo tal que pueda segmentar los datos de los parámetros que distingue el rostro para luego poder utilizar para actuar los servomotores. La clonación de la versión del programa se realizó para prevenir pérdida del código funcional y poder hacer uso de esto para recuperar el código en dado caso de que pierda el mismo código a la hora de estar realizando modificación encima de la versión actualizada.

La plataforma de Arduino enfocó en comunicación con los actuadores principalmente. Según los puertos de protocolo serial, dedicó cada puerto serial por una cadena específica de actuadores y sensores. Unas cadenas de actuadores basaron únicamente de los servomotores AX12A o de XL320 por cada puerto serial. Los servomotores son configurados previo a la implementación con la plataforma de Arduino y Python. La

configuración basó en habilitación de funcionalidades como el torque, notificación por medio de leds, y los identificadores respectivos según sea la ubicación del servomotor.

2. Sistema de reconocimiento de rostro. Para el desarrollo de sistema de reconocimiento de rostro, se realizó la investigación preliminar con el fin de identificar la mejor selección del algoritmo categorizados por diferentes objetivos. El criterio de la selección del algoritmo basó en reconocer un único rostro del usuario dentro de la región de detección permitido siendo flexible de determinar el usuario con intención de comunicar con el animatrónico. De los componentes adicionales para desarrollo del algoritmo, el algoritmo de extensión se seleccionó al OpenCV. Con las funciones de OpenCV que proveen son suficientes para detección del rostro del usuario.

En términos de iteraciones del desarrollo de sistema de reconocimiento de rostro basa en diferentes entornos de programación. El entorno de programación, aparte de Python, se utilizó el Matlab también con el fin de observar el rendimiento y la flexibilidad que puede proveer a diferencia de Python. Entonces, en caso de desarrollo con Matlab no llegó a implementar la comunicación serial y quedó en la fase de reconocimiento del rostro.

Se realizó la configuración de llamadas a las funciones de OpenCV para establecer la interfaz con el dispositivo observador. Observador en este contexto refiere a la cámara, cuál es el ojo del animatrónico. La configuración de la imagen de la cámara únicamente controlada es la resolución, y demás parámetros del sensor o la calidad de la imagen de la cámara dejó para que la cámara pueda adaptarse automáticamente según sea el ambiente de operación. La resolución se ajustó de tal manera que adapte de forma nativa la resolución al cuál la cámara puede operar sin algún problema por insuficiencia de recursos de potencia. Con todo esto entonces establece la configuración básica para poder desarrollar el algoritmo para distinguir el rostro del usuario dado la región de detección permitida.

Previo a avanzar con el desarrollo del algoritmo de reconocimiento de rostro, la configuración de la comunicación serial es definido. La comunicación serial utilizó la librería predeterminado de Python para establecer la conexión entre Arduino. Esto es por la necesidad de enviar los datos de los ejes extraídos de la operación de reconocimiento de rostro y convertirlo luego en plataforma de Arduino para movilización de los servomotores.

Para distinguir el rostro del usuario, se hizo el uso de “eXtensible Markup Language” o Lenguaje de Marcas Extensible en el cual posee los parámetros característicos que define la geometría del rostro de la persona. Este lenguaje tiene el formato de extensión “.xml” y es llamado antes de establecer la conexión con la cámara para dar definiciones predeterminadas antes de que pueda realizar operaciones. Luego, al tener los parámetros característicos de rostro y conexión establecido con la cámara, desarrollo el algoritmo para

operación de segmentación de los valores de los datos del rostro reconocido. Alrededor del rostro se dibuja un cuadrado que muestra que reconoció el rostro y este mismo cuadrado es utilizado para obtener los valores de las coordenadas sobre la imagen desplegada.

El algoritmo desarrolló en base a compensaciones fijas. El sistema de control fue considerado como parte del desarrollo para mejorar las respuestas de los datos decodificados a partir del cuadro de reconocimiento. Al contrario, desarrolló inicialmente por medio de las compensaciones fijas en donde al tener los datos del cuadro de reconocimiento segmentados en términos de los ejes de 2 dimensiones, se suma las compensaciones respectivas sobre los ejes. Los ejes son el plano en el cual el servomotor ofrece el grado de libertad para moverse. Las compensaciones fueron ajustadas de acuerdo a las pruebas con partes mecánicas de los ojos. Las primeras pruebas se realizaron con base en la cámara en una posición fija que realimenta los resultados de la ubicación del rostro del usuario. Entonces, los valores de la compensación fueron mejorando con prueba de poder notar que los ojos del usuario y de los ojos de animatrónico coinciden en la dirección de vista, dando sensación de que ambos están viendo.

El sistema de control es estudiado por la posición de la cámara. La cámara es posicionada dentro de los ojos en el cual la posición de la cámara varía según el movimiento de los ojos del animatrónico. Por lo cual, el algoritmo de la forma como realiza la implementación de las compensaciones de los ejes es utilizado para centrar con el dato del centroide del cuadro de reconocimiento. El tema de compensaciones lleva encadenado la región de detección.

La región de detección es definida entonces por medio del tamaño del cuadro de reconocimiento. Es decir, el espacio de detección es libre el cuál varía según la longitud perspectiva. Entonces, el rostro puede ser mucho más pequeño en ciertas regiones, a diferencia de regiones que dibuja el cuadro de reconocimiento más grande, resultando en pérdida de precisión de posicionamiento de los ojos y no logra involucrar al usuario a poder tener la sensación de estar interactuando de forma fluida. Por lo cual, la región de detección se limitó en un espacio específico con la relación del tamaño del cuadro de detección. Al tener cierto tamaño de cuadro de detección predice la ubicación espacial del usuario y ejecuta el proceso de obtener los datos de los ejes requeridos para movilizar los actuadores de los ojos en respectivos ejes para posicionar en centroide del cuadro de detección.

Al lograr definir la región de detección para reconocimiento del rostro y obtención de los datos respectivos de los ejes a convertirlos en valores angulares respectivos de los servomotores, se refina la configuración serial de tal forma que pueda enviar los datos de los ejes respectivos hacia la plataforma de Arduino. En la plataforma de Arduino, realiza la conversión de estos valores de los ejes extraídos a partir del cuadro de detección en valores angulares respectivos de los servomotores que representan el desplazamiento a realizar.

3. Sistema de reconocimiento de voz. El sistema de reconocimiento de voz, al igual que el desarrollo del sistema de reconocimiento de rostro, ejecutó con la selección del mejor algoritmo o librería con funciones para reconocimiento de voz. La investigación de librerías para funciones de reconocimiento de voz resultó en Snowboy, Jasper, PocketSphinx, IBM Watson API, y Google Speech API. Cada uno basó en diferentes plataformas de prueba, es decir de sistema operativo del dispositivo. El dispositivo que tomó en consideración para ejecución de las pruebas de cada API fue el Raspberry Pi con distribuciones de Linux y Computadora en general con sistema operativo Windows.

Entonces, el desarrollo de algoritmo de reconocimiento de voz enfocó con computadora en general con Windows con la plataforma de Python. Teniendo en cuenta la compatibilidad de librería o API, el Google Speech API es seleccionado como parte del desarrollo del algoritmo. El enfoque de realizar pruebas con computadora basó por lo común y la facilidad del acceso a desarrollar el código.

El código es desarrollado a partir de la inicialización de los parámetros principales de la propiedad de micrófono, la forma de acceder y mantener el archivo de audio parcialmente, y ajustes de reproducción. Los ajustes de reproducción basaron en configuraciones de tonalidad y la velocidad del audio digital cuando reproduce un archivo de audio o frase predeterminado. La conexión a la red es necesario para la operación del sistema y se estableció por medio de la función del Google Speech API. Al tener todos estos parámetros de configuración lista, realizó un listado de términos simples y compuestos para probar el nivel de precisión de detección.

Para validar la precisión de detección, se realizó la operación de comparación del término por medio de palabras claves previamente almacenadas. Es decir, realizó la comparación del audio de reconocido de la entrada con la variable predeterminada del término deseado ser reconocido correctamente en el medio digital. Al validar la precisión del reconocimiento de los comandos de audio, aumento la cantidad de palabras claves de detección para ampliar el uso de la función de reconocimiento de voz. Los comandos basaron en idea de activación de gestos del animatrónico, respuesta en audio, o combinación de las dos formas de comunicación previos. Estos son segmentados según sea los valores de identificación que representa el comando dicho por el usuario, y esto incluyó en vías de comunicación serial con estos números de identificación.

El modelo de prueba de sistema de reconocimiento de voz es diseñado de tal manera que sea fácil de distinguir el rendimiento y la probabilidad de fallos de detección con el uso del sistema. El diseño del modelo de prueba basó en diferentes niveles de complejidad. Los niveles de complejidad baso en cantidad de 30 iteraciones fijas a lo largo de cambio de complejidades con diferentes combinaciones de términos.

La mayor parte de las pruebas se dedicó a ejecutar con la computadora en general debido a la disponibilidad de librerías compatibilizadas con el sistema operativo Windows. La consideración de Raspberry Pi basó en criterio del alejarse de la computadora con base Windows que ocupa mayor espacio. Al contrario, Raspberry Pi que iba ser implementado consiste en un procesador de un único núcleo, el cuál limitó la libertad poder usar variedad distribuciones de Linux y el nivel de la compatibilidad baja con las librerías o APIs investigados. Al referir que el nivel de la compatibilidad es bajo es decir que no existen versiones disponibles para poder ejecutar en la distribución de Linux, el cuál al mismo tiempo la distribución de Linux es limitado para versión de Raspberry Pi de un solo núcleo de procesador. Además, hay necesidades de realizar modificaciones sobre la librería o el API para poder ejecutar con la limitación del rendimiento que proporciona el Raspberry Pi.

B. Diseño e implementación mecánica de brazo y garras de DANI

1. Iteraciones de los diseños del brazo. El primer diseño del brazo robótico se basó en el área del codo, específicamente parte del antebrazo y el brazo. Por lo que lo primero que se realizó fue un análisis dimensional y dinámico del diseño del artista para conocer cuál era la longitud y el área transversal que el brazo debía ocupar además de cuántos grados de libertad iba a tener. Para los mismos, se decidió que 4 grados de libertad eran suficientes para imitar, a grandes rasgos, el movimiento del brazo humano con el modelo estético realizado. Luego, se descargaron de una comunidad virtual de archivos CAD los servomotores y sus accesorios para aprovechar la modularidad que estos presentan en su encapsulado y así diseñar piezas que se ensamblen a su geometría. Los discos que se incluyeron cumplían la funcionalidad de brindar una estructura robusta y desarmable en un volumen donde no habría partes movibles. Esto se logró utilizando discos de MDF maquinados en una cortadora láser y con un eje de acero inoxidable de 3/16” ensamblados con tuercas del mismo diámetro. La geometría de los discos se basó en definir un diámetro equivalente al de la fibra de vidrio del brazo y varios patrones de corte para disminuir el peso. Luego está el accesorio del servomotor XL-320 cuyo objetivo era transmitir la potencia de dicho motor y brindar un soporte a toda la estructura inferior del brazo. En este punto específicamente fue importante identificar que, debido a que es una articulación de tipo bisagra, se tenía que ocupar poca área transversal para brindar la mayor flexión y extensión del brazo. Esto explica el porqué de la geometría de las piezas “Codo-Muñeca”. Haciendo un barrido visual en descenso se puede apreciar que cerca de la articulación (eje del motor) el área transversal ocupada es muy pequeña mientras que, al alejarse del motor, el diámetro ocupado aumenta hasta llegar al valor que tenía antes (el de los discos). Sin embargo, debido a la geometría de estas piezas, se terminó con dos piezas en voladizo.

Para el diseño 2 se terminó de diseñar el prototipo del brazo. Retomando la idea de un recorrido descendente por el brazo, la primera pieza que se encuentra es un accesorio para los servomotores AX-12A cuya geometría permite los movimientos de abducción y aducción del animatrónico. Se puede observar que

usando este accesorio lo que giraría sería el motor completo, lo que estructuralmente hablando está bien porque se utilizaría una pieza que está diseñada para ser un soporte en una articulación. Sin embargo, esto representa que el servomotor no solo debería mover el brazo sino su propio peso también. Cada servomotor AX12-A incluye un accesorio F333 que es el que sirve para ensamblar un motor Dynamixel a otro. Estos accesorios permitieron, precisamente, agregar otro servomotor lo que a su vez representa un grado de libertad más y en este caso se habilitó la extensión y flexión del brazo completo. Luego se encuentra el accesorio F2222, también incluido con los motores, que se utilizó para ensamblar los actuadores a la estructura del brazo. Sin embargo, en este punto se decidió mejorar la eficiencia del proceso de ensamble y desensamble. Por lo que se diseñó un acople hembra y un acople macho con dientes de forma que el ensamble fuera del tipo “Twist-and-Lock”, lo que permitió dividir el brazo en 3 partes principales: hombro, antebrazo, brazo (que de ahora en adelante se llamará “parte inferior del brazo”). Si se necesitaba cambiar una pieza en el antebrazo, por ejemplo, se podía retirar el mismo sin necesidad de utilizar herramientas. Esto permitiría que una vez con la piel puesta, se pudiera desensamblar manualmente. En la parte inferior del brazo se cambió las piezas Codo-Muñeca para adaptarlas a una geometría que siguiera cumpliendo la funcionalidad de la versión previa, que fuera capaz de darle soporte a otro servomotor XL-320 y que resolviera el problema del voladizo. Entonces se diseñó la pieza “Separador”, que está atornillada al motor XL-320 para brindarle soporte estructural tanto al mismo como a las piezas Codo-Muñeca, eliminando así el voladizo. Cabe mencionar que el nuevo motor les brindaría movimiento a los dedos del robot por medio de un mecanismo de hilos. Se decidió utilizar dicho mecanismo para ahorrar peso y mover los dedos con un solo actuador. La pieza “Separador 2” finaliza la estructura del brazo, eliminando completamente el voladizo mencionado y organizando los hilos de los dedos.

En la tercera iteración se agregó una parte cilíndrica al accesorio del hombro con el objetivo de facilitar su ensamble a la estructura del cuerpo, pero esto significaría tener que modificar la pieza original digitalmente e imprimirla por completo. En esta etapa se colocaron todos los elementos de sujeción para empezar con las simulaciones. También se puede ver que se diseñaron las piezas que irían dentro de las garras, estas solo darían una estructura y un volumen a las mismas (ya que en esta parte no hay fibra de vidrio) por lo que se imprimieron con muy poca densidad de PLA.

En el diseño 4 se decidió utilizar en el hombro el accesorio F2222 que viene incluido con los servomotores grandes debido a que también incluye 8 tornillos, que, si bien son pequeños, la fuerza se distribuye equivalentemente entre ellos y logran ser capaces de soportar el peso del brazo completo. Esto lo confirmaría el análisis de esfuerzos. Los acoples macho y hembra se actualizaron y en lugar de utilizar dientes se utilizó un mecanismo tipo conector eléctrico porque los dientes no eran adecuados para soportar la carga que se les aplicaba y por lo mismo, representaban el punto más crítico de todo el brazo cuando este se encontraba en voladizo. En la garra se cambiaron las piezas anteriores porque no se logró un método de sujeción adecuado con la piel que fuera no permanente y, además, que no deteriorara la misma. Entonces se decidió colocar tubos flexibles para cableado. Estos se modificaron para que tuvieran un pivote y para que

los hilos se pudieran sostener. Al final ocupaban el volumen necesario, eran livianos y no necesitaban elementos de sujeción.

Para el diseño final, se decidió retirar completamente la parte del antebrazo con el objetivo de disminuir el peso ya que haciendo pruebas físicamente, los servomotores no eran capaces de levantar 90° el brazo. Además, como dichos actuadores tienen auto-protección, estos se apagaban cuando percibían una sobrecarga o una temperatura elevada. El acople macho y hembra se cambió por un acople estándar de tubería roscada de 1" ya que los acoples tipo conector eléctrico presentaron muchos problemas en el proceso de la impresión y seguían siendo los principales puntos críticos del brazo en voladizo. Con una unión roscada, la resistencia a la flexión aumentó considerablemente (ver análisis de esfuerzos). El accesorio del XL-320 superior se actualizó agregándole 5 pines al diseño original y cambiando su geometría para ser colocado a presión porque el disco del actuador no está diseñado para utilizar tornillos. Con estas últimas modificaciones se obtuvo la versión final del brazo, siendo de bajo costo porque lo único considerable son los actuadores y de fácil maquinado habiendo utilizado impresión 3D donde el resto de piezas son estándar o son accesorios incluidos con los actuadores por el fabricante.

2. Análisis de esfuerzos. Para llevar a cabo el análisis de esfuerzos de la forma correcta, se siguieron los siguientes pasos:

1. Análisis de las posiciones críticas que tendría el brazo robótico.
2. Modelado del brazo como una geometría simple.
3. Modelado de los pesos de todas las piezas.
4. Obtención del diagrama de corte y momento.
5. Identificación de puntos críticos.
6. Definir los elementos de esfuerzos y secciones transversales.
7. Simulación de análisis de estrés.
8. Cálculo de esfuerzos y factores de seguridad.

3. Análisis de tensión para los actuadores de la garra

1. Modelar un setup para tensar los hilos sin el servo.
2. Calcular la tensión necesaria en los hilos para actuar cada dedo.
3. Calcular el torque total necesario para abrir y cerrar la garra.

4. Análisis de parámetros de los actuadores principales

1. Obtener el límite mínimo y máximo del ángulo para el servo AX-12A superior.
2. Hacer un barrido de ángulos en el rango establecido anteriormente.
3. Obtener el torque y el ángulo que devuelve el AX-12A.

4. Encontrar un modelo teórico para el torque en función del ángulo.
5. Comparar torques y ángulos teóricos con torques y ángulos experimentales para calcular porcentaje de error.
6. Medir tiempo de desactivación del servo en cada ángulo, de ocurrir alguno.

C. Diseño e implementación mecánica de gestos faciales de DANI

La elaboración del diseño e implementación de los mecanismos dentro del módulo diseño mecánico de la cabeza del megaproyecto DANI involucro un ciclo de análisis, selección, diseño y manufactura, basado en un conjunto de herramientas de diseño CAD y conceptos del área de esfuerzos y selección de materiales. La metodología empleada se resume de la siguiente manera:

1. Identificación de los gestos a emular por medio de diversos mecanismos.
2. Medición del espacio de trabajo dentro de la cabeza del dragón para la segmentación de los mecanismos.
3. Selección de mecanismo de ojos, para su debido análisis de esfuerzos y selección de materiales para el mecanizado.
4. Manufactura de piezas para mecanismo de los ojos.
5. Selección y análisis de mecanismo para la boca.
6. Manufactura de las piezas para el armado de la boca.
7. Análisis teórico de esfuerzos de ciertos puntos críticos como pernos de sujeción.
8. Selección y diseño de mecanismo para cejas.
9. Identificación de mecanismo para orejas.
10. Análisis final de los diversos mecanismos y validación de los mismos.

En los capítulos posteriores se detallan las particularidades en el diseño de cada experimento al igual que los resultados obtenidos para su validación.

Figura 48. Cabeza de “DANI”



VII. RESULTADOS

A. Componente de procesamiento de audio y video

1. Trade Study para selección de sistemas de reconocimiento

a. Trade Study del sistema de reconocimiento de voz. El Cuadro 14 presenta el resultado de la selección de Google Speech API con la implementación del método de Pugh de trade study, por medio de criterios de selección basado en la necesidad de desarrollo de sistema de reconocimiento de voz.

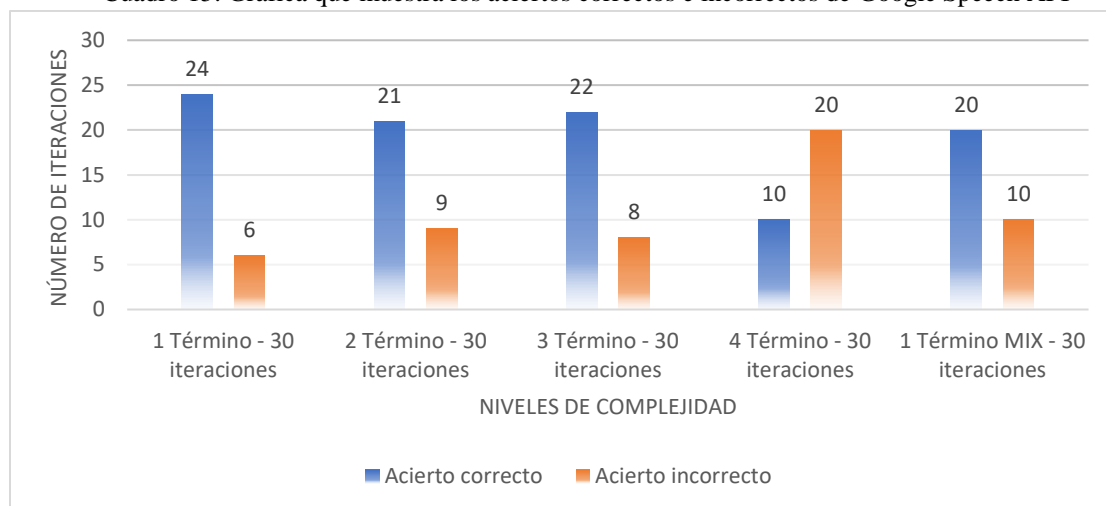
Cuadro 14. Trade study realizado del sistema de reconocimiento de voz

| Criterio de selección | Opciones de librerías / API de Reconocimiento de voz | | | | |
|---|--|--------|--------------|----------------|-------------------|
| | Snowboy | Jasper | PocketSphinx | IBM Watson API | Google Speech API |
| Compatibilidad del entorno de programación | - | - | S | S | + |
| Facilidad de integración | - | - | S | S | + |
| Diversidad de funciones | S | S | S | S | S |
| Soporte / Comunidad | S | S | S | S | + |
| Funcionamiento sin conexión a la red | + | - | S | - | - |
| Amplio entorno de programación | + | S | - | S | + |
| Mejora autónoma (Machine Learning) | + | + | S | + | + |
| Compatibilidad de plataforma (Windows, Mac, Linux (Raspberry)) | S | S | S | S | + |
| Personalización | + | + | + | S | S |
| Suma de positivos | 4 | 2 | 1 | 1 | 6 |
| Suma de negativos | 2 | 3 | 1 | 1 | 1 |
| Suma de iguales | 3 | 4 | 7 | 7 | 2 |
| Selección de librería / API | | | | | CHECK |

2. Estadísticas

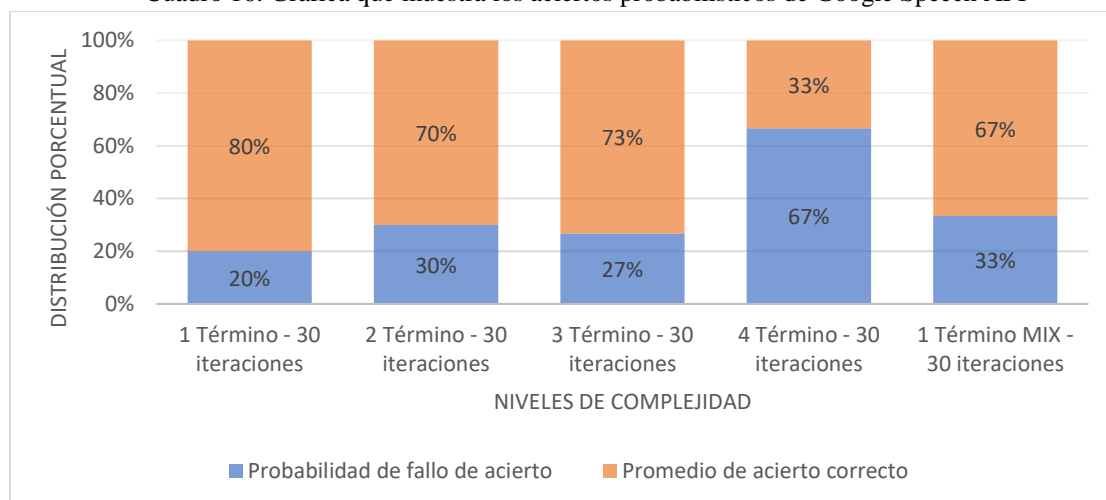
a. Estadística de Prueba de Aciertos del Google Speech API. El Cuadro 15 muestra estadística de aciertos correctos e incorrectos con la implementación de Google Speech API, con el modelo prueba previamente diseñado para validación del rendimiento del sistema.

Cuadro 15. Gráfica que muestra los aciertos correctos e incorrectos de Google Speech API



b. Estadísticas Probabilísticas de Aciertos del Google Speech API. El Cuadro 16 muestra la estadística probabilística de promedio de aciertos correctos y con probabilidad de fallo bajo el modelo de prueba con condiciones de sobrecarga.

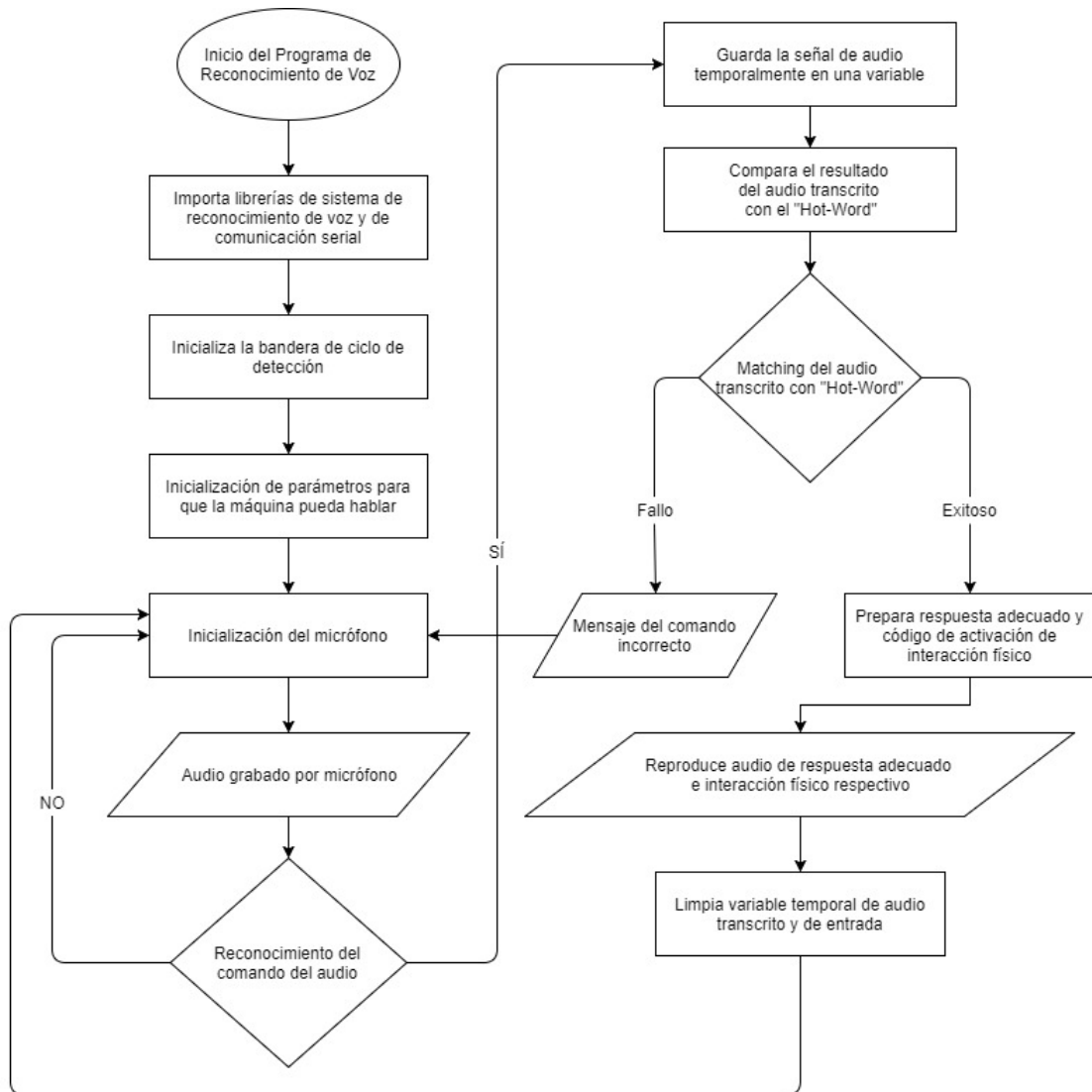
Cuadro 16. Gráfica que muestra los aciertos probabilísticos de Google Speech API



3. Diagramas de flujo

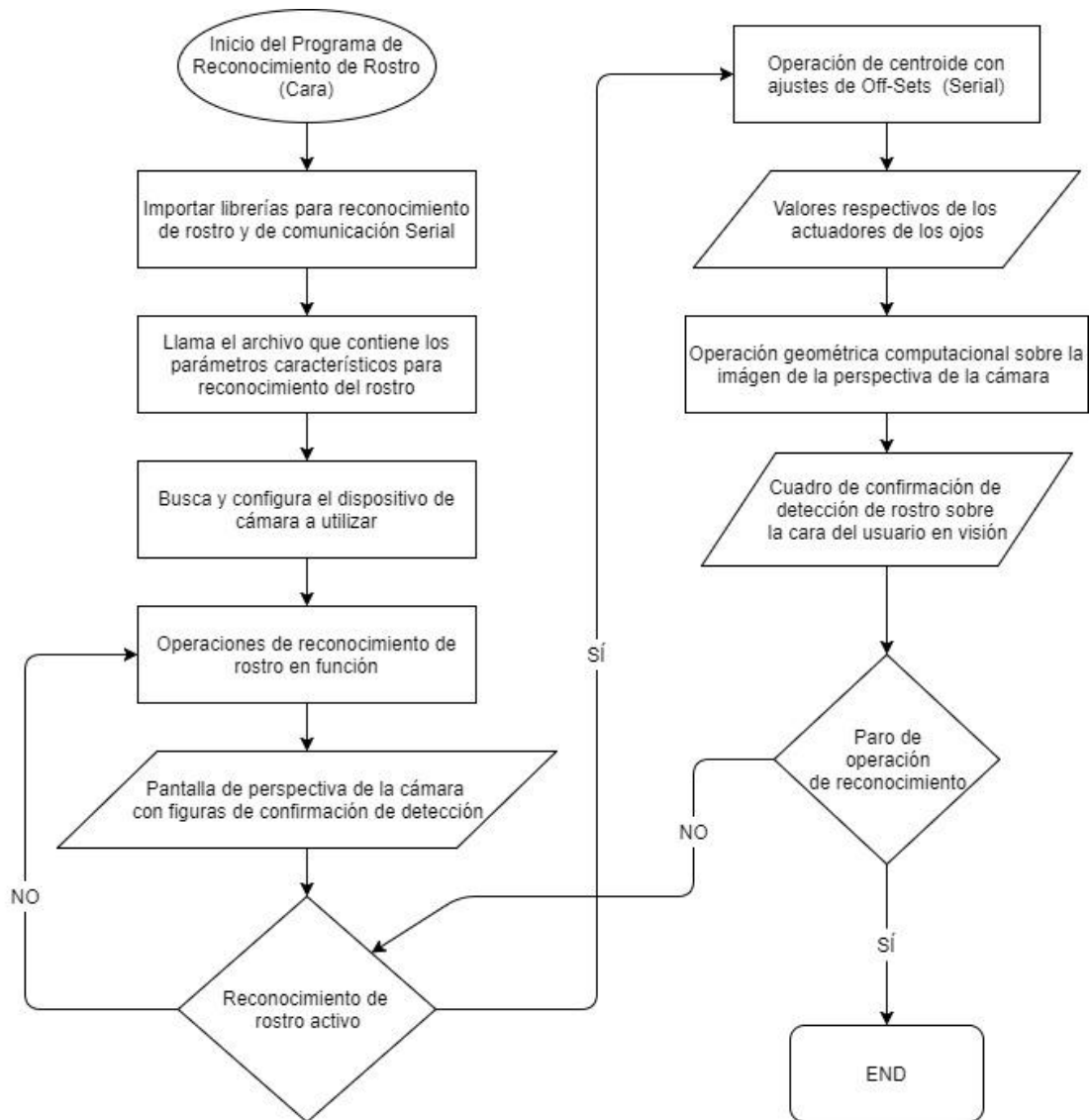
1. Diagrama de sistema de reconocimiento de voz. La Figura 49 describe el comportamiento o la operación del sistema de reconocimiento de voz para tener la capacidad de interactuar auditivamente con el humano mientras esta en operación, con la implementación de Google Speech API.

Figura 49. Diagrama de flujo del sistema de reconocimiento de voz por medio de Google Speech API



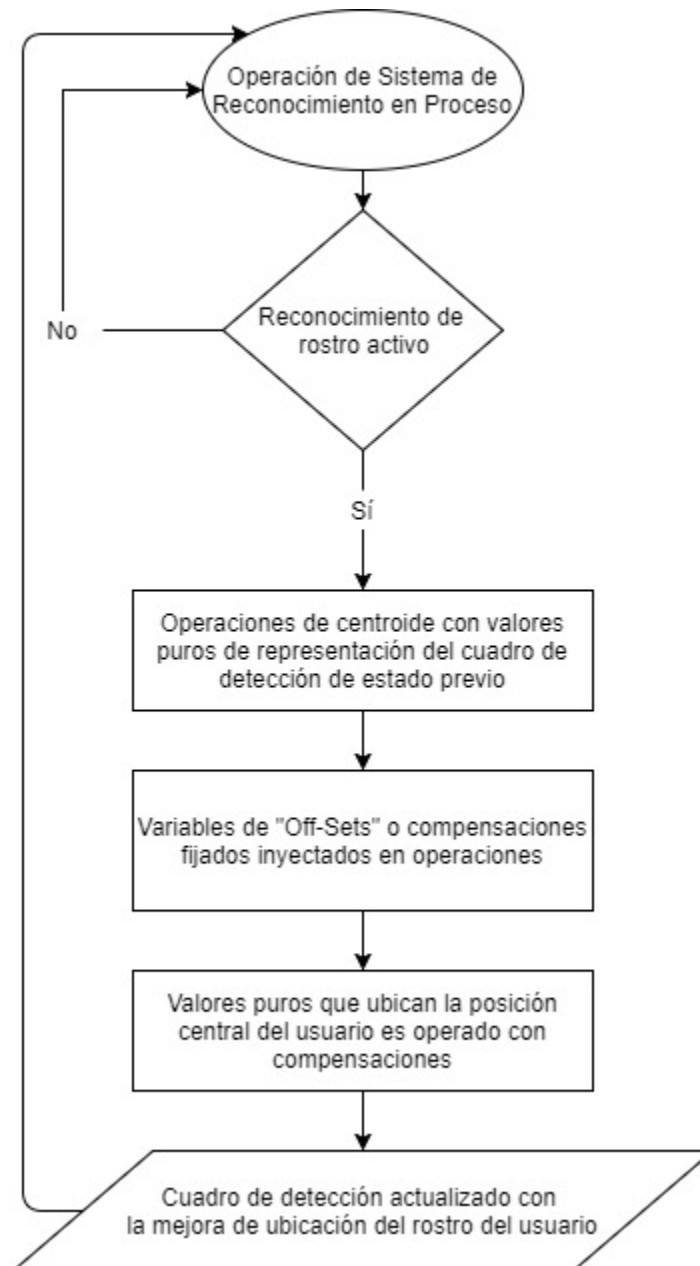
2. Diagrama de sistema de reconocimiento de rostro. La Figura 50 es el diagrama que describe el funcionamiento del sistema de reconocimiento de rostro con la implementación de OpenCV con el uso de algoritmo de compensaciones u “Off-sets”.

Figura 50. Diagrama de flujo del sistema de reconocimiento de rostro empleando OpenCV



3. Algoritmo de compensación del sistema de reconocimiento (Off-sets). La Figura 51 es la derivada del diagrama de flujo del sistema de reconocimiento de rostro, con el enfoque en la parte del algoritmo de compensaciones que ajusta el desplazamiento de los ojos del animatrónico.

Figura 51. Diagrama de flujo de algoritmo de compensaciones (Off-sets) del rostro



B. Pruebas de detección de rostro

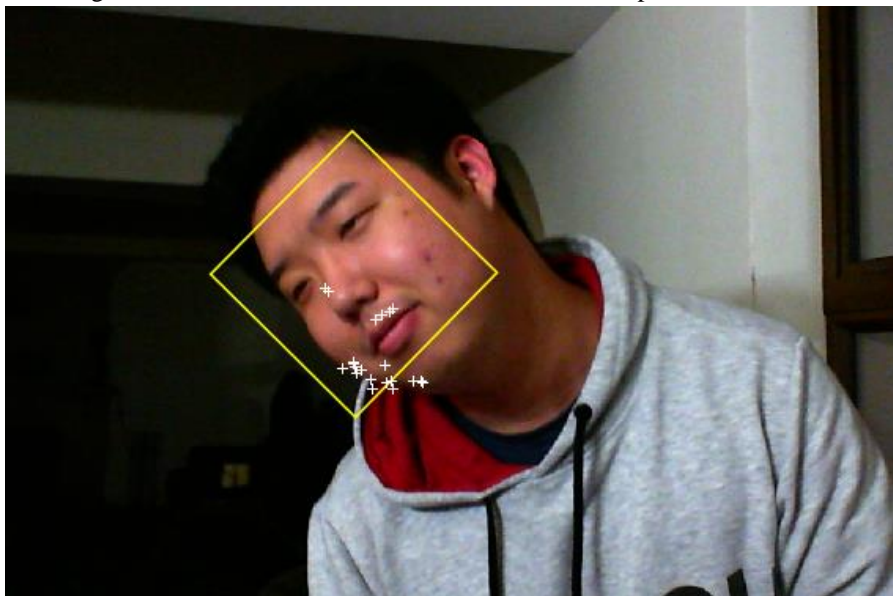
Las figuras 52 y 53 ilustran el funcionamiento del sistema de reconocimiento de rostro de perspectiva de los ojos de animatrónico, realizando la prueba desde la plataforma de Matlab.

