



Diseño e implementación de una celda de manufactura remota basada en un robot SCARA con aplicación de visión artificial para su aplicación en los laboratorios remotos.

España Males, Iván Darío y Peñafiel Sánchez, Alejandro Ricardo

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Trabajo de titulación, previo a la obtención del título de Ingeniero en Mecatrónica

Ing. Alulema Flores, Darwin Omar

16 de enero de 2023



Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Certificación

Certifico que el trabajo de titulación: "Diseño e implementación de una celda de manufactura remota basada en un robot SCARA con aplicación de visión artificial para su aplicación en los laboratorios remotos." fue realizado por los señores España Males, Iván Darío y Peñafiel Sánchez, Alejandro Ricardo; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 16 de enero de 2023

Firma:



Escaneado digitalmente por:
DARWIN OMAR
ALULEMA
FLORES

.....
Alulema Flores, Darwin Omar

C. C. 1002493334



Departamento de Ciencias de la Energía y Mecánica
Carrera de Ingeniería en Mecatrónica

Responsabilidad de Autoría

Nosotros, **España Males, Iván Darío** y **Peñañiel Sánchez, Alejandro Ricardo**, con cédulas de ciudadanía n°1720652013 y 1723155782, declaramos que el contenido, ideas y criterios del trabajo de titulación: **"Diseño e implementación de una celda de manufactura remota basada en un robot SCARA con aplicación de visión artificial para su aplicación en los laboratorios remotos"** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 16 de enero de 2023

España Males, Iván Darío

C.C.: 1720652013

Peñañiel Sánchez, Alejandro Ricardo

C.C.: 1723155782



Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Autorización de Publicación

Nosotros **España Males, Iván Darío y Peñafiel Sánchez, Alejandro Ricardo**, con cédulas de ciudadanía n°1720652013 y 1723155782, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Diseño e implementación de una celda de manufactura remota basada en un robot SCARA con aplicación de visión artificial para su aplicación en los laboratorios remotos"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Sangolquí, 16 de enero de 2023

España Males, Iván Darío

C.C.: 1720652013

Peñafiel Sánchez, Alejandro Ricardo

C.C.: 1723155782

Dedicatoria

Iván España

A mis padres quienes me brindaron la oportunidad de tener educación y quienes son pilar de mi desarrollo como persona.

Alejandro Peñafiel

Dedico este trabajo a mis padres, pilares fundamentales en el desarrollo personal y académico de mi persona, quienes con su cariño y apoyo he logrado alcanzar mis metas.

Agradecimiento

Iván España

A mi familia que estuvo y está pendiente de mí todo el tiempo, especialmente a mis padres; a mis amigos y profesores que me ayudaron durante este proceso.

Alejandro Peñafiel

A mi familia y amigos quienes forman parte importante de mi vida durante todos estos años, además de los profesores que marcaron enseñanzas importantes en mi vida.

Tabla de contenido

Capítulo 1: Generalidades	32
Introducción.....	32
Justificación e Importancia.....	33
Área de influencia.....	34
Objetivos.....	34
<i>Objetivo general</i>	34
<i>Objetivos específicos</i>	34
Alcance.....	35
Estado del arte	37
<i>Mapeo sistemático de la literatura</i>	38
<i>Método de investigación</i>	38
<i>Preguntas de investigación</i>	39
<i>Establecer los criterios de búsqueda primarios y secundarios</i>	40
<i>Definir la cadena de búsqueda</i>	41
<i>Estrategia para la extracción de datos</i>	43
<i>Clasificación de los artículos obtenidos</i>	43
<i>Resultados</i>	44
<i>Conclusiones del SMS</i>	46
<i>Otras conclusiones</i>	48
<i>Revisión semántica de la literatura</i>	49

	10
<i>Método de investigación</i>	49
<i>Preguntas de investigación</i>	50
<i>Establecer los criterios de búsqueda</i>	51
<i>Fijar los criterios de inclusión y exclusión</i>	52
<i>Estrategia para la extracción de datos</i>	53
<i>Clasificación de los artículos obtenidos</i>	53
<i>Resultados</i>	53
<i>Conclusiones del SLR</i>	56
<i>Otras conclusiones</i>	57
Glosario de términos	58
Capítulo 2: Marco teórico.....	59
Robots manipuladores	59
<i>Robot SCARA</i>	64
Cinemática del robot	66
<i>Cinemática directa</i>	66
<i>Matrices de transformación</i>	67
<i>Parámetros Denavit-Hartenberg</i>	69
<i>Cinemática inversa</i>	71
<i>Método geométrico</i>	71
Celdas de manufactura inteligente	73
Internet de las cosas	75

	11
<i>Protocolos de comunicación</i>	76
<i>Message Queuing Telemetry Transport (MQTT)</i>	77
Servicios WEB.....	79
<i>Amazon Web Services</i>	80
Visión artificial	82
<i>Open Computer Vision (OpenCV)</i>	83
<i>Evaluación de los resultados</i>	84
Capítulo 3: Diseño	86
Desarrollo de conceptos	86
<i>Casa de la calidad</i>	86
<i>Conclusiones de la casa de la calidad</i>	88
Diseño del subsistema mecánico	90
<i>Mecanismo de brazo</i>	91
<i>Mecanismo del antebrazo</i>	95
<i>Mecanismo rotatorio del efector final</i>	100
<i>Mecanismo elevador del efector final</i>	104
<i>Mecanismo del efector final</i>	107
<i>Modelamiento de las selecciones</i>	110
Mecanismo del brazo.....	110
Mecanismo del antebrazo.....	115
Mecanismo elevador y rotatorio del efector final.	119

Mecanismo del efector final.....	121
<i>Validación del diseño</i>	123
<i>Dimensionamiento de motores</i>	129
Diseño del circuito electrónico	138
<i>Selección de sensores</i>	138
<i>Selección de microcontrolador y microcomputador</i>	145
<i>Circuito electrónico</i>	148
Diseño del módulo de visión artificial.....	151
<i>Detección de color y forma</i>	155
<i>Estimación de la distancia</i>	158
Modelado de las plantas	161
<i>Modelamiento de las plantas</i>	162
Cálculo de cinemática directa e inversa	168
Diseño de la infraestructura de comunicación	173
Diseño de la interfaz humano máquina	182
<i>Pantalla Inicio</i>	185
<i>Pantalla Control manual</i>	186
<i>Pantalla Visión artificial</i>	187
<i>Pantalla Monitorización</i>	189
Capítulo 4: Implementación.....	191
Subsistema mecánico.....	191

Circuito electrónico.....	203
Subsistema de visión artificial.....	213
Arquitectura de comunicación.....	217
Interfaz humano máquina	229
Capítulo 5: Verificaciones	234
Pruebas de funcionamiento	234
<i>Pruebas del sistema de visión por computador.....</i>	<i>234</i>
Pruebas de color.....	234
Pruebas de forma.	236
Prueba de estimación de distancia.	237
Precisión de clasificación.....	239
Pruebas de conexiones eléctricas	243
Pruebas del subsistema mecánico	249
<i>Fase 1</i>	<i>250</i>
<i>Fase 2</i>	<i>259</i>
Pruebas de usabilidad con usuarios.....	267
Pruebas de carga del sistema	270
<i>Prueba con 10 usuarios</i>	<i>274</i>
<i>Prueba con 50 usuarios</i>	<i>275</i>
<i>Prueba con 100 usuarios</i>	<i>276</i>
<i>Prueba con 200 usuarios</i>	<i>277</i>

<i>Prueba con 500 usuarios</i>	278
<i>Prueba con 800 usuarios</i>	279
Pruebas unitarias de Node-Red.	282
<i>Prueba 1: Mensaje recibido</i>	282
<i>Prueba 2: Formar mensajes de posiciones</i>	285
<i>Prueba 3: Cinemática inversa</i>	285
<i>Prueba 4: Completar mensaje SCARA</i>	286
<i>Resultados de las pruebas unitarias</i>	288
Capítulo 6: Resultados	290
Conclusiones	290
Recomendaciones	291
Trabajos futuros	292
Bibliografía	293
Apéndices	302

Índice de tablas

Tabla 1 Preguntas de investigación del SMS.....	40
Tabla 2 Cadenas de búsqueda	41
Tabla 3 Preguntas de investigación para el SLR	50
Tabla 4 Términos de búsqueda para el SLR.....	51
Tabla 5 Simbología para las relaciones de la matriz QFD.....	88
Tabla 6 Simbología utilizada en la matriz QFD	88
Tabla 7 Resultados de la evaluación de la matriz QFD	89
Tabla 8 Subsistemas que conforman la celda de manufactura	89
Tabla 9 Ventajas y desventajas de la alternativa 1 del brazo	92
Tabla 10 Ventajas y desventajas de la alternativa 2 del brazo	93
Tabla 11 Ventajas y desventajas de la alternativa 3 del brazo	94
Tabla 12 Matriz de evaluación para las alternativas de solución al mecanismo del brazo	95
Tabla 13 Ventajas y desventajas de la alternativa 1 del antebrazo.....	96
Tabla 14 Ventajas y desventajas de la alternativa 2 del antebrazo.....	97
Tabla 15 Ventajas y desventajas de la alternativa 3 del antebrazo.....	99
Tabla 16 Matriz de evaluación para las alternativas de solución al mecanismo del antebrazo .	99
Tabla 17 Ventajas y desventajas de la alternativa 1 del mecanismo rotatorio del efector.....	101
Tabla 18 Ventajas y desventajas de la alternativa 2 del mecanismo rotatorio del efector.....	102
Tabla 19 Ventajas y desventajas de la alternativa 3 del mecanismo rotatorio del efector.....	103
Tabla 20 Matriz de evaluación para las alternativas de solución al mecanismo rotatorio del efector.....	103
Tabla 21 Ventajas y desventajas de la alternativa 2 del mecanismo rotatorio del efector.....	105
Tabla 22 Ventajas y desventajas de la alternativa 3 del mecanismo rotatorio del efector.....	106
Tabla 23 Matriz de evaluación para las alternativas de solución al mecanismo rotatorio del efector.....	106