Figura 52. Prueba de detección inicial con cuadro de detección recto



(Matlab – Microsoft LifeCam Cinema)

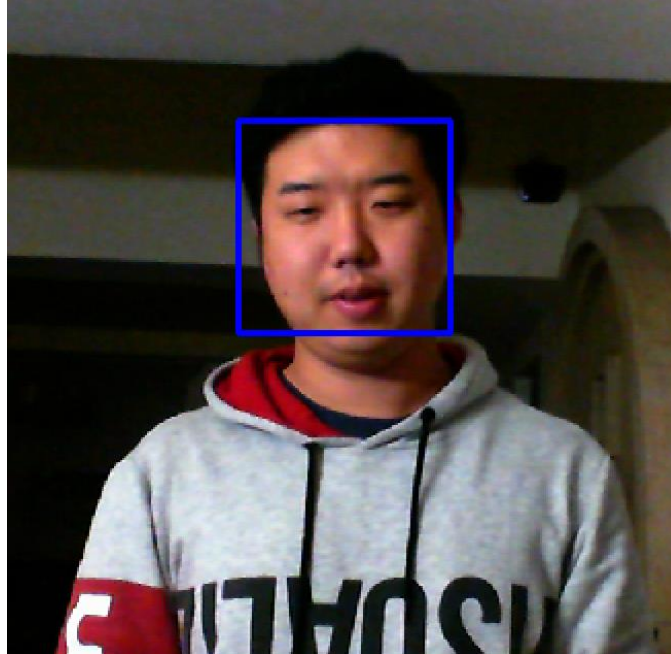
Figura 53. Prueba de detección flexible al cambio de posición del rostro



(Matlab – Microsoft LifeCam Cinema)

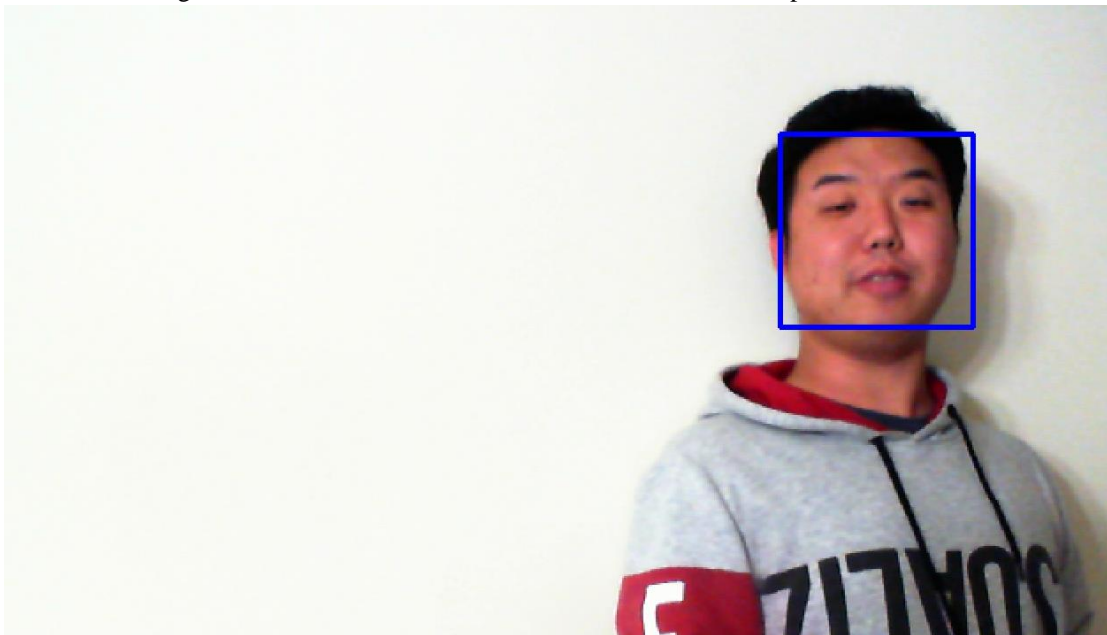
Las figuras 54 y 55 son el resultado de la implementación del algoritmo de reconocimiento de rostro en entorno de programación de Python desde la perspectiva de los ojos del animatrónico o la cámara. A diferencia del Matlab, en Python únicamente cambia el tamaño de cuadro de detección para obtener el centroide constante sin algún cambio brusco del valor debido a la deformación del cuadro de detección.

Figura 54. Prueba de detección inicial con cuadro de detección recto



(Python – Microsoft LifeCam Cinema)

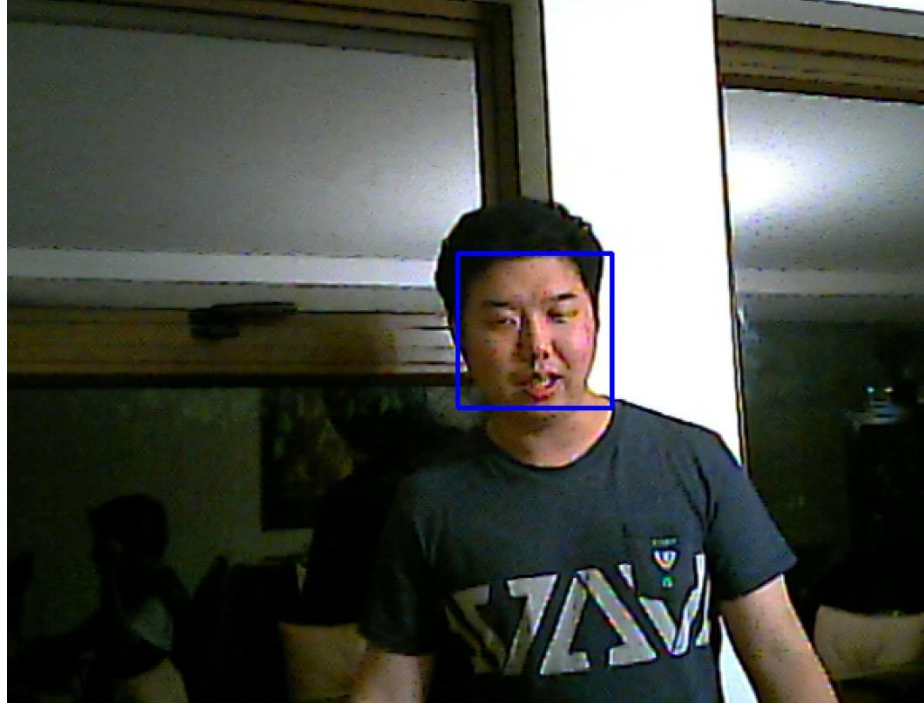
Figura 55. Prueba de detección con cuadro de detección en posición alterna



(Python – Microsoft LifeCam Cinema)

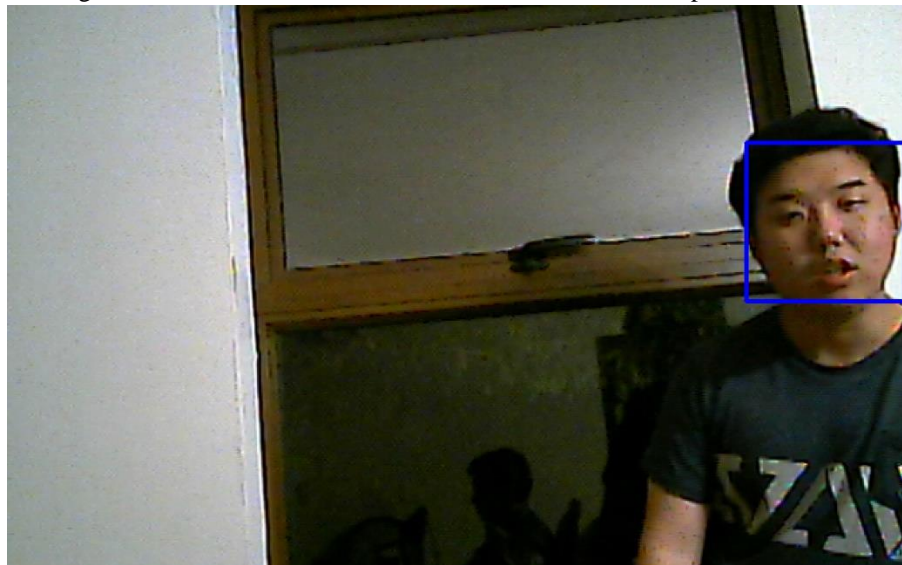
Las figuras 56 y 57 son resultados de utilizar el entorno de programación de Python y cámara OOSXXX Mini Security, el cual se implementa dentro del mecanismo de los ojos del animatrónico. A diferencia de resultados anteriores, esto muestra baja calidad de imágenes, pero el rendimiento del algoritmo se mantuvo constante.

Figura 56. Prueba de detección inicial con cuadro de detección recto



(Python – OOSXXX Mini Security Camera)

Figura 57. Prueba de detección con cuadro de detección en posición alterna



(Python – OOSXXX Mini Security Camera)

C. Resultado de brazos mecánicos de DANI

1. Iteraciones de los diseños del brazo. A partir de las consideraciones mencionadas en la metodología se obtuvieron los siguientes diseños del brazo mecánico:

Figura 58. Primeras 3 iteraciones del diseño del brazo mecánico

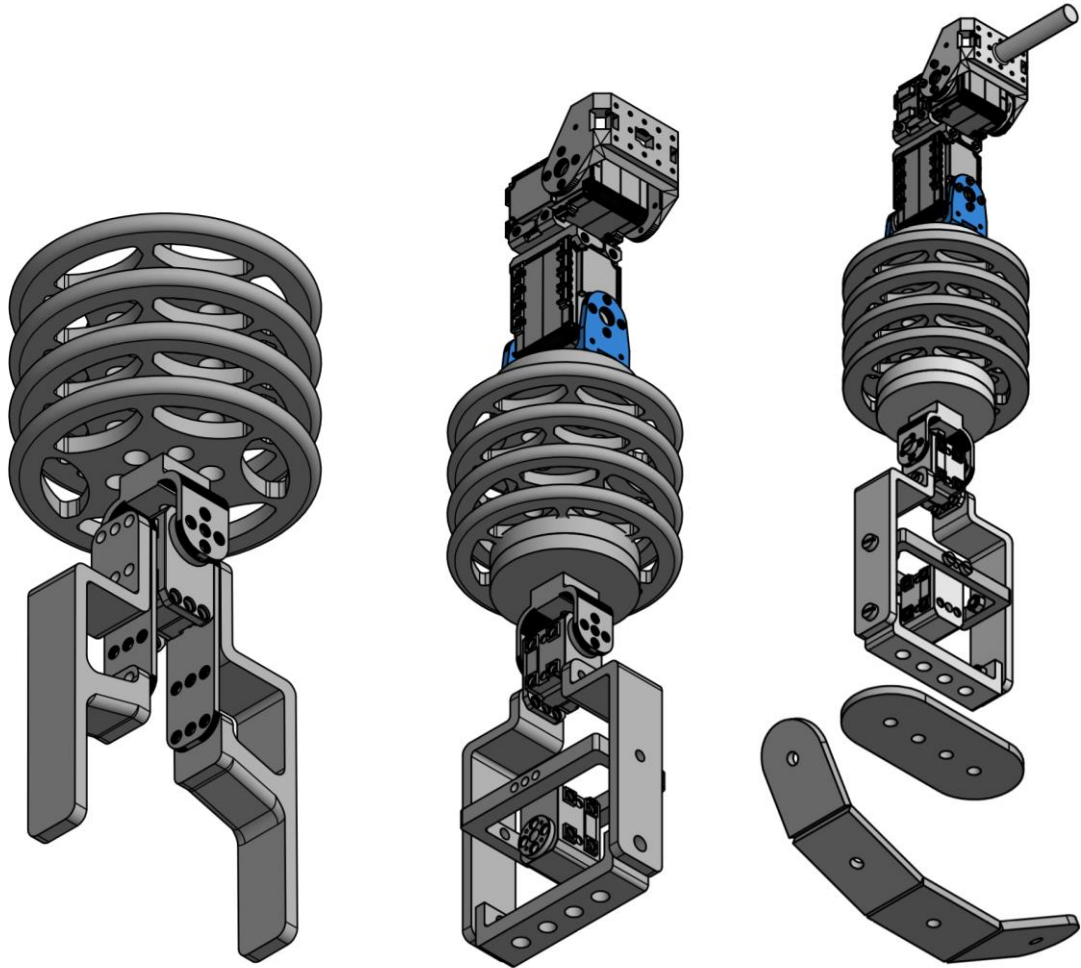
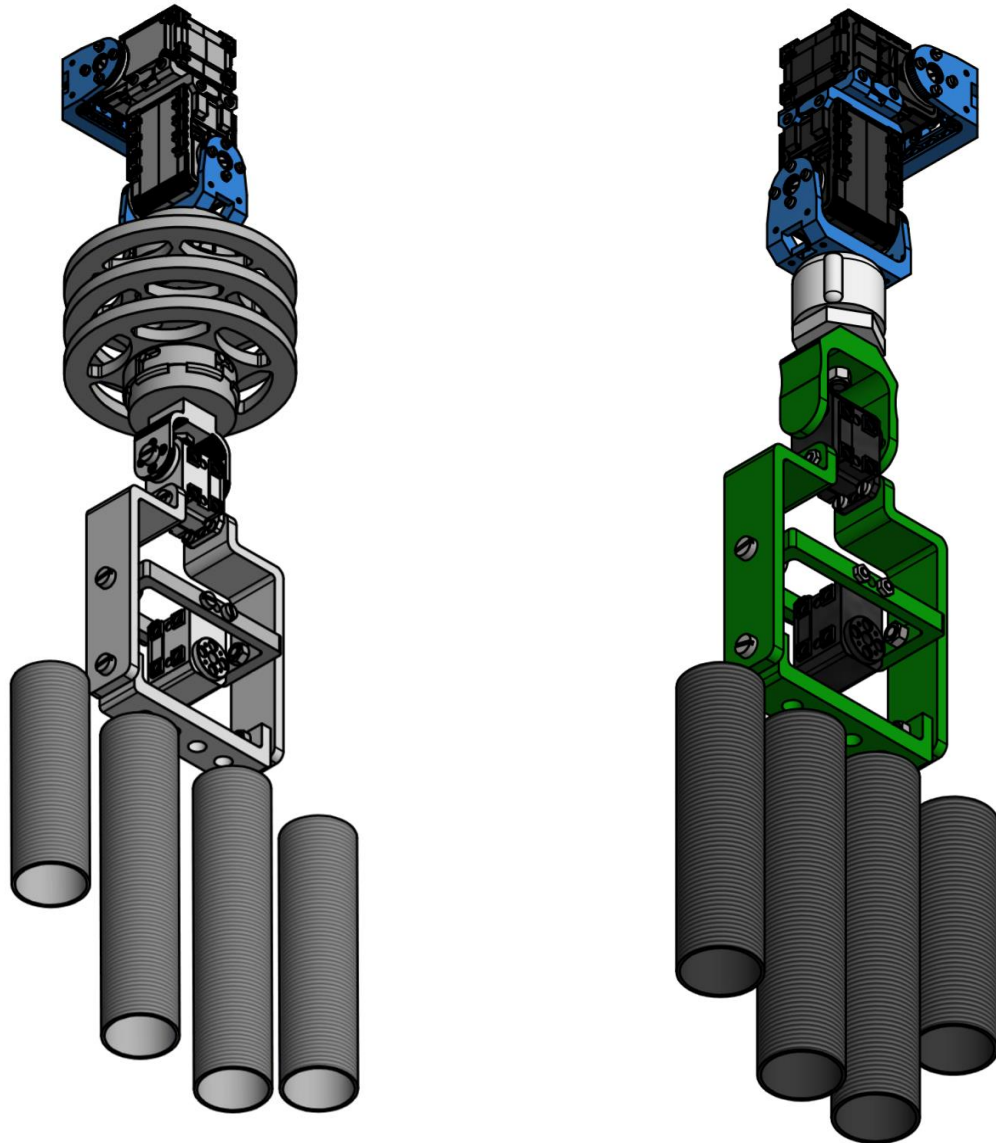
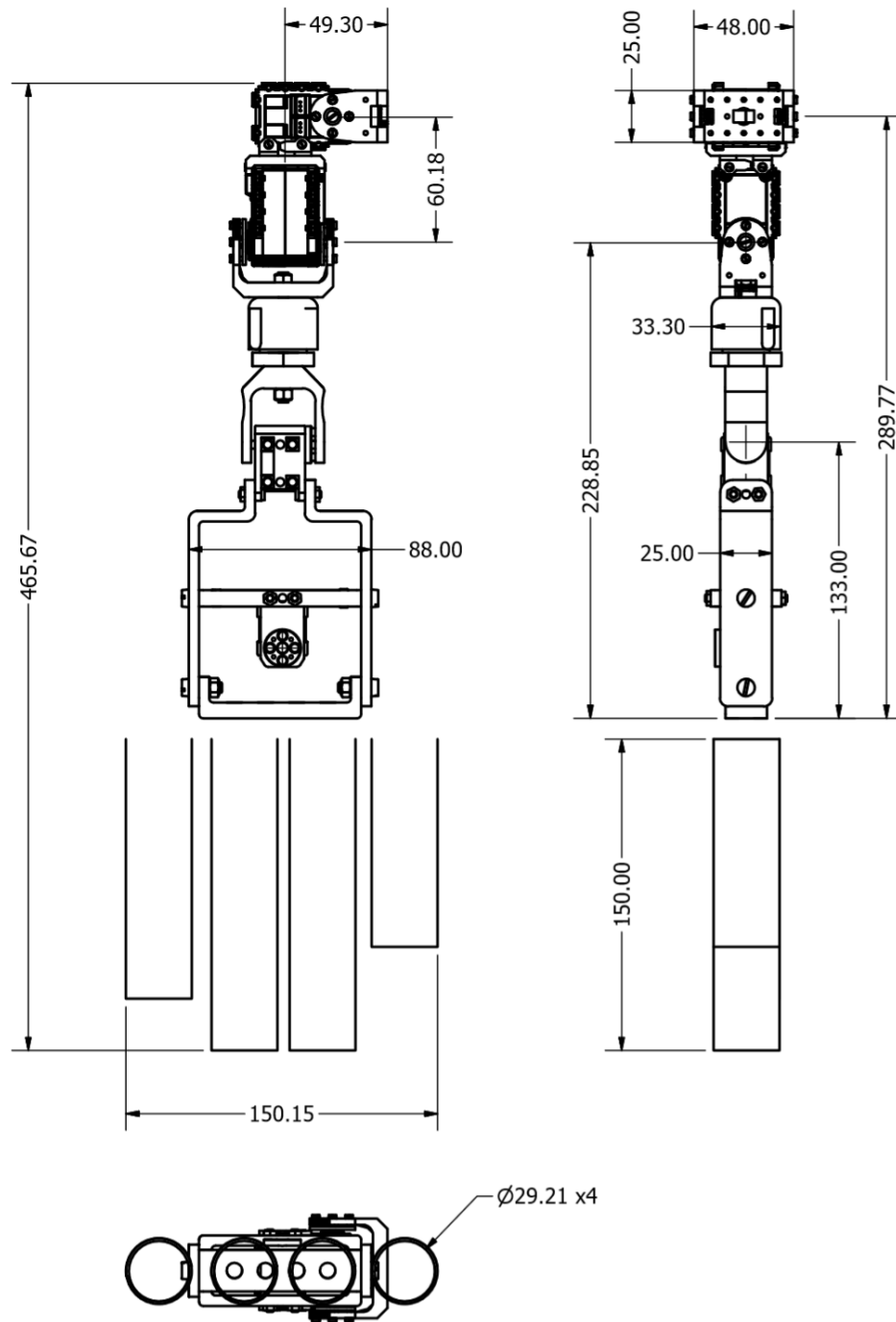


Figura 59. Últimas 2 iteraciones del diseño del brazo mecánico



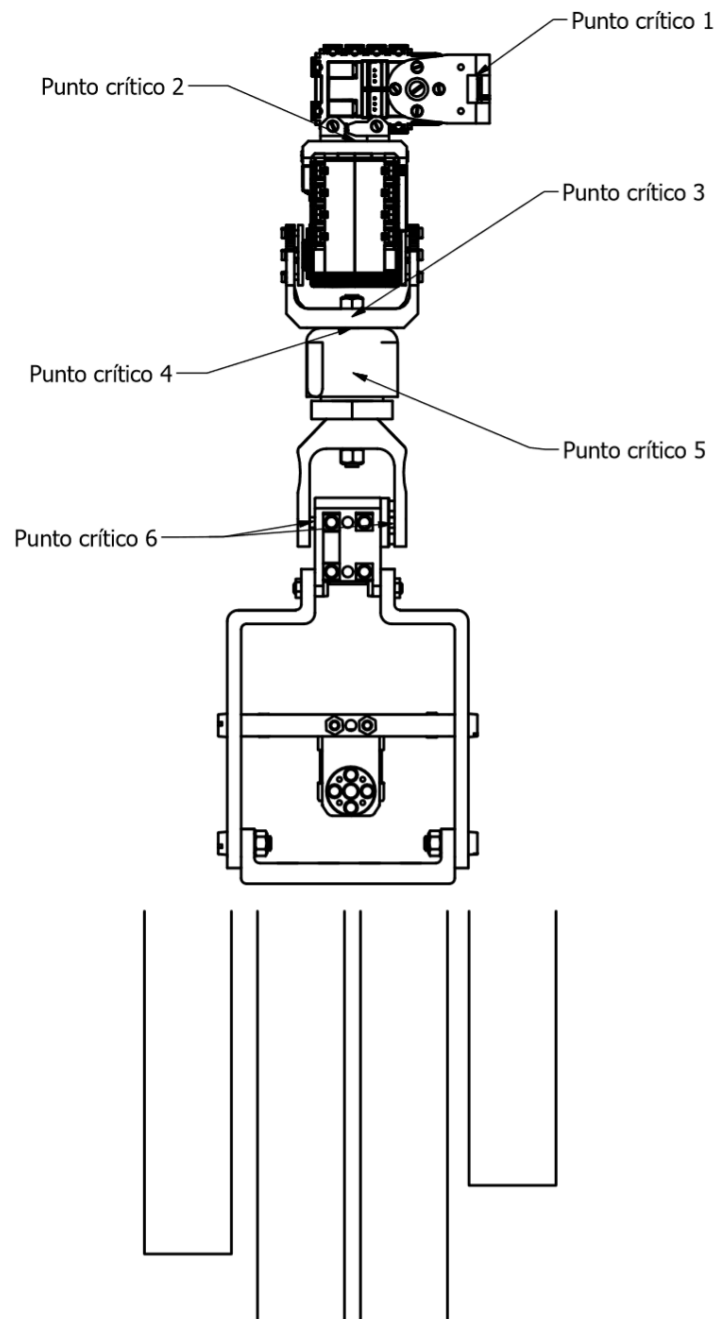
2. Dimensiones más relevantes. De las vistas ortogonales del brazo mecánico se obtuvieron las dimensiones más relevantes que se pueden observar a continuación:

Figura 60. Dimensiones más relevantes del brazo mecánico



3. **Puntos críticos para análisis de esfuerzos.** Los puntos críticos se ubicaron gracias a los diagramas de corte y momento para el brazo completamente en voladizo. En la Figura 61 solamente se ubican los puntos críticos, no se coloca el brazo en voladizo.

Figura 61. Ubicación de los puntos críticos del brazo mecánico



4. Diagrama de corte y momento en voladizo

Figura 62. Diagrama de corte y momento para el brazo completamente en voladizo

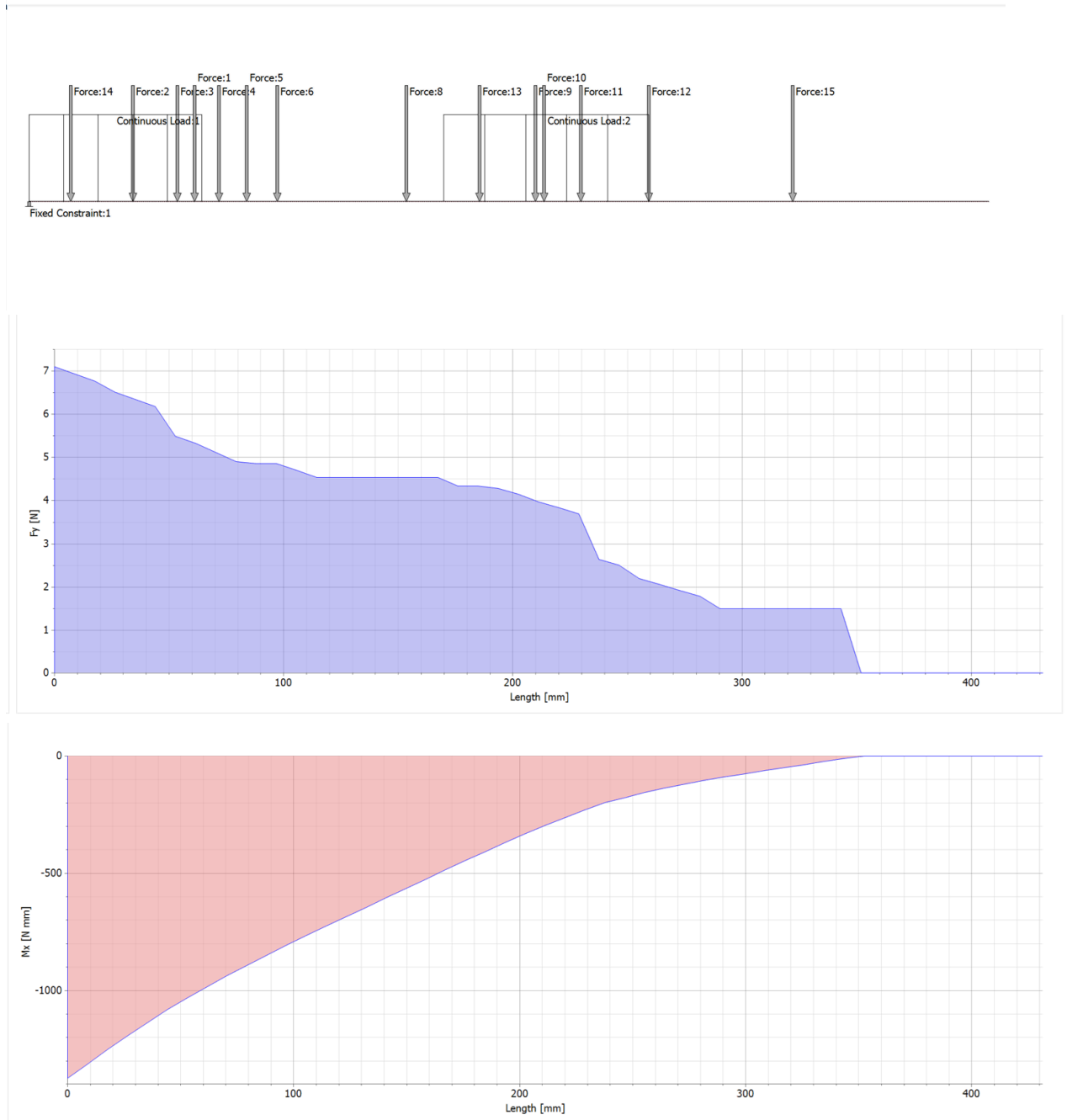


Figura 63. Resultados obtenidos del diagrama de corte y momento

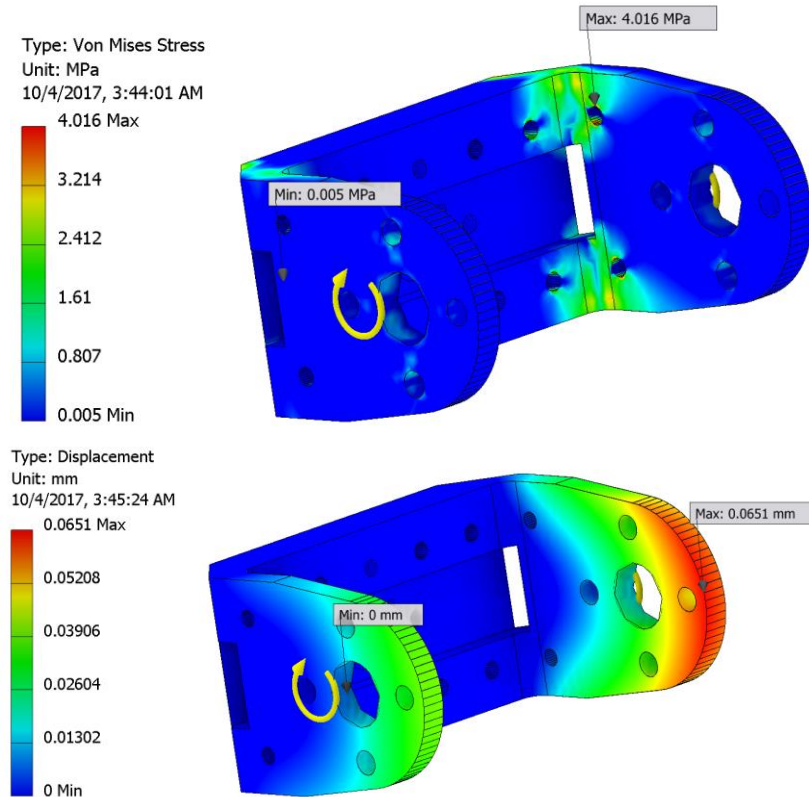
| Results | |
|---------------|----------------|
| Length | 440.000 mm |
| $F_{x_{max}}$ | -0.000 N |
| $F_{y_{max}}$ | 7.101 N |
| $F_{z_{max}}$ | 0.000 N |
| $M_{x_{max}}$ | -1373.870 N mm |
| $M_{y_{max}}$ | -0.000 N mm |
| $M_{z_{max}}$ | 0.000 N mm |

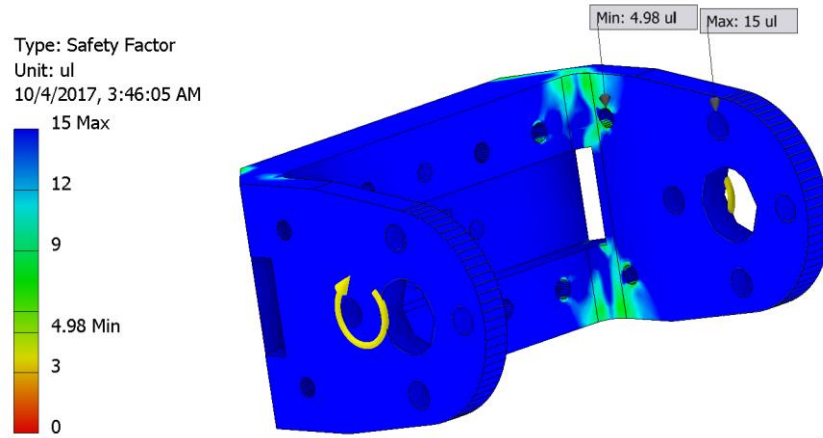
5. Análisis de esfuerzos

a. Punto crítico 1

- 1) Brazo horizontal. Resultados obtenidos para el punto crítico 1 con el brazo horizontal.

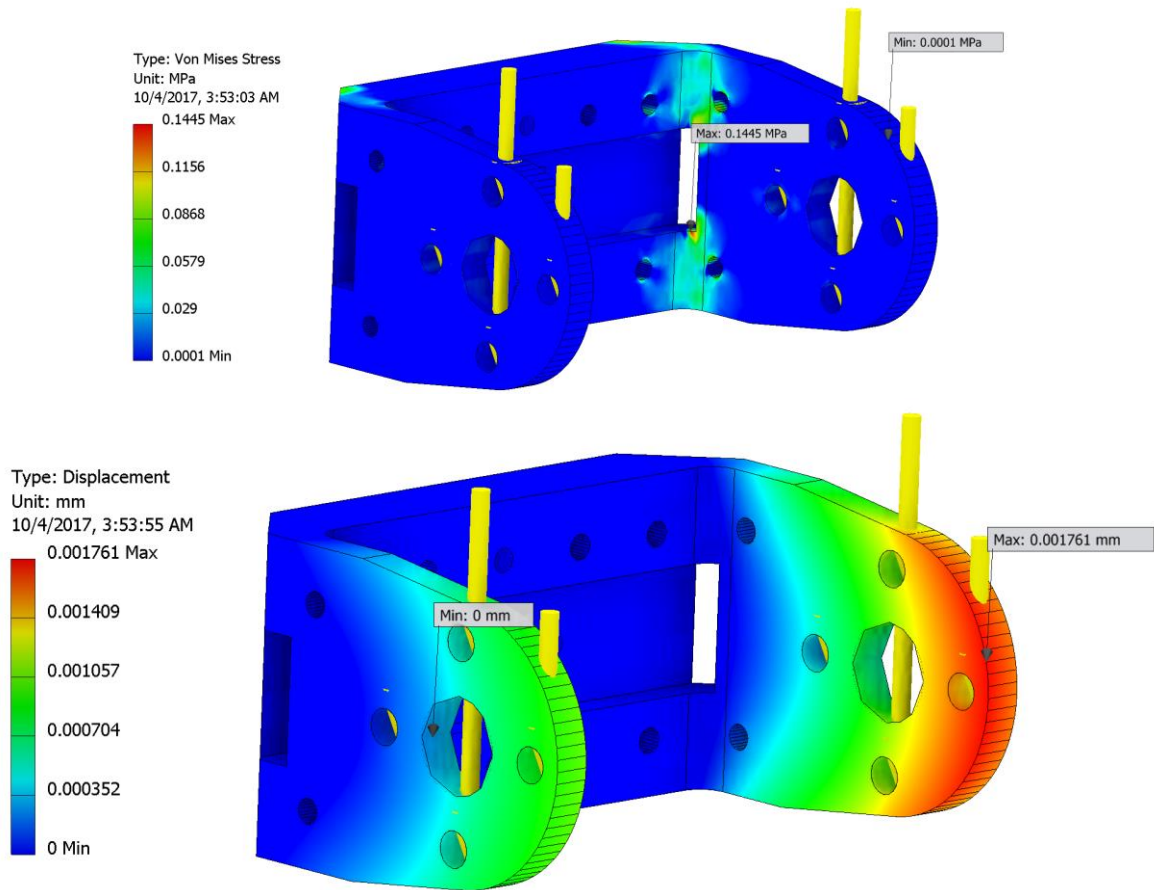
Figura 64. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 1.

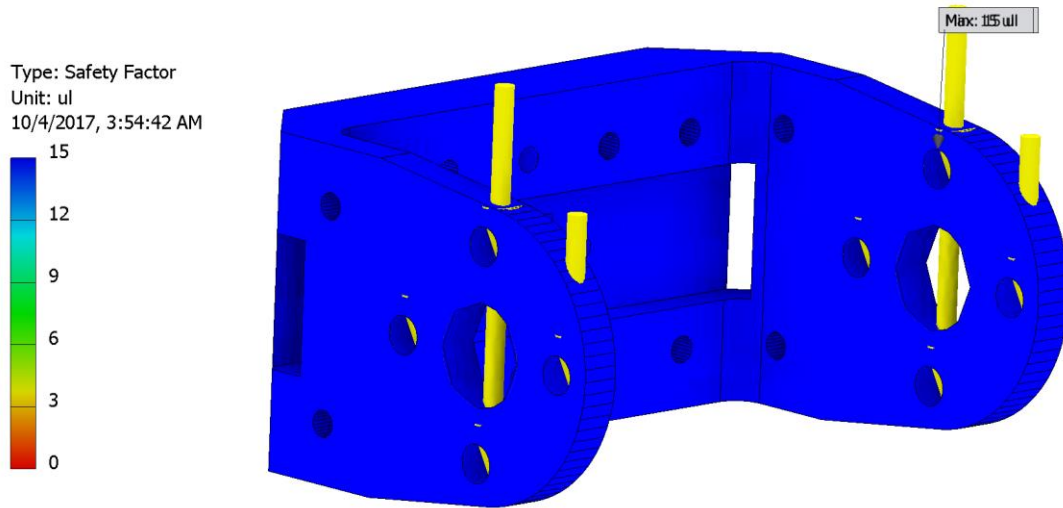




2) Brazo vertical. Resultados obtenidos para el punto crítico 1 con el brazo vertical.

Figura 65. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 1

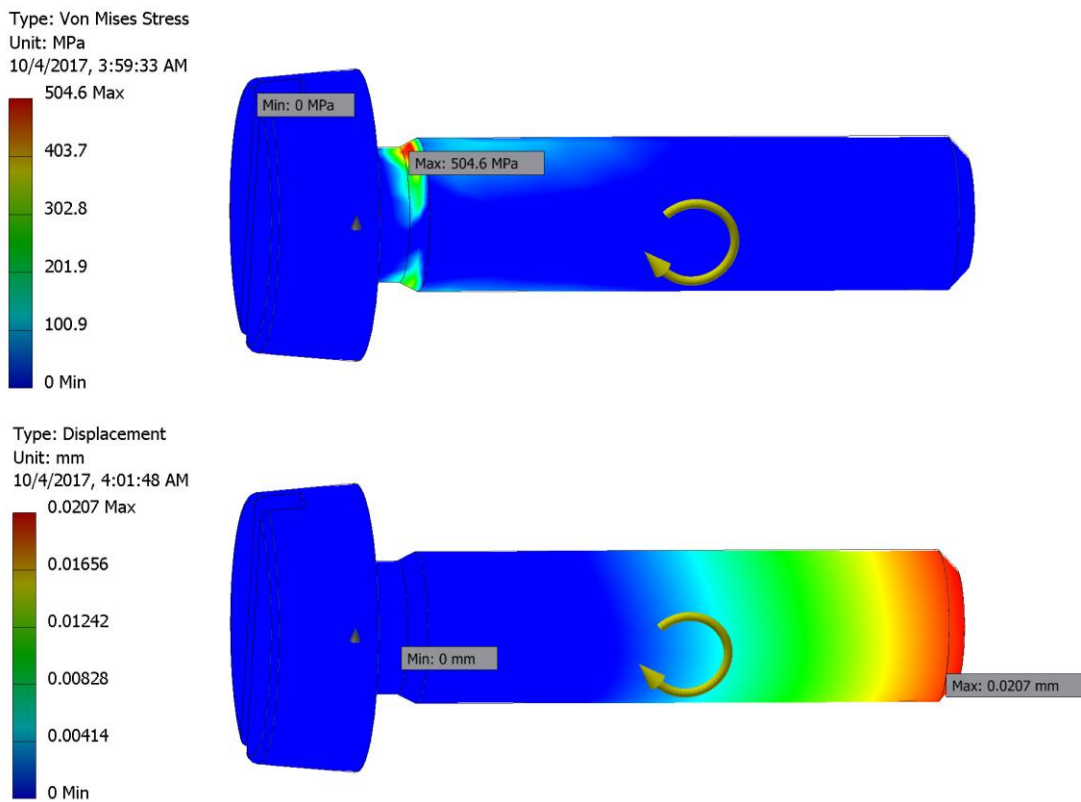


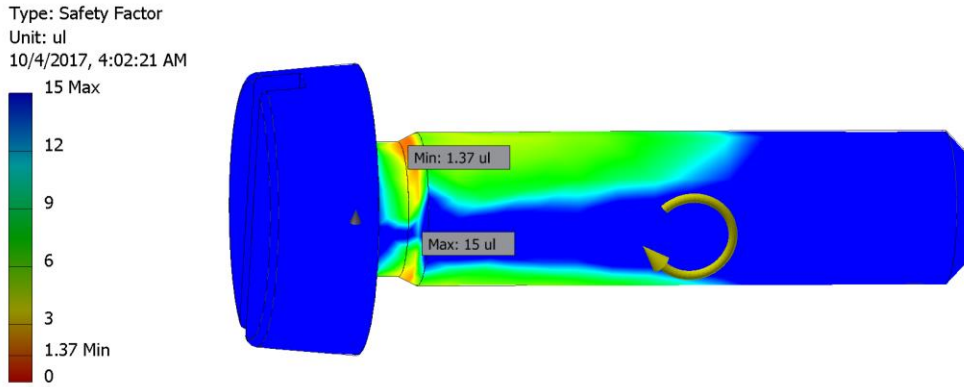


b. Punto crítico 2

- 1) Brazo horizontal. Resultados obtenidos para el punto crítico 2 con el brazo horizontal.

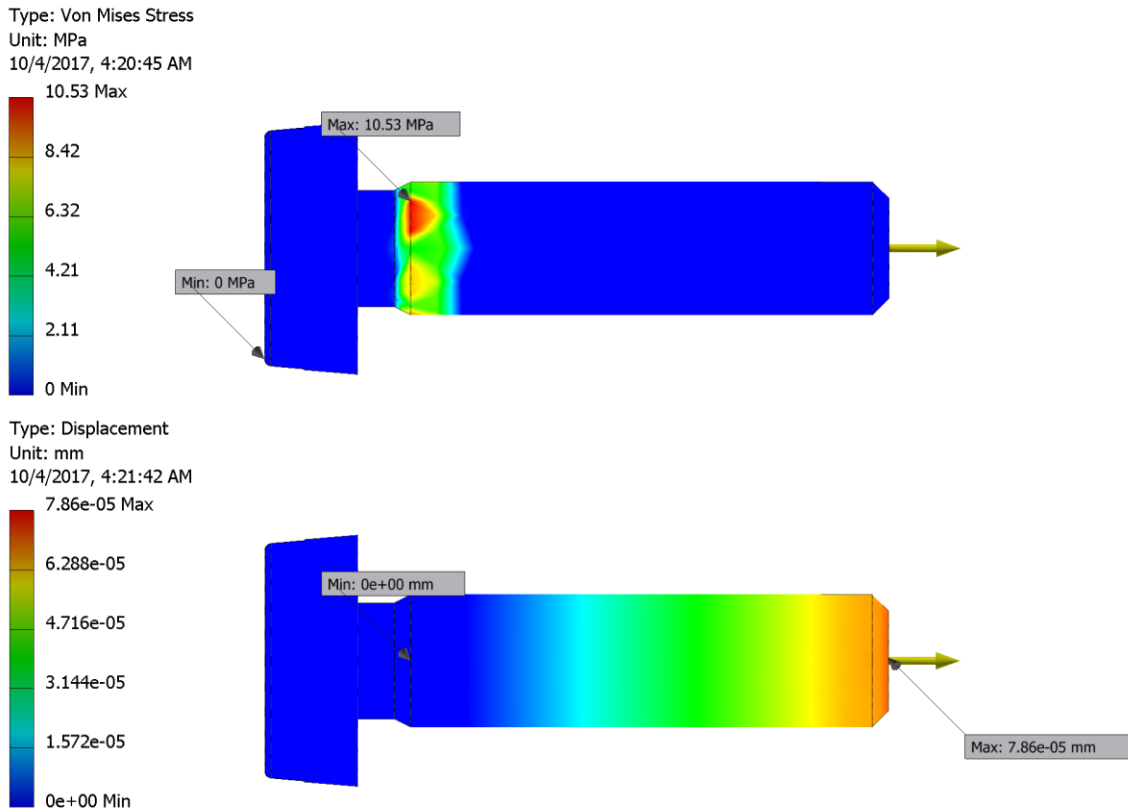
Figura 66. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 2

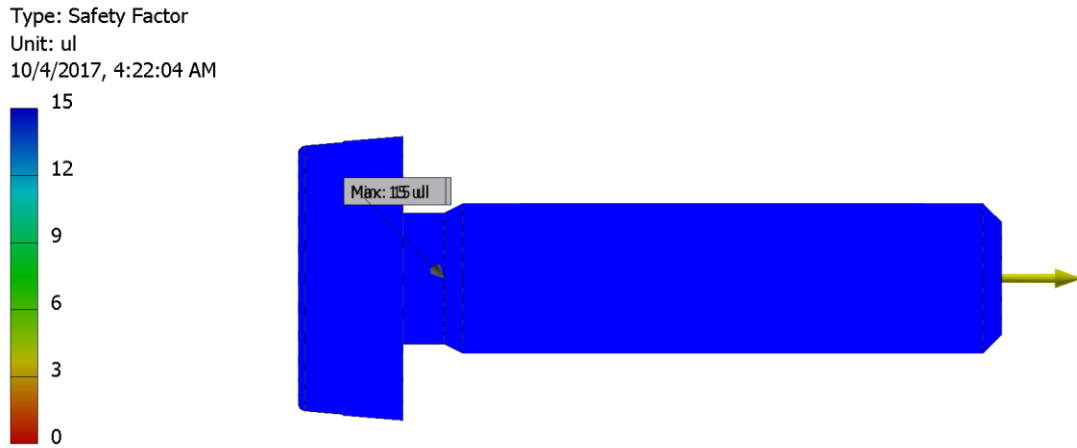




2) Brazo vertical. Resultados obtenidos para el punto crítico 2 con el brazo vertical.

Figura 67. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 2

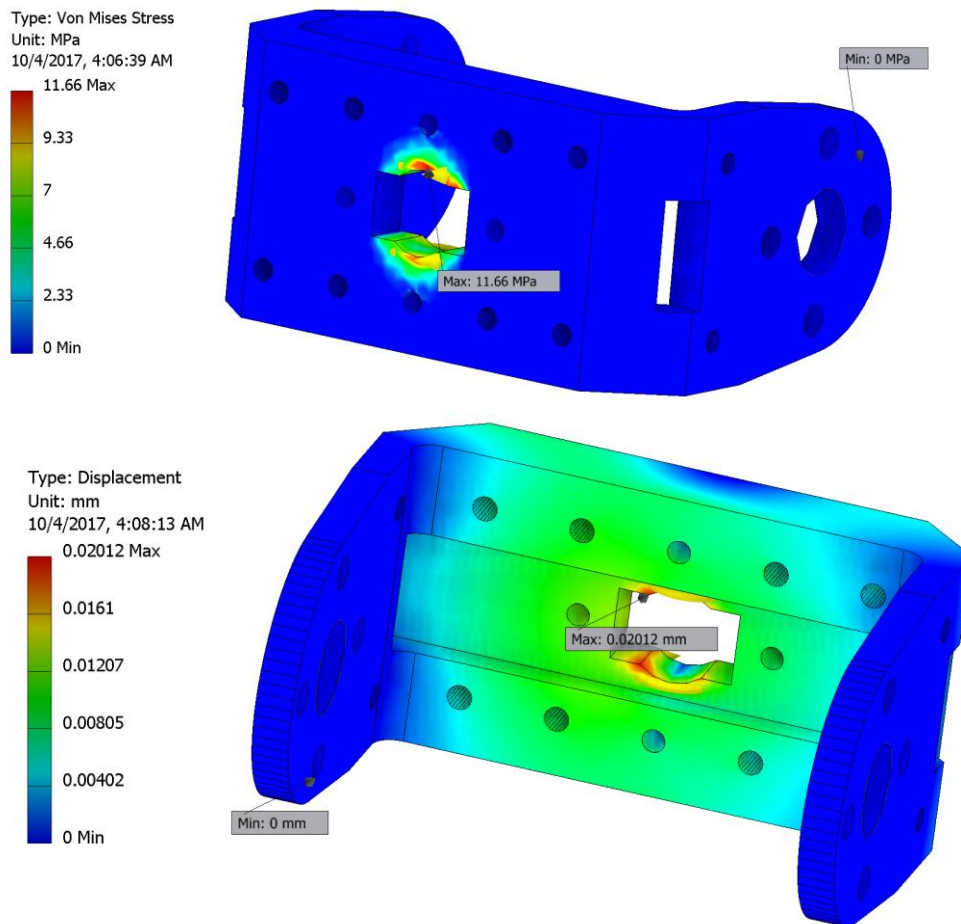


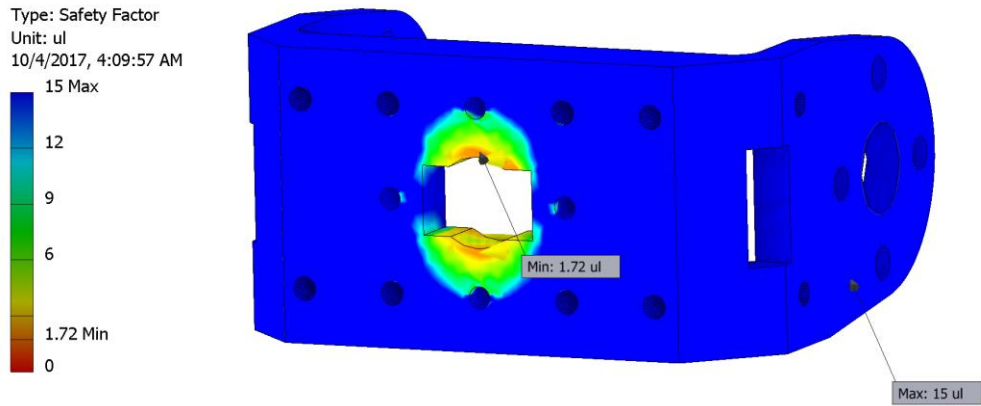


c. Punto crítico 3

- 1) Brazo horizontal. Resultados obtenidos para el punto crítico 3 con el brazo horizontal.

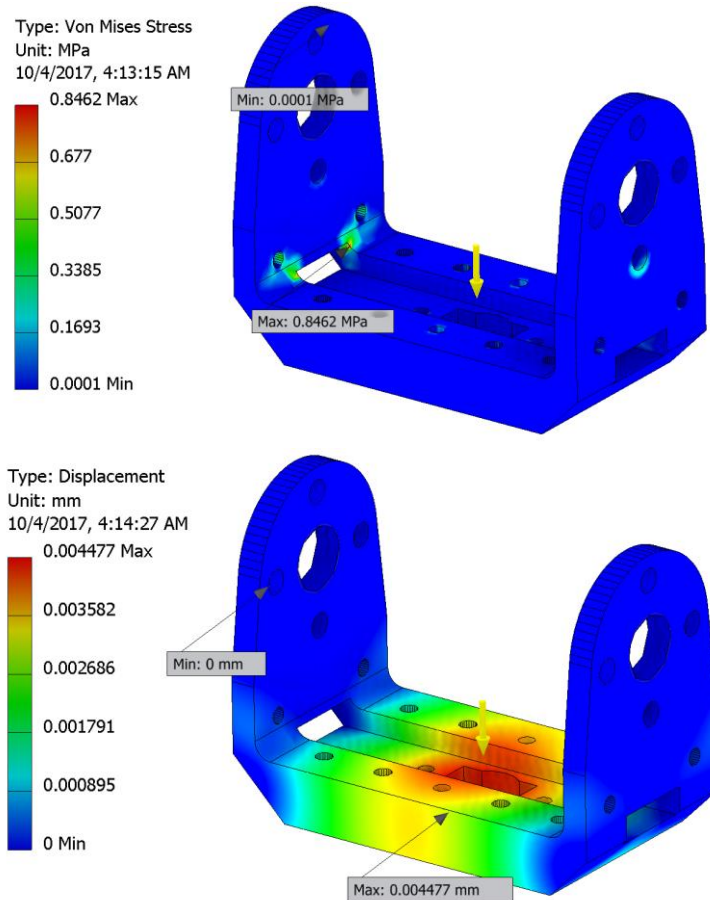
Figura 68. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 3

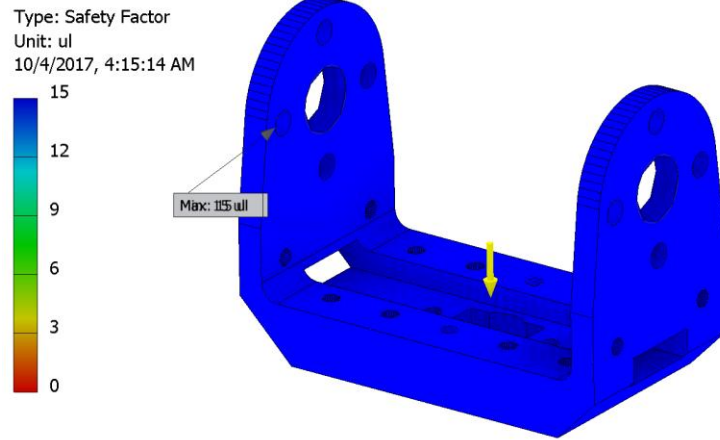




2) Brazo vertical. Resultados obtenidos para el punto crítico 3 con el brazo vertical.

Figura 69. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 3

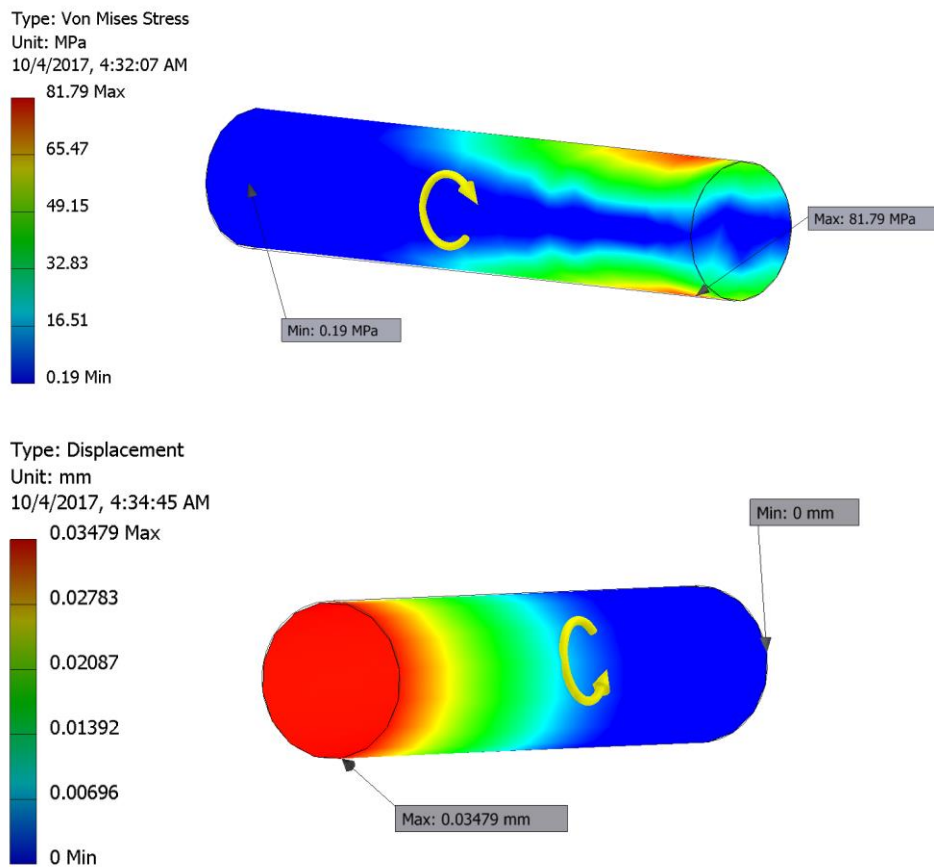


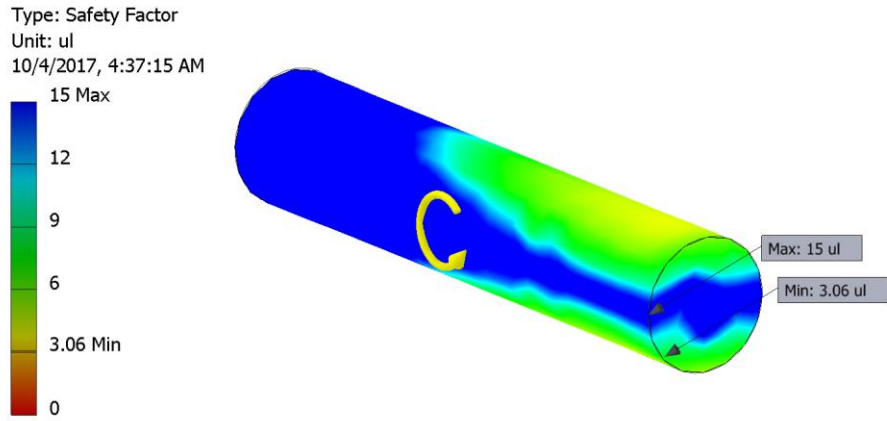


d. Punto crítico 4

- 1) Brazo horizontal. Resultados obtenidos para el punto crítico 4 con el brazo horizontal.

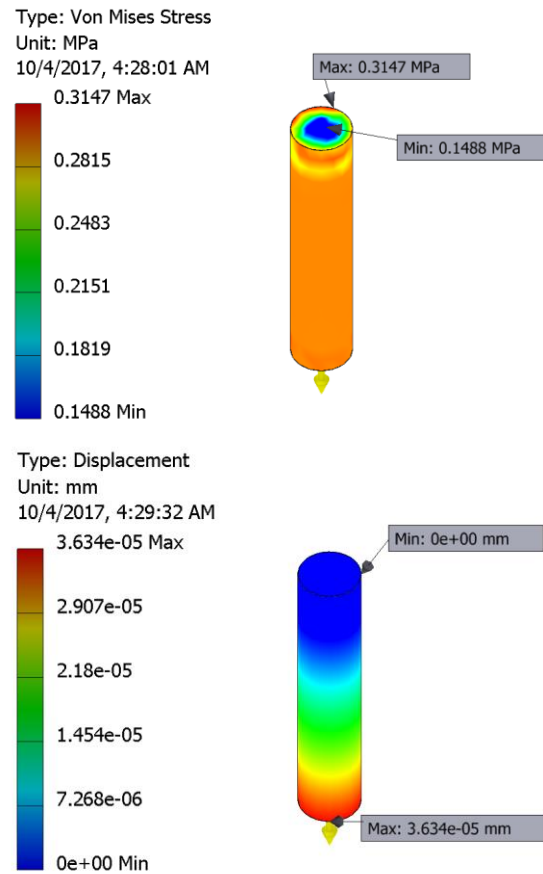
Figura 70. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 4

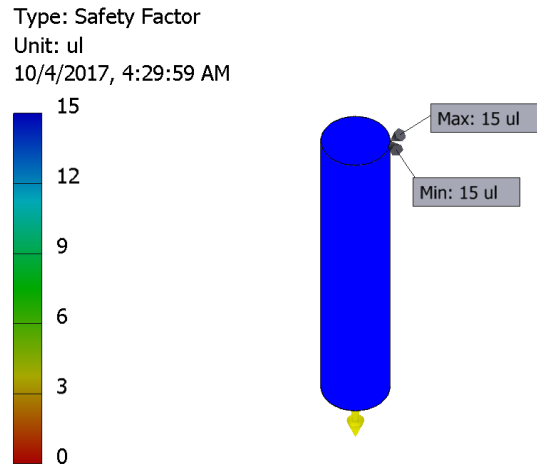




2) Brazo vertical. Resultados obtenidos para el punto crítico 4 con el brazo vertical.

Figura 71. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 4

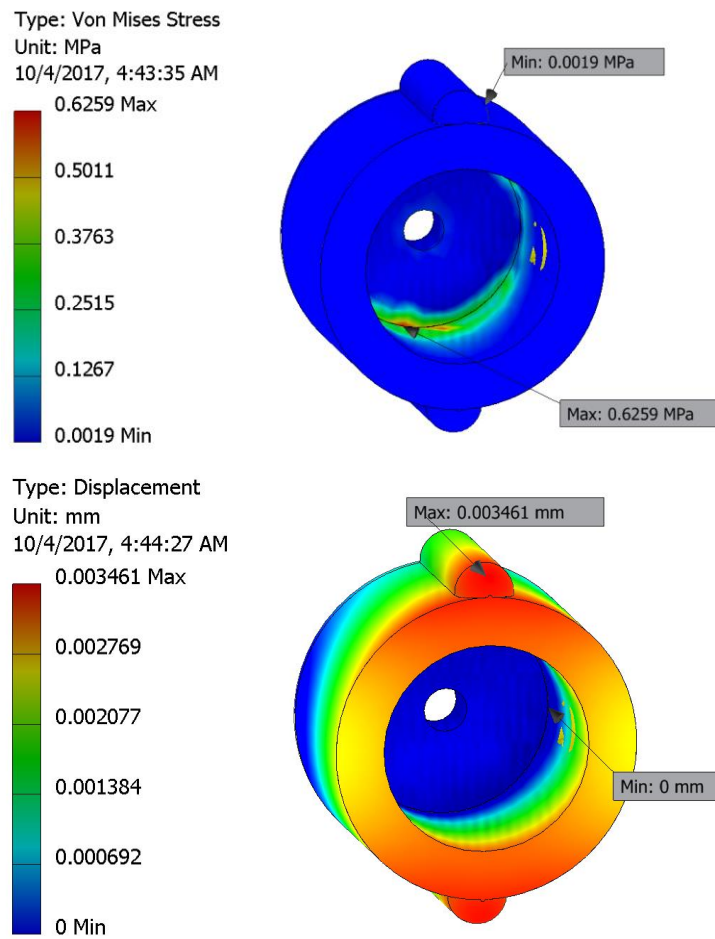


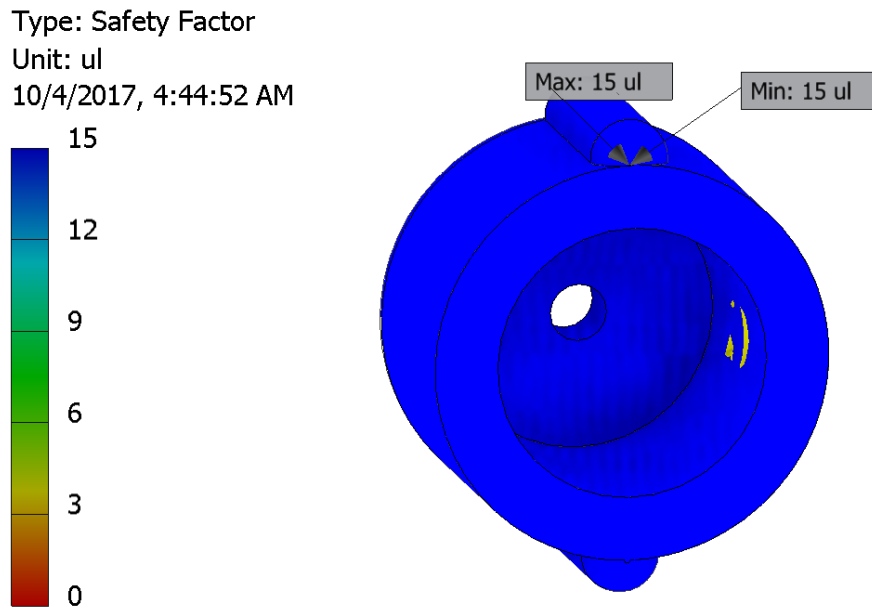


e. Punto crítico 5

1) Brazo horizontal. Resultados obtenidos para el punto crítico 5 con el brazo horizontal.

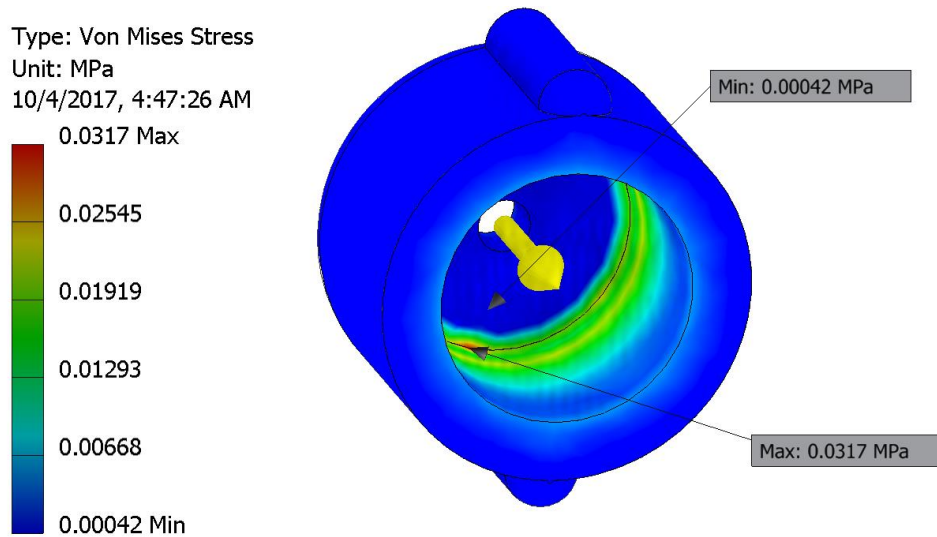
Figura 72. Resultados de esfuerzos, deformaciones y factor de seguridad del punto critico 5



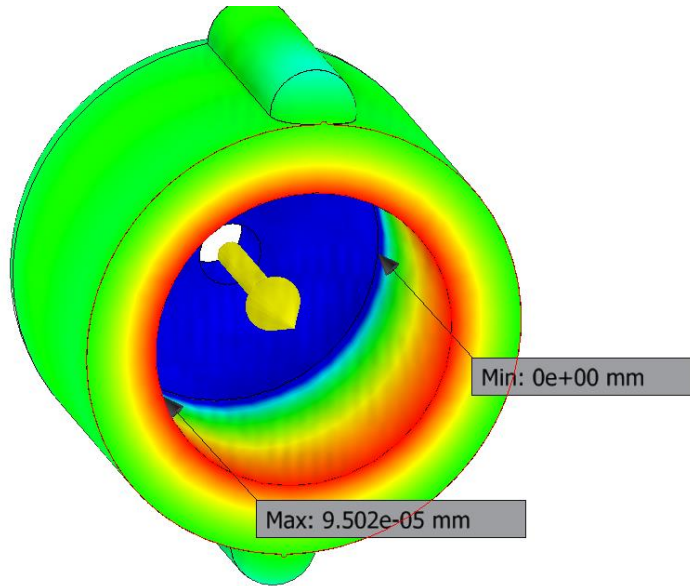
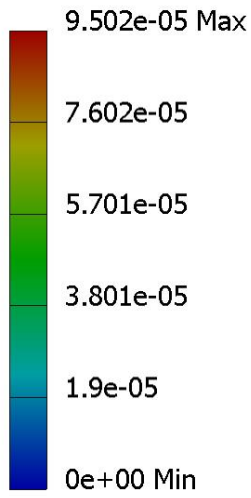


2) Brazo vertical. Resultados obtenidos para el punto crítico 6 con el brazo vertical.

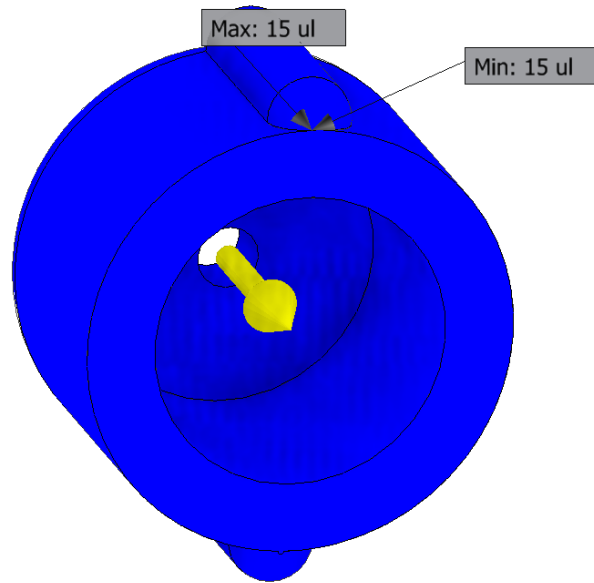
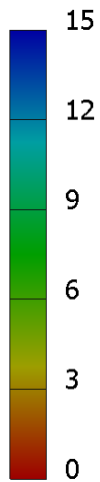
Figura 73. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 5



Type: Displacement
Unit: mm
10/4/2017, 4:47:47 AM



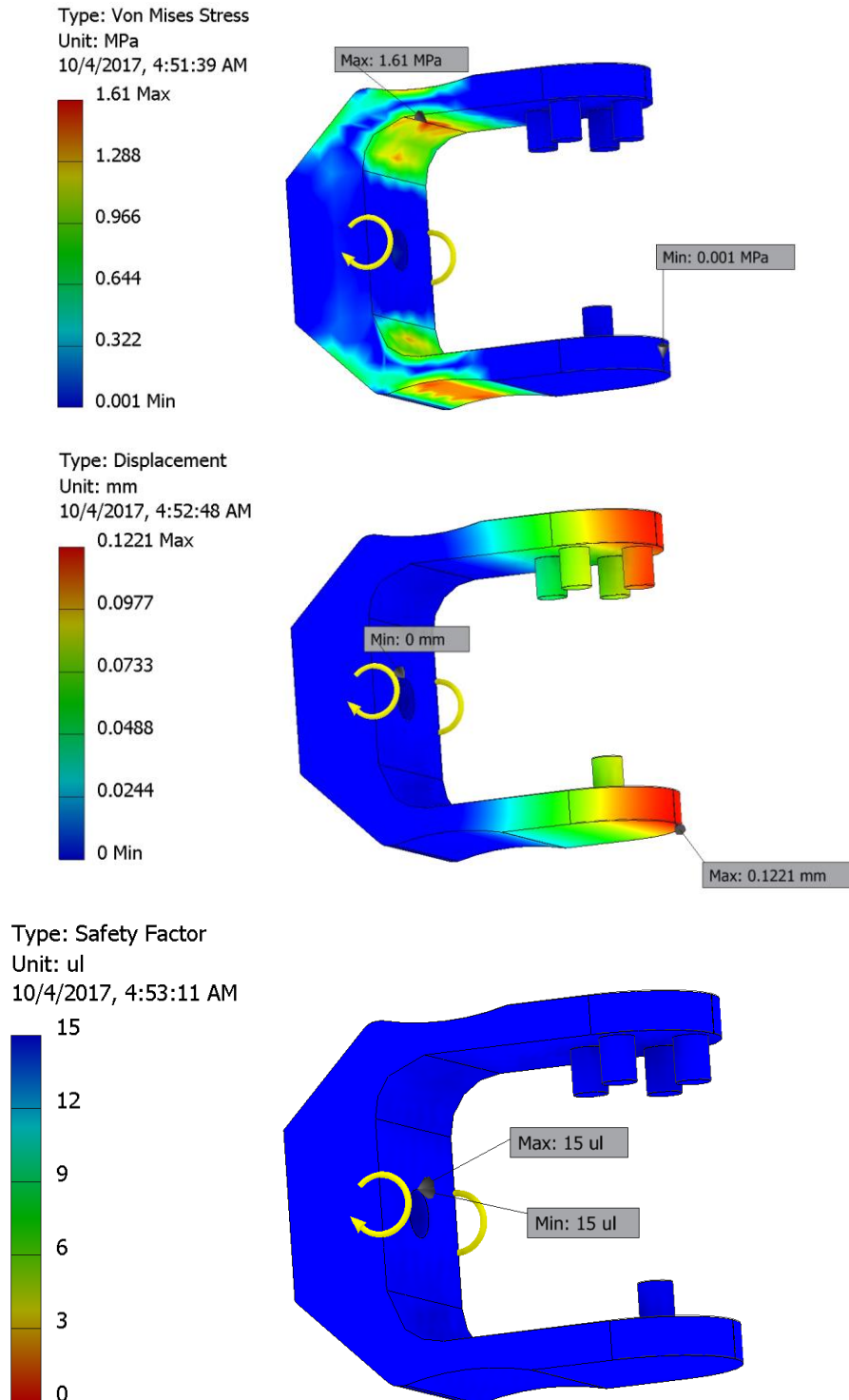
Type: Safety Factor
Unit: ul
10/4/2017, 4:48:17 AM



f. Punto crítico 6

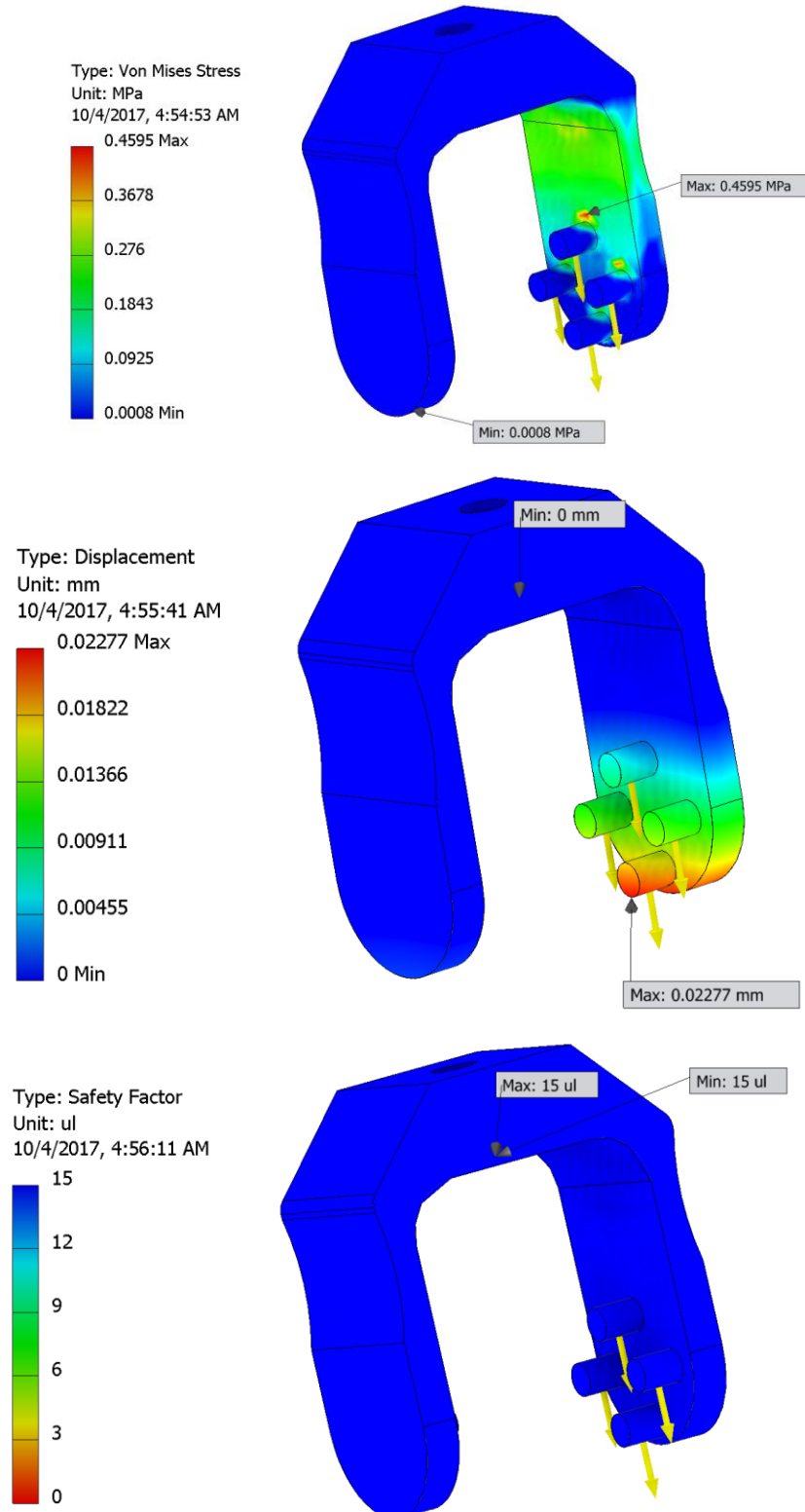
1) Brazo horizontal. Resultados obtenidos para el punto crítico 6 con el brazo horizontal.

Figura 74. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 6



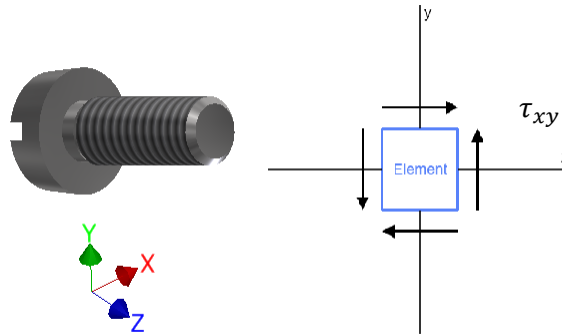
2) Brazo vertical. Resultados obtenidos para el punto crítico 6 con el brazo vertical.