Tabla 24 <i>Ventajas y desventajas de la alternativa 1 del mecanismo del efector final</i>	108
Tabla 25 <i>Ventajas y desventajas de la alternativa 2 del mecanismo del efector final</i>	109
Tabla 26 <i>Matriz de evaluación para las alternativas de solución al mecanismo del efector final</i>	109
Tabla 27 <i>Resumen de deflexiones obtenidas en brazo y antebrazo</i>	124
Tabla 28 <i>Características del motor seleccionado</i>	131
Tabla 29 <i>Características del motor seleccionado</i>	133
Tabla 30 <i>Características del motor seleccionado para el mecanismo rotatorio</i>	135
Tabla 31 <i>Características del motor seleccionado para el mecanismo elevador</i>	137
Tabla 32 <i>Ventajas y desventajas de la alternativa 1 del sensor de posición angular</i>	139
Tabla 33 <i>Ventajas y desventajas de la alternativa 2 del sensor de posición angular</i>	141
Tabla 34 <i>Matriz de evaluación para las alternativas de sensores para medición de posición angular</i>	141
Tabla 35 <i>Ventajas y desventajas de la alternativa 1 del sensor de distancia</i>	142
Tabla 36 <i>Ventajas y desventajas de la alternativa 2 del sensor de distancia</i>	143
Tabla 37 <i>Matriz de evaluación para las alternativas de sensores para medición de distancia</i>	144
Tabla 38 <i>Ventajas y desventajas de la alternativa 1 para microcontrolador</i>	146
Tabla 39 <i>Ventajas y desventajas de la alternativa 2 para microcontrolador</i>	147
Tabla 40 <i>Matriz de evaluación para las alternativas de placa microcontrolador</i>	147
Tabla 41 <i>Anchos de pista requeridos en base a la corriente nominal obtenida de hojas de datos</i>	150
Tabla 42 <i>Ventajas y desventajas de la alternativa 1 para el sistema de visión artificial</i>	151
Tabla 43 <i>Ventajas y desventajas de la alternativa 1 para el sistema de visión artificial</i>	152
Tabla 44 <i>Matriz de evaluación para las alternativas del sistema de visión artificial</i>	153
Tabla 45 <i>Resumen de la identificación de las funciones de transferencia</i>	168
Tabla 46 <i>Parámetros de Denavit - Hartenberg</i>	169

Tabla 47 <i>Ventajas y desventajas de la alternativa 1 para el microordenador</i>	175
Tabla 48 <i>Ventajas y desventajas de la alternativa 2 para el microordenador</i>	176
Tabla 49 <i>Matriz de evaluación para las alternativas de placa microordenador</i>	177
Tabla 50 <i>Ventajas y desventajas de la alternativa 1 de plataforma de servicios web</i>	178
Tabla 51 <i>Ventajas y desventajas de la alternativa 2 de plataforma de servicios web</i>	179
Tabla 52 <i>Matriz de evaluación para las alternativas de placa microordenador</i>	179
Tabla 53 <i>Matriz de observación de color</i>	235
Tabla 54 <i>Matriz de observación de formas</i>	236
Tabla 55 <i>Matriz de observación de estimación de distancia</i>	238
Tabla 56 <i>Matriz de observación combinada</i>	238
Tabla 57 <i>Resultados de las pruebas de precisión realizadas</i>	239
Tabla 58 <i>Estadísticas descriptivas de la precisión de calibración</i>	241
Tabla 59 <i>Comparación de valores de corriente reales y simulados</i>	249
Tabla 60 <i>Valores obtenidos de la lectura de ángulos en el brazo (GDL Q1)</i>	250
Tabla 61 <i>Resultado de las estadísticas descriptivas del GDL Q1</i>	251
Tabla 62 <i>Valores obtenidos de la lectura de ángulos en el antebrazo (GDL Q2)</i>	254
Tabla 63 <i>Estadísticas descriptivas del GDL Q2</i>	255
Tabla 64 <i>Valores de la medición de los GDL Q1 y Q2 de forma simultánea</i>	260
Tabla 65 <i>Estadísticas descriptivas de los GDL Q1 y Q2</i>	261
Tabla 66 <i>Resultados de la prueba de usabilidad</i>	268
Tabla 67 <i>Tabla de puntajes en la escala SUS</i>	269
Tabla 68 <i>Tabla resumen de las pruebas de carga</i>	280

Índice de figuras

Figura 1 <i>Arquitectura de comunicación implementada</i>	36
Figura 2 <i>Mapa relacional generado por VosViewer</i>	37
Figura 3 <i>Esquema del proceso de mapeo de información</i>	38
Figura 4 <i>Procedimiento del mapeo sistemático</i>	39
Figura 5 <i>Fases de clasificación de artículos del SMS</i>	44
Figura 6 <i>Publicaciones por año sobre celdas de manufactura remotas</i>	44
Figura 7 <i>Distribución geográfica de los artículos de interés de la fase 2 del SMS</i>	45
Figura 8 <i>Distribución geográfica de los artículos de interés de la fase 2 del SMS</i>	45
Figura 9 <i>Procedimiento del mapeo sistemático</i>	49
Figura 10 <i>Fases de clasificación de artículos del SLR</i>	53
Figura 11 <i>Publicaciones anuales de celdas de manufactura remotas</i>	54
Figura 12 <i>Publicaciones de laboratorios remotos desde 2017</i>	54
Figura 13 <i>Cuantificación de los artículos obtenidos de la fase 3 del SLR</i>	55
Figura 14 <i>Robot PUMA</i>	60
Figura 15 <i>Prótesis robótica de brazo</i>	60
Figura 16 <i>Tipos de articulaciones</i>	61
Figura 17 <i>Elementos que conforman a los robots manipuladores</i>	63
Figura 18 <i>Diferencia entre exactitud y precisión</i>	64
Figura 19 <i>Robot SCARA</i>	64
Figura 20 <i>Espacio de trabajo de manipulador RRP</i>	65
Figura 21 <i>Transformación realizada por la cinemática directa</i>	66
Figura 22 <i>Representación de los sistemas de referencia de los grados de libertad de un robot SCARA</i>	67
Figura 23 <i>Estructura de las matrices de rotación</i>	68

Figura 24 Estructura de las matrices homogéneas.....	68
Figura 25 Trazado del eje Z para la representación de Denavit - Hartenberg.....	69
Figura 26 Trazado del eje X para la representación de Denavit - Hartenberg.....	70
Figura 27 Parámetros de Denavit - Hartenberg.....	70
Figura 28 Transformación realizada por la cinemática inversa	71
Figura 29 Representación esquemática de los eslabones de un manipulador planar RR (primera solución).....	72
Figura 30 Representación esquemática de los eslabones de un manipulador planar RR (solución 2).....	72
Figura 31 Ejemplo de celda de manufactura.....	73
Figura 32 Proceso de producción automatizado.....	74
Figura 33 Implementación de IoT en la industria	76
Figura 34 Estructura del modelo OSI	77
Figura 35 Estructura de los mensajes en el protocolo MQTT.....	78
Figura 36 Proceso de publicación de mensajes por MQTT.....	78
Figura 37 Funcionamiento de los servicios web	79
Figura 38 Servicios ofrecidos por Amazon Web Services	80
Figura 39 Funcionamiento del servicio Amazon IoT Core	81
Figura 40 Funcionamiento del servicio Amazon S3.....	82
Figura 41 Brazo robótico con sistema de visión artificial	83
Figura 42 Matriz de confusión para sistemas de visión artificial.....	84
Figura 43 Matriz QFD	87
Figura 44 Requerimientos del subsistema mecánico	91
Figura 45 Alternativa 1 para el mecanismo de brazo.....	92
Figura 46 Alternativa 2 para el mecanismo de brazo.....	93
Figura 47 Alternativa 3 para el mecanismo de brazo.....	94

Figura 48 <i>Alternativa 1 para el mecanismo de antebrazo</i>	96
Figura 49 <i>Alternativa 2 para el mecanismo de antebrazo</i>	97
Figura 50 <i>Alternativa 3 para el mecanismo de antebrazo</i>	98
Figura 51 <i>Alternativa 1 para el mecanismo rotatorio del efector</i>	100
Figura 52 <i>Alternativa 2 para el mecanismo rotatorio del efector</i>	101
Figura 53 <i>Alternativa 3 para el mecanismo rotatorio del efector</i>	102
Figura 54 <i>Alternativa 1 para el mecanismo elevador del efector final</i>	104
Figura 55 <i>Alternativa 3 para el mecanismo elevador del efector final</i>	105
Figura 56 <i>Alternativa 1 para el mecanismo del efector final</i>	107
Figura 57 <i>Alternativa 2 para el mecanismo del efector final</i>	108
Figura 58 <i>Modelado 3D del mecanismo del brazo</i>	110
Figura 59 <i>Dimensiones del mecanismo del brazo</i>	111
Figura 60 <i>Dimensiones del mecanismo del brazo (Soporte del motor)</i>	111
Figura 61 <i>Simulación de esfuerzos de von Mises para el mecanismo del brazo</i>	112
Figura 62 <i>Simulación de desplazamientos para el mecanismo del brazo</i>	112
Figura 63 <i>Simulación de factores de seguridad para el mecanismo del brazo</i>	113
Figura 64 <i>Simulación de esfuerzos de von Mises para el acople del motor del brazo</i>	113
Figura 65 <i>Simulación de desplazamiento para el acople del motor del brazo</i>	114
Figura 66 <i>Simulación de factores de seguridad para el acople del motor del brazo</i>	114
Figura 67 <i>Mecanismo del antebrazo</i>	115
Figura 68 <i>Simulación de esfuerzos de von Mises para el mecanismo del antebrazo</i>	115
Figura 69 <i>Simulación de desplazamientos para el mecanismo del antebrazo</i>	116
Figura 70 <i>Simulación de factores de seguridad para el mecanismo del antebrazo</i>	116
Figura 71 <i>Acople del motor del antebrazo</i>	117
Figura 72 <i>Simulación de esfuerzos de von Mises para el acople del motor del antebrazo</i>	117
Figura 73 <i>Simulación de desplazamientos para el acople del motor del antebrazo</i>	118