Figura 75. Resultados de esfuerzos, deformaciones y factor de seguridad del punto crítico 6



6. Tornillos entre servomotor AX-12A superior con el accesorio F2222. Para el caso del brazo vertical:

Figura 76. Representación gráfica del elemento de esfuerzos en el tornillo



- Tipo de esfuerzo: Cortante puro
- Peso que soporta: 6.45 N
- Diámetro: 3 mm
- Cantidad: 2
- Material: Acero inoxidable
- Resistencia a la fluencia: 241 MPa
- Resistencia a la fluencia al corte: $0.577 \cdot S_y$

De la ecuación 5, se obtiene

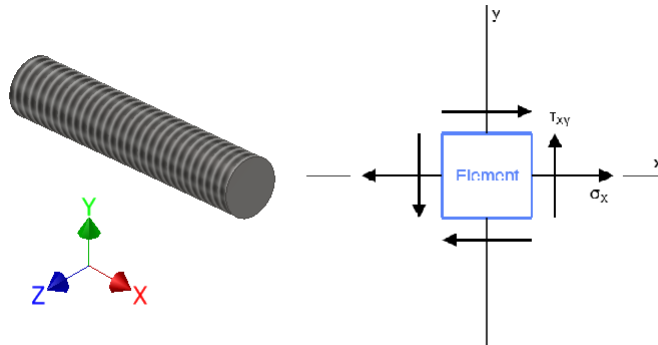
$$\tau_{xy} = \frac{F}{A} = \frac{6.45 \text{ N}}{2 * \pi * \frac{(3 \text{ mm})^2}{4}} = 0.46 \text{ MPa}$$

De la ecuación del factor de seguridad se obtiene

$$n = \frac{S_{sy}}{\tau_{xy}} = \frac{0.577 * (241 \text{ MPa})}{0.46 \text{ MPa}} = 304.95$$

a. Comprobación del punto crítico 4. Para el caso del brazo horizontal:

Figura 77. Representación gráfica del elemento de esfuerzos en el punto crítico 4



- Tipo de es.,fuerzo: Flexión
- Fuerza: 9.77 N
- Distancia del punto crítico al hombro (cero): 87 mm
- Diámetro: 4.76 mm
- Material: Acero inoxidable
- Resistencia a la fluencia: 241 MPa

De las ecuaciones de Momento se obtiene

$$M = F * distancia = (9.77N)(87mm) = 850 N * mm$$

$$c = \frac{diámetro}{2} = \frac{4.76 mm}{2} = 2.38 mm$$

$$I = \frac{\pi(diámetro)^4}{64} = \frac{\pi(4.76mm)^4}{64} = 25.26 mm^4$$

$$A = \pi c^2 = \pi(2.38mm)^2 = 17.82 mm^2$$

De la ecuación de Von Mises se obtiene

$$\sigma' = \sqrt{\sigma_x^2 + \sigma_x \sigma_y + \sigma_y^2 + 3\tau_{xy}^2} = \sqrt{\left(\frac{Mc}{I}\right)^2 + 3\left(\frac{F}{A}\right)^2}$$

$$\sigma' = \sqrt{\left(\frac{(850 N * mm)(2.38mm)}{(25.26mm^4)}\right)^2 + 3\left(\frac{9.77N}{17.82 mm^2}\right)^2} = 80.13 MPa$$

$$n = \frac{S_y}{\sigma'} = \frac{241 MPa}{80.13 MPa} = 3.0075$$

7. Análisis de esfuerzos para el resto de piezas críticas

Cuadro 17. Resultados del resto de piezas críticas

| Componente a analizar | Orientación del brazo | Tipo de esfuerzos | Resultados (MPa) | Factor de seguridad |
|--|-----------------------|-------------------|------------------|---------------------|
| Tornillos que unen los dos accesorios F333 | Vertical | Tensión pura | 0.235 | 1026.28 |
| | Horizontal | Flexión | 194.22 | 1.24 |
| Pines del accesorio del servomotor XL-320 superior | Vertical | Corte puro | 0.057 | 269.91 |
| | Horizontal | Flexión | 1.1 | 14.04 |
| Punto crítico 4 | Vertical | Tensión pura | 0.296 | 813.55 |
| | Vertical | Tensión pura | 0.018 | 2984.92 |
| Punto crítico 5 | Horizontal | Flexión | 0.696 | 346.44 |
| | Vertical | Corte puro | 0.078 | 3103.36 |
| Tornillos que sostienen el servomotor XL-320 superior a la pieza Codo-Muñeca | Vertical | Corte puro | 0.078 | 3103.36 |
| Componente F2222 superior | Horizontal | Flexión | 2.2 | 7.04 |

8. Análisis de tensión para los hilos de las garras

Cuadro 18. Resultados de tensión para los hilos de las garras y torque necesario para moverlos

| | Masas de 11g | Masas de 5.5 g | Total (g) | Peso (N) | Radio (mm) | Torque (N*mm) |
|--------|--------------|----------------|-----------|----------|------------|---------------|
| Dedo 1 | 3 | 2 | 44 | 0.43164 | 15 | 6.4746 |
| Dedo 2 | 19 | 0 | 209 | 2.05029 | 15 | 30.75435 |
| Dedo 3 | 19 | 0 | 209 | 2.05029 | 15 | 30.75435 |
| Dedo 4 | 5 | 3 | 71.5 | 0.701415 | 15 | 10.52123 |
| TOTAL | | | | | | 78.50453 |

Al sustituir valores en la ecuación de la conservación de la energía y derivarla respecto al tiempo se obtiene la aceleración angular que se puede sustituir en la ecuación del péndulo simple, obteniendo:

$$\tau = -g * m * L * \text{sen}(\theta)$$

donde,

g = gravedad

m = masa del brazo

L = longitud del brazo

θ = ángulo deseado de abducción

9. Estudio de carga teórica vs. carga experimental del servomotor AX-12A

Cuadro 19. Carga teórica vs carga experimental del AX-12A superior

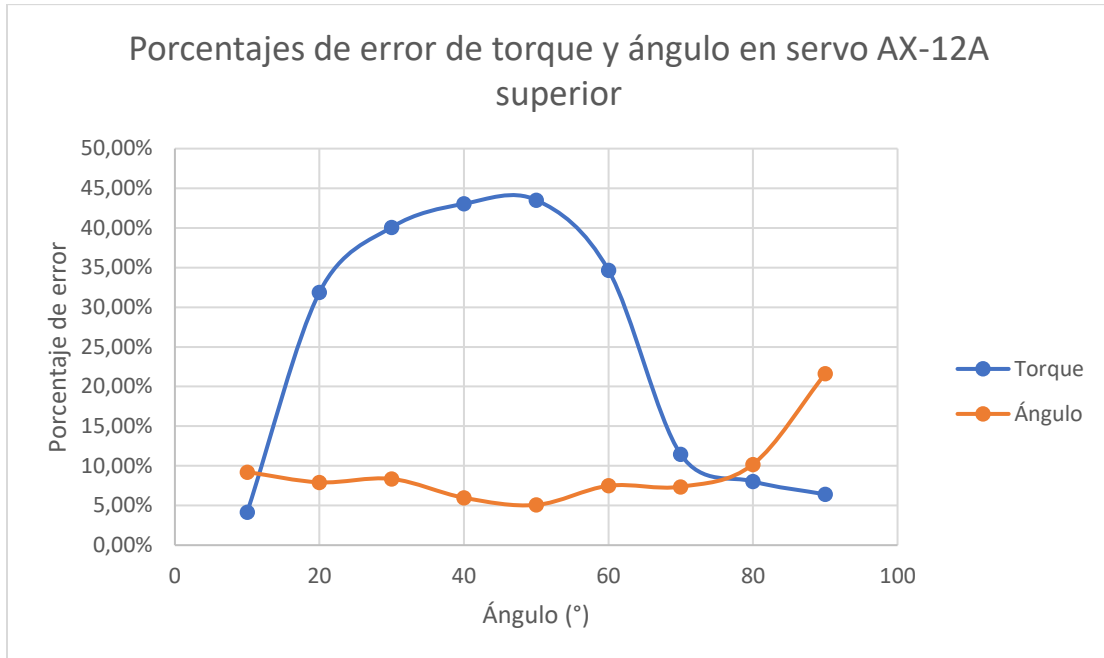
| | Ángulo | Teórico (N*m) | Experimental (bits, 0 a 1023) | Experimental (N*m) | Porcentaje de error |
|---|--------|---------------|-------------------------------|--------------------|---------------------|
| 1 | 10 | 0.2447868 | 160 | 0.234604106 | 4.16% |
| 2 | 20 | 0.4821360 | 224 | 0.328445748 | 31.88% |
| 3 | 30 | 0.7048357 | 288 | 0.42228739 | 40.09% |
| 4 | 40 | 0.9061193 | 352 | 0.516129032 | 43.04% |
| 5 | 50 | 1.0798710 | 416 | 0.609970674 | 43.51% |
| 6 | 60 | 1.2208113 | 544 | 0.797653959 | 34.66% |
| 7 | 70 | 1.3246579 | 800 | 1.173020528 | 11.45% |
| 8 | 80 | 1.3882554 | 1023 | 1.5 | 8.05% |
| 9 | 90 | 1.4096714 | 1023 | 1.5 | 6.41% |

10. Estudio de ángulo teórico vs. ángulo logrado para servomotor AX-12A

Cuadro 20. Ángulo teórico vs experimental del servomotor AX-12A superior

| | Ángulo Teórico (°) | Ángulo Experimental (bits, 497 a 173) | Experimental (°) | Porcentaje de error |
|---|--------------------|---------------------------------------|------------------|---------------------|
| 1 | 10 | 462 | 9.07780979 | 9.22% |
| 2 | 20 | 426 | 18.4149855 | 7.93% |
| 3 | 30 | 391 | 27.4927953 | 8.36% |
| 4 | 40 | 352 | 37.6080691 | 5.98% |
| 5 | 50 | 314 | 47.4639769 | 5.07% |
| 6 | 60 | 283 | 55.5043227 | 7.49% |
| 7 | 70 | 247 | 64.8414985 | 7.37% |
| 8 | 80 | 220 | 71.8443804 | 10.19% |
| 9 | 90 | 225 | 70.5475504 | 21.61% |

Figura 78. Porcentaje de error del torque y ángulo del servo AX-12A superior

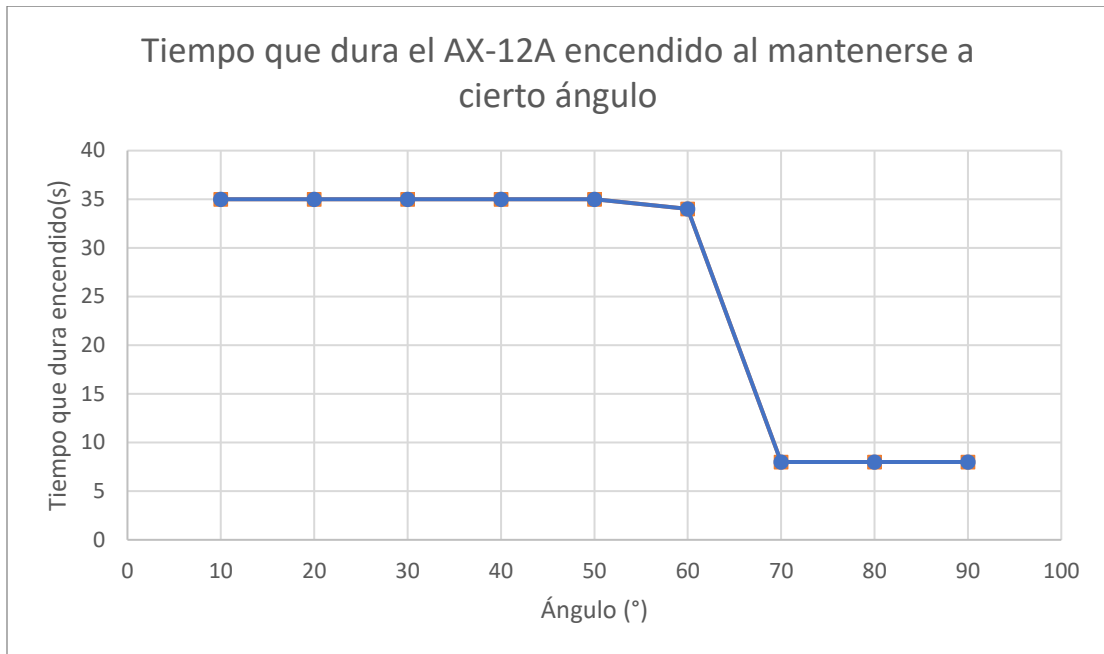


11. Tiempo que dura el AX-12A a un ángulo definido

Cuadro 21. Tiempo que se mantuvo encendido el servo AX-12A superior antes de apagarse por sobrecarga

| | Ángulo Teórico (°) | Tiempo antes de apagarse (asumiendo máx. = 35s) |
|---|--------------------|---|
| 1 | 10 | 35 |
| 2 | 20 | 35 |
| 3 | 30 | 35 |
| 4 | 40 | 35 |
| 5 | 50 | 35 |
| 6 | 60 | 34 |
| 7 | 70 | 8 |
| 8 | 80 | 8 |
| 9 | 90 | 8 |

Figura 79. Tiempo que se mantuvo encendido el AX-12A antes de apagarse a cierto ángulo



VIII. ANÁLISIS DE RESULTADOS

A. Componente de procesamiento de audio y video

1. Análisis de Trade Study del sistema de reconocimiento de voz. El trade study de la Tabla 3 muestra los criterios de selección de las opciones de librerías o API de reconocimiento de voz, y la confirmación de la selección con el Google Speech API como la mejor opción de API. El método de trade study implementado es el método de Pugh, en el cual en base a los criterios de selección se escoge la opción con mayor de diferencia de puntos entre la suma de positivos y negativos. El Google Speech API como la mejor selección se determinó mediante evaluación de los criterios de selección que distingue la versatilidad de la librería o API. El Cuadro 14 muestra el método de Pugh de trade study.

Los criterios de selección en consideración son la compatibilidad del entorno de programación, facilidad de integración, diversidad de funciones, soporte o comunidad, funcionamiento sin conexión a la red, amplio entorno de programación, mejora autónoma, personalización y compatibilidad del sistema. Cada uno tomado en consideración abarca diferentes perspectivas al verificar y analizar el requisito a cumplir con el objetivo de reconocimiento de voz. Debido a que el sistema de reconocimiento de voz cambia su forma de funcionar según sea el objetivo.

Entonces, la compatibilidad del entorno de programación se refiere a la flexibilidad de adaptarse a diferentes de plataformas de código. Sea en ambiente de Python, Java, C++, C# e incluyendo Matlab con su uso de lenguajes de los previos de derivación de C. Snowboy y Jasper provee formatos en lenguajes para posibilidad del uso con la computadora con Windows y dedicado para entorno de Raspberry; al contrario, la estructura del código y archivos encadenados para el funcionamiento no es amigable con el entorno de sistema operativo Windows y en ciertas distribuciones de Linux. Pocketsphinx e IBM Watson API provee calidad similar de compatibilidad del entorno de programación especialmente con Python, aunque requiere códigos bases para funcionamiento correcto. Google Speech API provee la mayor compatibilidad de los entornos de programación en especial a Python y configuración es adaptado al lenguaje al cual es desarrollado el sistema. Por lo cual, Google Speech API lleva el único punto positivo.

La facilidad de integración implica el uso de librería o API sin necesidad de alguna modificación sobre el archivo de funcionamiento del mismo para luego poder integrar al lenguaje que está siendo desarrollado. Al igual que anterior, Snowboy y Jasper obtienen puntos negativos por la necesidad de configurar específicamente la librería en el entorno que se opera para garantizar el funcionamiento correcto de los mismos. PocketSphinx e IBM Watson API proveen mayor modularidad cuanto al uso de la API o librería, aunque requiere configuración alterna en el cual requiere conexión o archivos terceros adicionales implicando en aumento del volumen de operación. Google Speech API lleva el único punto positivo debido a que requiere

el API únicamente y llamar la función de este API mismo para tener listo la configuración y desarrollar en base a misma estructura sin estar encadenando con librerías o APIs terceros.

Las funciones en mayor parte son los que definen la flexibilidad de la librería o API a adaptarse en numerosos escenarios de operaciones. Snowboy, Jasper, PocketSphinx, IBM Watson API y Google Speech API proveen diversidad de funciones previamente integradas que ofrecen de parte de los desarrolladores mismos. La cantidad de funciones no define la calidad concreta de librerías, pero existe probabilidad en que una librería o API resulte con una función específicamente necesitado a diferencia demás librerías o APIs. Por lo tanto, se evaluó de la forma que todas las librerías presentes en trade study son similares entre sí.

Como los desarrolladores, el soporte y la comunidad es de un factor importante a considerar debe a que son los quienes realizan la realimentación del funcionamiento de la librería o API. Al tener mayor soporte y una comunidad amplia infiere entonces mayor información sobre la librería o API y rendimiento sólido en las implementaciones. Snowboy, Jasper y PocketSphinx poseen comunidades abierto por los desarrolladores y aparte por los usuarios de estos. En caso de IBM Watson API proviene desde una industria y aparte poseen comunidad de usuarios de esta API el cual asegura el funcionamiento y resolución de problemas de la librería o API. En caso de Google Speech API, que lleva el punto positivo, proviene igualmente de una industria. La diferencia basa en que Google Speech API posee un vasto de comunidades, y es la fuente y recurso principal en mayoría de dispositivos y sistemas operativos.

El funcionamiento sin conexión a la red pueda ser un factor opcional pero determinante según el entorno de implementación. Es decir, al requerir la conexión a la red es posible que reduzca el uso de recursos locales para operaciones computacionales; mientras que al no tener la conexión a la red infiere operaciones computacionales únicamente local requiriendo dispositivo computacional de mayor potencia. Adicionalmente, el tema de privacidad encuentra en este factor, por lo cual, si el usuario no desea transferir datos relacionados a la voz del mismo a la red, la única opción es utilizar una librería o API sin el requerimiento de conexión a la red. Jasper, IBM Watson API y Google Speech API requiere conexión obligatoria debido a que realizan operaciones en los servidores propios y en medio del proceso de esta operación transfiere los archivos del usuario en los bases de datos de los mismos. PocketSphinx da la opción de poder realizar operaciones localmente sin conexión a la red o con conexión a la red. A diferencia de los demás, Snowboy funciona localmente sin necesidad de la conexión a la red, dando ventaja de modificación abierta del desarrollo de programa sin algún límite.

La compatibilidad del entorno de programación es importante reconocer, aunque es obsoleto si el entorno de programación no es amplio. Es decir, si el entorno de programación a cuál se pueda desarrollar es limitado, el nivel de la compatibilidad del mismo no es un factor considerable debido que las opciones están bastante limitadas. Snowboy y Google Speech API ofrece posibilidad de desarrollar en diferentes lenguajes e

incluyendo Python que usan comúnmente los desarrolladores. Jasper e IBM Watson API proveen en entornos de programaciones predeterminados que es posible extender la comunicación en otros entornos de programación, aunque la libertad de modificación de estos códigos predeterminados o conversión en un entorno de programación diferente requiere realizar desarrollos adicionales para poder mantener la lógica sin perder la eficiencia de ejecución del mismo. PocketSphinx tiene enfocado el entorno de programación en el cual es posible integrar en otros entornos de programación; al contrario, traer entornos de programación distinto al base de lenguaje desarrollado es necesario modificar varias lógicas de algoritmo para la posibilidad de mantenerlos.

Compatibilidad de plataforma es otro factor adicional diferente de compatibilidad de entorno de programación que enfoca en sistemas operativos a cuál la ejecución de la librería o API operan. Para Snowboy, Jasper, PocketSphinx, e IBM Watson API no funciona en todas las plataformas, aunque no es necesario ni requisito que funcione en todas las plataformas, pero si provee garantía de funcionamiento en plataformas de Windows o Raspberry Pi. Mientras que Google Speech API provee compatibilidad de funcionamiento correcto en Raspberry Pi y Windows. Por el cual el Google Speech API lleva el punto positivo únicamente.

La cuestión de mejora de la librería normalmente depende de las actualizaciones que realizan los desarrolladores. Algunas librerías o API, a diferencia de dependerse únicamente de actualizaciones, mejora el rendimiento de algoritmo de forma autónoma. Es decir, la mejora autónoma que realiza se hace por medio de proceso de “Machine Learning” o “Deep Learning”. Estos conceptos basan en mejorar el rendimiento de tal aspecto por medio de usos que se realizan. PocketSphinx, a diferencia demás librerías o API no utiliza este método de mejora, aunque tiene paquetes de actualizaciones bastante sólidos. Snowboy, Jasper, IBM Watson API y Google Speech API mejora el rendimiento de sus sistemas de reconocimiento de voz por medio de estos métodos de aprendizaje.

Por último, la personalización que ofrece la librería o API pueda que ayude a desarrollar de forma más flexible o rígida dependiendo de la aplicación que pueda realizar. IBM Watson API y Google Speech API, aunque son buenos APIs para el sistema de reconocimiento de voz, carece de ofrecer la habilidad de crear módulos personalizados e implementarlos. Snowboy, Jasper y PocketSphinx son los ejemplares en cuánto a nivel de personalización y un buen ejemplo es el Snowboy que ofrece creación del módulo propio para diversidad de aplicaciones posibles que el desarrollador o al usuario le pueda idear.

B. Análisis de estadísticas

1. Estadística de prueba de aciertos del Google Speech API. Con base en el modelo de prueba de sistema de reconocimiento de voz obtuvo el Cuadro 15. Esta estadística basa en prueba de aciertos con la implementación del Google Speech API. Entonces existen aciertos correctos e incorrectos como categorías

para separar entre reconocimientos correctos e incorrectos. La primera complejidad se basa en 1 término – “Hola” – con 30 iteraciones y tuvo 24 aciertos correctos y 6 incorrectos. La segunda complejidad se basa en 2 términos – “¿cómo estás?” – con 30 iteraciones y tuvo 21 acierto correctos y 9 incorrectos. La tercera complejidad basa en 3 términos – “Hola, ¿cómo estás?” con 30 iteraciones y tuvo 22 aciertos correctos y 8 incorrectos. La cuarta complejidad basa en 4 términos – “¿Cuál es tu nombre?” – con 30 iteraciones y tuvo 10 aciertos correctos y 20 incorrectos. La quinta complejidad basa en un término distinto – “Selfie” – con 30 iteraciones y tuvo 20 aciertos correctos y 10 incorrectos.

El ambiente de la prueba se realizó en un lugar donde existe la presencia de ruidos externos en los cuales no son comandos o voz de parte del usuario. Entonces, los tres primeros niveles de complejidad muestran mayor número de aciertos correctos a comparación de acierto incorrectos presentes. Al combinar con tres términos en máximo ofrece un nivel de reconocimiento bastante consistente. Al contrario, la cuarta complejidad la cuál combina cuatro términos resultó dificultando al sistema de reconocimiento con mayor número de aciertos incorrectos. Posiblemente mientras la ventana de detección, la señal de audio mezcla con ruidos de pico resultando en una tonalidad de audio representativo distinto a lo que realmente es la señal con tonalidad original de comando de voz. La última complejidad se basó en un término que mezcla lenguaje de inglés y con tonalidad español. El término es uno único, pero resulta ser más complejo a diferencia del término lingüísticamente español únicamente. En fin, el sistema de reconocimiento de voz muestra un rendimiento óptimo en máximo de combinaciones de tres términos diferentes y de términos lingüísticamente combinados.

2. Estadísticas probabilísticas de aciertos del Google Speech API. Al igual que la estadística de prueba de aciertos, esta estadística baso a partir del mismo modelo de prueba, pero con un enfoque distinto al anterior. Es bueno conocer la cantidad de los aciertos correctos e incorrectos de tal forma que los resultados son posibles de observar directamente. Al contrario, no provee alguna información general o estándar de que tan preciso es el sistema de reconocimiento con una probabilidad para predicción del comportamiento del reconocimiento. Entonces, elaboró una estadística que toma en cuenta el aspecto probabilístico del rendimiento de sistema de reconocimiento de voz.

Del Cuadro 16, se puede observar los resultados probabilísticos de cada nivel de complejidad. La categoría característica ve con la probabilidad de fallo de acierto y promedio de acierto correcto. La probabilidad de fallo de acierto tiene función de ver que tan probable es el sistema de reconocimiento tiende a fallar cada 30 iteraciones. El promedio de acierto correcto predice probabilidad de tener aciertos correctos cada 30 iteraciones. Entonces, las tres primeras complejidades presentan promedio de aciertos correctos mayores de 70% con probabilidad de fallo de acierto máximo de 30%. Esto es seguro interpretar en que por lo menos 70% de veces en cada 30 iteraciones el sistema de reconocimiento de voz tendrá aciertos correctos o podrá interpretar el comando del usuario correctamente con tendencia de posibilidad de fallo de acierto con una probabilidad máxima de 30% veces cada 30 iteraciones.

En caso del cuarto nivel de complejidad presentan promedio de acierto correcto bastante baja con 33% a comparación de probabilidad de fallo de acierto de 67% cada 30 iteraciones al realizar un comando del presente nivel o combinaciones de mayor cantidad de términos. La última complejidad, utilizando la misma idea de combinación lingüístico inglés y español, resulta que tiene un promedio de aciertos correctos de 67% con la probabilidad de 33% en fallo de aciertos cada 30 iteraciones de comando que sea realicen con el presente nivel de complejidad. Por lo tanto, el sistema de reconocimiento rinde considerablemente funcional mientras tanto funcione a máximo de nivel de complejidad en el cual combina tres términos; debido a que a partir de combinación de 4 términos pierde la precisión por mezcla de ruido en la señal de comando.

C. Descripción analítica de diagramas de flujos

1. Diagrama de sistema de reconocimiento de voz. La Figura 49 se muestra el resultado final con la idea general del algoritmo de sistema de reconocimiento de voz hasta enviarse identificadores con respectivos valores de servomotores. El algoritmo entonces utiliza el Google Speech API y la comunicación serial para dar configuración inicial de los parámetros de la máquina y del micrófono. Entonces primero revisa si hubo detección de voz y procesa de tal manera que el audio contiene la señal de la voz audible. Al lograr reconocer que la señal es de un usuario con comando, entra en operación de comparación del audio transcrito en texto con términos o frases predeterminados. Estos términos o frases predeterminados son las claves o “Hot-Word” que realiza la detección con términos claves para ejecutar un evento.

Entonces si la comparación falla, el sistema regresa al inicio de detección de la señal de audio de comando: y si la comparación cumple, prepara el código de activación del evento respectivo y responde adecuadamente al usuario. Al terminar este proceso, limpian las variables parciales y de texto transcrito para liberar el proceso de reconocimiento de voz para poder asegurar segmentación de recepción de órdenes secuenciales a partir de los usuarios.

2. Diagrama de sistema de reconocimiento de rostro. La Figura 50 muestra la versión del sistema de reconocimiento de rostro desarrollado en el entorno de programación de Python. Al igual que un sistema de reconocimiento de voz, esto hace el uso de las librerías en los cuales son OpenCV y comunicación serial. A diferencia de sistema de reconocimiento de voz, este sistema de reconocimiento de rostro utiliza el “.xml” con parámetros característicos del rostro del usuario. En la versión presente de la Figura 49, expone una pantalla de la perspectiva de la cámara para verificar y observar como la cámara ve e ilustra el cuadro de detección del rostro.

Entonces cuando reconoce que hay el rostro del usuario presente dentro de la región de detección, marca el cuadro de detección sobre el rostro del usuario. Del cuadro de detección desplegado, obtiene el centroide para segmentar los valores en respectivos ejes permitidos de movilidad de servomotores. La operación entonces procesa de forma cíclica y termina el proceso de detección si detecta un evento de paro de operación

durante el sistema de reconocimiento de rostro. Para los ajustes del centroide, se utilizó las compensaciones u “Off-sets” para mejorar la ubicación de los servomotores durante la operación.

3. Algoritmo de compensación del sistema de reconocimiento (Off-sets). La Figura 51 muestra el fragmento del sistema de reconocimiento de rostro de parte del algoritmo de compensaciones. Las compensaciones, u “Off-sets”, son variables de ajustes o de calibración de posicionamiento de los ojos del animatrónico. Esto se deriva de imitar ajuste o calibración durante la operación de seguimiento del rostro del usuario. Los cuadros de detección es el resultado visual desde la perspectiva de la cámara cuando reconoce el rostro del usuario. Este cuadro, sin alguna compensación, de perspectiva de la cámara no se notará la diferencia. A diferencia del resultado de las posiciones de los servomotores.

Las compensaciones son tomadas en efecto en los actuadores y el cuadro de detección, implicando en actualizaciones sobre los desplazamientos en los valores de las posiciones de los ejes. La calibración del valor de las compensaciones resultó ser manual, debido a que cada valor de esto respectivo al eje es fijo. La comprobación de los valores de las compensaciones se colaboró visualmente respecto al punto de vista del ojo de animatrónico con la perspectiva del usuario. Entonces, este algoritmo dedicado al funcionamiento de compensaciones calibra el sistema de reconocimiento de rostro, resultando en mejor seguimiento de rostro. Las Figuras 106 a la 107 del anexo muestran la ubicación de instalación de la cámara OOSXX y el respectivo sistema de servos que toma el efecto de algoritmo de compensaciones para adecuar las posiciones finales de los ojos. Esto dependiendo del tipo de animatrónico y la ubicación de la persona, los mecanismos efectúen respectivamente las compensaciones. Adicionalmente, las Figuras 108 a la 111 del anexo muestran la configuración mecánica de la cámara integrada internamente dentro del ojo de animatrónico.

D. Análisis de prueba de detección de rostro

Las pruebas de detección de rostro se dividen en fases según sea la combinación de la cámara y el entorno de programación. La primera fase de prueba de reconocimiento de rostro utilizó el Matlab y la cámara Microsoft LifeCam Cinema. El uso de Matlab basó en comprobar la funcionalidad de reconocimiento de rostro para desarrollo de la estructura del sistema de reconocimiento de rostro. Matlab mismo proporciona librerías y APIs para reconocimiento del rostro y la combinación de esto con el desarrollo enfocado flexibilidad resultó ser funcional. La flexibilidad en este contexto basaba en que, según la inclinación del rostro de usuario, el cuadro de detección seguía deformar de tal forma que contenga el rostro del usuario dentro de la región de detección. Los puntos son marcados al identificar parámetros característicos del rostro del usuario y guiaba al posicionamiento del cuadro de detección sobre el rostro del usuario. Al contrario, debido al entorno de programación que debe ser bajo en peso y poder realizar operaciones con bajos recursos, esta combinación únicamente desarrolló dentro de la fase de prueba y estudio del sistema de reconocimiento de rostro. Las imágenes de perspectiva de la cámara resultante se observan con las Figuras 52 y 53.

La combinación de implementación funcional cumpliendo con los requerimientos de ser bajo peso y funcional con operaciones simples que rinde suficientemente con bajos recursos, fue el entorno de programación de Python con la librería OpenCV y la cámara Microsoft LifeCam Cinema. Esto con un archivo “.xml” que contiene los parámetros característicos del rostro de la persona logra reconocer el rostro del usuario. Las funciones de OpenCV hacen el uso de la extensión de los parámetros característicos e ilustra el cuadro de detección del usuario. Adicionalmente, logró delimitar la región de detección para que el algoritmo inicie la operación de seguimiento del rostro siempre y cuando encuentre a una distancia de 1m a 1.3m. Por otro lado, previamente en Matlab era posible deformar el cuadro de detección según sea la inclinación del rostro del usuario; pero en este entorno de programación de Python debido a la necesidad de comprimir mayor cantidad operaciones sin comprometer con el rendimiento de reconocimiento de rostro, el cuadro de detección no se deforma y queda recto en todo el tiempo de seguimiento. Las Figuras 54 y 55 muestran los resultados de esta fase.

Después de la combinación previa con entorno de programación de Python y la cámara Microsoft LifeCam Cinema, realizó el cambio de la cámara a OOSXXX Mini Security Camera. Esta cámara ya es el dispositivo que se implementa en el animatrónico y aplicó los mismos parámetros y configuraciones que con la cámara Microsoft LifeCam Cinema. El resultado de seguimiento de rostro resultó ser similar al utilizar OOSXXX Mini Security Camera que Microsoft LifeCam Cinema. Aunque en la detección con OOSXXX Mini Security Camera resulta ser más difícil a diferencia de Microsoft LifeCam Cinema por diferencia de especificación que posee cada cámara. La cámara OOSXXX tiene la misma resolución que la de LifeCam Cinema, pero ofrece baja calidad en equilibrio de contrastes, de color, la densidad de píxeles, y capacidad de recibir la luz para mejor calidad de imágenes. Por el cual, teniendo la calidad de imagen más baja que la de LifeCam Cinema resulta que el usuario debe encontrarse en un ambiente con iluminación notable dentro de la región de detección. Las Figuras 56 y 57 son los resultados de esta fase.

La eficacia del sistema de reconocimiento de rostro se puede observar por medio de la cantidad de marcos por segundo o “Frames-per-Second”. A mayor cantidad de marcos por segundo ofrece respuesta más rápida y mayor cantidad de información dado en fracciones del tiempo. Entonces, los algoritmos de compensaciones inyectan a la operación de posicionamiento de los ojos con mayor cantidad de información por cada marco de imagen, ayudando que el movimiento de los ojos sea más suave y responsiva. Entonces, la medición de respuesta frecuencia que se observa al cambiar de posición de uno a otro usando las dos cámaras para observar la eficiencia de funcionamiento de seguimiento es posible analizarse de diferencia de unidades en fracciones de segundo.

E. Análisis de resultados de brazos mecánicos

1. **Análisis de esfuerzos.** Las posiciones críticas del brazo robótico dependen de los grados de libertad que posea el mismo y, además, es en donde se encuentra la mayor exigencia de torque para los servomotores y de resistencia para las piezas. Por ese motivo, se escogieron dos: el brazo completamente en voladizo hacia un lado y el brazo vertical cuando no se está moviendo. Ahora bien, cuando el brazo está completamente en voladizo se puede modelar como una viga empotrada en un extremo tomando los siguientes supuestos:

- No se va a realizar cálculo de deflexión ni pendiente porque dependen del material y del perfil de la viga.
- No existen cargas axiales.
- No hay esfuerzos de torsión.
- No existe falla por pandeo.
- La viga está en flexión pura.

Así, se tomó en cuenta el peso de cada pieza como una fuerza puntual que pasa por su centroide. Pero hay que resaltar que las fuerzas aplicadas en puntos bastante cercanos se sumaron por simplificación de cálculos. A la piel y la fibra de vidrio se le consideró como una carga distribuida porque esta tiene contacto a lo largo de toda su superficie interna con el brazo robótico que se diseñó. Además, al considerar que las fuerzas pasan por el centroide de las piezas, también se está considerando que cada una tiene una distancia determinada desde el empotramiento hasta el eje de dicha fuerza. Lo que se traduce, a su vez, en la sumatoria de varios momentos flexionante.

Así, ya que no se va a calcular deflexión ni pendiente, se escogió una viga tubular de perfil cuadrado pudiendo ser realmente de perfil circular o cualquier otro por la misma razón (pero siempre de la misma longitud que el brazo robótico). Y ya que el par es independiente de dichos parámetros se procedió a encontrar el Diagrama de Corte y Momento. De este análisis se encontró que el momento máximo en el empotramiento es de aproximadamente $1,375 \text{ N}\cdot\text{mm}$, lo que es aproximadamente el 92% de la capacidad de carga que tiene el servomotor AX-12A. Esto indica que, si bien el motor es capaz de soportar el brazo completamente en voladizo, es muy probable que se caliente y que sufra una sobrecarga ante cualquier perturbación en las fuerzas. Además, debido a que los Dynamixel tienen protección contra sobrecarga, tendría sentido que se apaguen. Por lo que, lo recomendado sería cambiar de motores o reducir el peso del brazo.

El diagrama también ayudó para definir una región en donde cualquier pieza y elemento de unión que se ubicara dentro de dicha región se considerara crítico. Esta región se definió a partir de los $500 \text{ N}\cdot\text{mm}$

aproximadamente porque en el cálculo de esfuerzos, cualquier pieza con un momento menor al mencionado, tenía un factor de seguridad lo suficientemente grande como para no tomarla en cuenta.

Sabiendo esto, se escogieron 6 puntos críticos tomando en cuenta que fueron los factores de seguridad más bajos en todo el brazo. A continuación, se seleccionaron los elementos de esfuerzos que estuvieran en las secciones transversales de las piezas críticas. Estas secciones debían ser seleccionadas con cuidado para no incurrir en errores al momento de simularlo en Inventor y de calcularlo teóricamente. Parte de esas consideraciones es que hay que tomar en cuenta que el esfuerzo va a ser diferente para diferentes posiciones del brazo (las posiciones críticas). Entonces, a cada pieza crítica se le aplicaron dos análisis de estrés, uno para la posición crítica del brazo completamente en voladizo y otra para la posición vertical del brazo. Es necesario mencionar que para piezas iguales ubicadas en diferentes posiciones a lo largo del brazo y con la misma funcionalidad, bastó con calcular los esfuerzos en una sola pieza porque si la más exigida soporta la carga, la otra pieza (con menor carga) no debe fallar. Cada análisis contaba con configuraciones diferentes de restricciones y vectores de fuerza, que, a su vez, representaron una mayoría de esfuerzos cortantes puros y de tensión pura para el brazo vertical y una mayoría de esfuerzos flexionantes para el brazo en voladizo.

Finalmente, del cálculo teórico se comprobó que Inventor suele ser un poco más conservador con los resultados. Ya que, de todas formas, basados en el factor de seguridad como un indicador, los resultados coinciden en que la pieza más propensa a fallar primero son los tornillos que sostienen los dos accesorios F333. En este caso para resolver el problema habría que reducir peso o aumentar la cantidad de tornillos utilizados. Pero esta última es una alternativa un poco más complicada porque el accesorio no cuenta con más agujeros para poner más tornillos. Habría que modificarla y luego imprimirla.

2. Análisis de tensión para los actuadores de la garra. En este caso debido a que los servomotores XL-320 no lograron devolver información fue necesario tensar los hilos de los dedos con masas conocidas colgando de ellos. Se realizó un análisis bastante simplificado porque no se requiere precisión en el ángulo de giro de los dedos, solamente la actuación de los mismos, y todos al mismo tiempo. Sin embargo, siempre se calculó el torque requerido para mover 4 dedos con el objetivo de asegurar que el servo XL-320 fuera suficiente para la aplicación porque no es lo mismo mover 1 dedo que mover 4 dedos, las torques necesarias para cada uno se suman. De hecho, en el Cuadro 18 se puede observar que para distintos dedos se necesita torques diferentes. Esto puede ser tanto por fricción del pivote de los dedos como por el tamaño de los mismos.

3. Análisis de parámetros de los actuadores principales. En el estudio de los servomotores AX-12A fue necesario evaluar si estaban generando el torque y el ángulo necesario. Entonces, dado que físicamente el robot tiene limitaciones en su movimiento de abducción se debió establecer límites para el ángulo del motor y así evitar sobrecargas o accidentes. Aprovechando que los servos AX-12A sí devuelven información, se programó que enviaran el torque y el ángulo instantáneo en el que se encontraban. Entonces con esa información, ya se puede hacer un listado de posiciones y torques devueltas por el motor para compararlos luego con el modelo matemático. Sin embargo, como los motores reciben y devuelven información en 10 bits, solo se podían ingresar valores de 0 a 1023, lo que puede resultar confuso entonces se decidió mapear los valores a un rango adecuado. Para los ángulos se estableció un rango de mapeo de 0 a 90° que son los límites de abducción y aducción. El torque si se decidió ingresar con valores de 0 a 1023 porque la relación es lineal entre dicho rango y el rango de torque que puede brindar el servo (0 a 1.5 N*m). Entonces tal como lo decía el fabricante, para un valor de 512, el servo solo generaba 0.75 N*m.

Lo interesante de este análisis fue el comportamiento que tuvo el torque a lo largo del barrido de ángulos, el cual se puede observar en la Figura 78. De este resultado se puede notar que el modelo teórico decía que era necesario determinado torque mientras que el servo decía que en realidad necesitaba menos torque del calculado. Desde el punto de vista de eficiencia esto es conveniente porque se está necesitando menos torque del esperado. Pero el motivo por el cual ocurrió esto, fue porque al utilizar el modelo matemático había que asumir ciertas variables como la masa en el extremo del brazo, lo que afecta la inercia. Además, la masa del brazo, que según el modelo del péndulo simple no debería tener masa, en este caso si tiene.

En cuanto al ángulo, el porcentaje de error se mantuvo debajo del 10%. A 90° se encontró un aumento del porcentaje porque las pruebas se realizaron con cinco segundos de intervalo entre una posición y la otra, lo que aumentó la temperatura del servomotor perdiendo parte de su torque nominal. Esto se comprobó enviándole al servomotor un ángulo de 90° después de estar una hora apagado.

Un dato importante que hay que mencionar es que el servomotor se apagaba luego de estar cierto tiempo a un ángulo constante. Entonces, fue necesario realizar un breve estudio de dichos tiempos, lo que produjo la Figura 79. Para el análisis se asumió que el animatrónico no iba a tener los brazos en la misma posición por más de 35 segundos, siendo éste, el límite para la gráfica. Sabiendo lo anterior, a 60° el motor se empezó a apagar automáticamente en menos de 35 segundos. Teniendo una caída brusca a ocho segundos de duración para un ángulo mayor o igual de 70°. A partir de los datos que brinda el fabricante se determinó que 1,375 N*mm es tan cercano al valor nominal de los motores que los mismos disparan la alarma de sobrecarga (el led se queda parpadeando cuando el motor se apaga) y se apagan. Por lo que hay que concluir que los brazos no podrán estar en abducción total por más de ocho segundos, siendo la causa unos motores funcionales, pero no ideales para el brazo robótico que se construyó.

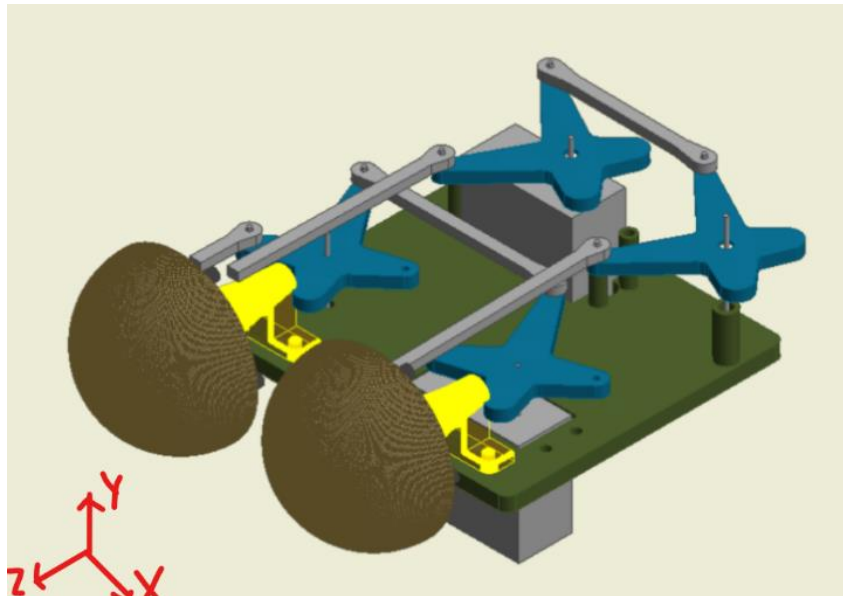
IX. Análisis de resultados de mecanismos para emular gestos faciales

A. Resultados de ojos de DANI

Para determinar la mejor selección del mecanismo de los ojos, se definió los requerimientos los cuales comprendían el espacio de trabajo, la ubicación el mecanismo de los ojos, y también emplear materiales dúctiles y livianos para no generar un peso excesivo y también que sea ensamblable.

Como fase previa se elaboró un prototipo de animatrónico en forma de burro, para el cual se diseñó un mecanismo con la movilidad de los ojos la cual se limitó a dos grados de libertad, movimiento en X y Y, para dicho diseño se empleó únicamente dos servomotores los cuales permitieran la movilidad deseada. A continuación, se puede observar el diseño de la primera iteración del mecanismo de los ojos:

Figura 80. Mecanismo de ojos



En la Figura 80 se puede observar que se utilizó dos servomotores debido al espacio disponible y con base a investigación previa de mecanismos para animatrónicos se analizó de forma en que se pudieran emplear conceptos de mecanismos de cuatro barras para tener la misma movilidad en dos ejes y así transmitir el movimiento tanto horizontal como vertical de un ojo a otro (Figura 81). Se analizó el mecanismo de cuatro barras por la movilidad que genera de la manivela al balancín, la cual por tener las mismas distancias los eslabones, se genera el mismo movimiento como se puede observar en la Figura 82, lo cual permite girar ambos ojos la misma cantidad de movimiento tanto horizontal como vertical.

Figura 81. Mecanismos de cuatro barras analizado

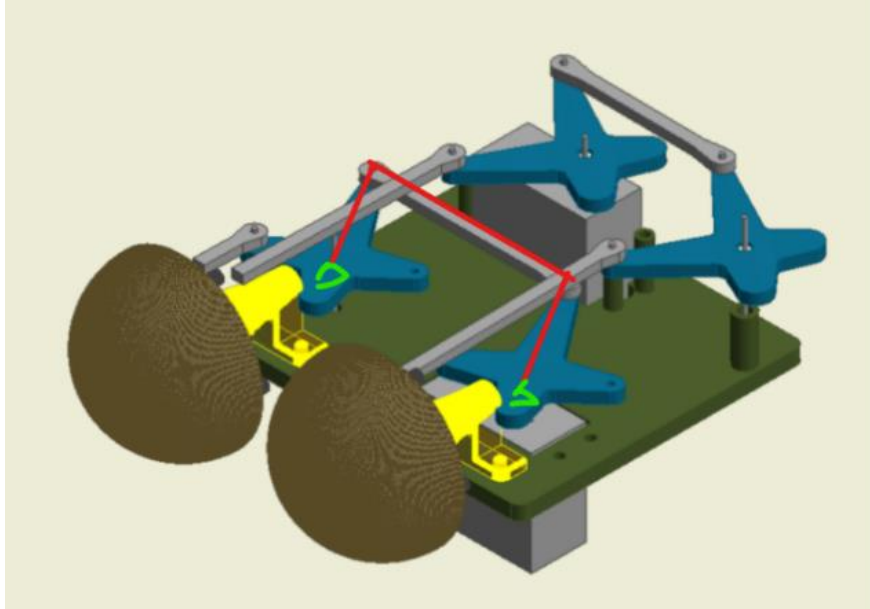
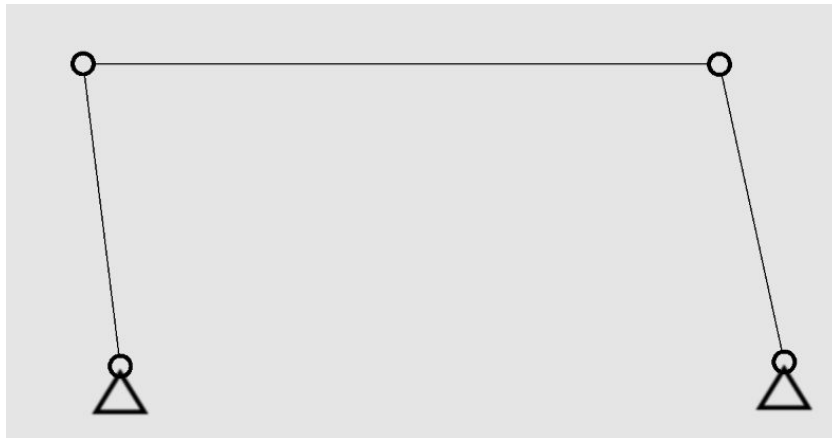


Figura 82. Mecanismo cuatro barras, movimiento horizontal y vertical de los ojos



Para la segunda iteración se tomó dimensiones del área de la cabeza (Figura 83) para así diseñar con base en ellas y poder acoplar el mecanismo a la cabeza, a este nuevo mecanismo se le diseñó el movimiento de parpados el cual permite tener mayor movilidad y así emular de mejor manera el comportamiento de los ojos.

Figura 83. Dimensiones de la Cabeza, Vista Frontal

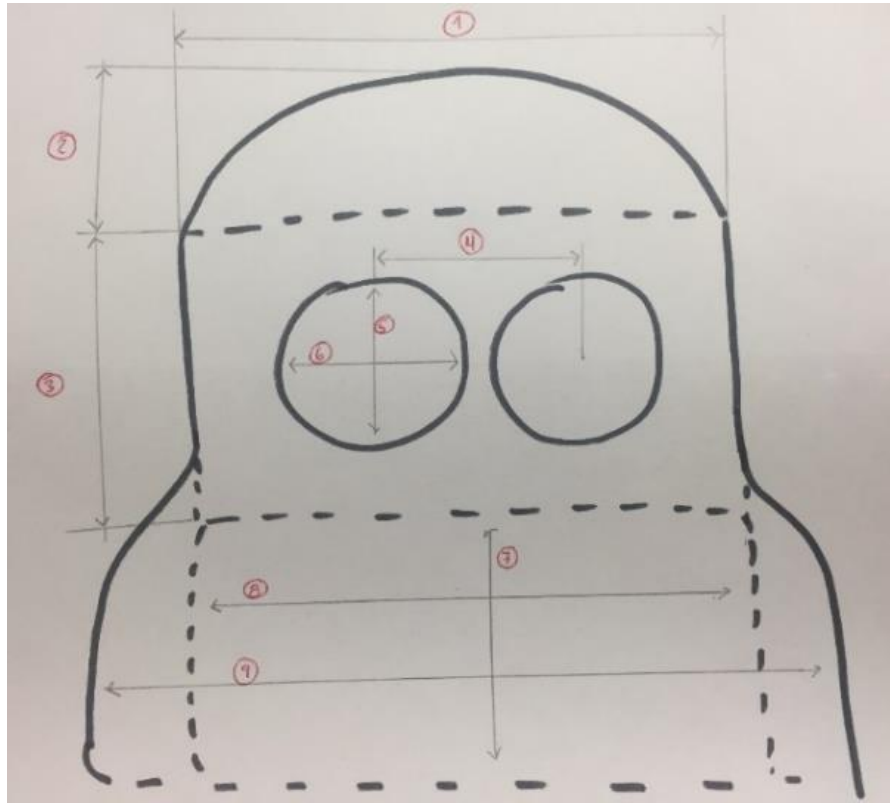
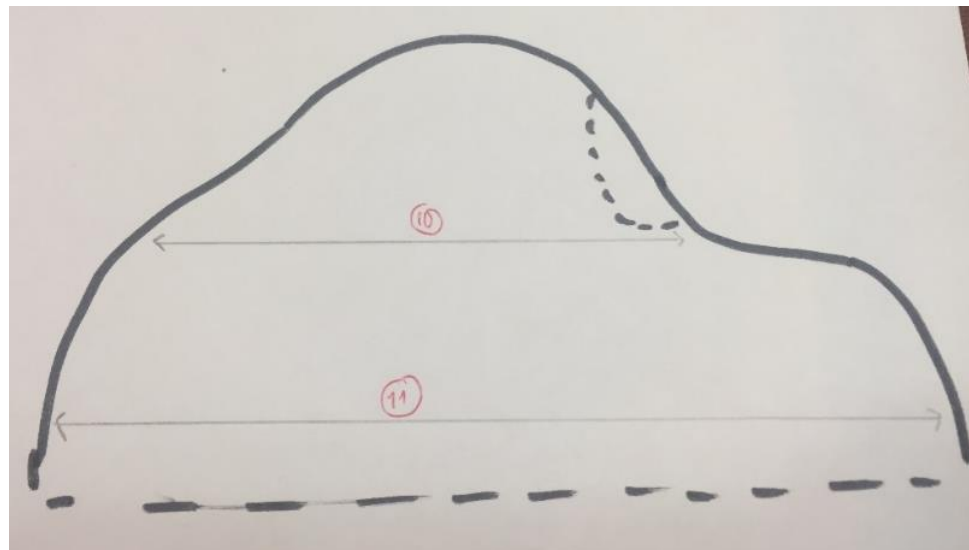


Figura 84. Dimensiones de la cabeza, vista lateral



Cuadro 22. Valor de las dimensiones de la cabeza

| Número | Dimensión (cm) |
|--------|----------------|
| 1 | 14.20 |
| 2 | 2.10 |
| 3 | 5.50 |
| 4 | 7.19 |
| 5 | 4.65 |
| 6 | 4.65 |
| 7 | 11.1 |
| 8 | 16.8 |
| 9 | 21.4 |
| 10 | 16 |
| 11 | 26 |

Al conocer del área de trabajo se prosiguió a diseñar el nuevo mecanismo para los ojos el cual contiene como movimiento adicional el de los párpados, siempre utilizando la geometría de los eslabones para representarlos como mecanismo de 4 barras. Esto permite utilizar menos servomotores, por lo tanto, se seleccionó los servomotores Dynamixel XL-320, primero por su tamaño y además se comunican en serie con otros servomotores lo cual facilita el ensamblado de los mismos, además su forma física permite que se acoplen con mayor facilidad a cualquier pieza.

Por otra parte, su torque máximo es 0.39 N m lo cual es más que suficiente ya que por no haber cargas representativas y despreciables en comparación a los torques de los servomotores no se necesita movilizar altas cargas únicamente eslabones con dimensiones menores a los 5cm solo para realizar la movilidad deseada.

Figura 85. Mecanismo de ojos diseñado para DANI

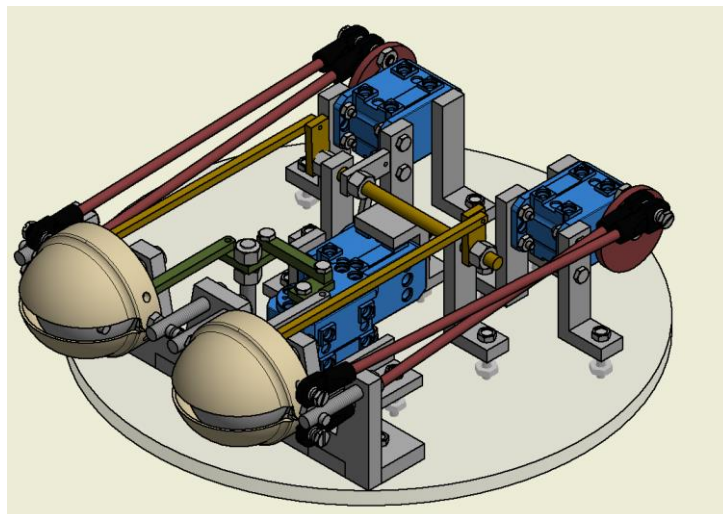
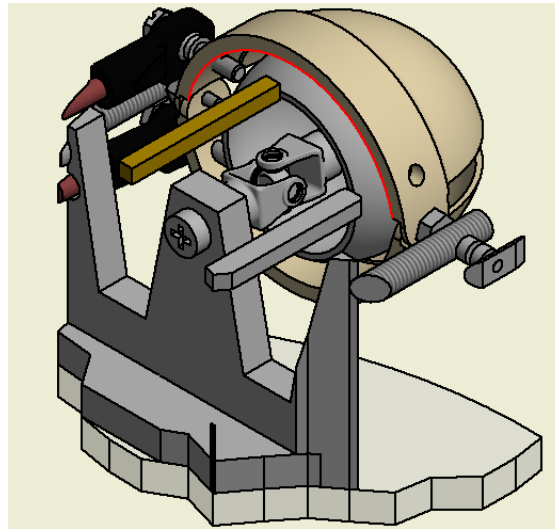


Figura 86. Ensamble del ojo junto al párpado



Como se puede observar en la Figura 86, se empleó diversos métodos de unión para cumplir con el requisito de ensamblable, como primer punto se utilizó diversos pernos métricos, los cual facilitan el armado del mecanismo, para los ojos se adaptó las juntas universales las cuales proporcionan dos grados de libertad lo que permitió tener la movilidad deseada, horizontal y vertical en el ojo. Además, se seleccionó juntas esféricas las cuales se acoplaron a los párpados, esto porque la superficie es curva entonces estas juntas se adaptan de mejor manera y así permitir la correcta apertura de los párpados. Se seleccionó como materiales de fácil mecanizado y fácil acceso el MDF y PLA, debido a que también posee entre sus propiedades físico-mecánicas la baja densidad en comparación con otros plásticos y maderas, lo cual permite tener un mecanismo de bajo peso. Para el mecanizado se utilizó la cortadora laser y las impresoras 3D de la Universidad del Valle de Guatemala para así fabricar las piezas.

Al analizar el mecanismo completo (Figura 85) se observó que, debido a sus dimensiones, por ser un elemento pequeño que únicamente tiene el propósito de movilidad y no de soportar cargas, no se presentan cargas considerables que puedan dañar el mecanismo, esto debido a que los factores de seguridad de los proveedores de pernos, juntas son altos debido a que están fabricados para resistir altos esfuerzos. Pero en este caso únicamente se tomó un punto que puede llegar a ser crítico el cual es el perno que sostiene el ojo, en el cual se analizó el esfuerzo flexionante en el perno empleando un cálculo sencillo con la siguiente ecuación:

$$\sigma = \frac{M*c}{I} \quad (12)$$

Donde:

I: Segundo momento de inercia, sección circular

C: Distancia al centro del círculo, es decir el radio

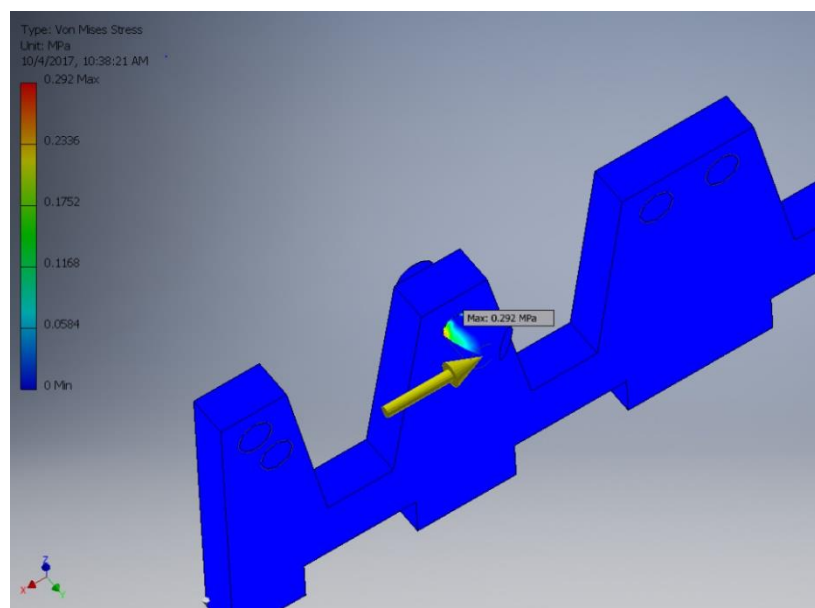
M: Momento flector el cual es generado por la carga del ojo

Para obtener la carga que soportaría el perno se midió la masa de cada elemento (Cuadro 23) para así obtener el peso total y colocar la carga en el extremo del perno. Teóricamente se obtuvo un valor de esfuerzo flexionante debido a la carga de 0.36 MPa, al analizarlo en el software de diseño CAD (Figura 87) se obtuvo un valor experimental de 0.292 esto es debido a las consideraciones del análisis que se tomó como una viga empotrada, únicamente está sometido a flexión pura y el material es isotrópico y homogéneo.

Cuadro 23. Masas y peso total de la carga del ojo

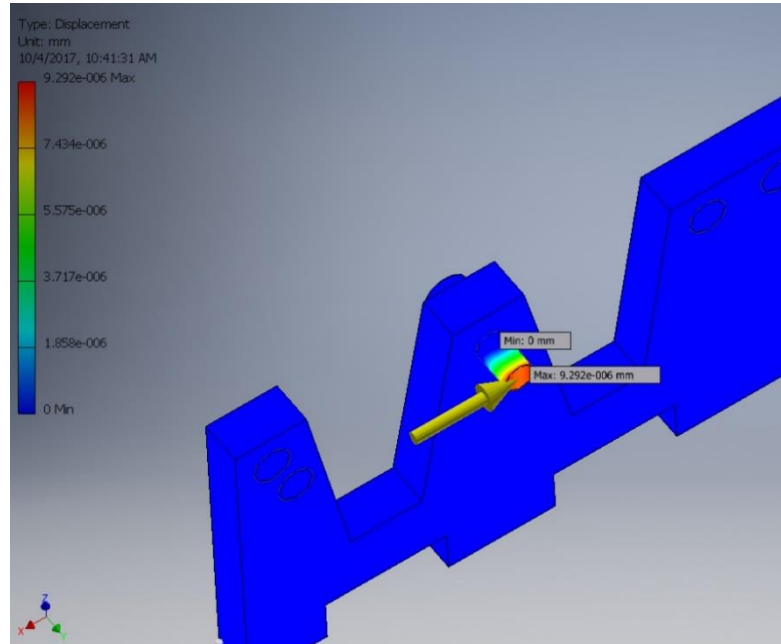
| Descripción | Masa (g) | Masa (Kg) | Peso (N, kg*m/s ²) |
|--------------------------------|----------|-----------|--------------------------------|
| Ojo | 3.9 | 0.0039 | 0.03822 |
| Junta Universal con Accesorios | 11.5 | 0.0115 | 0.1127 |
| Tornillo M4 | 4.1 | 0.0041 | 0.04018 |
| Junta Esférica | 2.7 | 0.0027 | 0.02646 |
| Base café | 37.8 | 0.0378 | 0.37044 |
| Peso Total (N) | - | - | 0.1911 |

Figura 87. Análisis de flexión en el perno



También al analizar la deformación generada (Figura 88), se observó un valor máximo de $9.29e-6$ mm lo cual es insignificante e imperceptible a la vista humana, por lo tanto, el perno soporta la carga generada.

Figura 88. Deformación del perno



A continuación, se presenta en la Figuras 89 y 90 fotos de la fabricación de las piezas diseñadas:

Figura 89. Ensamble de junta universal con el ojo que sostiene la cámara



Figura 90. Ensamble de componente interno del ojo



Para mejor visualización del mecanismo, como sus dimensiones relevantes se puede referir a la sección de anexos donde se encuentran el juego de planos de todo el mecanismo.

1. Discusión. Iniciado con la primera iteración sirvió como base para realizar la segunda iteración ya que se observó que se podía agregar movilidad a los párpados sin necesidad de modificar las dimensiones de la cabeza, por lo tanto, esto proporcionó una mejor emulación del gesto.

Como se visualizó en las figuras 85 y 86, el mecanismo empleado para los ojos cumplió con todos los requerimientos de espacio, utilización de materiales de fácil mecanizado, de fácil acceso y ensamblable, por lo tanto, fue satisfactorio porque se logró la movilidad deseada gracias al análisis de movimiento de los eslabones de cuatro barras. También se analizó que, por las altas propiedades físicas y mecánicas de los materiales y componentes para métodos de unión, hace que el mecanismo tenga una duración alta de funcionamiento.

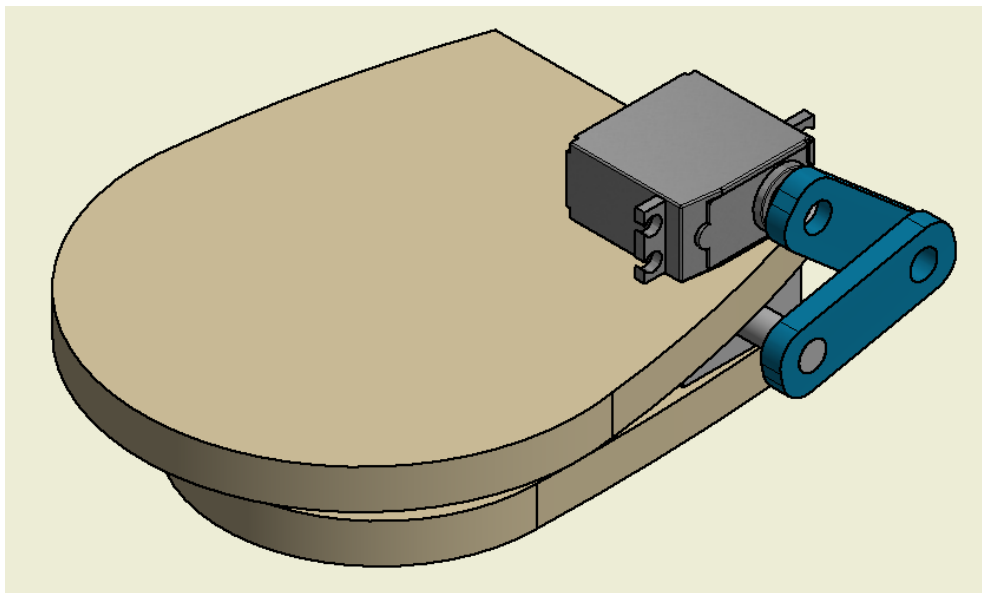
Al analizar los puntos críticos y esfuerzos presentes realmente no hay cargas considerables a las cuales vaya a estar sometido el mecanismo ya que no hay ningún objeto o fuerza externa que vaya a generar un aumento en los esfuerzos, por lo tanto, solo se analizó el perno que soporta el ojo, y se puede observar en la Figura 88, que la deformación es imperceptible por lo tanto los resultados obtenidos son válidos para un correcto funcionamiento del mecanismo de los ojos.

B. Resultados boca DANI

Para determinar la mejor selección del mecanismo de la boca, fue necesario definir los requerimientos los cuales comprendían el espacio de trabajo, la ubicación del mecanismo de la boca, y también emplear materiales dúctiles y livianos para no generar un peso excesivo y también que sea ensamblable. También que el torque que puede proporcionar el servomotor logre levantar el peso de la mandíbula, para así generar el gesto deseado.

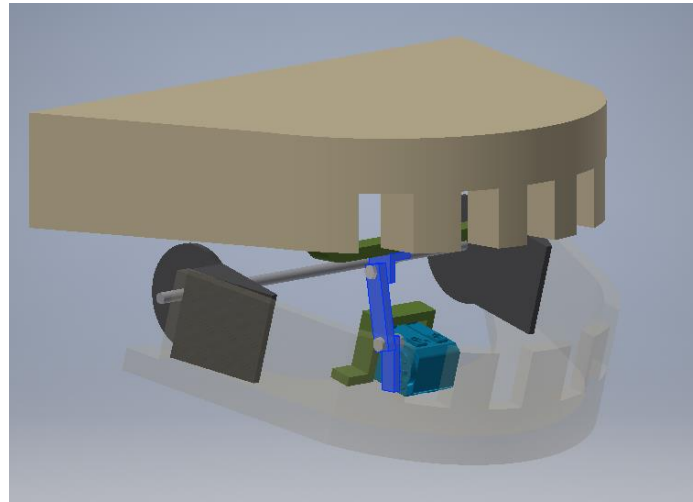
Como fase previa se elaboró un prototipo de animatrónico en forma de burro, para el cual se elaboró un mecanismo con la movilidad de la boca, el cual solo tiene un grado de libertad. Para dicho diseño se empleó únicamente un servomotor el cual permitiera la movilidad deseada, sin embargo, la colocación del servomotor no permitió que se abriera la mandíbula, lo suficiente para ser apreciada visualmente. A continuación, se puede observar el diseño de la primera iteración del mecanismo de la boca:

Figura 91. Mecanismo de boca, primera iteración



Luego se tomó como referencia los videos adquiridos del gorila animatrónico de Daniele Tirinnzani. Con base en el espacio y movilidad se diseñó el nuevo mecanismo colocando el servo en la mandíbula y no en la parte superior, esto generó que el ángulo de apertura de la boca fuera mayor y más realista, ya que al investigar la apertura promedio de los humanos es de aproximadamente 45 mm, por lo tanto, se optó a desarrollar una segunda iteración del mecanismo.

Figura 92. Mecanismo de la boca de DANI



Para el análisis del mecanismo, se puede observar en la figura 92, que únicamente son dos eslabones los cuales giran uno respecto al otro gracias a la rotación del servomotor, un análisis más profundo sería modelarlo como un mecanismo de cuatro barras, pero en este caso haría falta un eslabón, por lo tanto, se optó a realizar un análisis de ángulo de giro del servomotor con respecto al de apertura de la boca y además la distancia de apertura de la boca. Para realizar dicho análisis se definió el rango de giro del servo que va de 0° a aproximadamente 105° , luego en el software CAD se analizó en ángulo máximo de apertura de la mandíbula para que el mecanismo siguiera funcionando correctamente el cual fue de 20° (Figura 93). Seguido se tomó el ángulo de 45° , 75° del servomotor y se midió el ángulo de apertura, al conocer ya dichos datos, se realizó una interpolación lineal la cual permitió conocer el ángulo de apertura del servo a distintos ángulos de giro del servomotor, los datos se pueden observar en el Cuadro 24.

Figura 93. Ángulo de apertura y distancia máxima de la boca

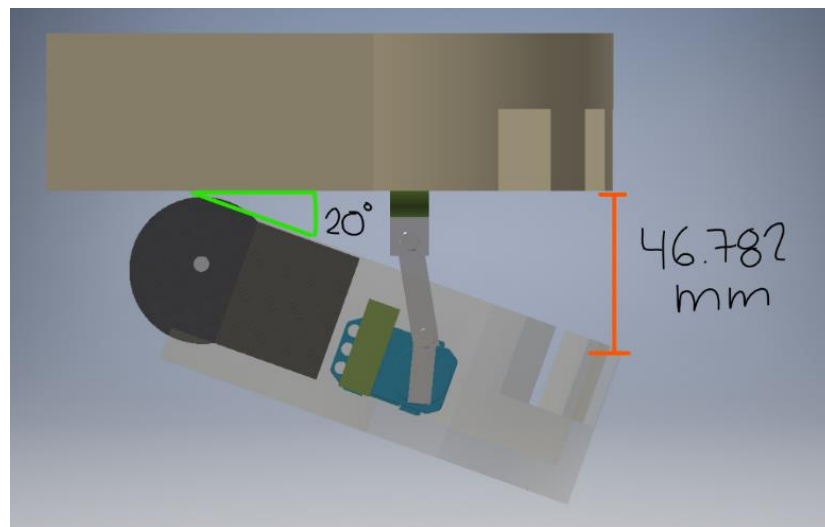


Figura 94. Boca de "DANI"



Cuadro 24. Ángulo de giro de la boca y apertura

| Angulo giro servo (deg) | Angulo apertura (deg) | Distancia apertura (mm) |
|-------------------------|-----------------------|-------------------------|
| 0.000 | 0.000 | 0.000 |
| 25.000 | 4.762 | 13.650 |
| 35.000 | 6.667 | 17.580 |
| 45.000 | 8.571 | 21.168 |
| 55.000 | 10.476 | 26.656 |
| 65.000 | 12.381 | 30.274 |
| 75.000 | 14.286 | 34.122 |
| 85.000 | 16.190 | 38.454 |
| 95.000 | 18.095 | 42.244 |
| 105.000 | 20.000 | 46.782 |

Al graficar la relación de ángulo de giro y ángulo de apertura se puede observar que tienen una relación lineal, así como también el ángulo de apertura con la distancia de apertura, esto es de utilidad para próximos experimentos como seguimiento de audio por medio de la boca entonces esto haría que la boca simule una conversación.

Figura 95. Ángulo de giro vs ángulo de apertura

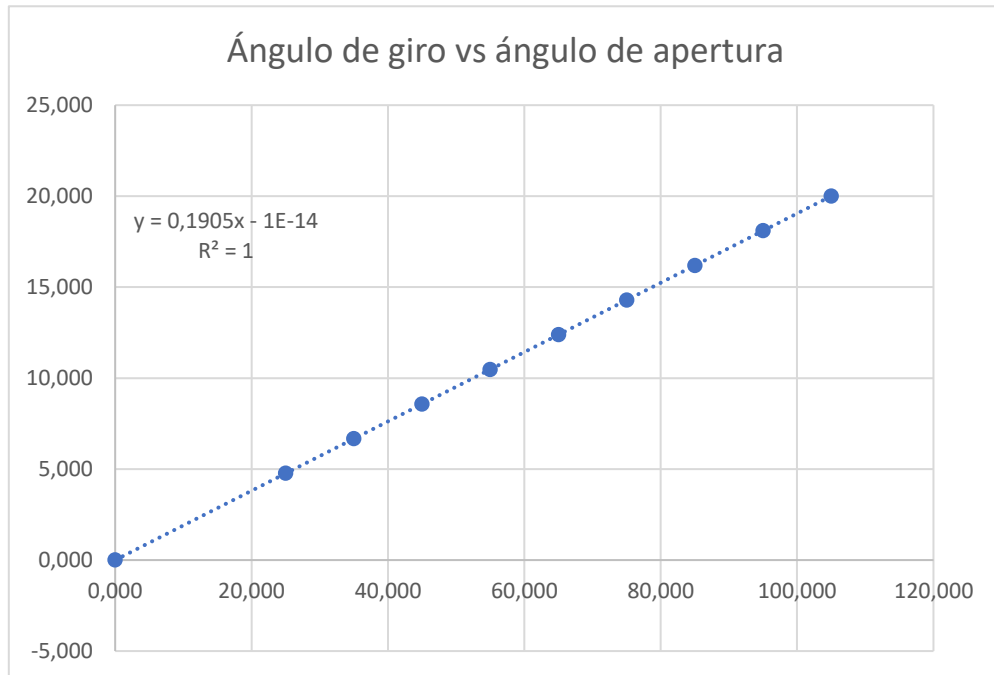
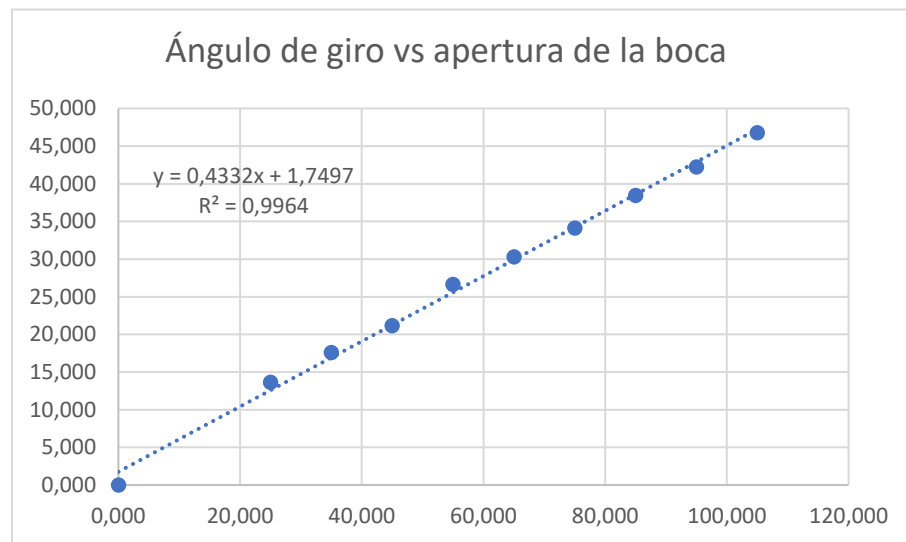


Figura 96. Ángulo de giro vs apertura de la boca



Seguido no se realizó un análisis de esfuerzos ya que igualmente no está soportando cargas considerables como se mencionó en el diseño mecánico de los ojos entonces los métodos de unión utilizados poseen factores de seguridad altos de parte de los proveedores entonces son lo suficientemente resistentes para la aplicación, sin embargo, si se realizó el análisis del torque que requiere el servo para levantar la mandíbula y cerrarla.

Para dicho análisis se analizó como un péndulo simple y se llegó a la siguiente relación empleando la el análisis con la segunda ley de Newton de forma rotacional de un cuerpo y además utilizando conservación de la energía.

$$\tau = mgh * \text{sen}(\theta) \quad (13)$$

Donde:

M: es la masa total de la mandíbula con el mecanismo

G: La gravedad

H: la distancia del pivote al extremo de la mandíbula

θ : Ángulo de apertura de la boca

Al evaluar la ecuación anterior en 20° ángulo máximo de apertura se encontró un torque de 0.12 mucho menor a lo que puede proporcionar el servo (Cuadro 25) por lo tanto el servo seleccionado Dynamixel XL-320 cumple con los requisitos para la correcta emulación del gesto.

Cuadro 25. Valores para análisis del torque

| Descripción | Valor | Unidad |
|-------------------------|---------|--------|
| Masa mandíbula | 0.08630 | Kg |
| Torque teórico | 0.12354 | N*m |
| Torque máximo del servo | 0.39000 | N*m |
| Angulo máximo apertura | 20 | Deg |
| Distancia de pivote | 0.16000 | m |

1. Discusión. Al dar inicio con la primera iteración, sobresalió la posición del servomotor ya que no permitía una apertura adecuada, la cual realizara una movilidad deseada. Entonces se dio inicio con una segunda iteración la cual mantenía el mismo principio sin embargo se colocó el servomotor en la mandíbula para que la configuración de eslabones permitirá una mayor movilidad.

Los materiales utilizados para su fabricación fueron PLA para los eslabones para que no añadiera peso extra, debido a que tiene una densidad inferior a otros plásticos por lo tanto se analizó el torque requerido para cerrar completamente la mandíbula obteniendo un valor de 0.12 N*m lo cual verifica la selección del servomotor Dynamixel XL-320 ya que este posee un torque máximo de 0.39 N*m pero como no es

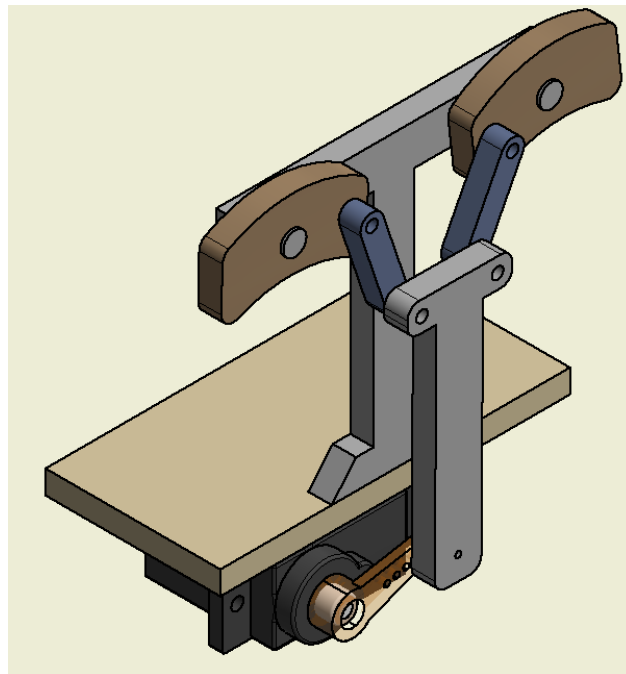
recomendable trabajar al 100% de la capacidad, el valor obtenido cumple con los requerimientos planteados y así emular de mejor manera la boca de DANI.

También se analizó el ángulo de giro del servomotor vs el ángulo de apertura de la boca, dichos ángulos tienen una tendencia lineal por lo tanto se logró encontrar la relación de ángulo de apertura de la boca vs la distancia de apertura, esta es importante ya que se puede relacionar a la biología de una persona humana que el promedio de apertura es de 45 mm y la obtenida a un ángulo de 20° fue de 46 mm, entonces el resultado obtenido es acorde y válido para la emulación del gesto facial.

C. Resultados cejas

Como primera iteración se elaboró el mecanismo de cejas de la Figura 97, el cual se descartó debido a la fragilidad por su forma elaborada y por los materiales utilizados. Debido a esto se optó a realizar el mecanismo de una forma distinta el cual fuera capaz de realizar el movimiento adecuado para así mover la piel que se le aplique encima (Figura 48) y emular el gesto de manera correcta.

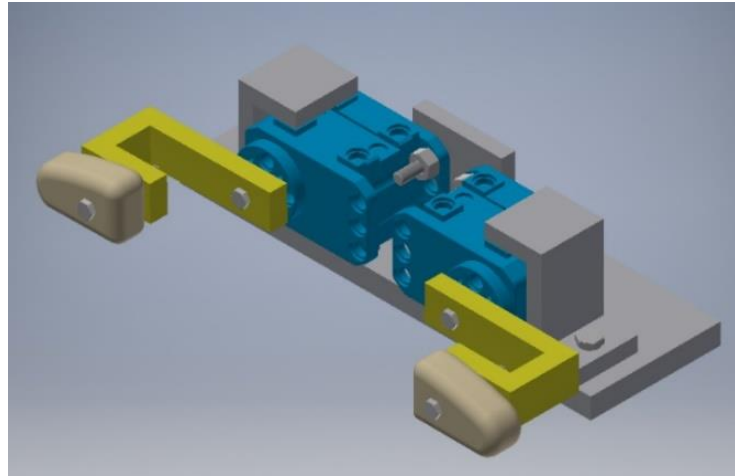
Figura 97. Mecanismo cejas, primera iteración



Ya teniendo en cuenta que la movilidad de las cejas tendrá que soportar el peso de la piel se diseñó una segunda iteración como se puede observar en la Figura 98. Se emplearon dos servos en el diseño para que cada uno mueva una ceja y así distribuir el peso de la piel en ambos servos. Cabe mencionar que la piel elaborada fue por un artista nacional, la cual es de esponja con una aplicación de pintura acrílica para que sea

moldeable, entonces la carga generada no va a generar una deformación ni esfuerzos significativos que dañen el mecanismo.