Figura 74	<i>Simulación de factores de seguridad para el acople del motor del antebrazo</i>	118
Figura 75	<i>Mecanismo elevador y rotatorio del efector</i>	119
Figura 76	<i>Simulación de esfuerzos de von Mises para el mecanismo elevador rotatorio</i>	120
Figura 77	<i>Simulación de desplazamientos para el mecanismo elevador rotatorio</i>	120
Figura 78	<i>Simulación de factores de seguridad para el mecanismo elevador rotatorio</i>	121
Figura 79	<i>Mecanismo del efector final</i>	121
Figura 80	<i>Simulación de esfuerzos de von Mises para el mecanismo del efector final</i>	122
Figura 81	<i>Simulación de desplazamientos para el mecanismo del efector final</i>	122
Figura 82	<i>Simulación de factores de seguridad para el mecanismo del efector final</i>	123
Figura 83	<i>Mallado adaptativo del mecanismo del brazo</i>	124
Figura 84	<i>Curva de convergencia de la simulación para el mecanismo del brazo</i>	125
Figura 85	<i>Mallado adaptativo del acople del motor del brazo</i>	125
Figura 86	<i>Curva de convergencia de la simulación para el acople del motor del brazo</i>	125
Figura 87	<i>Mallado adaptativo del mecanismo del antebrazo</i>	126
Figura 88	<i>Curva de convergencia de la simulación para el mecanismo del antebrazo</i>	126
Figura 89	<i>Mallado adaptativo del acople del motor del antebrazo</i>	126
Figura 90	<i>Curva de convergencia de la simulación para el acople del motor del antebrazo</i>	127
Figura 91	<i>Mallado adaptativo del mecanismo elevador rotatorio</i>	127
Figura 92	<i>Curva de convergencia de la simulación para el mecanismo elevador rotatorio</i>	128
Figura 93	<i>Mallado adaptativo del mecanismo del efector final</i>	128
Figura 94	<i>Curva de convergencia de la simulación para el mecanismo del efector final</i>	128
Figura 95	<i>Propiedades de masa del mecanismo del brazo</i>	130
Figura 96	<i>Motor seleccionado para el brazo</i>	131
Figura 97	<i>Propiedades de masa para la estructura del antebrazo</i>	132
Figura 98	<i>Motor seleccionado para el antebrazo</i>	133
Figura 99	<i>Propiedades de masa de la estructura del mecanismo rotatorio</i>	134

Figura 100 <i>Motor seleccionado para el mecanismo rotatorio</i>	135
Figura 101 <i>Propiedades de masa del mecanismo rotatorio</i>	136
Figura 102 <i>Motor seleccionado para el mecanismo elevador</i>	137
Figura 103 <i>Sensor tipo encoder incremental</i>	139
Figura 104 <i>Sensor tipo encoder óptico</i>	140
Figura 105 <i>Sensor ultrasónico HC-SR04</i>	142
Figura 106 <i>Sensor infrarrojo tipo SHARP</i>	143
Figura 107 <i>Interruptores finales de carrera del robot SCARA</i>	145
Figura 108 <i>Placa ESP-WROOM-32</i>	146
Figura 109 <i>Placa Arduino MEGA 2560</i>	147
Figura 110 <i>Diagrama de conexiones eléctricas</i>	149
Figura 111 <i>Diseño de la placa de circuitos</i>	150
Figura 112 <i>Esquema de la alternativa 1 para el sistema de visión artificial</i>	151
Figura 113 <i>Esquema de la alternativa 1 para el sistema de visión artificial</i>	152
Figura 114 <i>Proceso de umbralización de las imágenes</i>	154
Figura 115 <i>Detección de vértices del espacio de trabajo</i>	154
Figura 116 <i>Corrección de perspectiva de la imagen</i>	155
Figura 117 <i>Determinación de valores del filtro HSV</i>	156
Figura 118 <i>Imagen con filtro HSV aplicado</i>	156
Figura 119 <i>Contornos detectados dentro de la imagen</i>	157
Figura 120 <i>Objetos detectados e identificados por forma y color</i>	158
Figura 121 <i>Objetos detectados dentro de la imagen</i>	159
Figura 122 <i>Obtención de forma, color coordenadas de los centros de los objetos en la imagen</i>	160
Figura 123 <i>Determinación de objetos alcanzables e inalcanzables por el robot SCARA</i>	161
Figura 124 <i>Señales de salida y entrada del brazo del robot</i>	162

Figura 125 <i>Características de la función de transferencia del brazo obtenida por modelamiento</i>	163
Figura 126 <i>Seguimiento de la salida de la función de transferencia del brazo hacia los datos medidos</i>	164
Figura 127 <i>Respuesta en el tiempo de la función de transferencia del brazo</i>	164
Figura 128 <i>Señales de salida y entrada del antebrazo</i>	165
Figura 129 <i>Características de la función de transferencia del antebrazo obtenida por modelamiento</i>	166
Figura 130 <i>Seguimiento de la salida de la función de transferencia del brazo hacia los datos medidos</i>	166
Figura 131 <i>Respuesta en el tiempo de la función de transferencia del antebrazo</i>	167
Figura 132 <i>Esquema del robot SCARA basado en el método Denavit - Hartenberg</i>	169
Figura 133 <i>Matrices homogéneas individuales del robot SCARA</i>	170
Figura 134 <i>Matriz homogénea total del robot SCARA</i>	170
Figura 135 <i>Matriz homogénea del robot SCARA con valores de longitud</i>	171
Figura 136 <i>Esquema de los eslabones del robot SCARA</i>	172
Figura 137 <i>Esquema de la arquitectura de comunicación</i>	173
Figura 138 <i>Placa Raspberry Pi Zero</i>	175
Figura 139 <i>Placa Raspberry Pi 3B+</i>	176
Figura 140 <i>Arquitectura de comunicación completa</i>	180
Figura 141 <i>Contenido de las pantallas de la HMI</i>	184
Figura 142 <i>Esquema de la pantalla Inicio</i>	186
Figura 143 <i>Esquema de la pantalla Control Manual</i>	187
Figura 144 <i>Esquema de la pantalla Visión artificial</i>	189
Figura 145 <i>Esquema de la pantalla Monitorización del robot</i>	190
Figura 146 <i>Soporte del motor del mecanismo elevador</i>	191

Figura 147 <i>Acople del motor del antebrazo y soportes de encoders</i>	192
Figura 148 <i>Piezas del mecanismo del efector final</i>	192
Figura 149 <i>Piezas del mecanismo rotatorio</i>	193
Figura 150 <i>Chumaceras utilizadas en brazo y antebrazo</i>	193
Figura 151 <i>Derecha: Soporte de la estructura de los indicadores LED. Izquierda: Cubiertas de los indicadores LED</i>	194
Figura 152 <i>Proceso de impresión 3D de las piezas de cubierta</i>	195
Figura 153 <i>Piezas de cubierta terminadas</i>	195
Figura 154 <i>Cubierta del antebrazo del robot SCARA</i>	196
Figura 155 <i>Tubo redondo metálico utilizado para la base del robot</i>	197
Figura 156 <i>Piezas de base del robot y del antebrazo cortadas por láser</i>	197
Figura 157 <i>Platina utilizada para la estructura de soporte del eje del brazo</i>	198
Figura 158 <i>Construcción de la estructura mecánica del robot SCARA</i>	198
Figura 159 <i>Parte interna del antebrazo del robot SCARA</i>	199
Figura 160 <i>Interior del antebrazo del robot SCARA</i>	199
Figura 161 <i>Robot SCARA construido</i>	200
Figura 162 <i>Estructura de soporte de la cámara y reflector</i>	200
Figura 163 <i>Soporte para la estructura de los indicadores LED</i>	201
Figura 164 <i>Robot SCARA situado en la mesa de trabajo</i>	202
Figura 165 <i>Piezas utilizadas para el proceso de recolección de objetos</i>	202
Figura 166 <i>Placa electrónica del antebrazo</i>	203
Figura 167 <i>Proceso de grabado de la placa electrónica que se aloja en la caja de control</i>	204
Figura 168 <i>Placa electrónica después de retirar cobre excedente</i>	204
Figura 169 <i>Placa electrónica ubicada en la caja de control</i>	205
Figura 170 <i>Drivers de los motores del brazo y antebrazo</i>	206
Figura 171 <i>Circuito de los indicadores LED</i>	206

Figura 172 Interior de la caja de control.....	207
Figura 173 Parte posterior de la caja de control	207
Figura 174 Diagrama GRAFCET del funcionamiento del robot SCARA	208
Figura 175 Diagrama de flujo de la programación del microcontrolador Arduino	209
Figura 176 Estado 0 de operación del robot SCARA.....	209
Figura 177 Estado 1 de operación del robot SCARA.....	210
Figura 178 Estado 2 de operación del robot SCARA.....	210
Figura 179 Estado 3 de operación del robot SCARA.....	210
Figura 180 Estado 4 de operación del robot SCARA.....	211
Figura 181 Estado 5 de operación del robot SCARA.....	211
Figura 182 Función para poner en la posición inicial al robot.....	212
Figura 183 Función que separa el mensaje recibido por el puerto serial.....	212
Figura 184 Función que forma el mensaje con las lecturas de los sensores y lo envía por el puerto serial	213
Figura 185 Diagrama de flujo de la programación del algoritmo de visión artificial	215
Figura 186 Umbrales en espacio HSV para filtro del espacio de trabajo y figuras	216
Figura 187 Preprocesamiento y binarización por el método de Canny.....	216
Figura 188 Transformación de la imagen para normalizar en el espacio de trabajo	216
Figura 189 Detección y clasificación de figuras válidas en el array de trabajo	217
Figura 190 Cálculo de la distancia de los objetos de prueba al centro del robot SCARA	217
Figura 191 Menú principal del servicio Amazon EC2	218
Figura 192 Menú de creación de instancias de Amazon EC2	218
Figura 193 Instalación del Node – RED en la instancia de Amazon EC2	219
Figura 194 Asignación de IP elástica a la instancia.....	220
Figura 195 Configuración del servicio Amazon IoT Core	221
Figura 196 Nodos de envío de información por MQTT.....	221

Figura 197 Configuración del broker MQTT en Node - RED.....	222
Figura 198 Asignación de las credenciales de Amazon IoT Core en Node - RED.....	222
Figura 199 Nodos utilizados para la recepción de instrucciones.....	223
Figura 200 Nodos utilizados para el uso de la visión artificial	223
Figura 201 Nodos utilizados para el control general.....	224
Figura 202 Nodos utilizados para la recepción de información	225
Figura 203 Nodos utilizados para el control manual.....	226
Figura 204 Nodos utilizados para el manejo de la visión artificial	227
Figura 205 Nodos de cámaras de visión artificial y monitorización	228
Figura 206 Nodos utilizados para el control general.....	228
Figura 207 Página de inicio de sesión.....	229
Figura 208 Pantalla de Inicio	230
Figura 209 Pantalla de Monitorización del robot.....	231
Figura 210 Pantalla de visión artificial.....	232
Figura 211 Pantalla de Control manual.....	233
Figura 212 Proceso de detección de colores	235
Figura 213 Proceso de detección de formas.....	236
Figura 214 Proceso de estimación de distancia	237
Figura 215 Histograma de precisión.....	241
Figura 216 Diagrama de caja de la precisión.....	242
Figura 217 Gráfica de normalidad de los valores de precisión.....	242
Figura 218 Simulación del circuito electrónico.....	244
Figura 219 Simulación del consumo de corriente de los motores de los grados de libertad...	245
Figura 220 Simulación del circuito de los indicadores LED.....	245
Figura 221 Simulación del consumo de corriente del microordenador	246

Figura 222 <i>Medición conjunta de corriente de los motores de los grados de libertad Q1, Q2 y Q3.....</i>	247
Figura 223 <i>Medición de corriente del motor a pasos.....</i>	247
Figura 224 <i>Medición de consumo de corriente del microordenador.....</i>	248
Figura 225 <i>Medición del consumo de corriente del circuito de los LED's indicadores</i>	248
Figura 226 <i>Histograma de los valores de error porcentual del GDL Q1</i>	252
Figura 227 <i>Gráfica de dispersión de la precisión del GDL Q1</i>	252
Figura 228 <i>Gráfica de normalidad de la precisión del GDL Q1</i>	253
Figura 229 <i>Histograma de valores de precisión del GDL Q1</i>	253
Figura 230 <i>Prueba de normalidad de los valores de precisión de Q1.....</i>	254
Figura 231 <i>Histograma del error porcentual del GDL Q2</i>	256
Figura 232 <i>Gráfica de dispersión de la precisión del GDL Q2</i>	257
Figura 233 <i>Prueba de normalidad del error absoluto del GDL Q2</i>	257
Figura 234 <i>Histograma de precisión del GDL Q2</i>	258
Figura 235 <i>Gráfica de normalidad de la precisión del GDL Q2</i>	258
Figura 236 <i>Diagrama de caja de la precisión de los GDL Q1 y Q2.....</i>	259
Figura 237 <i>Histograma de error porcentual del GDL Q1</i>	262
Figura 238 <i>Histograma de error porcentual del GDL Q2</i>	263
Figura 239 <i>Histograma de precisión del GDL Q1</i>	263
Figura 240 <i>Histograma de precisión del GDL Q2</i>	264
Figura 241 <i>Gráfico de caja de la precisión de Q1 y Q2</i>	264
Figura 242 <i>Gráfica de dispersión de la precisión de Q1.....</i>	265
Figura 243 <i>Gráfica de dispersión de la precisión de Q2.....</i>	265
Figura 244 <i>Gráfica de normalidad de la precisión de Q1</i>	266
Figura 245 <i>Gráfica de normalidad de la precisión de Q2</i>	266
Figura 246 <i>Escala SUS.....</i>	269

Figura 247 <i>Prueba de carga de la página web de Node - RED</i>	271
Figura 248 <i>Configuración de la herramienta Gatling</i>	271
Figura 249 <i>Código generado de la prueba de carga</i>	272
Figura 250 <i>Código generado de la prueba de carga (Escenario de simulación)</i>	272
Figura 251 <i>Código generado de la prueba de carga (Usuarios a simular)</i>	273
Figura 252 <i>Prueba de carga con 1000 usuarios</i>	273
Figura 253 <i>Número de peticiones por segundo para 1000 usuarios</i>	274
Figura 254 <i>Prueba de carga con 10 usuarios</i>	274
Figura 255 <i>Número de peticiones por segundo para 10 usuarios</i>	275
Figura 256 <i>Prueba de carga con 50 usuarios</i>	275
Figura 257 <i>Número de peticiones por segundo para 50 usuarios</i>	276
Figura 258 <i>Prueba de carga con 100 usuarios</i>	276
Figura 259 <i>Número de peticiones por segundo para 100 usuarios</i>	277
Figura 260 <i>Prueba de carga con 200 usuarios</i>	277
Figura 261 <i>Prueba de carga con 200 usuarios</i>	278
Figura 262 <i>Prueba de carga con 500 usuarios</i>	278
Figura 263 <i>Número de peticiones por segundo para 500 usuarios</i>	279
Figura 264 <i>Prueba de carga con 800 usuarios</i>	279
Figura 265 <i>Número de peticiones por segundo para 800 usuarios</i>	280
Figura 266 <i>Gráfica de dispersión de las peticiones de las pruebas de carga</i>	281
Figura 267 <i>Nodos de recepción de instrucciones de la visión artificial</i>	282
Figura 268 <i>Resultado esperado de la primera prueba unitaria</i>	283
Figura 269 <i>Algoritmo de comparativa del resultado obtenido con el esperado de la primera función</i>	283
Figura 270 <i>Algoritmo de comparativa del resultado obtenido con el esperado de la segunda función</i>	284

Figura 271 <i>Algoritmo de comparativa del resultado obtenido con el esperado de la tercera función.....</i>	284
Figura 272 <i>Flujo implementado en Node RED para formar mensajes.....</i>	285
Figura 273 <i>Algoritmo de comparativa del resultado obtenido con el esperado de la función de formar mensajes.....</i>	285
Figura 274 <i>Nodo en Node-RED para el cálculo de la cinemática inversa.....</i>	286
Figura 275 <i>Algoritmo de comparativa del resultado obtenido con el esperado del cálculo de la cinemática inversa.....</i>	286
Figura 276 <i>Nodos para el envío de las instrucciones hacia el microordenador.....</i>	287
Figura 277 <i>Algoritmo de comparativa del resultado obtenido con el esperado del orden de las instrucciones.....</i>	287
Figura 278 <i>Algoritmo para verificar el funcionamiento de las funciones de Node - RED.....</i>	288
Figura 279 <i>Resultado de las pruebas unitarias.....</i>	288

Resumen

El trabajo describe el proceso para el diseño e implementación de una celda de manufactura remota basada en un robot SCARA a través del internet de las cosas y la industria 4.0. El robot será capaz, mediante visión artificial, de realizar acciones automáticas y capaz de cambiar su comportamiento para los requerimientos del usuario.

El objetivo de hacer celdas de manufactura remota es poder controlar y operar el robot sin limitaciones por la presencialidad. Además de aumentar la experiencia de estudiantes y personal de la industria en el manejo de robots manipuladores industriales y procesos automatizados por medio de visión artificial.

En el primer capítulo del trabajo se definen algunas generalidades, como la justificación e importancia, alcance, estado del arte, entre otros. En el siguiente capítulo se trabaja un poco más del marco teórico requerido para el entendimiento del trabajo.

En el tercer y cuarto capítulo se trata el diseño e implementación respectivamente, mostrando de manera detallada los procesos de selección para un diseño de acuerdo con la casa de la calidad, realizando análisis de factibilidad. Con la implementación se muestra la evidencia de la construcción integral del robot manipulador y los nodos utilizados en Node-Red.

El siguiente capítulo muestra las pruebas que se realizaron a los subsistemas para poder validar su funcionamiento en aspectos de hardware (simulación de resistencia mecánica y conexiones eléctricas), software (Visión por computador, control, carga del sistema) y usabilidad con el usuario.

Finalmente, las conclusiones y recomendaciones que se obtuvieron al llevar a cabo el trabajo; además de trabajos futuros en los que se pueda expandir el presente trabajo.

Palabras Clave: SCARA, celda de manufactura remota, IoT, industria 4.0, Visión artificial, Node-Red.

Abstract

This work describes the process for the design and implementation of a remote manufacturing cell based on a SCARA robot through the Internet of Things and Industry 4.0. The robot will be able, through artificial vision, to perform automatic actions and be able to change its behavior to the user's requirements.

The objective of making remote manufacturing cells is to be able to control and operate the robot without limitations due to being present. In addition to increasing the experience of students and industry personnel in the use of industrial manipulator robots and automated processes through artificial vision.

In the first chapter is defined some generalities, such as justification and importance, scope, state of the art, among others. In the next chapter we work a little more on the theoretical framework required for the understanding of the work.

In the third and fourth chapters, the design and implementation, respectively, showing in detail the selection processes for a design in accordance with the house of quality, carrying out feasibility analysis. With the implementation, the evidence of the integral construction of the manipulator robot and the nodes used in Node-Red is shown.

The following chapter shows the tests that were carried out on the subsystems in order to validate their operation in terms of hardware (simulation of mechanical resistance and electrical connections), software (computer vision, control, system load) and usability.

Finally, the conclusions and recommendations that were obtained when carrying out the work; in addition to future works in which the present work can be expanded.

Keywords: SCARA, remote manufacturing cell, IoT, industry 4.0, artificial vision, Node-Red.

Capítulo 1: Generalidades

Introducción

El presente trabajo estaba enfocado en diseñar e implementar una celda de manufactura para la manipulación de piezas metálicas por medio de un robot SCARA de cuatro grados de libertad, el cual permite su control de manera remota a través de internet desde una interfaz humano máquina (HMI) desarrollada en Node-RED, que utiliza el protocolo de comunicación MQTT para permitir el envío y recepción de información de los sensores e instrucciones que el brazo debe ejecutar.

Esta interfaz cuenta con un modo de control manual para que el usuario decida qué posición debe tomar el robot, así como un modo de control por visión artificial, que detecta e identifica piezas en el espacio de trabajo por forma y color, y en base al algoritmo programado, clasifica por forma o color enviando instrucciones de forma automática hacia el robot para que las recoja y posicione.

El desarrollo de este trabajo pretende ser de utilidad para laboratorios remotos de la Universidad de las Fuerzas Armadas - ESPE, puesto que como consecuencia de la pandemia del COVID - 19, existió distanciamiento social que impidió que personas relacionadas con el ámbito industrial en el cual se manipulan robots de forma cotidiana se vean afectadas al no contar con infraestructura física con la cual desarrollar tareas y tests de funcionamiento.