Figura 98. Mecanismo de las cejas



El espacio de trabajo es mucho más reducido para las cejas que para los demás mecanismos entonces no se podía diseñar un mecanismo complejo por lo que solo se acoplo un eslabón que sujete un bloque de material de fibra de vidrio del cual está fabricada la cabeza de DANI y que gire para arriba y abajo. El material utilizado para los eslabones es PLA el cual tiene una densidad baja y permite que el ensamble sea liviano y los métodos de unión fueron pernos para cumplir con el requerimiento de ser un mecanismo ensamblable.

1. Discusión. Este mecanismo fue el más sencillo para su elaboración ya que solo se necesitaba que el servomotor seleccionado levantara un bloque del material de la cabeza para emular el gesto de las cejas, se seleccionaron materiales de fácil mecanizado y a la vez de fácil acceso el diseño del mecanismo elaborado cumple con los requerimientos de espacio y es válido para emular el gesto facial.

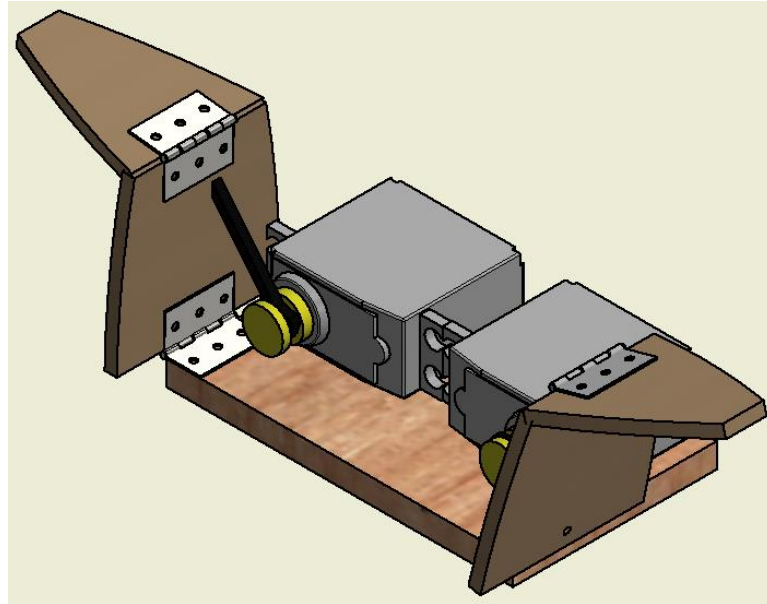
Realmente el análisis de esfuerzo fue nulo ya que en comparación del torque que puede proporcionar el servomotor siendo de $0.39 \text{ N}\cdot\text{m}$ es en comparación a la carga que pueda generar la piel de material blando es insignificante por lo tanto es válido el mecanismo elaborado.

D. Resultados orejas

El presente capítulo describe el proceso de selección y diseño del mecanismo de las orejas, con un énfasis en la movilidad del mismo, al igual que las dificultades encontradas que imposibilitaron su implementación en el animatrónico DANI.

Para esta sección se realizó un mecanismo para las orejas, el cual se diseñó con dos servomotores, el cual hace girar el servomotor y enrollar en un disco hilo de pescar y este se tensa y jala las orejas, este diseño se fabricó como prototipo previo y dio resultados adecuados, generando una movilidad adecuada.

Figura 99. Mecanismo de orejas



Sin embargo, al tener el diseño del dragón el material del que fueron hechas y la posición por parte del artista guatemalteco (Figura 48), se imposibilita la colocación de un mecanismo para generar el movimiento además por el tamaño de las orejas y su forma se decidió no implementar el mecanismo.

E. Resultados estructura DANI

Con base en los resultados obtenidos en el módulo del diseño mecánico del brazo de DANI se tomaron las fuerzas y momentos que actuaban directamente a la estructura los cuales son una fuerza de 6.45 N y un momento de 1375 N mm estas están ubicadas en los extremos de la estructura cuando el brazo se encuentra extendido hacia abajo y en voladizo (Figura 100).

Antes de realizar la soldadura se analizó los esfuerzos presentes en base a las cargas para así determinar la garganta de la soldadura, debido a la fuerza y el momento las soldaduras están sometidas a compresión y flexión.

Figura 100. Aplicación de cargas

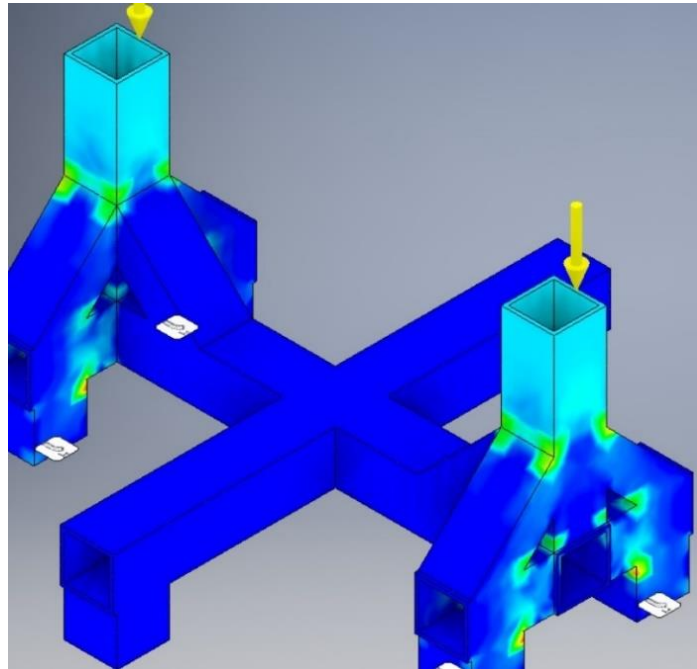
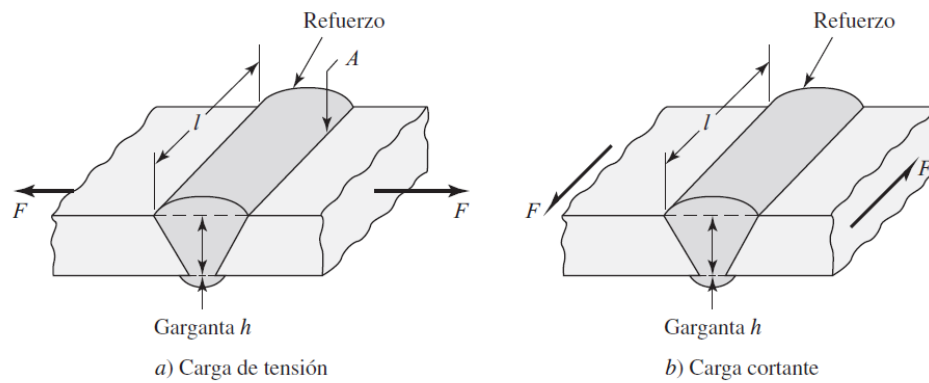


Figura 101. Garganta de soldadura, carga a tensión y cortante

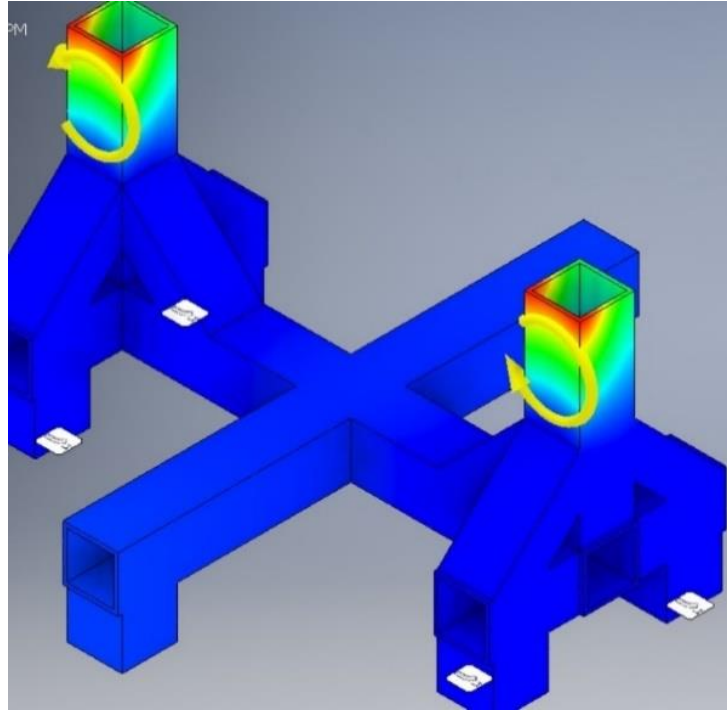


Para la soldadura que está sometida a compresión es la unión en T donde se calculó el esfuerzo el cual es de 0.25 MPa a partir de eso se utilizó la ecuación siguiente para calcular la garganta de la soldadura, esta significa la altura del cordón a realizar.

$$\sigma = \frac{F}{h l} \quad (14)$$

Al despejar h , se encontró que la garganta del cordón tenía que ser de 0.98 mm entonces se prosiguió a analizar la flexión de la soldadura por parte del momento encontrado, para este cálculo se utilizó el valor de h de 0.98 en la fórmula obteniendo un valor de esfuerzo por flexión en la soldadura de 78 MPa. Se tomó un factor de seguridad de 5 para que la soldadura fuera de 5 mm el cual es válido para soportar los esfuerzos generados y tener seguridad de que la soldadura no va a fallar.

Figura 102. Momentos en la estructura



Por último, se seleccionó el electrodo 6013 el cual cumplió con los requerimientos, para finalmente realizar la soldadura, luego se siguió un proceso de pulido y masillado para dejarlo presentable y estético.

X. CONCLUSIONES

A. Conclusiones de procesamiento de audio y video

Se logró el desarrollo del programa que interprete las señales provenientes de dispositivos de audio y video, por medio de la implementación de API y la librería en cada sistema de reconocimiento. La plataforma de desarrollo principal basó en el entorno de programación de Python y Arduino IDE. La idea general de la arquitectura de desarrollo de código se observa con las figuras 49, 50 y 51; y el funcionamiento de visual del sistema de reconocimiento de rostro.

El estudio para la selección de la mejor opción para el reconocimiento de voz se determinó mediante el trade study dado los criterios de selección planteados con la idea de cumplir con el requisito del funcionamiento de sistema de reconocimiento de voz. El método de trade study implementado fue el método de Pugh. En el cual, en base al estudio realizado, la mejor selección fue el Google Speech API. Además, logró implementar en el entorno de programación de Python y realizó las estadísticas de prueba del rendimiento de Google Speech API. Los Cuadros 14 y 15 muestran las estadísticas de prueba en base al modelo, y el Cuadro 16 muestra el trade study de realizado para la selección de la mejor opción para el reconocimiento de voz.

Se logró desarrollar el programa con la implementación de recursos de software existentes para el reconocimiento facial por medio de una cámara. El recurso o la librería implementada fue el OpenCV por medio del uso de archivo de extensión “.xml” que trae definido los parámetros característicos del rostro de una persona. Adicionalmente, el algoritmo de compensaciones se logró implementar en el sistema de reconocimiento de rostro mejorando la calidad de posicionamiento de los servomotores. El dispositivo de la cámara implementada es la OOSXXX Mini Security Camera, aunque ofrece baja tasa de marcos por segundo, no resultó en algún factor comprometedor para el funcionamiento correcto del sistema de reconocimiento de rostro. Las figuras 56 y 57 muestran la perspectiva resultante de la cámara OOSXXX con cuadro de detección en presencia. Las figuras de 103 a la 105 del anexo muestran la especificación técnica. Las figuras 106 al 111 muestran la instalación de la cámara físicamente con la ubicación de sistema de actuadores que toma el efecto del algoritmo de compensaciones.

B. Conclusiones de diseño de brazos mecánicos

- Fue posible realizar los brazos robóticos para el diseño de un artista nacional.
- Se seleccionó PLA como material para emplear en la manufactura ya que el requerimiento de torque no superó sus esfuerzos últimos de corte y normales.

- Al requerir un 92% de la capacidad de los servomotores estos no fueron los ideales para el diseño de brazos establecido, por lo que se recomienda seleccionar otros más potentes o reducir peso a la estructura.
- Se identificaron 6 puntos críticos en la estructura con una mayor probabilidad de falla de un factor de seguridad de 1.37 en los tornillos del punto crítico 2 para el brazo completamente en voladizo.
- A pesar de que los brazos llegan al ángulo con un porcentaje de error máximo de 10%, solo son capaces de moverse en posiciones que duren menos de 8 segundos a ángulos mayores a 70° a partir de su posición neutral.

C. Conclusiones de diseño de gestos faciales

- Se logró diseñar los mecanismos de ojos, cejas y boca los cuales emulan el comportamiento de gestos fáciles cumpliendo con el mayor requerimiento que fue el espacio de trabajo.
- No fue posible realizar el diseño e implementación de las orejas debido al diseño llevado a cabo por parte del artista nacional.
- La selección de materiales fue la óptima debido al fácil manejo para el mecanizado como su fácil acceso, los materiales fueron MDF y PLA dichos materiales fueron válidos para la fabricación de los mecanismos debido a sus propiedades físicas y mecánicas ya soportan las cargas mínimas despreciables que se presentan y cumplen con los requerimientos de bajo peso.
- El análisis de esfuerzo y movilidad fue el adecuado ya que se pudo observar que las cargas presentes son despreciables ya que son muy pequeñas en comparación con el torque que puede dar el servomotor, por lo tanto, la movilidad no se ve afectada y todos los mecanismos emulan de forma correcta los gestos faciales.

XI. RECOMENDACIONES

A. Recomendaciones para procesamiento de audio y video

Para desarrollo posterior, el sistema de reconocimiento de rostro podría basarse en enfocar en ampliación de la investigación de las cámaras. Cámaras con mejores resoluciones y sensores para mantener la calidad de la imagen capturado manteniendo la dimensión mínima con la consideración de la instalación mecánica. Además, podría ser interesante implementar la tecnología de reconocer los objetos en el entorno e interactuar con ello para comunicar intuitivamente con el usuario. Adicionalmente, interacción de múltiples instancias al haber 2 o más usuarios dentro de la región de detección así mejorando la calidad de comunicación y mostrar que la perspectiva presente sea el área completa de reconocimiento de rostro. El sistema de reconocimiento de audio podrá basar en comprensión de frases más complejas y aumentar el nivel de precisión de reconocimiento de comandos o dichos de los usuarios, eliminando por completo el límite de comandos en lo mínimo y procesarlo dado el orden completo de parte del usuario

B. Recomendaciones para diseño de brazos mecánicos

- Se recomienda diseñar el brazo robótico después de tener el modelo estético por el artista nacional. Este último debe ser realizado con una capa interna de un material rígido para poder acoplarse, y, además, que utilice materiales con poca fricción en las articulaciones.
- Se recomienda cambiar los servomotores por unos que ofrezcan un par por lo menos 5 veces mayor al necesario, según el fabricante, para movimientos suaves.
- Se recomienda, para brazos que requieran poco peso, una estructura que ocupe el área transversal del brazo necesaria ocupando el menor volumen interno posible.
- En las áreas de las articulaciones, se recomienda para la estructura tener una transición de mucha área transversal ocupada a otra más pequeña y luego regresar al área ocupada transversal previa.
- En piezas iguales y con la misma funcionalidad, se recomienda sólo calcular esfuerzos para la pieza más crítica.
- Se recomienda utilizar elementos de unión más robustos para el área del hombro.
- Se recomienda establecer los límites de ángulo para los grados de libertad de los brazos con el fin de establecer un volumen de trabajo para el manipulador y evitar percances por sobrecarga.
- Se recomienda hacer un diseño de brazo robótico que facilite su ensamble y desensamble con la piel colocada.
- Se recomienda comprobar el análisis de esfuerzo teóricamente y en un software de diseño

C. Recomendaciones para diseño mecánico de gestos faciales

- Se recomienda buscar artistas nacionales con mayor experiencia en la elaboración de moldes y que tengan el conocimiento de piel látex.
- Al seleccionar el ente a realizar para convertirse en animatrónico identificar el área para las orejas para que sean posibles su diseño.
- Para el mecanismo de las cejas, se puede reducir a un servomotor y así reducir espacio.
- Es recomendable utilizar un MDF a prueba de humedad si se piensa trabajar en exteriores.
- Para movimiento más fluidos o que se necesite mayor soporte de carga utilizar cilindros neumáticos.

XII. BIBLIOGRAFÍA

- AITIM. 2015. *Tableros de fibras de densidad media*. http://infomadera.net/uploads/productos/informacion_general_426_mdf.pdf, [29/09/2017].
- AITIM. 2015. *Tableros de madera de densidad media MDF*. https://www.cscae.com/area_tecnica/aitim/enlaces/documentos/Tableros_Fibras%20MDF_15.06.2015.pdf. [02/10/2017].
- Alpaydan, Ethem. 2014. *Introduction to Machine Learning*. 3a ed. Cambridge, MA: The MIT Press. 615 págs.
- Arduino. *Arduino Software (IDE)*. <https://www.arduino.cc/en/Guide/Environment> [Junio de 2017]
- Asociación Dental Mexicana. 2011. *Fuerza de Mordida*. <http://www.medigraphic.com/pdfs/adm/od-2012/od122c.pdf>. [24/09/17].
- Baturone, Anibal O.; 2001. «Introducción». *Robótica: Manipuladores y robots móviles*. Barcelona: Marcombo. págs. 11-13.
- Budynas, Richard G.; y Nisbett, Keith J.; 2008. «Análisis de carga y esfuerzo». *Diseño en ingeniería mecánica de Shigley*. México: Libros McGraw-Hill de México. págs. 67-121.
- Cambridge in Colour. *Camera Exposure*. <http://www.cambridgeincolour.com/tutorials/camera-exposure.htm> [Septiembre de 2017]
- Castro, Miguel A. *PLA: Material, interés y consejos de impresión*. <http://www.dima3d.com/pla-material-interes-y-consejos-de-impresion-3/> [Consultado el 5 de octubre del 2017]
- CMUSphinx. *Building an application with Pocket Sphinx*. <https://cmusphinx.github.io/wiki/tutorialpocketsphinx/> [Junio de 2017]
- Dima 3D. 2015. *¿Qué es el PLA y porque resulta interesante en impresión 3D?* <http://www.dima3d.com/pla-material-interes-y-consejos-de-impresion-3/>, [29/09/2017].
- Fogaça, Carlos Lane. 1998. *Apertura máxima de boca: estudio clínico*. <http://bases.bireme.br/cgi-bin/wxislind.exe/iah/online/?IsisScript=iah/iah.xis&src=google&base=LILACS&lang=p&nextAction=lnk&exprSearch=296586&indexSearch=ID>. [24/09/17].

- González, Víctor R. 2003. *Servomotores*.
http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/motores_servo.htm.
 [24/09/17].
- Goodfellow, Ian; B. Yoshua y C. Aarón. 2016. *Deep Learning*. Cambridge, MA: The MIT Press. 777 págs.
- Google. *Google Cloud Speech API Documentation*. <https://cloud.google.com/speech/docs/> [Junio de 2017]
- Hervas Villalba. *Tecnología Industrial I. Procedimientos de Fabricación*.
<https://iesvillalbahervastecnologia.files.wordpress.com/2011/03/conformado-por-union-de-piezas.pdf>, [30/09/2017].
- HXX. *Materiales de impresión 3D (1): PLA (ácido poliláctico)*. <http://hxx.es/2015/03/12/materiales-de-impresion-3d-i-pla-acido-polilactico/> [Consultado el 5 de octubre del 2017]
- IBM. *IBM Bluemix Docs*. <https://console.bluemix.net/docs/services/speech-to-text/index.html#about> [Junio de 2017]
- J. Nilsson, Nils. 2005. *Introduction To Machine Learning*. Stanford University. 188 págs.
- Kim, Phil. 2017. *Mat Lab Deep Learning. With Machine Learning, Neural Networks and Artificial Intelligence*. Seoul: Apress. 149 págs.
- Knight, Will. *Robots inteligentes junto a trabajadores en fábricas de automóviles*.
<https://www.technologyreview.es/s/3778/robots-inteligentes-junto-trabajadores-en-fabricas-de-automoviles> [Consultado el 24 de septiembre del 2017]
- Lazalde, Alan. *Robots militares autónomos: inicia el debate*. http://www.eldiario.es/turing/Robots-militares_0_107790085.html [Consultado el 21 de septiembre del 2017]
- LeCun, Yann; B. Yoshua y H. Geoffrey. 2015. «*Deep Learning*». *REVIEW*, Vol. 521. Macmillan Publishers Limited.
- Lee, H.-H.; 2017. «*Introduction*». *Finite Element Simuations with ANSYS Workbench 17*. Estados Unidos: SDC Publications. págs. 9-50.
- Madera Santana. 2015. *Características MDF*. <https://www.maderasantana.com/caracteristicas-tableros-madera-mdf/>, [2/10/2017].
- Masisa. *Masisa: MDF*. <http://old.masisa.com/mex/productos/masisanatural/mdf/> [Consultado el 5 de octubre del 2017]

- Masisa. 2017. *MDF*. http://www.masisa.com/ven/wp-content/files_mf/1419819864FichaT%C3%A9cnicaMDP.pdf, [29/09/2017].
- Math Works. *MATLAB Product Design*. https://www.mathworks.com/help/matlab/learn_matlab/product-description.html [Junio de 2017]
- Microsoft. *Microsoft LifeCam Cinema Technical Datasheet*. https://media.flixcar.com/f360cdn/Microsoft-303383594-Microsoft-TDS_LifeCamCinema.pdf [Agosto de 2017]
- Moore, Keith L.; y Agur, Anne M.R.; 2002. «*Introducción a la Anatomía Clínica*». *Fundamentos de Anatomía con orientación clínica*. Buenos Aires: Médica Panamericana. págs. 2-19.
- Mott, Robert L. 2006. *Diseño de elementos de máquinas*. 4ta edición. México. Pearson educación. Págs. 83-122.
- Mott, Robert L.; 2006. «*Análisis de Esfuerzos y Deformaciones*». *Diseño de elementos de máquinas*. México: Pearson Educación. págs. 83-122.
- Normani, Franco. *Pendulum Physics*. <http://www.real-world-physics-problems.com/pendulum-physics.html> [Consultado el 6 de octubre del 2017]
- Norton, Robert L. 2009. *Fundamentos de Cinemática, Diseño de maquinaria*. 4ta ed. Mc Graw Hill. México. Págs. 27-50.
- OOSXXX. *Mini Camera, OOSXXX Mini Security Camera [600 TVL] Hidden Camera, Mini Camera Mini CCTV Spy Camera Pinhole Camera for Security Surveillance*. <http://www.securitysuppliesoutlet.com/unit/12701684/> [Agosto de 2017]
- OpenCV. OpenCV. <http://opencv.org/about.html> [Junio de 2017]
- Ordoñez. Raynack Valencia. 2016. *Diseño de Esqueleto Animatrónico*. Universidad Nacional Autónoma de México. Facultad de Ingeniería Mecatrónica.
- Pérez Camps. *Máquinas de grabado y corte láser*. http://www.perezcamps.com/es/maquinas-de-grabado-y-corte-laser_586 [Consultado el 5 de octubre del 2017]
- Prosector. *Ácido Poliláctico PLA*. <https://plastics.ulprosector.com/es/generics/34/c/t/acido-polilactico-pla-properties-processing>, [30/09/2017].

- Prospector. *Polylactic Acid (PLA) Typical Properties*.
<https://plastics.ulprospector.com/generics/34/c/t/polylactic-acid-pla-properties-processing>
 [Consultado el 5 de octubre del 2017]
- Python. *sys - System specific parameters and functions*. <https://docs.python.org/2/library/sys.html> [Junio de 2017]
- Python. *The Python Standard Library*. <https://docs.python.org/2/library/index.html#library-index> [Junio de 2017]
- Quora. *What is an API?* <https://www.quora.com/What-is-an-API-4> [Junio de 2017]
- ROBOTIS. *DYNAMIXEL XL-320*. <http://www.robotis.us/dynamixel-xl-320/>, [24/09/2017]
- Robotis. *Robotis: e-Manual*. <http://support.robotis.com/en/> [Consultado el 5 de octubre del 2017]
- RobotShop. *Dynamixel AX-12A Smart Serial Servo*. <http://www.robotshop.com/en/dynamixel-ax-12a-smart-servo-serial.html> [Consultado el 5 de octubre del 2017]
- Saha, Shubhro. *Jasper. Control Everything with your Voice*.
<http://jasperproject.github.io/documentation/api/> [Junio de 2017]
- Shigley. 2011. *Diseño en ingeniería mecánica*. 9na ed. Mc Graw Hill. México DF.
- Shih, R. 2016. *Autodesk Inventor 2017 and Engineering Graphics: An Integrated Approach*. SDC Publications.
- Shih, Randy H.; 2016. «Introduction». *Autodesk Inventor 2017 and Engineering Graphics: An Integrated Approach*. Estados Unidos: SDC Publications. págs. 1-3 a 1-22.
- Siciliano, Bruno y K. Oussama. 2011. *Robotics, Vision and Control. Fundamental Algorithms in MATLAB*. Stanford, CA: Springer. 697 págs.
- Siciliano, Bruno, et al. 2010. *Robotics: Modelling, Planning and Control*. Londres: Springer. 632 págs.
- Siciliano, Bruno; Michael J. Grumble y Michael A. Johnson. 2010. *Robotics. Modelling, Planning, and Control*. Natick, MA: Springer. 643 págs.
- Snow boy. *Snow boy, a Customizable hotword Detection Engine*. <http://docs.kitt.ai/snowboy/#introduction>
 [Junio de 2017]
- Starr, Taggart, Evers. 2009. *Biología La unidad y la diversidad de la vida*. 12va Ed. Cengage Learning.

Tirinnanzi Daniele. 2000. *Movie Animatronics. Building mechanical mask*. <http://www.daniele-tirinnanzi.com/store.php>, [29/09/2017]

Trossen Robotics. *Dynamixel XL-320 Robot Actuator*. <http://www.trossenrobotics.com/dynamixel-xl-320-robot-actuator> [Consultado el 5 de octubre del 2017]

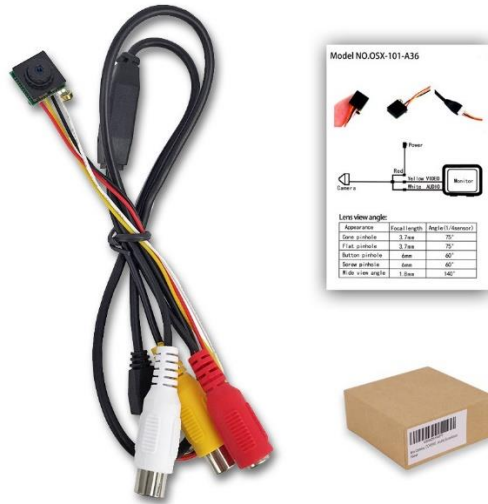
Villaverde M. *PLA*. <http://www.eis.uva.es/~biopolimeros/alberto/pla.htm>, [30/09/17]

W3C. *JSpeech Grammar Format*. <https://www.w3.org/TR/jsgf/> [Junio de 2017]

XIII. ANEXOS

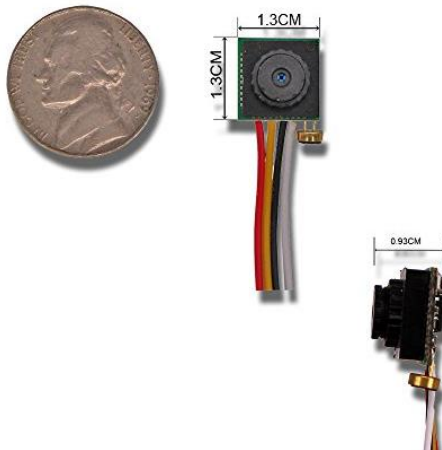
A. Anexo para el Procesamiento de Audio y Video

Figura 103. Vista general de OOSXXX Mini Security Camera [600TVL]



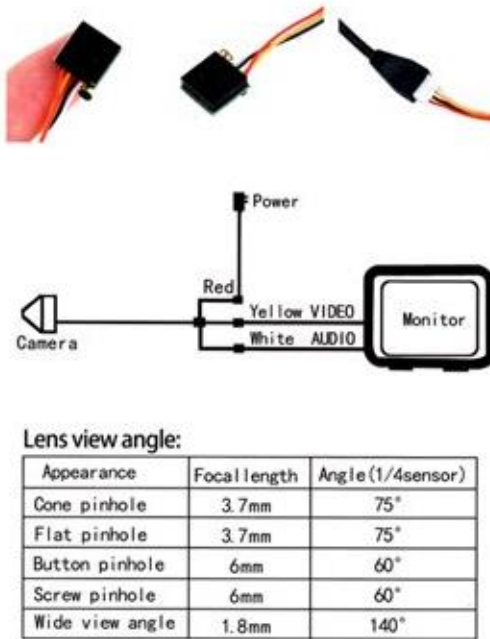
(OOSXXX)

Figura 104. Dimensionamiento de OOSXXX Mini Security Camera



(OOSXXX)

Figura 105. Especificación de conexión eléctrica de OOSXXX Mini Security Camera



(OOSXXX)

Figura 106. Ubicación de la cámara (C) OOSXXX para el sistema de reconocimiento de rostro

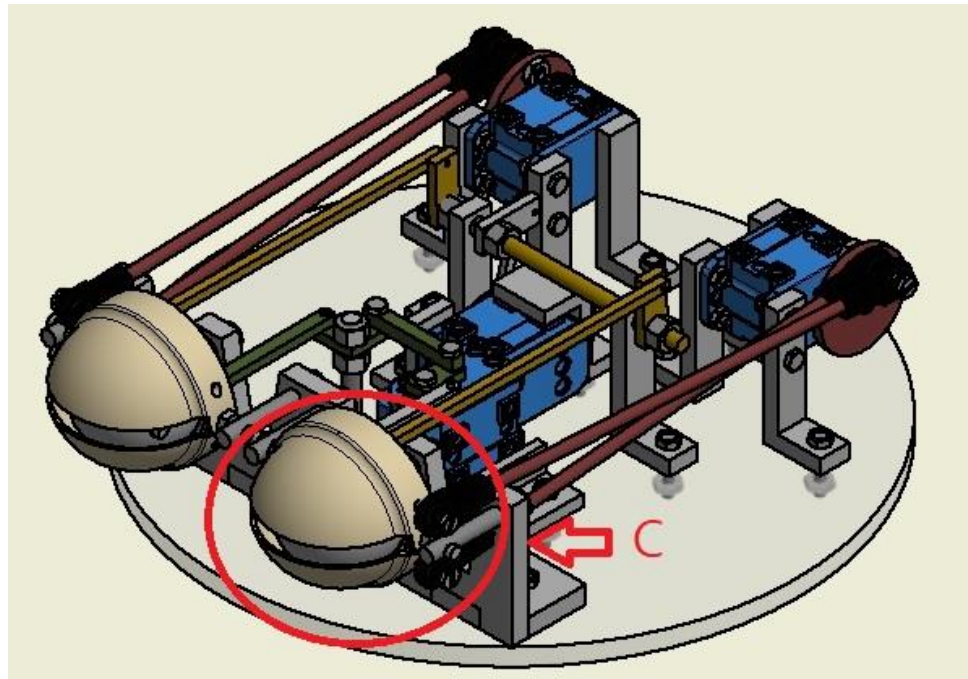


Figura 107. Ubicación de la cámara (A) OOSXXX, (B) Vista aérea de los ojos

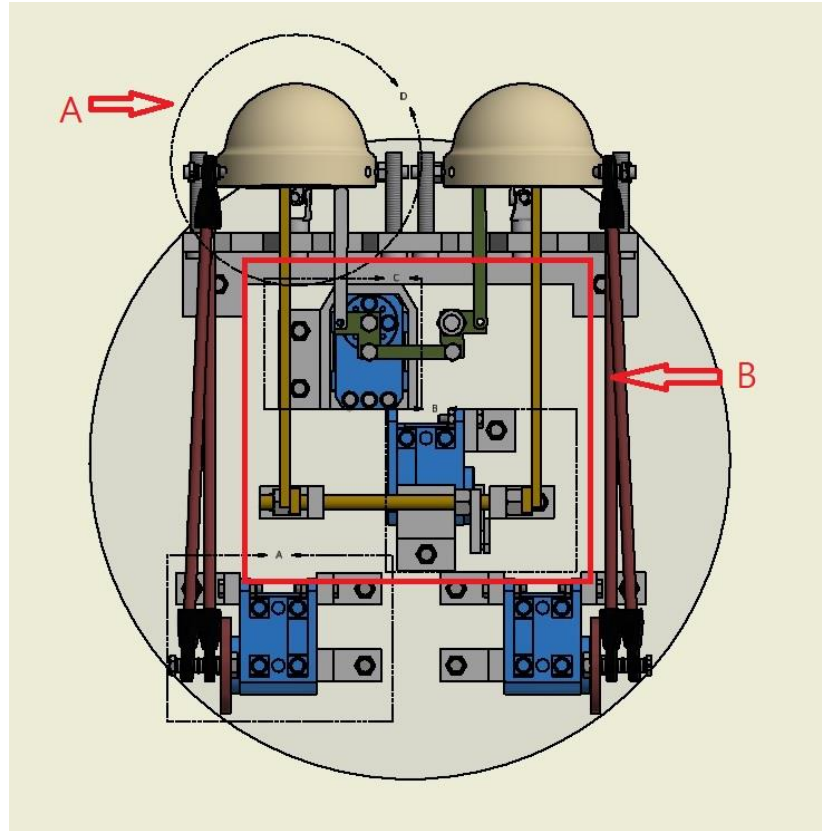


Figura 108. Cámara OOSXXX sujeta al mecanismo interno del ojo



Figura 109. Cámara OOSXX sujeto al mecanismo, vista lateral

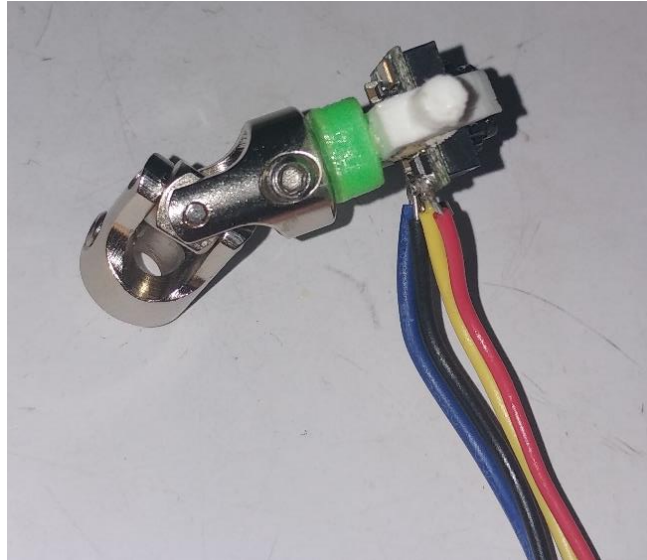


Figura 110. Cámara OOSXX integrada dentro del ojo




Figura 111. Cámara OOSXXX vista externa frontal




Figura 112. Especificaciones de Microsoft LifeCam Cinema (Microsoft)

Microsoft®

LifeCam Cinema™





Technical Data Sheet

| Version Information | |
|------------------------------------|---|
| Product Name | Microsoft® LifeCam Cinema™ |
| Product Version | Microsoft LifeCam Cinema |
| Webcam Version | Microsoft LifeCam Cinema |
| Product Dimensions | |
| Webcam Length | 2.20 inches (55.9 millimeters) |
| Webcam Width | 1.81 inches (46.0 millimeters) |
| Webcam Depth/Height | 1.58 inches (40.0 millimeters) |
| Webcam Weight | 3.36 ounces (95.3 grams) |
| Webcam Cable Length | 72.0 inches +6+0 inches (1829 millimeters +152+0 millimeters) |
| Compatibility and Localization | |
| Interface | Compatible with USB 2.0 High Speed specification |
| Operating Systems ¹ | <ul style="list-style-type: none"> • Microsoft Windows® 8.1, Windows 8, Windows RT 8.1, Windows RT 8, and Windows 7 • Macintosh OS X v10.7-10.9 • Android 3.2 and 4.2 <p>¹Advanced functionality not available with all devices and/or operating systems. See compatibility information at: www.microsoft.com/hardware/compatibility.</p> |
| Top-line System Requirements | <p>Requires a PC that meets the requirements for and has installed one of these operating systems:</p> <ul style="list-style-type: none"> • Microsoft Windows 8.1, Windows 8, or Windows 7 <p>For VGA video calling:</p> <ul style="list-style-type: none"> • Intel Dual Core 1.6 GHz or higher • 1 GB of RAM • 1.5 GB • USB 2.0 required • Windows-compatible speakers or headphones <p>For 720p HD recording:</p> <ul style="list-style-type: none"> • Intel Dual Core 3.0 GHz or higher • 2 GB of RAM • 1.5 GB • USB 2.0 required • Windows-compatible speakers or headphones <p>You must accept License Terms for software download. Please download the latest available software version for your OS/hardware combination.</p> <p>Internet access may be required for certain features. Local and/or long-distance telephone toll charges may apply. Software download required for full functionality of all features.</p> <p>Internet functions (post to Windows Live™ Spaces, send in e-mail, video calls), also require: Internet Explorer® 6/7/8 browser software required for installation; 25 MB hard drive space typically required (users can maintain other default Web browsers after installation)</p> |
| Compatibility Logos | <ul style="list-style-type: none"> • Compatible with Microsoft Windows 8 and Windows RT • Optimized for Microsoft Lync • Skype Certified |
| Software Localization | Microsoft LifeCam software version 3.0 may be installed in Simplified Chinese, Traditional Chinese, English, French, German, Italian, Japanese, Korean, Brazilian Portuguese, Iberian Portuguese, Russian, or Spanish. If available, standard setup will install the software in the default OS language. Otherwise, the English language version will be installed. |
| Windows Live™ Integration Features | |
| Video Conversation Feature | Windows Live call button delivers one touch access to video conversation |
| Call Button Life | 10,000 activations |
| Webcam Controls & Effects | LifeCam Dashboard provides access to animated video effects and webcam controls |
| Windows Live Integration Features | Windows Live Photo Gallery integration • Take a photo with LifeCam Software, then with one click open Photo Gallery to edit, tag and share it online Windows Live Movie Maker integration • Record a video with LifeCam Software and start a movie project on Movie Maker with just one click to then upload it to your favorite networking site |
| Imaging Features | |
| Sensor | CMOS sensor technology |
| Resolution | <ul style="list-style-type: none"> • Motion Video: 1280 x 720 pixels video • Still Image: Up to 5 megapixel (2880x1620 pixels, interpolated) photos* |
| Imaging Rate | Up to 30 frames per second |
| Field of View | 73° diagonal field of view |
| Imaging Features | <ul style="list-style-type: none"> • Digital pan, tilt, and zoom • Auto focus, range from 6" to infinity • Automatic image adjustment with manual override |
| Audio Features | |
| Audio Features | Integrated uni-directional digital microphone |
| Frequency Response | 200Hz to 8000Hz, +/- 4dB |