Durante el proceso de diseño e implementación se realizaron simulaciones y pruebas de funcionamiento de los diferentes subsistemas contemplados en la estructura tanto física como de software para que su el producto final cumpla con los requerimientos establecidos.

Definición del Problema

La industria 4.0 se encuentra en constante desarrollo, introduciendo nuevos conceptos y tecnologías que son adquiridas por el campo de las líneas de producción, como el manejo de información a través del uso de nubes computacionales, uso del Internet de las Cosas (IoT),

nanotecnologías, entre otras. En vista de este desarrollo, la preparación de personas capacitadas para manipular máquinas, específicamente robots, dentro del ámbito educativo o industrial es una de las responsabilidades de las instituciones de educación.

La situación de pandemia provocada por el COVID 19 provocó que exista distanciamiento social, lo que impidió que personas involucradas en el ámbito industrial donde se requiere la manipulación de robots, ya sea con fines didácticos o para producción puedan acceder a estos equipos para la realización de tareas o ensayos. De acuerdo con la UNESCO, el año 2020 impactó negativamente a la educación por la falta de presencialidad en las clases, afectando de especial manera a las carreras técnicas al no permitir que los estudiantes refuercen conocimientos con trabajos prácticos dentro de las instituciones (Instituto Internacional para la Educación Superior en América Latina y el Caribe, 2020).

Mantener un paradigma de educación virtual dentro de carreras técnicas donde es necesario desarrollar trabajos prácticos dentro de infraestructuras con los materiales adecuados impacta negativamente en la preparación de los futuros profesionales al limitarse en muchos casos al uso de simuladores del funcionamiento de robots. El desarrollo de una celda de manufactura remota, cuyo uso puede ser desde diferentes localidades podría ayudar a solventar esta situación, permitiendo que personas involucradas en el ámbito industrial donde se requiere la manipulación de robots puedan acceder a uno sin quebrantar el distanciamiento social y normas de bioseguridad.

Justificación e Importancia

La situación actual que ha vivido la humanidad desde el 2020 debido a la pandemia del COVID - 19 ha impedido que las actividades se realicen de forma presencial, afectando negativamente a las personas que se inician en el ámbito de procesos relacionados a la industria 4.0, que involucra la manipulación de los robots industriales.

Con el objetivo de solucionar la falta de acceso a las instalaciones que cuentan con el equipamiento destinado a procesos industriales realizados por robots manipuladores debido al COVID-19, se desarrolló e implementó una celda de manufactura remota basada en un robot SCARA de cuatro grados de libertad, que permite su uso por medio de tecnologías IoT, además de una aplicación de la visión artificial para la detección e identificación de piezas dentro del espacio de trabajo por forma o color.

El diseño implementado consta de una arquitectura modular que permite el acceso al manipulador físico de forma remota. El diseño de la plataforma permite que se replique en escala de forma que esté disponible para aumentar el acceso a estas tecnologías a nivel nacional, apoyando al proceso de enseñanza, de modo que aporte al plan de Creación de oportunidades del gobierno (2021 - 2025) con su objetivo de potenciar las capacidades de la ciudadanía y promover una educación innovadora, inclusiva y de calidad en todos los niveles.

Área de influencia

El presente trabajo aporta con una alternativa para la aplicación de conocimientos de robótica, redes industriales e internet de las cosas de manera conjunta al incluir estos tres grandes tópicos de conocimiento dentro del diseño e implementación de la celda de manufactura remota basada en un robot SCARA.

Objetivos

Objetivo general

Diseñar e implementar una celda de manufactura remota basada en un robot SCARA con aplicación de visión artificial para su aplicación en los laboratorios remotos.

Objetivos específicos

- Diseñar el robot SCARA a través de herramientas computacionales para los cálculos necesarios y utilización de software CAD para la estructura.

- Seleccionar los materiales para la construcción y componentes eléctricos y electrónicos necesarios en base al diseño del robot SCARA.
- Diseñar una plataforma de hardware para un robot SCARA que permita la manipulación de los instrumentos contenidos en la celda de manufactura.
- Diseñar la interfaz de control telemático de la plataforma, empleando servicios web y visión artificial.
- Validar el diseño del robot mediante la construcción del mismo y a través de la realización de pruebas de funcionamiento de la estructura física y la electrónica de control.

Alcance

Se diseñó e implementó una celda de manufactura remota basada en un robot SCARA de cuatro grados de libertad, la cual puede ser manipulada desde una interfaz gráfica diseñada en el dashboard de Node - RED. Se puede acceder a esta interfaz desde cualquier dispositivo que cuente con conexión a internet por medio de una página web de inicio de sesión que redirecciona hacia las páginas de flow y dashboard de Node-RED. La interfaz humano máquina (HMI) cuenta con tres pantallas principales: Monitorización, que permite visualizar el estado del robot en base a las lecturas de todos los sensores, además de reflejar el estado del robot por medio de un gemelo digital que indica que actuadores se encuentran activos; Control manual que permite el ingreso de valores para los grados de libertad disponibles y enviar la orden hacia el robot; y Visión artificial, que permite la captura una imagen de la superficie sobre la que se encuentra el robot, y detecta e identifica las piezas que se encuentran dentro del espacio de trabajo del SCARA, estima la posición a partir de cálculos de coordenadas y envía instrucciones de forma automática al robot para que se posicione, las recoja y las deposite en un receptáculo destinado para las piezas en un orden que puede ser determinado por el

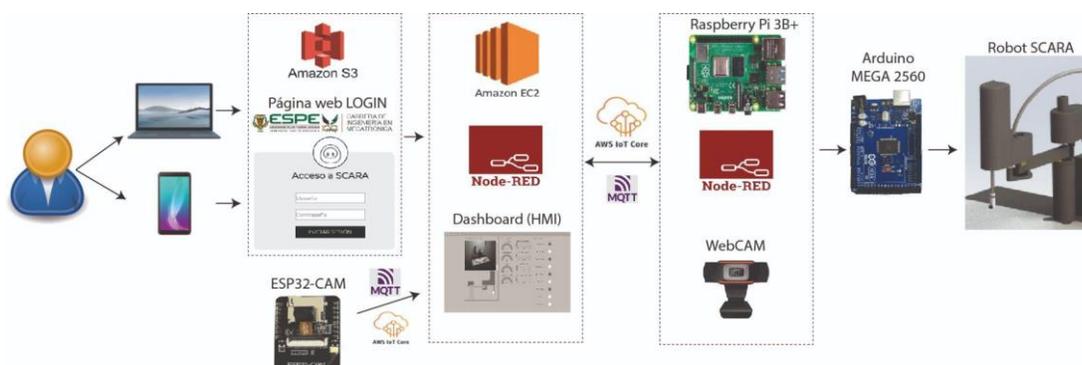
usuario. Por defecto el robot toma las piezas en base a un orden preestablecido de colores y los coloca en un recipiente; este comportamiento puede ser modificado por el usuario para que se coloquen las piezas en diferentes posiciones dentro del espacio de trabajo del robot, ya sea por colores, forma o el orden que el usuario determine que es el adecuado para su aplicación.

La implementación de la monitorización en tiempo real incluye una placa ESP32-CAM, que transmite en video el estado actual, el cual es visible en las tres pantallas de la interfaz, con la finalidad de que el usuario siempre tenga conocimiento de cómo se encuentra funcionando el brazo robótico. La implementación de la visión artificial incluye una cámara web que apunta desde la parte superior de la estructura hacia la superficie que soporta al robot, lo que le permite capturar una imagen de todo el espacio de trabajo, y posteriormente por algoritmos estima la posición de cada una de las piezas que se encuentren accesibles al robot.

La arquitectura implementada para el funcionamiento de la celda de manufactura se muestra en la Figura 1, donde se detalla cómo se conectan los elementos que permiten que el robot SCARA funcione de manera remota y se pueda monitorizar en tiempo real.

Figura 1

Arquitectura de comunicación implementada



El usuario accede a una página web para iniciar sesión y acceder al flow y al dashboard (HMI) de Node-RED alojado en una instancia de Amazon EC2, la cual envía y recibe

El mapeo de información cuenta con dos partes fundamentales, la primera que es la búsqueda de artículos relacionados con el tema presentado a través de un Mapeo sistemático de la literatura (Systematic Mapping Study, SMS), y la posterior revisión sistemática de la literatura (Systematic Literature Review, SLR), mientras que la segunda consiste en exponer los conceptos de celda de manufactura remota y laboratorios remotos, y se establecen las propiedades con las que debe cumplir la celda de manufactura para ser manipulada de forma remota y utilizable dentro de un laboratorio. En la Figura 3 se aprecia el procedimiento completo del mapeo de información.

Figura 3

Esquema del proceso de mapeo de información



Mapeo sistemático de la literatura

Al realizar la revisión de estudios y trabajos de investigación que toman la temática de los laboratorios remotos, se puede observar que no existe un estándar que guíe las soluciones propuestas, por lo que es necesario realizar un mapeo sistemático de literatura para obtener conocimiento de qué técnicas, herramientas y conceptos se está utilizando sobre los laboratorios remotos, con énfasis en celdas de manufactura y los complementos que utilicen, siendo estos sensores, actuadores, entre otros.

Método de investigación

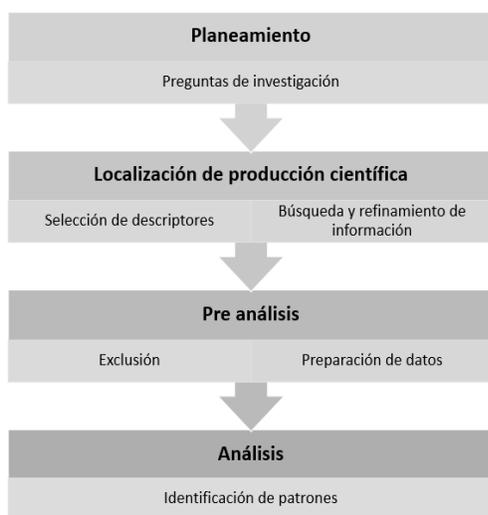
El SMS es un método que permite clasificar resultados obtenidos dentro de una investigación de una temática amplia al categorizar los datos obtenidos y al responder

preguntas de investigación previamente planteadas que permiten obtener la información específica que se requiere, disminuyendo el tiempo y esfuerzo de búsqueda.

El SMS utilizado se aplicó para buscar información acerca de qué hardware, protocolo de comunicación, entornos de desarrollo y visualización, y actuadores se está utilizando para el diseño de celdas de manufactura implementadas dentro de laboratorios remotos. El mapeo sistemático se realizó en base al procedimiento que exponen Navarro y Ramírez (2018), que se muestra en la Figura 4.

Figura 4

Procedimiento del mapeo sistemático



Preguntas de investigación

El objetivo del SMS realizado es determinar qué herramientas de hardware, protocolos de comunicación, entorno de desarrollo y visualización, y manipuladores están siendo utilizados en la actualidad para el desarrollo de celdas de manufactura remotas que sean viables dentro de un laboratorio remoto. En la Tabla 1 se recopilaron las preguntas que se formularon y el objetivo de cada una de ellas.

Tabla 1*Preguntas de investigación del SMS*

Pregunta de investigación	Objetivo
¿Qué plataformas de hardware se están utilizando para la programación remota de manipuladores dentro de celdas de manufactura?	Conocer la variedad de hardware existente y determinar el que mejor se adapta a las necesidades propias de la celda de manufactura remota.
¿Qué protocolos existen para realizar la comunicación entre el operador y la celda de manufactura para su manipulación?	Determinar el protocolo de comunicación que se adapte a las necesidades de la celda de manufactura remota.
¿Qué entornos de desarrollo existen para implementar aplicaciones para la manipulación de celdas de manufactura?	Determinar el entorno de desarrollo que mejor se adapta a las necesidades propias entre las opciones que se estén utilizando en trabajos previos.
¿Qué entornos de visualización existen para implementar interfaces gráficas para la utilización de celdas de manufactura remotas?	Conocer las formas en las que se puede presentar una interfaz gráfica al usuario, que está destinada para operar una celda de manufactura de forma remota.
¿Qué tipo de manipuladores robóticos existen para la implementación de celdas de manufactura remotas?	Determinar la variedad de manipuladores robóticos que son implementados dentro de celdas de manufactura remotas.

Establecer los criterios de búsqueda primarios y secundarios

Para este paso del SMS, se seleccionó la base de datos de trabajos de investigación Web of Science, cuyo repositorio permitió desarrollar el estado del arte del presente trabajo.

Esta base de datos fue escogida por su variedad de trabajos recopilados de diferentes revistas científicas tales como IEEE, Elsevier, etc, de la cual se pudo obtener una cantidad considerable de documentos que ofrecen información generalizada de cómo se realizaron los trabajos previos.

Definir la cadena de búsqueda

Para obtener información relacionada con el tema del presente trabajo dentro del repositorio se estructuraron diferentes cadenas de búsqueda en base a palabras clave que ayuden a realizar la investigación.

Las cadenas de búsqueda utilizadas para responder las preguntas de investigación se formulan en base a las palabras clave que presentan en la Tabla 2.

Tabla 2

Cadenas de búsqueda

Término principal	Términos alternativos
Celda de manufactura	Manufacturing Cell OR Machining Cell
Plataforma de hardware	Hardware platform OR Physical platform
Protocolo de comunicación	Communication Protocol OR Network Protocol OR Communication OR Protocol
Entorno de desarrollo	IDE OR Integrated Development Environment
Entorno de visualización	Graphical User Interface OR Graphical Environment OR User Interface
Manipulador robótico	Robot manipulator OR Robotic manipulator

(Manufacturing Cell OR Machining Cell) **AND** (Hardware platform **OR** Physical platform)
AND (Communication Protocol **OR** Network Protocol **OR** Communication **OR** Protocol) **AND**

(Integrated Development Environment **OR** IDE) **AND** (Graphical User Interface **OR** Graphical Environment **OR** User Interface) **AND** (Robot manipulator **OR** Robotic manipulator)

Para la cadena estructurada no se encontraron resultados, por lo que se plantearon tres cadenas alternativas con fragmentos de la cadena completa.

1. Cadena de búsqueda 1

(Machining Cell **OR** Manufacturing Cell) **AND** (Communication Protocol **OR** Network Protocol **OR** Communication **OR** Protocol) **AND** (Interface **OR** User Interface **OR** Graphical User Interface)

2. Cadena de búsqueda 2

(Communication Protocol **OR** Network Protocol **OR** Protocol **OR** Communication) **AND** (Hardware platform or Physical platform) **AND** (Integrated Development Environment **OR** IDE)

3. Cadena de búsqueda 3

(Machining Cell **OR** Manufacturing Cell) **AND** (Robot Manipulator **OR** Robot **OR** Robotic Manipulator) **AND** (Communication Protocol **OR** Network Protocol **OR** Protocol **OR** Communication) **AND** (User Interface **OR** Graphical User Interface)

Fijar los criterios de inclusión y exclusión

El objetivo de un SMS es recopilar gran cantidad de información sin profundizar en un determinado tema, por lo que los criterios de inclusión y exclusión utilizados permiten analizar documentos desde el año 2016 hasta junio de 2022.

Los criterios de inclusión que se seleccionaron para que un documento sea considerado dentro del trabajo se enumeran a continuación:

- Publicaciones desde junio de 2016

- Publicaciones que pertenecen a las ramas “Robotics”, “Engineering Electrical Electronic”, “Automation Control Systems”, “Telecommunications” o “Engineering Manufacturing”
- Documentos que muestren la tecnología, plataforma o protocolos utilizados para las celdas de manufactura remota o los laboratorios remotos.
- Cualquier tipo de publicación (artículos, grabaciones, conferencias, etc.)

Por otro lado, los criterios que se seleccionaron para la exclusión de documentos a ser incluidos se describen a continuación:

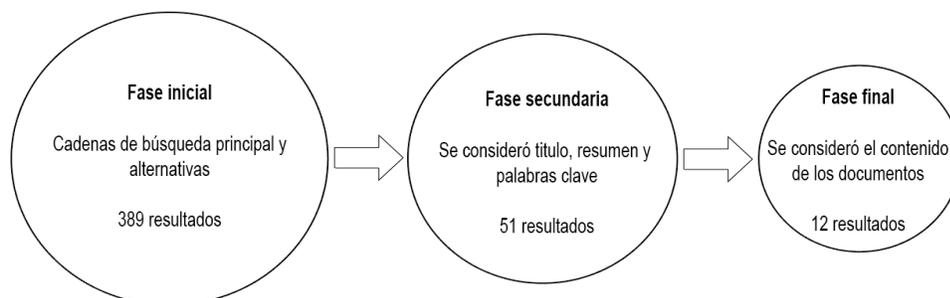
- Artículos relacionados con “Computer Science” o “Machine learning”
- Los documentos donde el principal objetivo es la identificación y desarrollo de un sistema de control de una celda de manufactura o manipuladores remotos.
- Los documentos donde se enfoquen en el desarrollo de nuevas celdas de manufactura con control in-situ.

Estrategia para la extracción de datos

El volumen de información obtenido de la búsqueda de información es considerable, por lo que se realiza una ficha para la extracción de información de la bibliografía obtenida en base a las preguntas que se plantearon en el SMS.

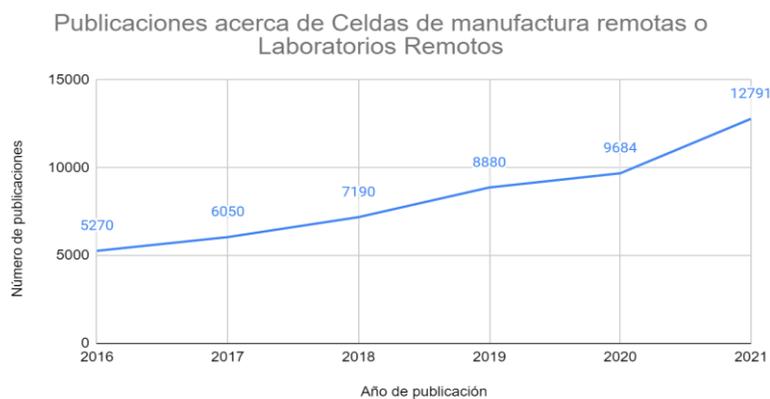
Clasificación de los artículos obtenidos

La Figura 5 refleja las fases de clasificación de artículos obtenidos de la búsqueda de información, junto a los criterios utilizados y el número de artículos obtenidos.

Figura 5*Fases de clasificación de artículos del SMS***Resultados**

Al aplicarse el procedimiento de filtrado de información se obtuvieron 12 trabajos de interés, que están relacionados con la temática manejada de celdas de manufactura remotas de forma parcial o total.

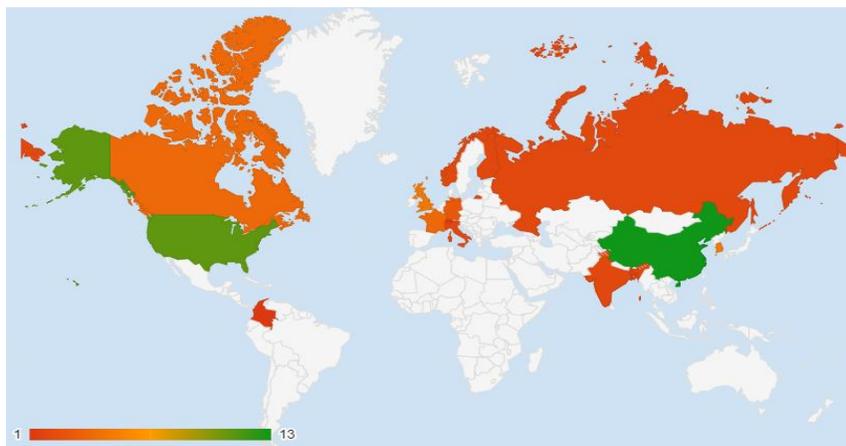
El primer dato que se extrae de este SMS es la determinación del número de artículos por año que se relacionan con las celdas de manufactura remotas o laboratorios remotos. La Figura 6 demuestra que el número de artículos escritos anualmente acerca de las temáticas buscadas aumenta cada año.