B. Anexo para diseño de los brazos mecánicos

Figura 113. Fotografía del brazo armado



Figura 114. Fotografía del setup para las pruebas de los dedos



Figura 115. Explosión del brazo completo

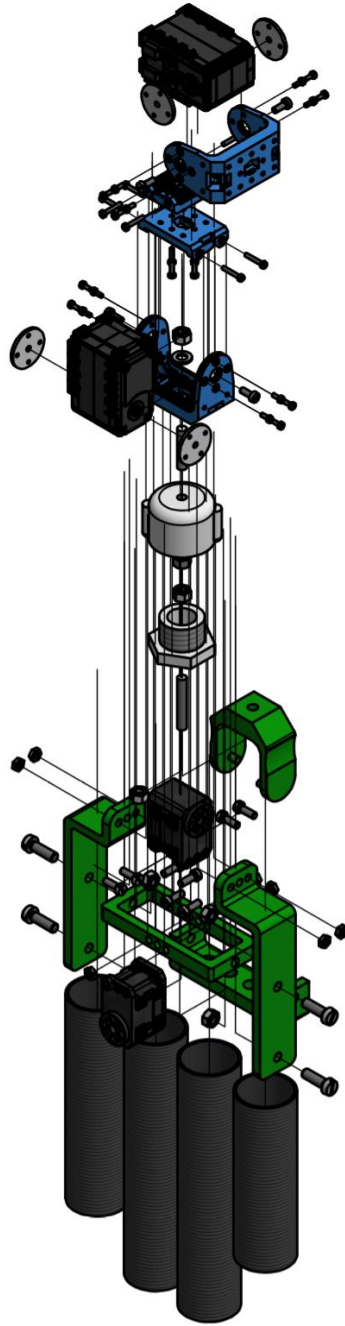


Figura 116. Dimensiones de todo el brazo armado

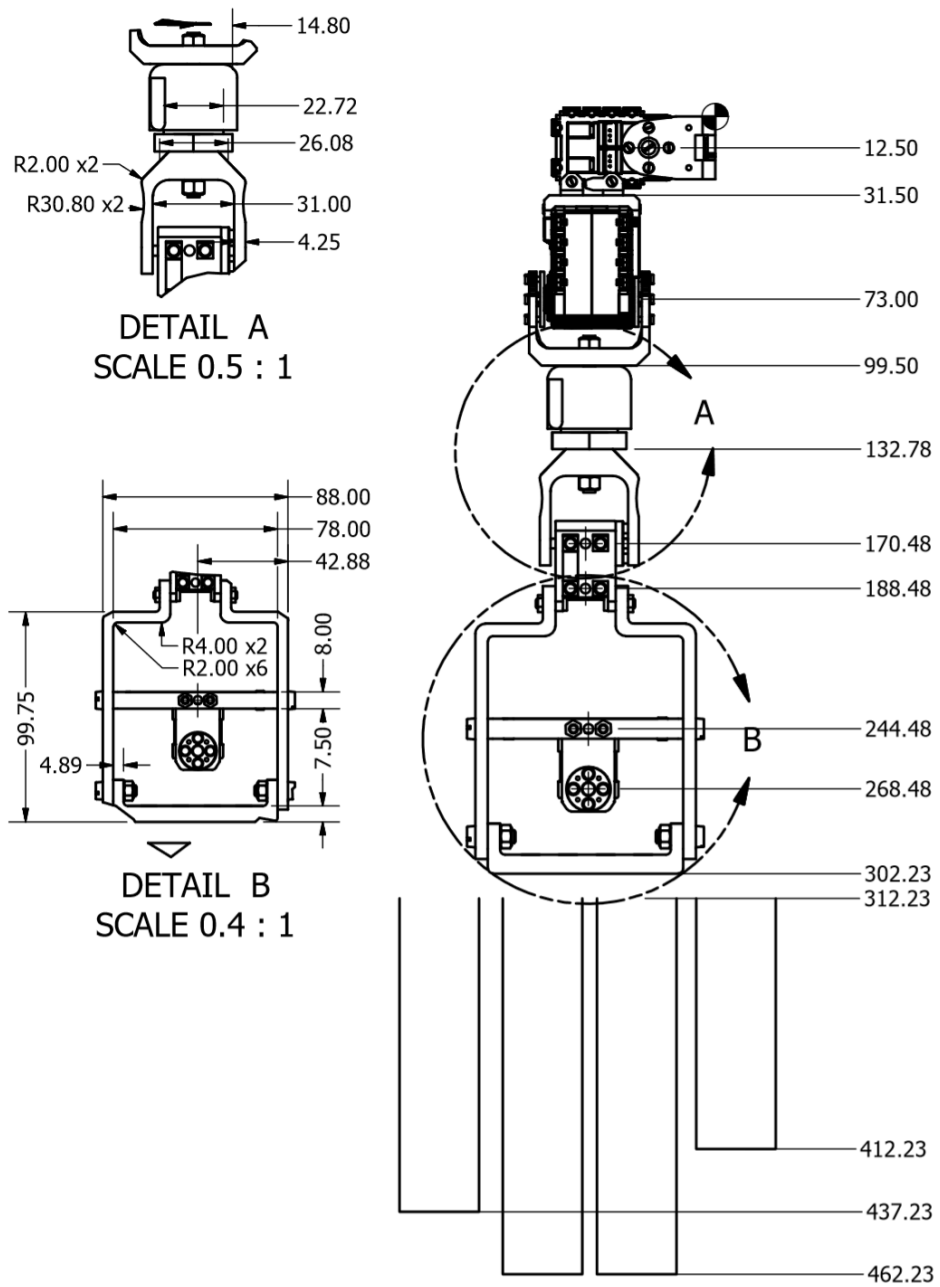


Figura 117. Dimensiones de todo el brazo armado desde otra vista

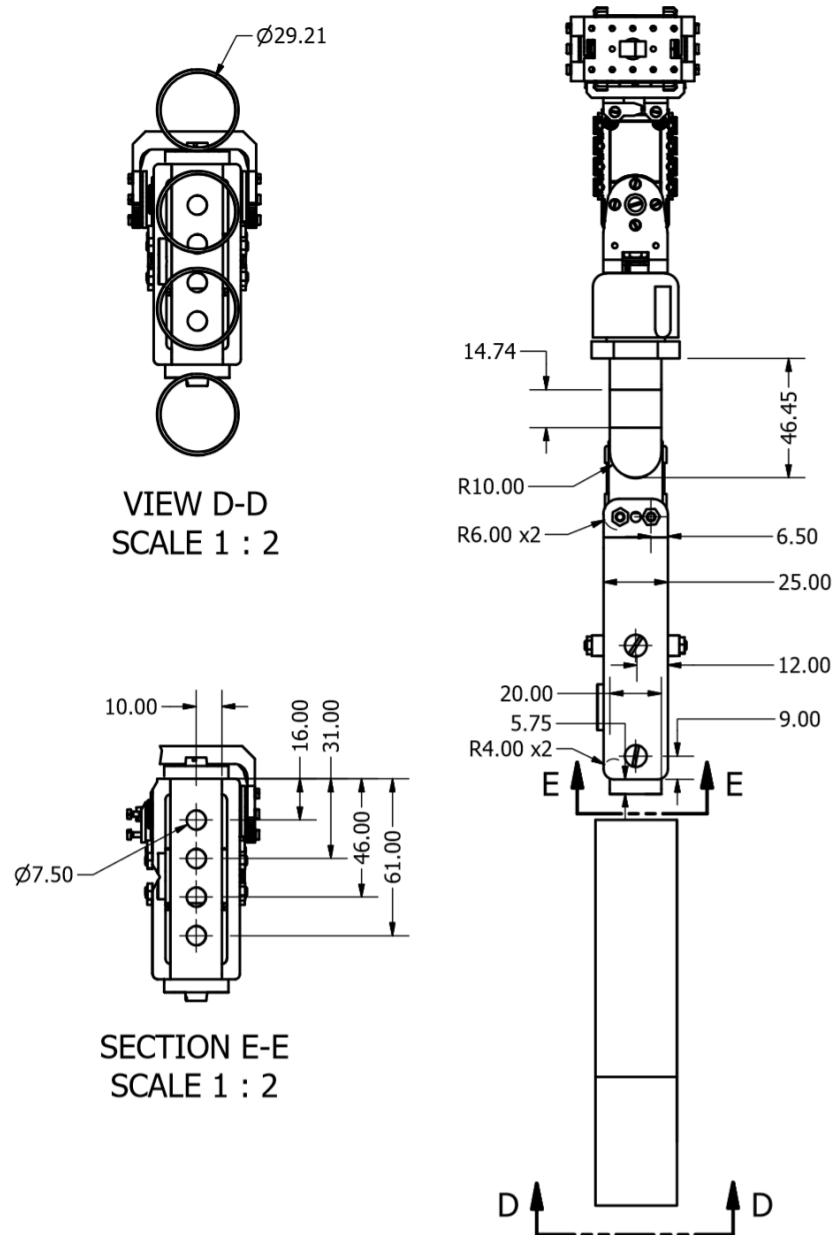


Figura 118. Pieza F2222, estándar (no se indican dimensiones)

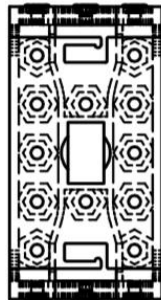
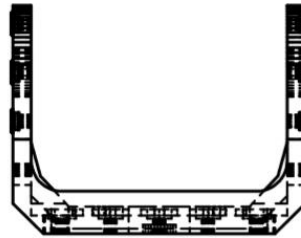
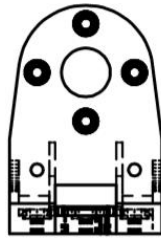
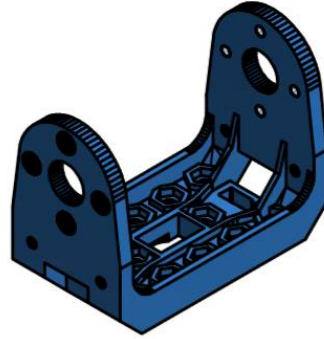
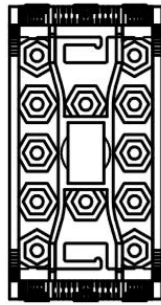


Figura 119. Accesorio para el servomotor AX-12A

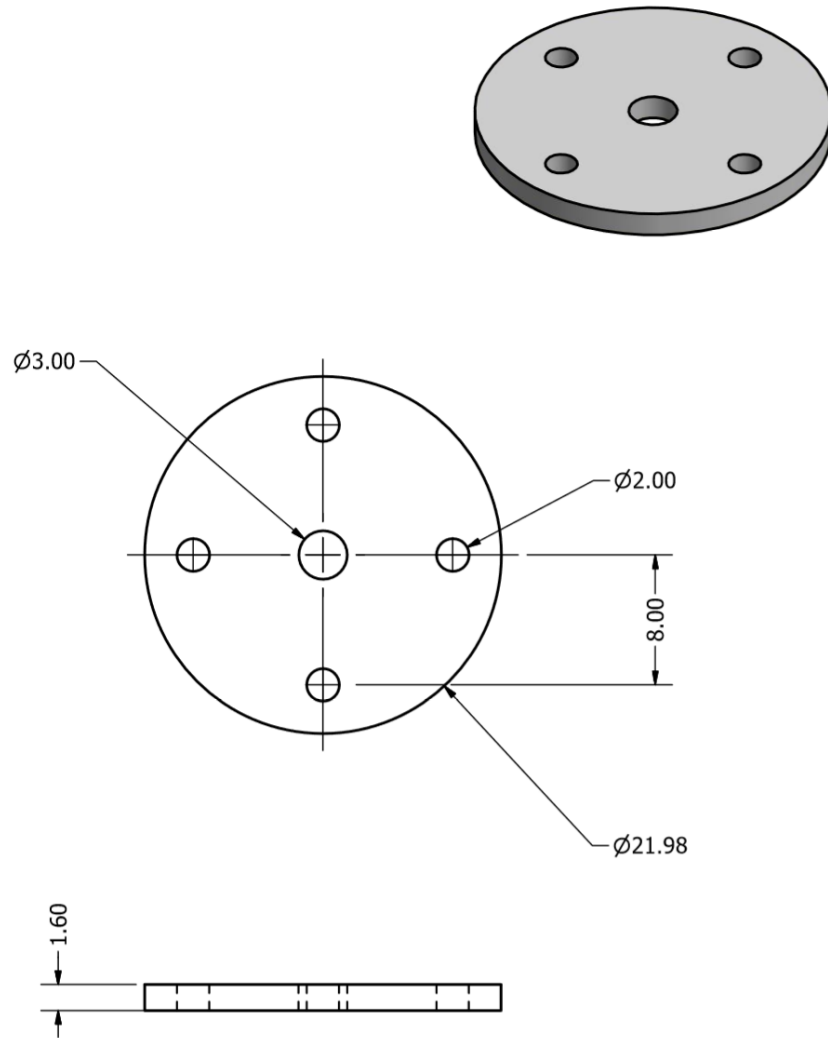


Figura 120. Servomotor AX-12A, estándar (no se indican dimensiones)

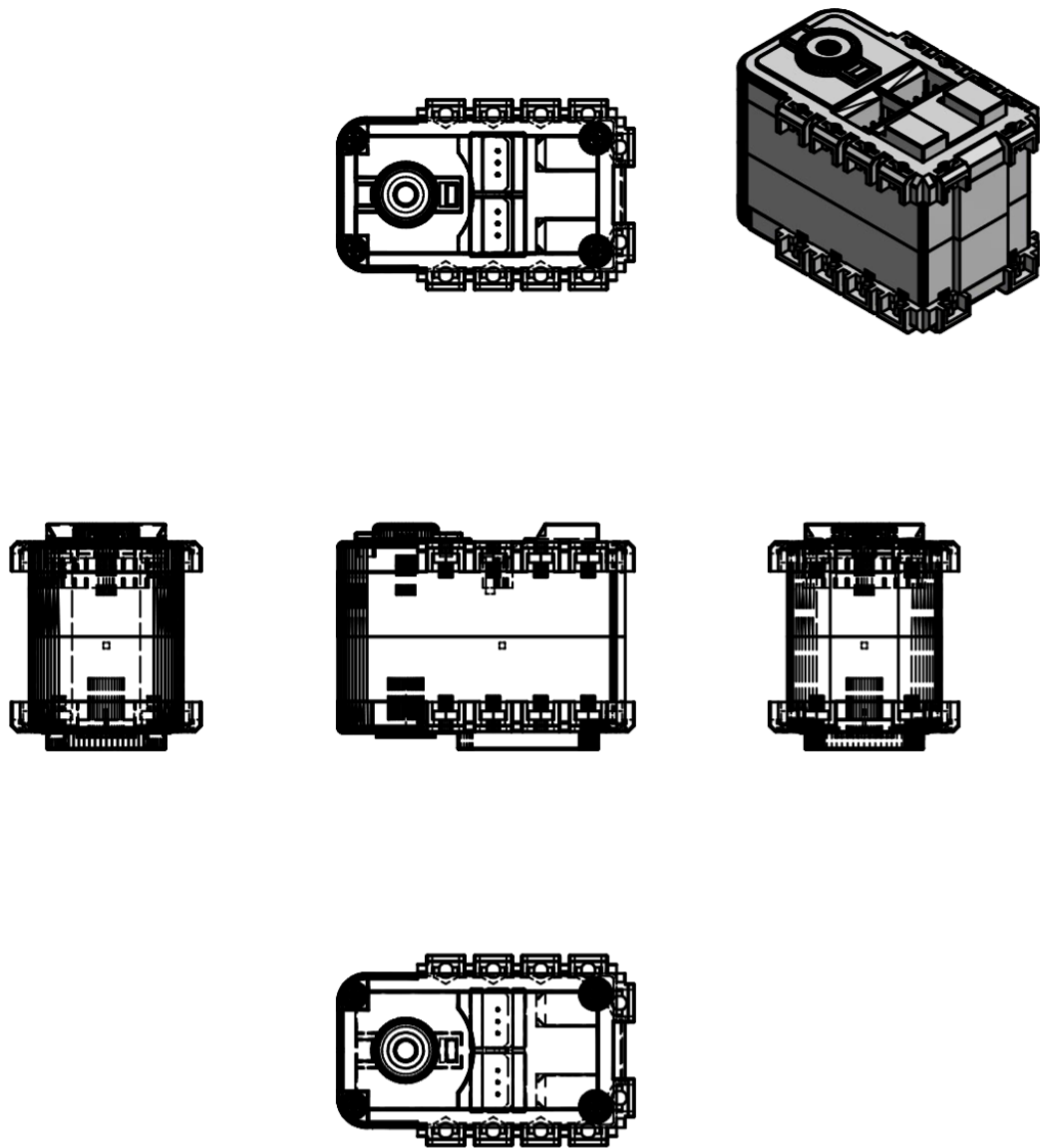


Figura 121. Pieza F333, estándar (no se indican dimensiones)

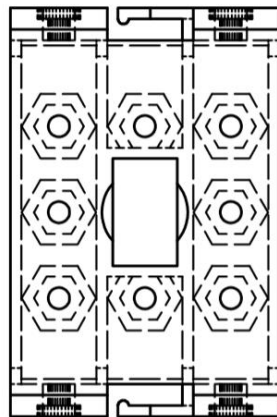
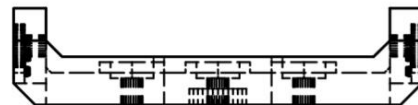
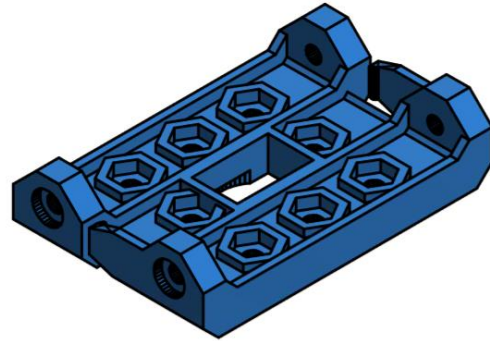
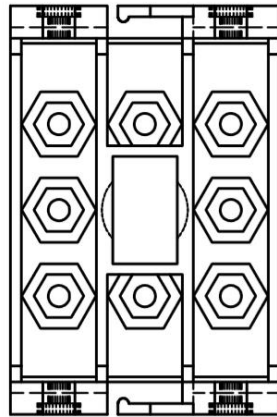


Figura 1222. Pieza acople hembra, estándar (no se indican dimensiones)

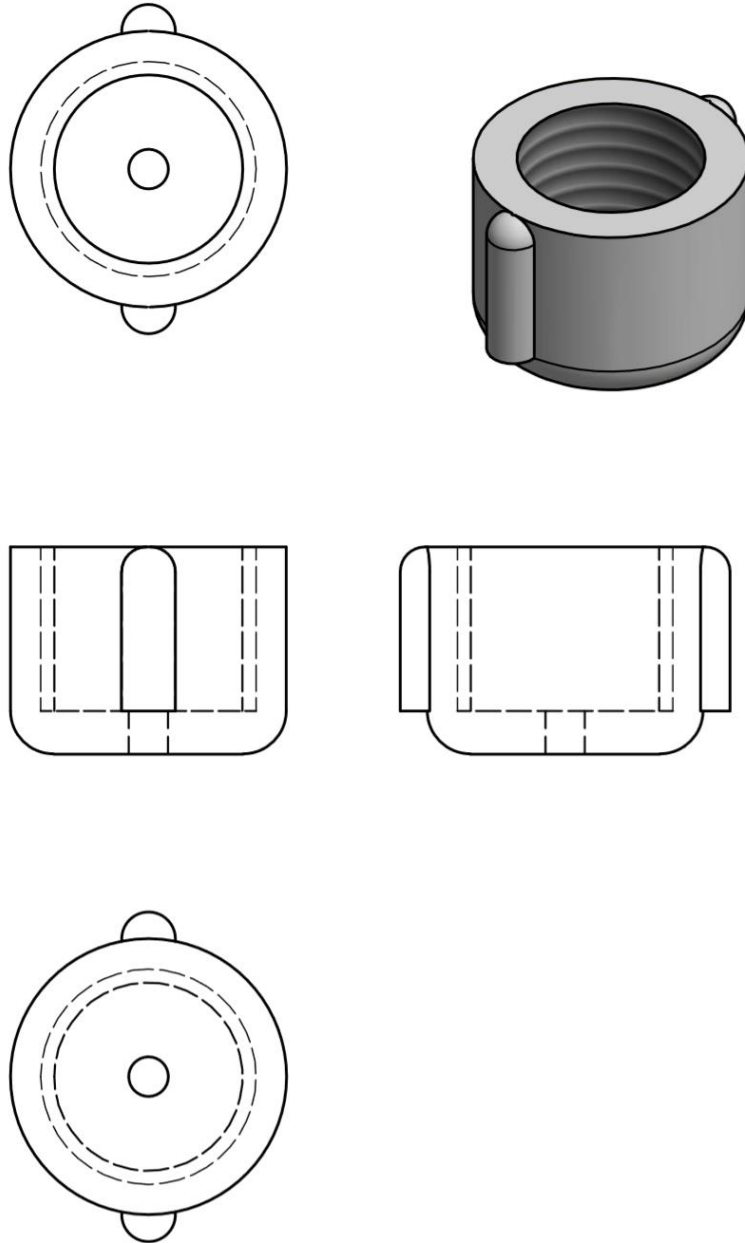


Figura 123. Pieza acople macho, estándar (no se indican dimensiones)