Figura 6*Publicaciones por año sobre celdas de manufactura remotas*

En la fase 2 del mapeo sistemático se obtuvieron 51 artículos de interés, que geográficamente se distribuyen como se muestra en la Figura 7.

Figura 7

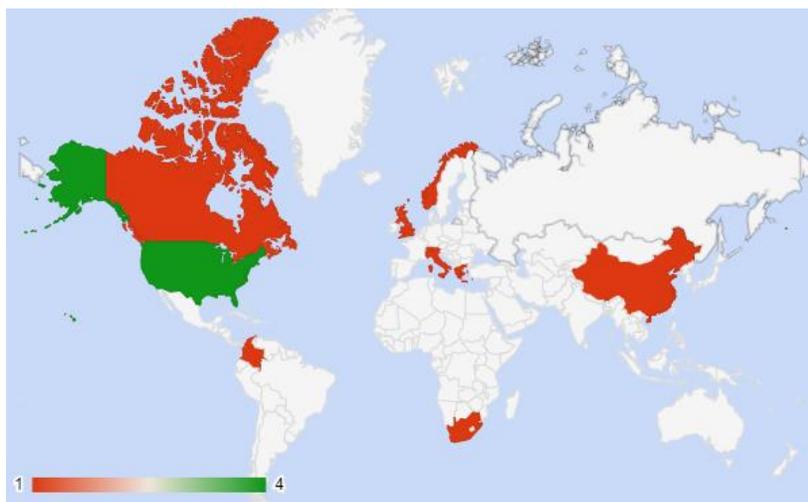
Distribución geográfica de los artículos de interés de la fase 2 del SMS



En la fase 3 del mapeo sistemático se obtuvieron 13 artículos de interés basándose en el contenido total de cada uno. Esta documentación en base a la región donde fueron publicados se distribuye como se muestra en la Figura 8.

Figura 8

Distribución geográfica de los artículos de interés de la fase 2 del SMS



Conclusiones del SMS

El SMS desarrollado permitió dar respuestas a las preguntas de investigación previamente planteadas, con sustanciación en trabajos previos realizados y publicados en el repositorio utilizado.

P1: ¿Qué plataformas de hardware se están utilizando para la programación remota de manipuladores dentro de celdas de manufactura?

Las plataformas de hardware utilizadas para el desarrollo de manipuladores remotos, son controladores lógicos programables (PLC) (Candell, Kashef, Yongkang, & Sebt, 2019; Xia, et al., 2020), el microcontrolador ESP32 que cuenta con posibilidad de conexión a redes Wi-Fi (Vargas, Cuero, & Torres, 2020; Ramya, Purushothama, & Prakash, 2020), y matrices de puertas programables (FPGA) para realizar circuitos electrónicos modificables (Sklivanitis, et al., 2018; Magyari & Chen, 2021).

P2: ¿Qué protocolos existen para realizar la comunicación entre el operador y la celda de manufactura para su manipulación?

Se pueden mencionar los protocolos Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Message Queuing Telemetry Transport (MQTT), siendo este último el utilizado en conjunto al microprocesador ESP32 (Vargas, Cuero, & Torres, 2020; Ramya, Purushothama, & Prakash, 2020).

P3: ¿Qué entornos de desarrollo existen para implementar aplicaciones para la manipulación de celdas de manufactura?

El entorno de desarrollo depende exclusivamente del microcontrolador que se utilizó, o con el que cuenta el manipulador robótico involucrado. Un ejemplo del segundo caso es de los manipuladores KUKA, que sus comportamientos son programados en entornos de desarrollo del mismo fabricante, como es KUKA SIMPRO (Zhu, Pires, & Azar, 2019).

Los trabajos que involucran el uso de PLC's recurren al entorno de desarrollo de la marca del controlador, como sucede con los PLC Siemens que utilizan el entorno de TIA Portal

(Xia, et al., 2020), mientras que los que utilizan microcontroladores de Arduino o ESP utilizan el Arduino IDE. Trabajos donde se utilizan plataformas de hardware de National Instruments como la NI DAQ utilizan el Multi-Bundle IDE que ofrece el fabricante.

Además de los entornos de desarrollo mencionados, si la aplicación requiere utilizar un lenguaje de programación aparte, se recurre a IDE's que soportan Java, Python, C++, C#, Verilog, entre otros.

P4: ¿Qué entornos de visualización existen para implementar interfaces gráficas para la utilización de celdas de manufactura remotas?

En el caso de las interfaces existió gran variedad de software utilizado, como son NI LabView, Mathworks MATLAB con su extensión Simulink, entornos de desarrollo que permiten la programación en lenguajes C# o Python. Adicionalmente los artículos que utilizaron PLC's del fabricante Siemens recurrieron a utilizar la extensión WinCC de Tia Portal (Xia, et al., 2020), mientras que los trabajos en los que se manipulaban robots KUKA, la interfaz de usuario se encontró en el entorno de Kuka User Interface.

La documentación que describe el uso del protocolo MQTT para las comunicaciones refleja que para el diseño de interfaces gráficas se recurrió a diseñar páginas web que cumplan la función (Vargas, Cuero, & Torres, 2020), o aplicaciones para Android o iOS en el caso de dispositivos móviles (Ramya, Purushothama, & Prakash, 2020).

P5: ¿Qué tipo de manipuladores robóticos existen para la implementación de celdas de manufactura?

Para tareas como la manipulación de objetos dentro de celdas de manufactura, o para procesos de manufactura aditiva se recurre a manipuladores de seis grados de libertad con pinzas como efector final (Abdeetal & Kermani, 2019; Xia, et al., 2020).

Para tareas de desmontaje de productos, la bibliografía refleja que se utilizan manipuladores colaborativos de tres grados de libertad, con un efector final que permita el cambio de herramienta necesaria para la tarea de desmontaje (Liu, et al., 2019).

Otras conclusiones

De la revisión bibliográfica se pudo extraer conclusiones aparte de las que responden las preguntas planteadas previamente, que son de interés para el desarrollo del presente trabajo

- Se pudo evidenciar que el tema de celdas de manufactura remotas es de interés para la comunidad científica, al comprobar que anualmente aumentan las publicaciones del tema en el website Web of Science; aunque, en base a los artículos filtrados para la extracción de datos se refleja que estos trabajos no toman un enfoque común, haciendo que no exista una estandarización de los productos desarrollados.
- La inteligencia artificial es un tema que se aborda constantemente dentro de los artículos encontrados, ya sea para adiestramiento del manipulador para posterior toma de decisiones en base a datos que recibe, o mediante el uso de visión artificial para que el manipulador cuente con información del entorno que lo rodea en su zona de trabajo.
- Cada artículo presenta una solución a una necesidad en específico, por lo que existe dificultad en encontrar una línea de tendencia que siguen todos los productos desarrollados. Tal es el caso, que para algunos trabajos se desarrolló software propio como respuesta a la necesidad que fue resuelta dentro de la documentación.

Reuniendo lo obtenido de las conclusiones extraídas de la búsqueda de información se puede afirmar que cada trabajo presenta su propia solución a un problema en específico, sin seguir un estándar y son destinadas a tareas concretas. Adicionalmente los manipuladores utilizados son brazos robóticos de seis grados de libertad o de tres grados de tipo RRR, de los cuales no todos tienen aplicación dentro de laboratorios remotos, lo cual permite inferir que el trabajo presente es de interés al utilizar un manipulador de tres grados de libertad diferente al tipo RRR y que está destinado a ser utilizado dentro de laboratorios remotos.

Revisión semántica de la literatura

El SMS permitió visualizar de forma cualitativa y cuantitativa los trabajos que incluyen la temática de celdas de manufactura remota y laboratorios remotos, sin embargo, no todos los trabajos poseen la característica de ser destinado para practicantes, y tampoco ser flexible para poder cambiar el comportamiento del manipulador.

Para conocer los protocolos y metodologías que se están utilizando en los nuevos laboratorios y celdas de manufactura remota con manipuladores se plantea una revisión semántica de la literatura (en adelante SLR), el mismo que permite integrar resultados de varios trabajos de forma objetiva y sistemática dentro del estado del arte.

Método de investigación

El método de investigación que se realiza para el SLR consta de tres actividades, como se observa en la Figura 9: planificación, que traza un plan de trabajo para el SLR y equilibra la carga de trabajo para cada miembro del trabajo; ejecución, encargada de extraer, seleccionar y sintetizar los datos de los trabajos a analizar durante el SLR; finalmente, se realiza una actividad donde se analiza e interpreta los resultados.

Figura 9

Procedimiento del mapeo sistemático



Preguntas de investigación

La realización del SLR se ve motivado por conocer a profundidad los protocolos, tecnologías y plataformas usadas en las celdas de manufactura remota y los laboratorios remotos. En la Tabla 3 se recopilan las preguntas formuladas para esta etapa de búsqueda de información.

Tabla 3

Preguntas de investigación para el SLR

Pregunta de investigación	Objetivo
¿Qué plataformas de hardware se están utilizando para la programación remota de manipuladores dentro de celdas de manufactura?	
RQ1: ¿Qué tecnología en plataforma de hardware se usa para laboratorios remotos?	Conocer cuál es la plataforma que se puede utilizar para un laboratorio remoto
RQ2: ¿Qué tecnología en plataforma de hardware se usa para celdas de manufactura remotas?	Conocer cuál es la plataforma que se puede utilizar para un laboratorio remoto enfocado en las celdas de manufactura
¿Qué protocolos existen para realizar la comunicación entre el operador y la celda de manufactura para su manipulación?	
RQ3: ¿Qué protocolo se usa para la comunicación remota de manipuladores?	Conocer el protocolo de comunicación de los manipuladores remotos
RQ4: ¿Qué técnicas se utilizan para la realización de laboratorios remotos enfocados en celdas de manufactura?	Conocer el estado del arte de los laboratorios remotos enfocados a la ingeniería en manufactura

Establecer los criterios de búsqueda

En la Tabla 4 se encuentran los términos seleccionados para la cadena de búsqueda del SLR, mismos que buscan responder las preguntas de investigación planteadas previamente.