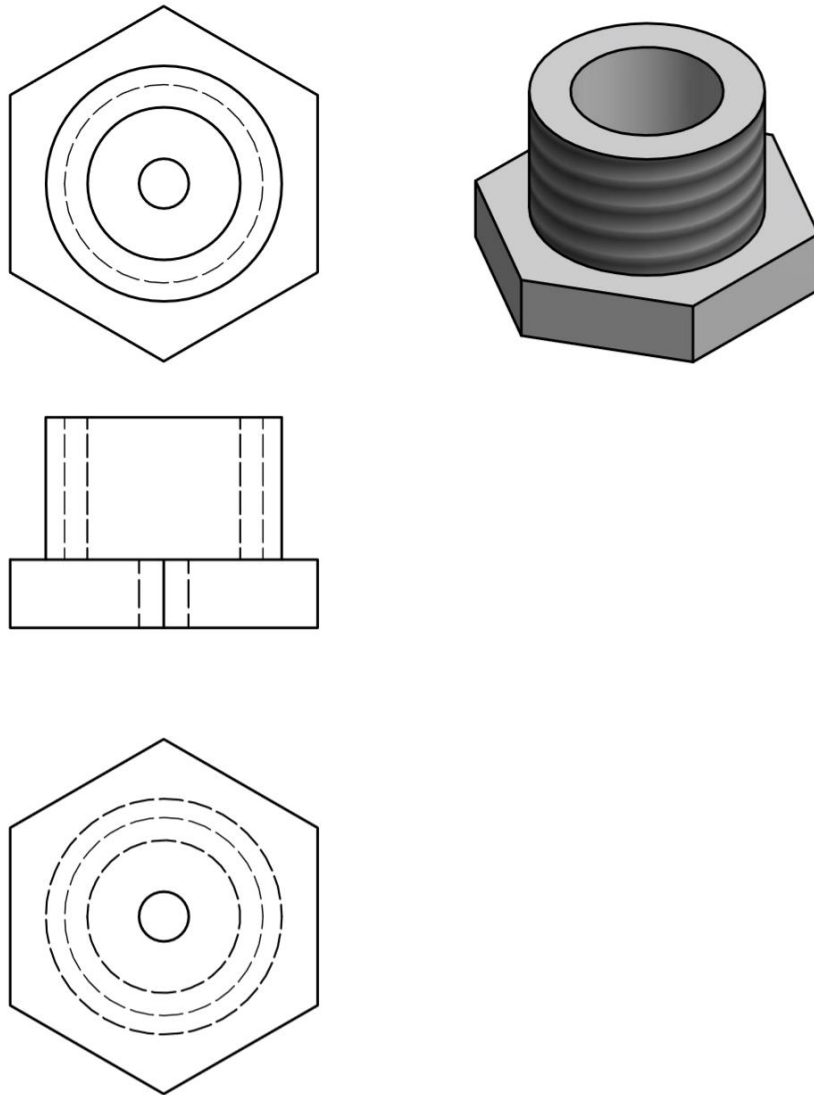


Figura 1243. Accesorio del servomotor XL-320

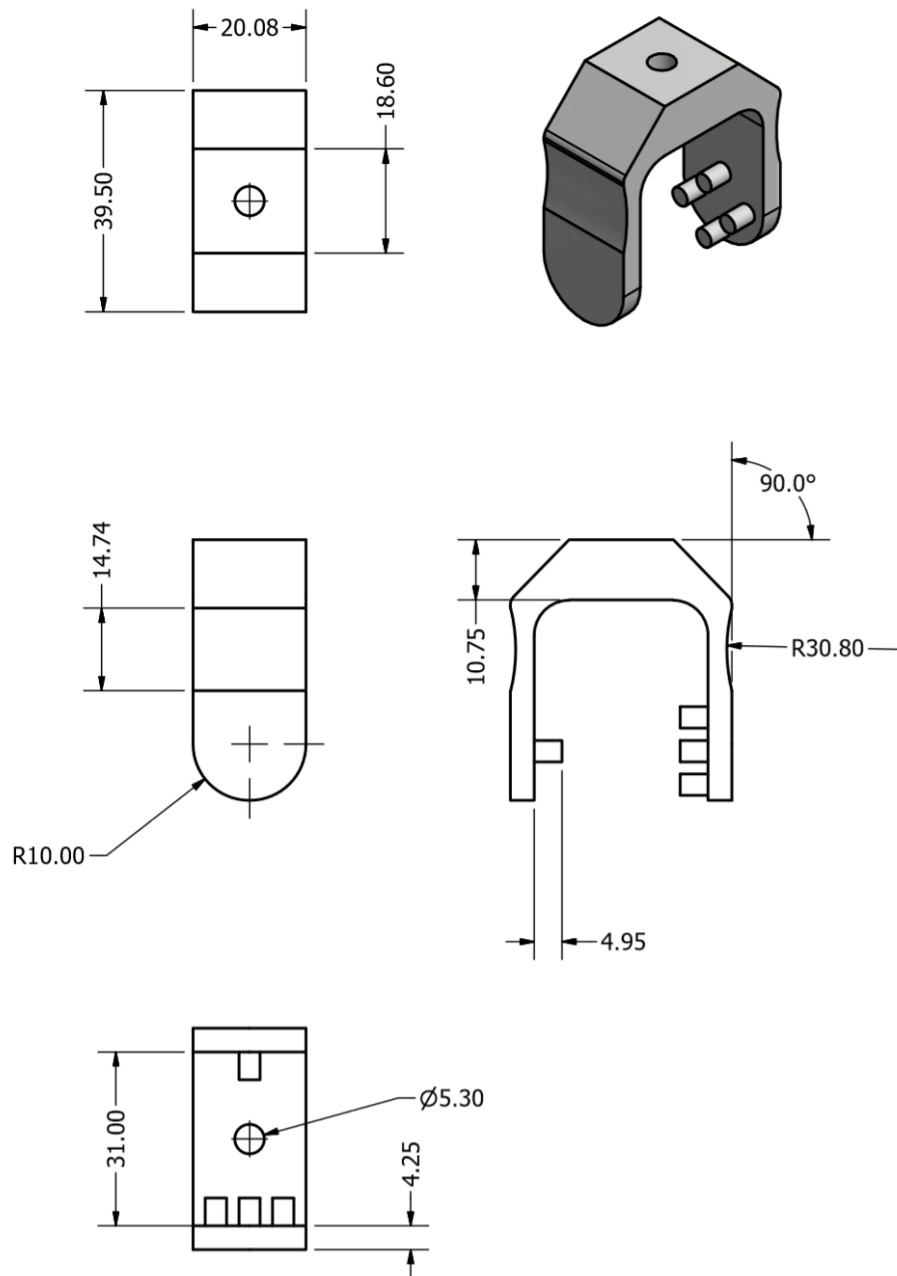


Figura 125. Servomotor XL-320

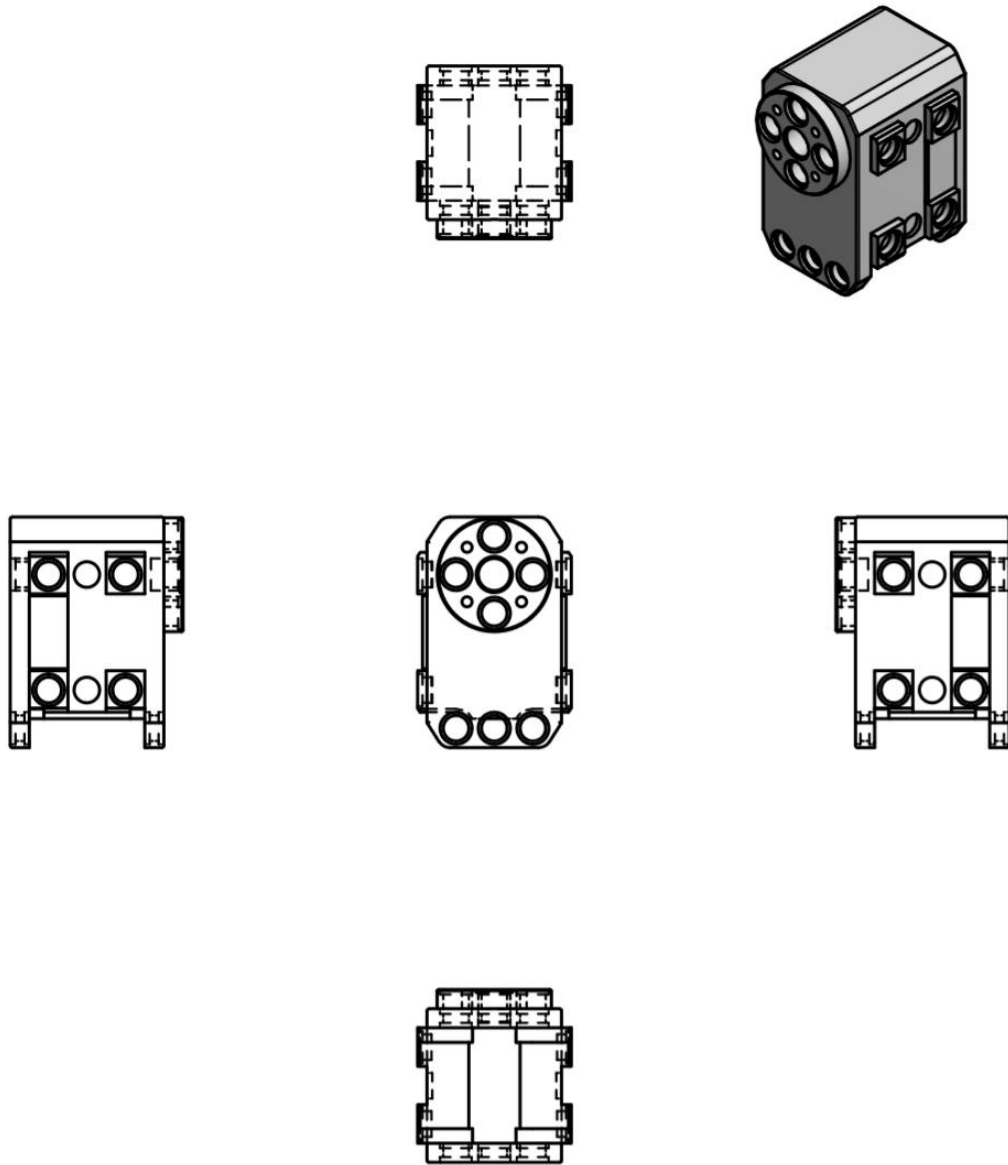


Figura 126. Pieza codo-muñeca

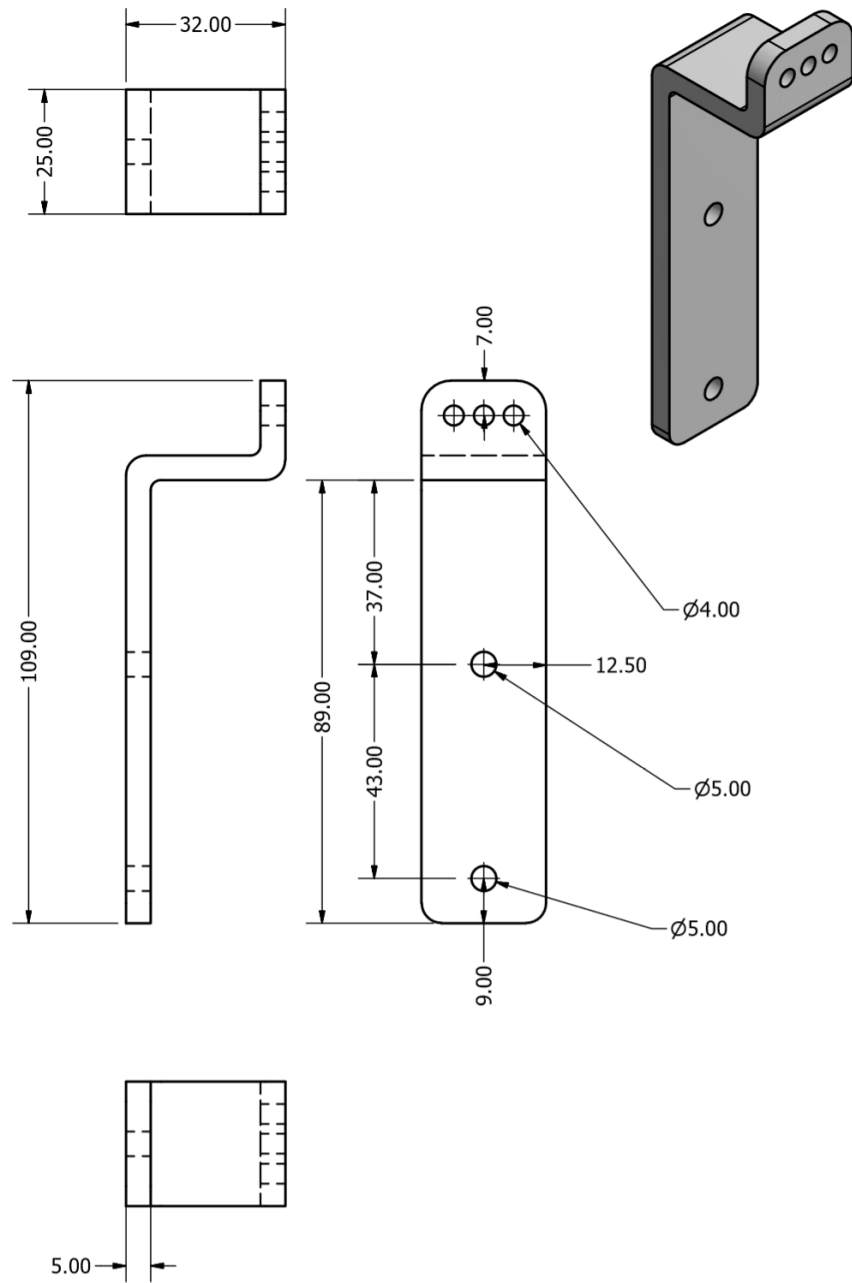


Figura 127. Pieza separador

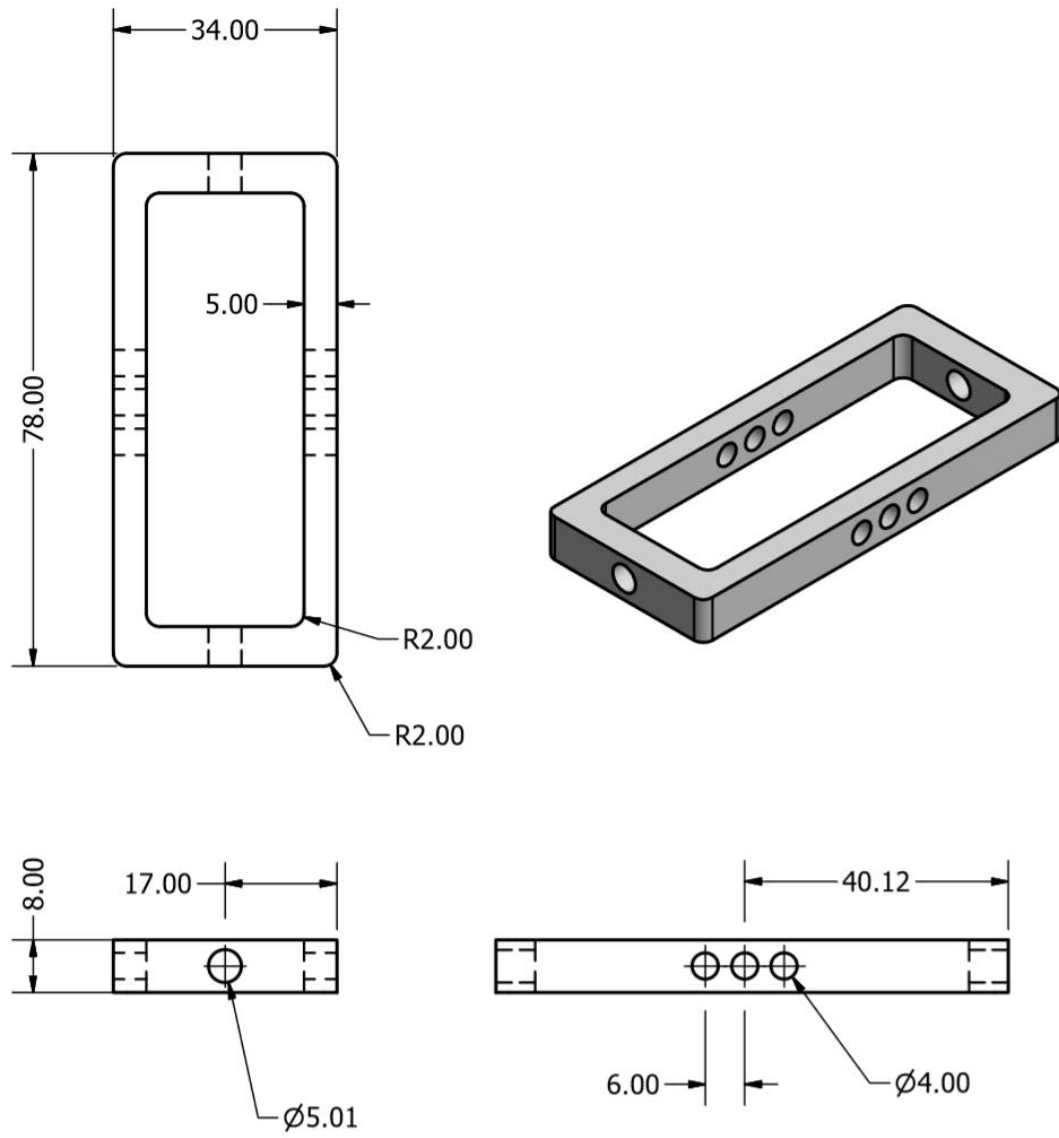


Figura 128. Pieza separador 2

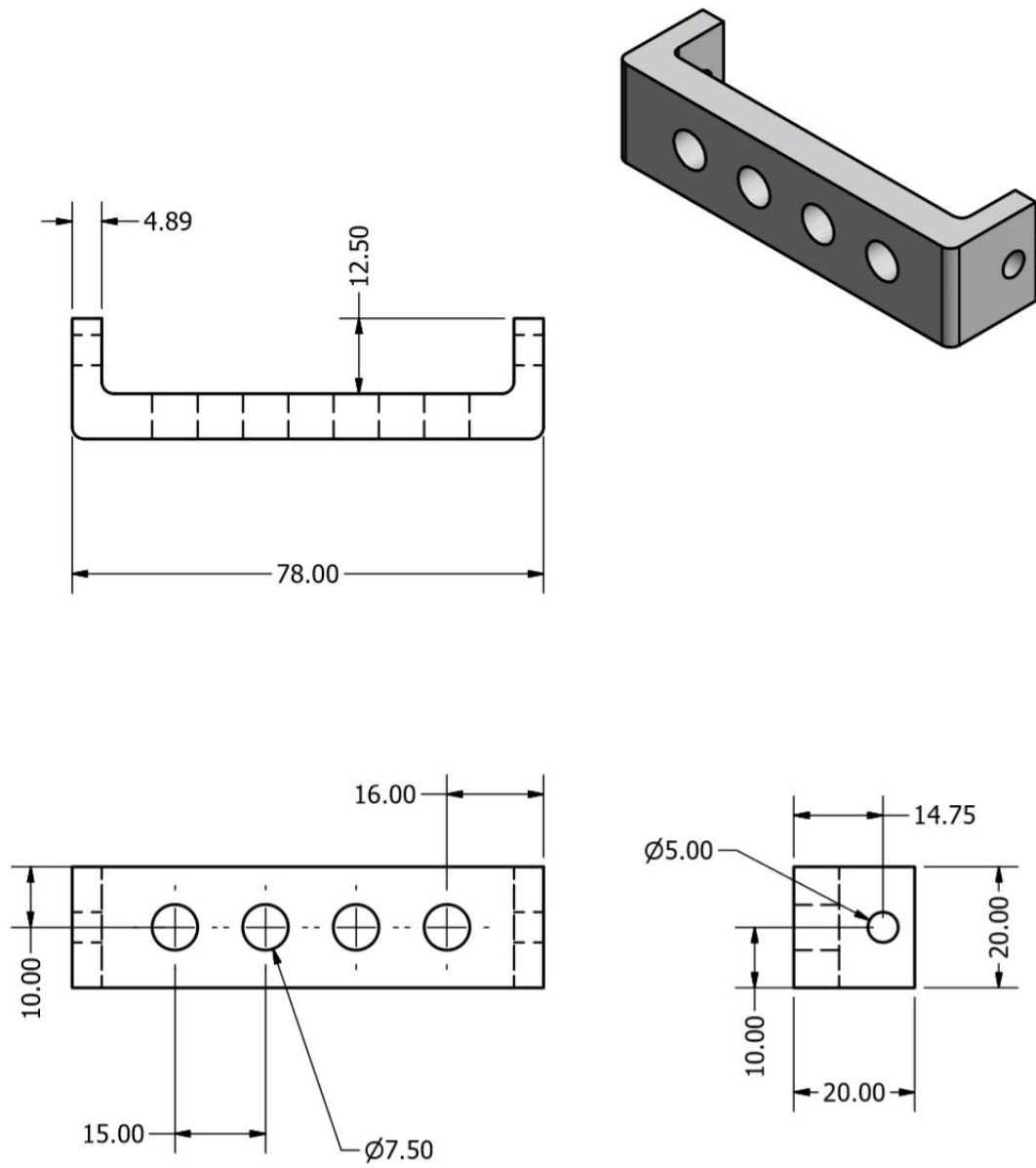
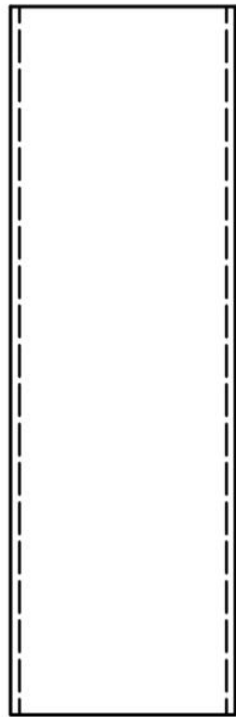
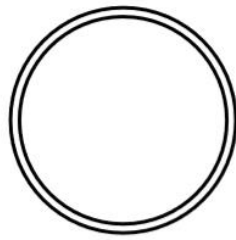
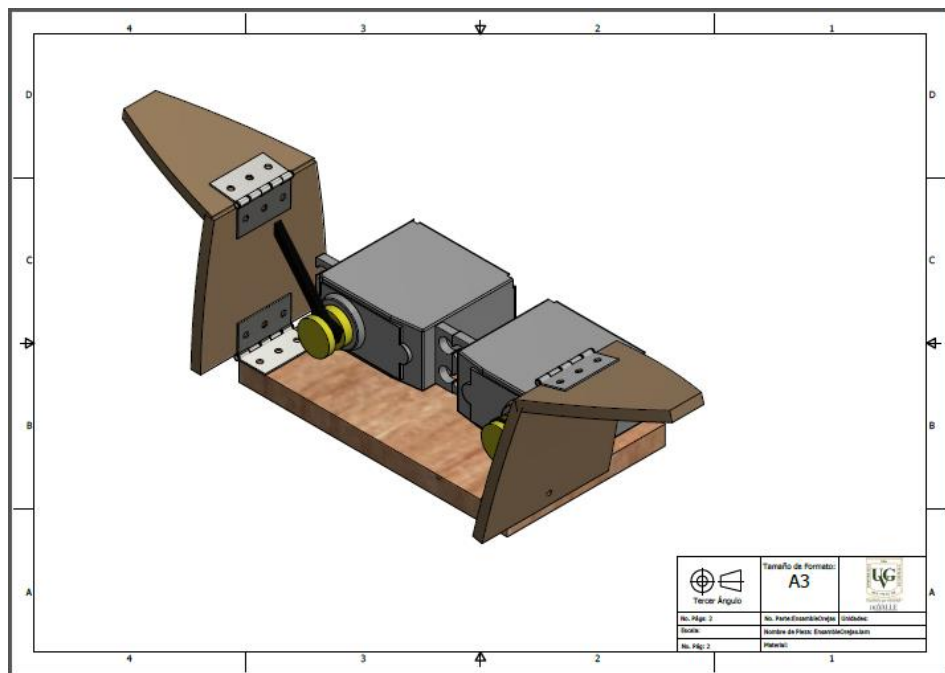
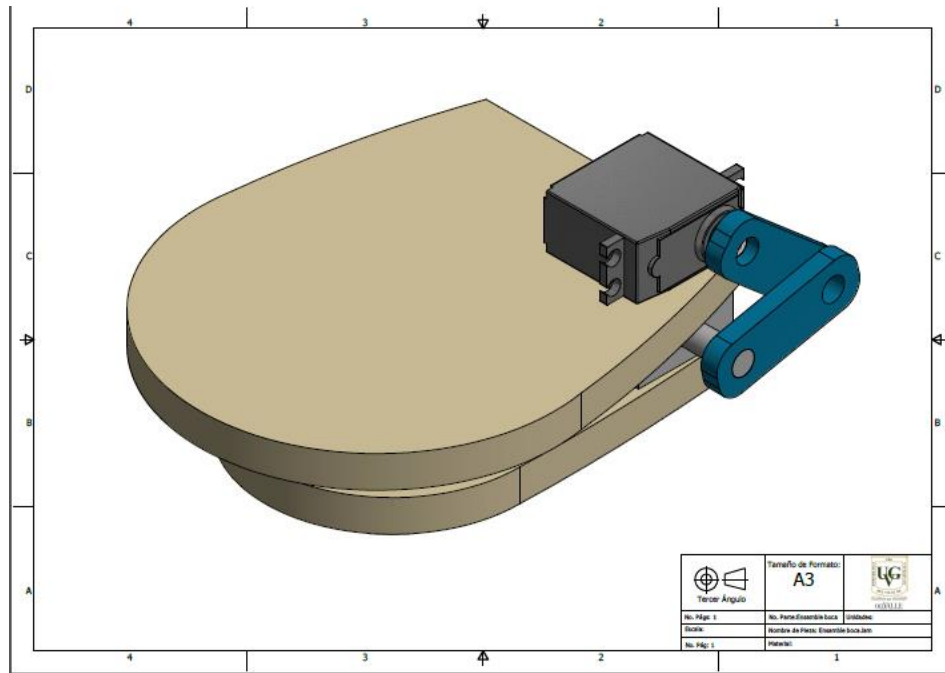
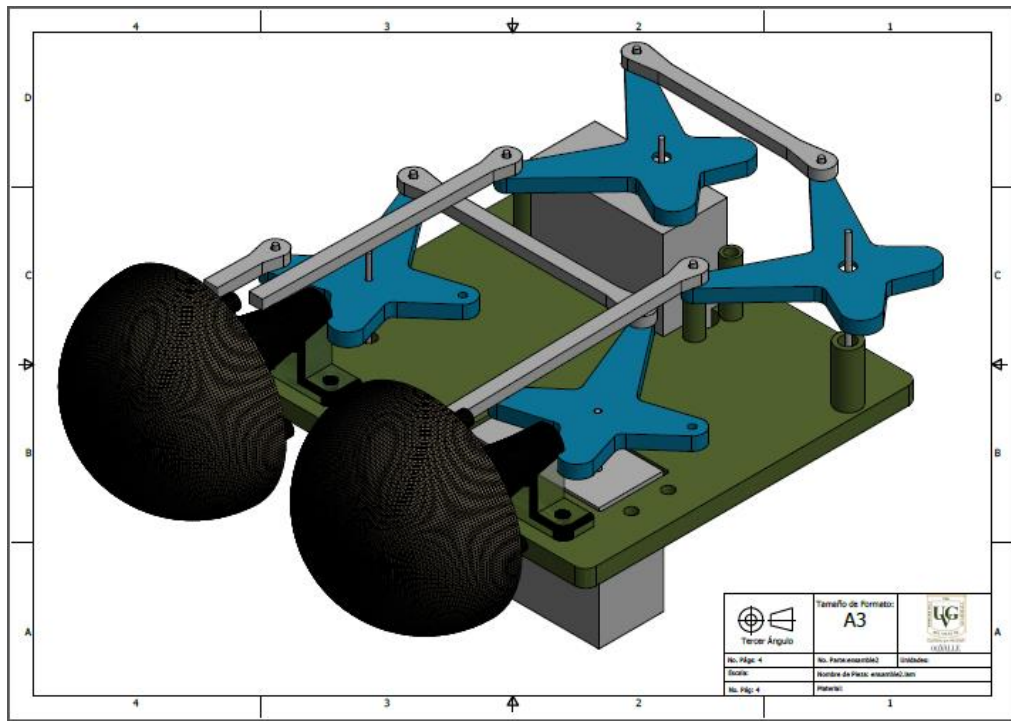
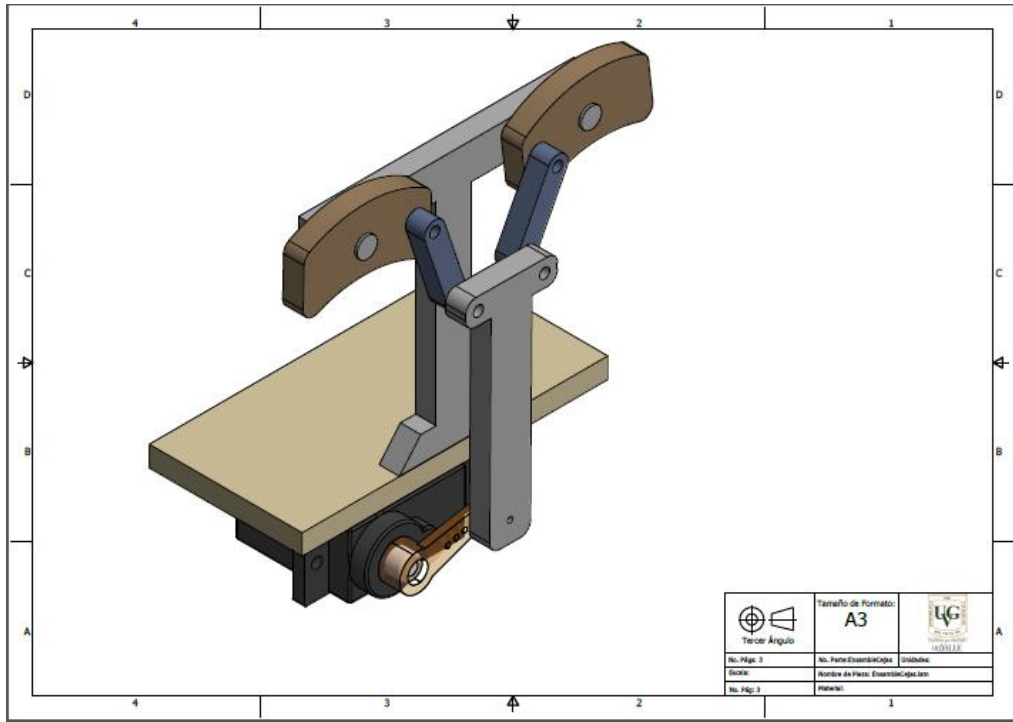


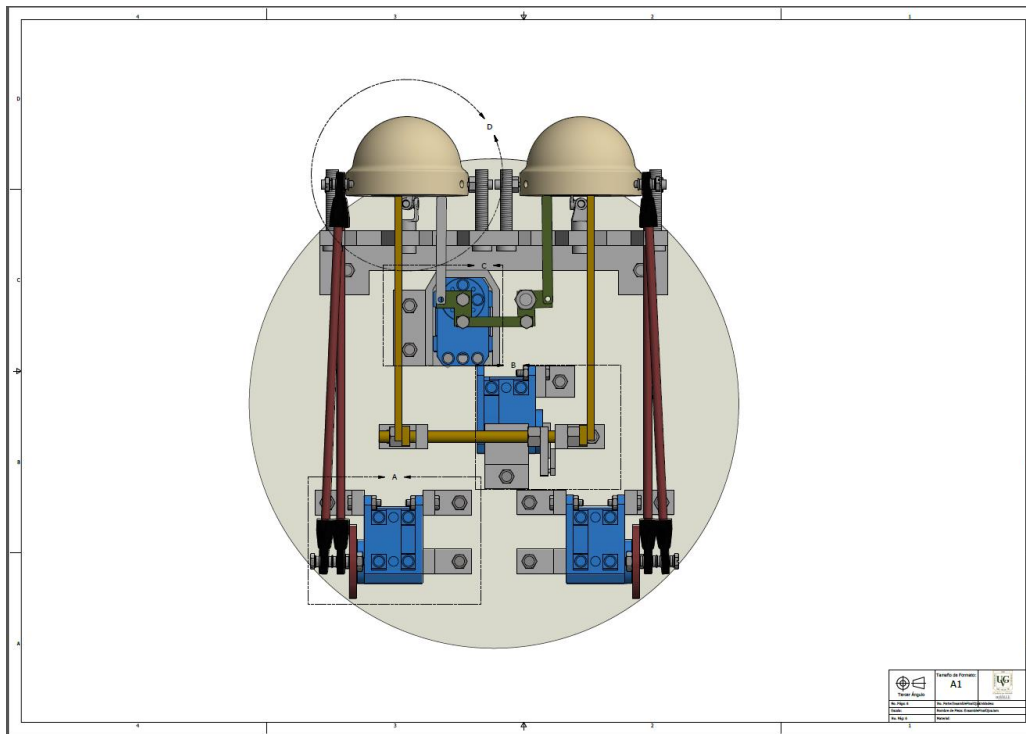
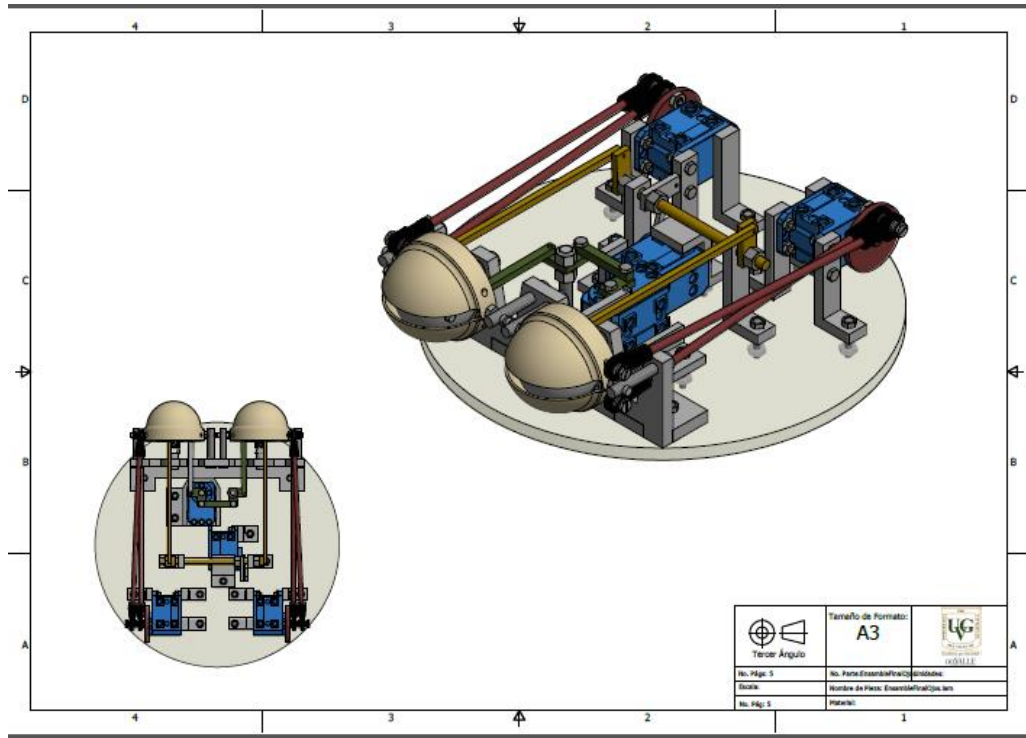
Figura 129. Estructura de los dedos

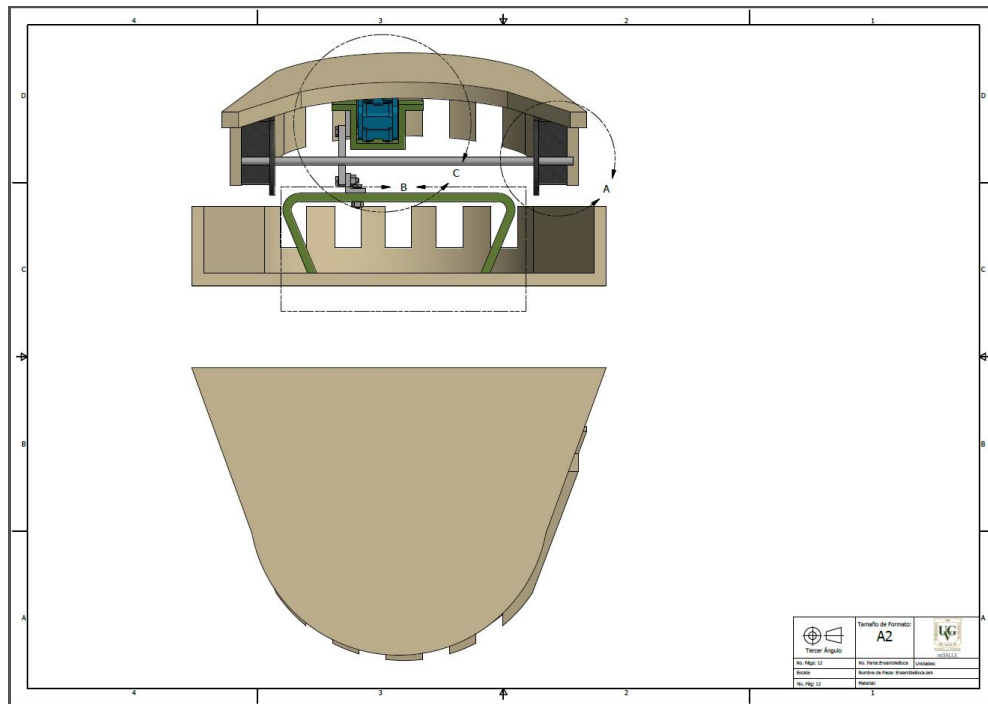
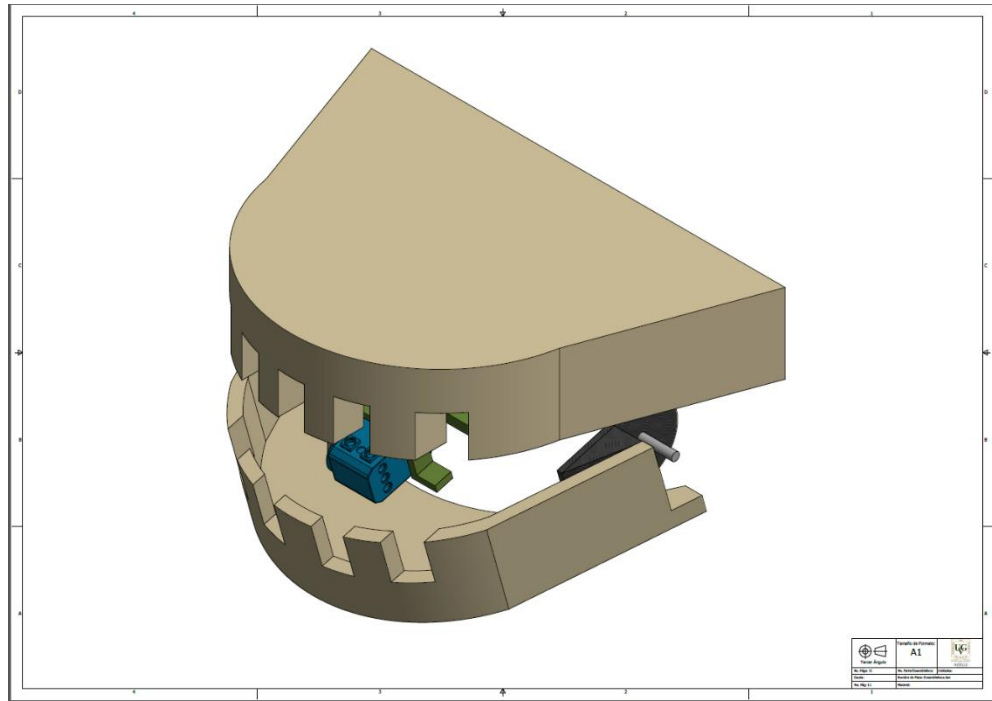


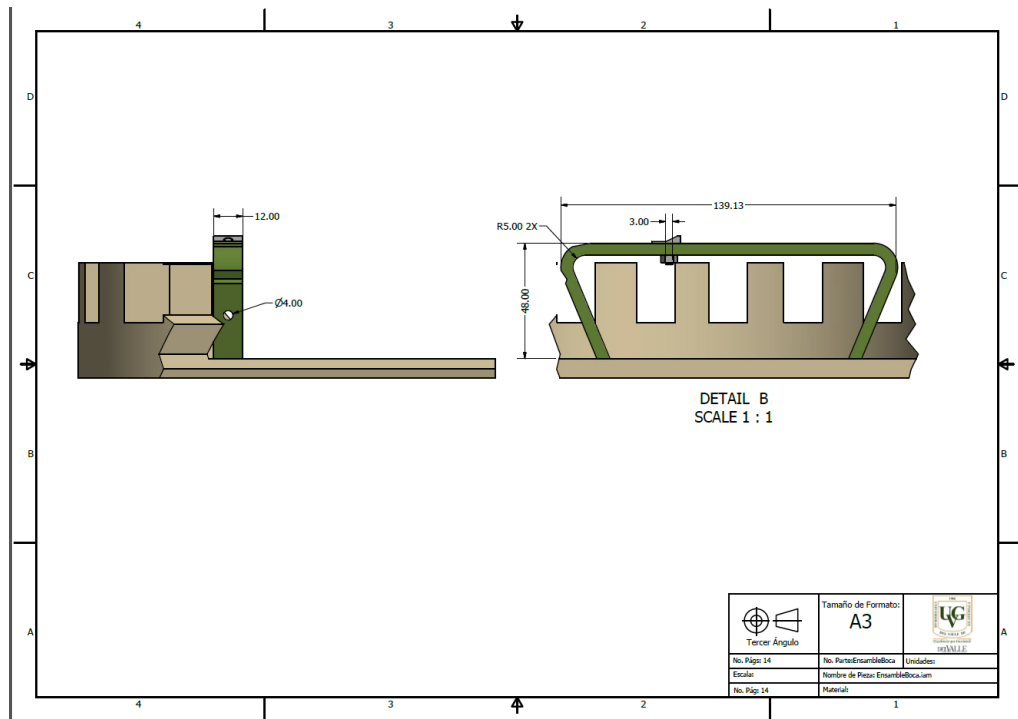
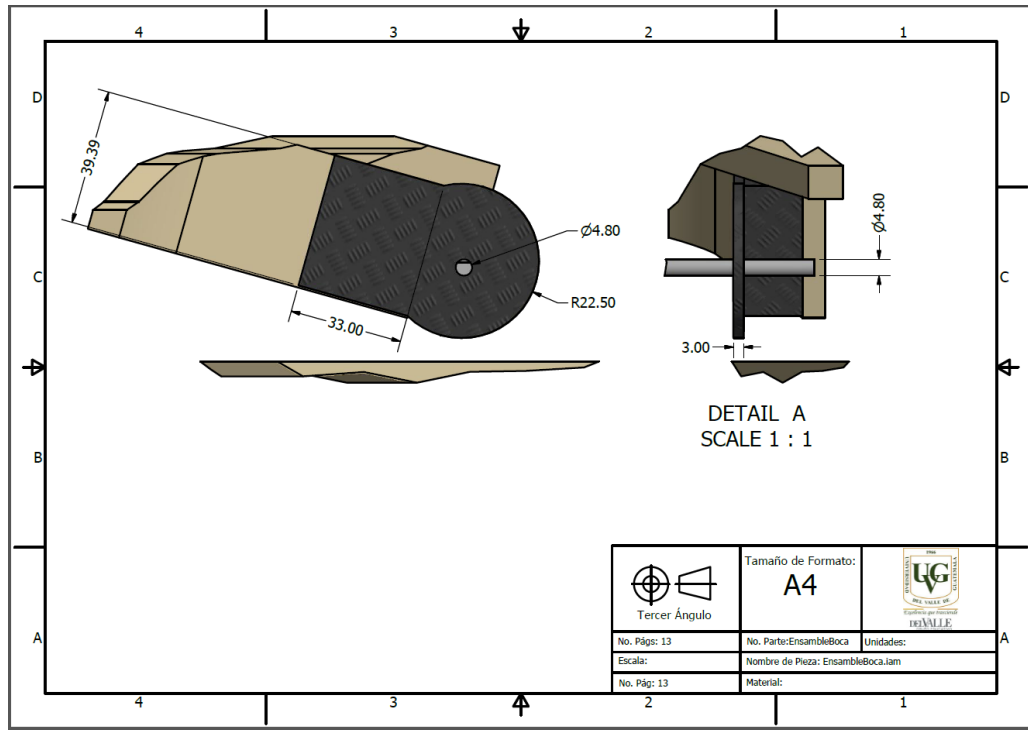
C. Planos mecanismos gestos faciales

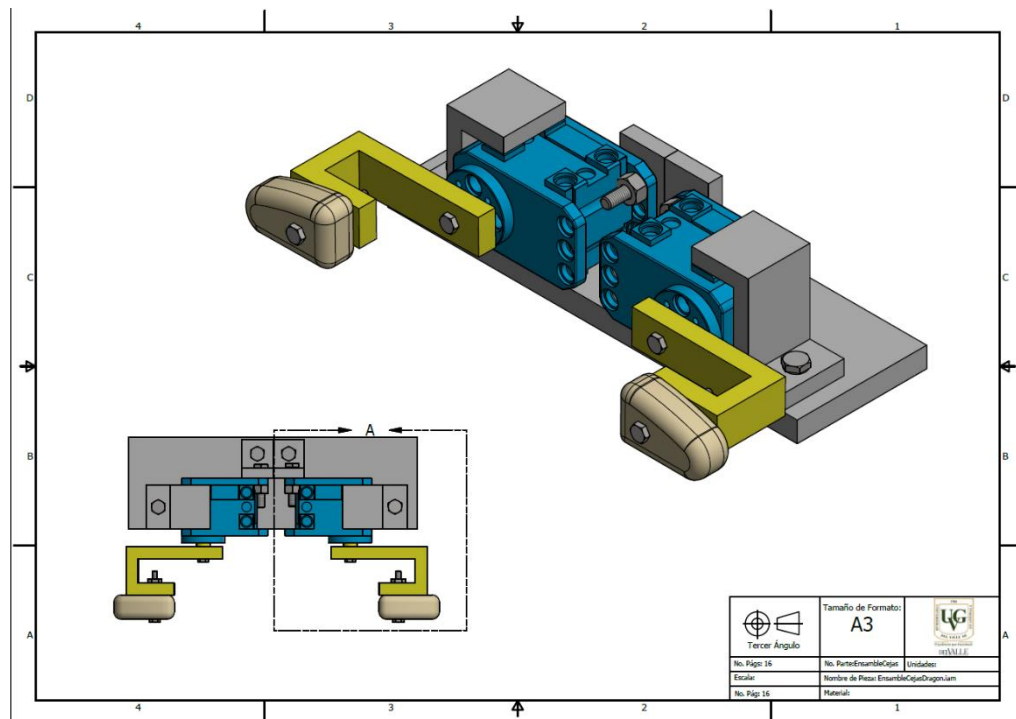
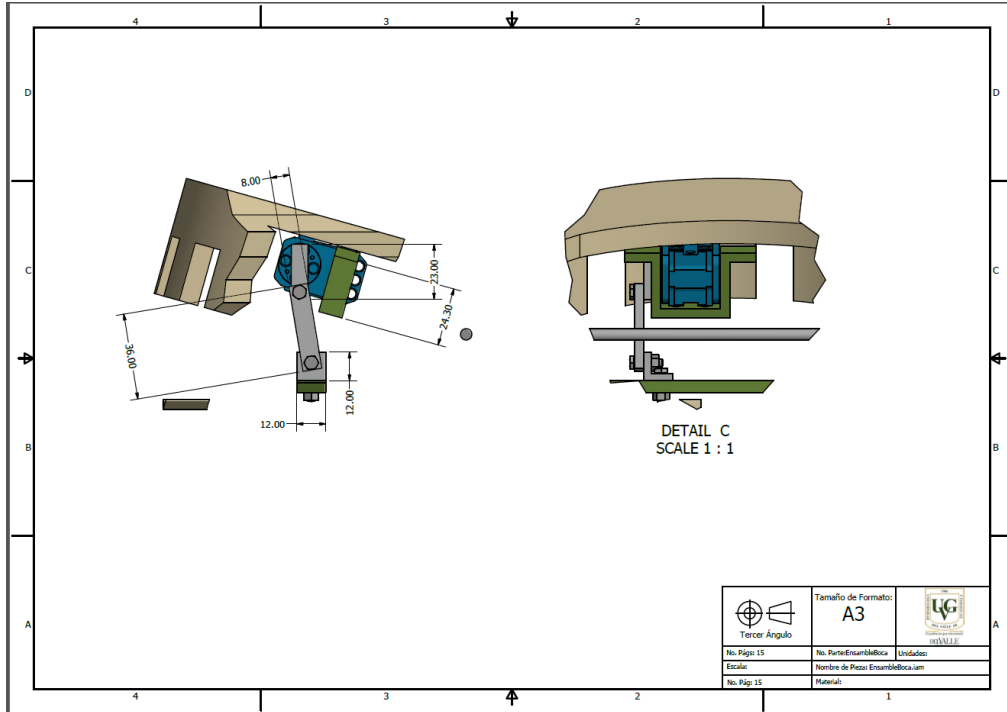












XIV. GLOSARIO

- **Abducción:** Es el movimiento de alejar un miembro del plano medio del cuerpo a través del plano frontal.
- **Aducción:** Es la aproximación al plano medio a través del plano frontal. Para los dedos se refiere a la aproximación entre ellos.
- **Animatrónico:** Se refiere a un robot interactivo ya sea con apariencia humana o animal con el fin de interactuar y proveer entretenimiento a las personas.
- **API:** Es el conjunto de definiciones de subrutinas, protocolos, y herramientas para construcción de software de aplicación. En términos generales, es el conjunto de métodos de comunicación claramente definidos entre varios componentes de software.
- **AX12A:** Es el servomotor, que es un actuador rotacional, ofrecido por Dynamixel (ROBOTIS Bioloid) de la versión actualizada de AX12+.
- **C:** Es de propósito general, de lenguaje de programación de computadora imperativa, el cual soporta programación estructurada, enfoque de variable léxico o recursión, mientras que el sistema de tipo estático que previene operaciones no deseados.
- **C++:** Es el lenguaje de programación de propósito general, que caracteriza por funciones de programaciones genéricas, imperativas y orientadas a objeto. Mientras que proveen las facilidades para manipulaciones de baja nivel de memoria.
- **CAD:** Se refiere al diseño asistido por computadora.
- **Cadena Cinemática:** Se refiere a un ensamble de eslabones y juntas interconectados de modo que produzcan un movimiento controlado en respuesta a un movimiento suministrado.
- **CCD:** Las siglas representan “Dispositivo de Carga Acoplada” es un dispositivo para el movimiento de carga eléctrica, usualmente desde el dispositivo al área en el cual la carga puede ser manipulado como conversión en un valor digital.
- **Circunducción:** Es la combinación de flexión, extensión, aducción y abducción que se traduce en el movimiento circular de un miembro formando un cono.
- **CMOS:** Las siglas son “Semiconductor Complementario de Óxido Metálico”, es la tecnología para construcción de circuitos integrados, y es usado en microprocesadores, microcontroladores, RAM estático u otros circuitos lógicos digitales.
- **Deformación:** Se refiere al resultado de la aplicación de una fuerza a un elemento de cierto material el cual si pasa la región elástica del mismo tiende a deformarse.
- **Dúctil:** Se refiere a los materiales que tienen alto coeficiente de elasticidad, es decir soportan altas deformaciones y vuelven a su estado original.
- **Esfuerzo:** Se refiere a la reacción causada por una fuerza en un área.
- **Eslabón:** Se refiere a un cuerpo rígido que posee por lo menos dos nodos que son puntos de unión con otros eslabones.

- eXtensible Markup Language: Es un lenguaje de marco que define un conjunto de reglas para codificación de documentos en un formato que es legible por humano y por la máquina. La especificación de W3C XML 1.0 y entre otras especificaciones, todos estos son estándares gratuitos abiertos, que define XML.
- Extensión: Es el aumento del ángulo entre dos partes del cuerpo que también se puede interpretar como el “enderezamiento” de dichas partes.
- FLANN: Es la librería para búsqueda de aproximación rápida de vecinos más cercanos en espacios de altas dimensiones. Está escrito en C++ y contiene enlaces de C, Matlab y Python.
- Flexión: Se refiere encorvamiento transitorio que experimenta un sólido elástico por la acción de una fuerza que lo deforma.
- FPS: Las siglas son “Frames per Second”, es la frecuencia o la tasa al cual las imágenes consecutivas llamadas marcos son desplegados en un despliegue animado.
- Frágil: Se refiere a un material que no soporta altos grados de deformación tienen a ser quebradizos y romperse rápido.
- gpu: Es el circuito electrónico especializado diseñado para manipular rápidamente y alterar la memoria para acelerar la creación de imágenes en un destinado en una pila de marcos para salida de un dispositivo de despliegue.
- Hardware: Es la parte física o componentes de una computadora, tales como la tarjeta de madre, tarjeta de video, almacenamiento de datos de computadora, tarjetas de sonido y entre otros periféricos físicos.
- Homogéneo: Se refiere a materiales los cuales poseen características de composición y estructura uniforme.
- ISO: Las siglas son “Estandarización de Organización Internacional”, el cual es el cuerpo principal que gobierna los estándares de las tasas de sensibilidades para sensores de las cámaras.
- Isotrópico: Se refiere a un material el cual sus propiedades no cambian en la dirección en que se le coloque.
- Junta: Se refiere a la conexión entre dos o más eslabones (en sus nodos), la cual permite algún movimiento, o movimiento potencial, entre los eslabones conectados
- Máquina: Se refiere al grupo de mecanismos que están diseñados para producir y transmitir fuerzas significativas.
- Matlab: De las siglas de “Matrix Laboratory”, es entorno de computación numérica de múltiples paradigmas, y es lenguaje de programación propietario desarrollado por MathWorks.
- MDF: Se refiere a la Madera de densidad media para ambientes secos.
- Mecanismo: Se refiere a un dispositivo que transforma el movimiento en un patrón deseable y por lo general desarrolla fuerzas muy bajas y transmite poca potencia.

- OpenCV: Es la librería de funciones de programaciones principalmente enfocadas en visión de computadora en tiempo real. Originalmente desarrollado por Intel y actualmente es mantenido por Itseez. La librería es la plataforma cruzado y uso es gratis bajo la licencia Open-Source BSD.
- PLA: Se refiere al ácido Poliláctico el cual es un plástico de baja densidad y fácil mecanizado.
- Pronación: Movimiento de la mano y el antebrazo basado en una rotación medial en la que la palma termina hacia el frente.
- Python: Es un lenguaje de programación de alto nivel ampliamente usado para programación de propósito general.
- RAM: Las siglas representan la memoria de acceso aleatorio y es la forma del almacenamiento de datos de computadora que almacena instrucciones del programa frecuentemente usado para aumentar la velocidad general del sistema.
- Rotación lateral: Es el giro de la parte superior de una extremidad.
- Rotación medial: Es el giro de la parte inferior de una extremidad.
- Rotación: Es el movimiento de una parte del cuerpo a través de su eje longitudinal.
- Software: Es la parte del sistema de computadora que consiste de datos o instrucciones de computadora, a diferencia a hardware físico al cual el sistema está construido.
- STT Engine: Es un motor para operación de conversión de Speech-to-Text o de voz a texto.
- Torsión: Se refiere a una fuerza que genera que el elemento sufra de un torcimiento bajo su línea de acción.
- Trade Study: Es la actividad de un equipo multidisciplinario para identificar las soluciones técnicas más equilibrados entre un conjunto de soluciones viables propuestos.
- TTS Engine: Es un motor para operación de conversión de Text-to-Speech o de texto a voz.
- XL320: Es el servomotor, que es un actuador rotacional, ofrecido por Dynamixel (ROBOTIS Bioloid).