Tabla 4

Términos de búsqueda para el SLR

Término principal	Términos alternativos
Celda de manufactura	Manufacturing Cell OR Machining Cell
Manipulador robótico	Robot manipulator OR Robotic manipulator OR SCARA
Laboratorios remotos	Remote Laboratory OR Remote Lab OR Remote

Basado en la tabla anterior, la cadena de búsqueda queda estructurada de la siguiente forma:

(Manufacturing Cell **OR** Machining Cell) **AND** (Robot manipulator **OR** Robotic manipulator **OR** SCARA) **AND** (Remote Laboratory **OR** Remote Lab **OR** Remote)

Debido a que la estructura de búsqueda presentada anteriormente no arroja resultados, se decidió formar cadenas de búsqueda que permita encontrar mejores resultados, quedando de la siguiente manera:

1. Cadena de búsqueda 1

(Manufacturing Cell **OR** Machining Cell) **AND** (Robot manipulator **OR** Robotic manipulator **OR** SCARA)

2. Cadena de búsqueda 2

(Manufacturing Cell **OR** Machining Cell) **AND** (Remote Laboratory **OR** Remote Lab **OR** Remote)

Fijar los criterios de inclusión y exclusión

La búsqueda pretende encontrar información más actualizada y con mayor profundidad en los temas de interés de las preguntas planteadas, por lo que se ha ajustado el tiempo de los documentos para poder ser considerados como relevantes, siendo el año 2017 la fecha máxima para la búsqueda de trabajos.

A continuación, se presenta la lista de las consideraciones para revisar un trabajo para el estado del arte:

- Publicaciones desde junio de 2017
- Que pertenezcan a las ramas “Robotics”, “Automation Control Systems”, “Engineering Electrical Electronic”, “Telecommunications “Engineering Manufacturing”
- Documentos que muestren la tecnología, plataforma o protocolos utilizados para las celdas de manufactura remota o los laboratorios remotos.
- Solamente artículos de revistas

Así también, se tiene en consideración los siguientes criterios para excluirlos trabajos a ser revisados:

- Artículos relacionados con “Computer Science” o “Machine learning”
- Los documentos donde el principal objetivo es la identificación y desarrollo de un sistema de control de una celda de manufactura o manipuladores remotos.
- Los documentos donde se enfoquen en el desarrollo de nuevas celdas de manufactura con control in-situ.
- Documentos duplicados o que provienen de la misma fuente.

Estrategia para la extracción de datos

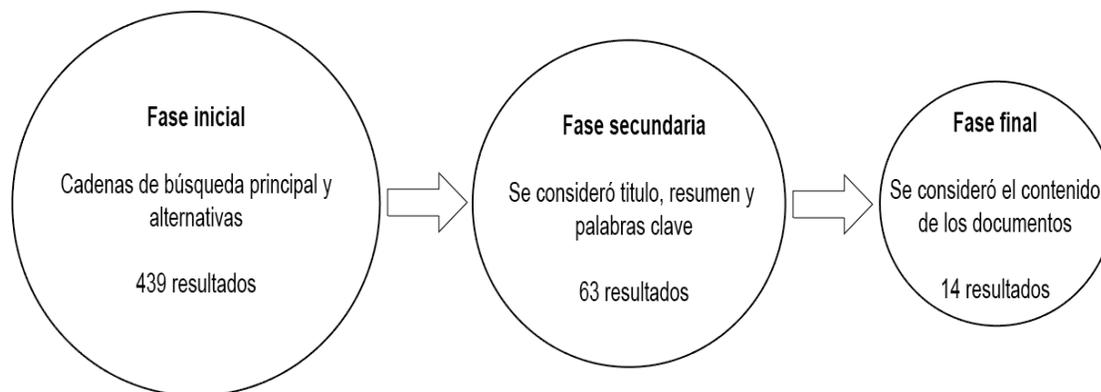
Para la extracción de datos, se elaboró una ficha similar a la utilizada en el SMS para extraer información relevante de los trabajos encontrados. El SLR cuenta con una reducción en la cantidad de documentos que se busca, pero más específica en cuanto a la información que se requiere, por lo que la ficha consta con criterios de extracción de información más precisa para descartar artículos no relacionados o duplicados.

Clasificación de los artículos obtenidos

La Figura 10 refleja las fases de clasificación de artículos obtenidos de la búsqueda de información, junto a los criterios utilizados y el número de artículos obtenidos.

Figura 10

Fases de clasificación de artículos del SLR

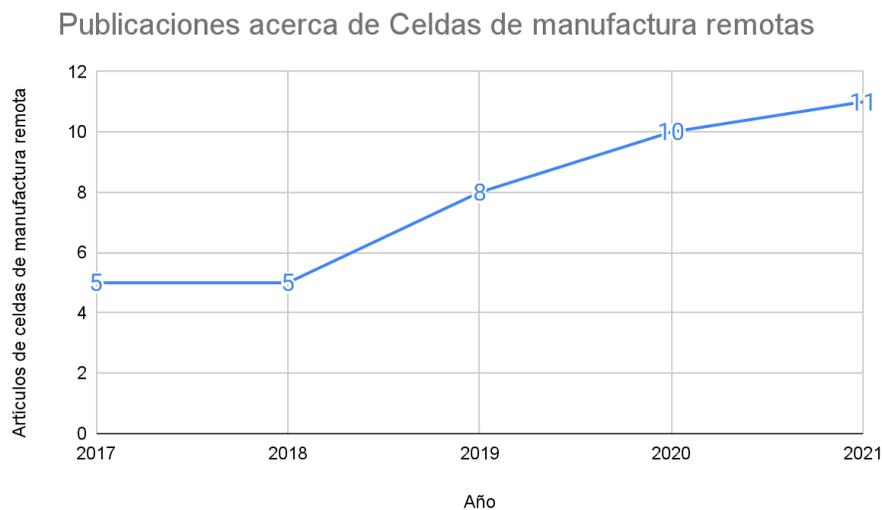


Resultados

Como antecedente de la cantidad de trabajos previos relacionados con las celdas de manufactura publicados en Web of Science hasta el año 2021, se encontraron 39, con un incremento en los últimos años, revelando el interés por la industria y la novedad en el tema. La Figura 11 muestra la tendencia creciente de publicaciones sobre el tema desde el 2017 hasta 2021.

Figura 11

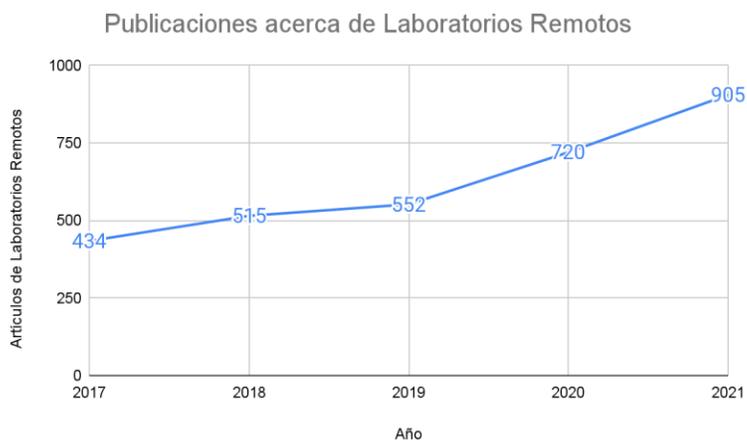
Publicaciones anuales de celdas de manufactura remotas



De igual forma, se analizó la cantidad de trabajos encontrados para los laboratorios remotos, con un total de 3126 trabajos hasta el año 2021, notando un claro incremento para el año 2020 y 2021, tal como se refleja en la Figura 12.

Figura 12

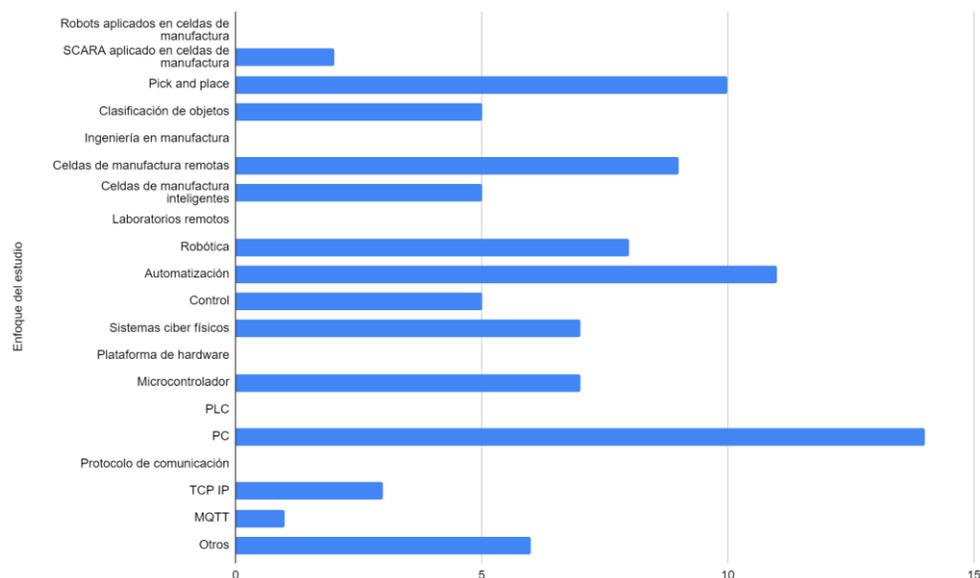
Publicaciones de laboratorios remotos desde 2017



La Figura 13 muestra el resultado de analizar y cuantificar la documentación en base a la filtración realizada durante la etapa 3 del SLR.

Figura 13

Cuantificación de los artículos obtenidos de la fase 3 del SLR



Existen trabajos que relacionan los robots en las celdas de trabajos con robots Pick and Place principalmente, seguido de la clasificación de objetos, siendo estos robots con al menos 6-DOF, adicionalmente se encontró trabajos con robots SCARA para operaciones de Pick and Place.

Dentro de este enfoque del estudio, se tiene la relación con la ingeniería en manufactura, siendo el interés saber si se está orientando esta ingeniería hacia las celdas de manufactura remotas o celdas de manufactura inteligentes. Como resultado se obtuvo una mayor cantidad de trabajos enfocados a las celdas de manufactura remota, aunque este resultado podría estar sesgado por los criterios de búsqueda.

Conclusiones del SLR

La revisión del estado del arte que se dio durante la fase del SLR permite dar una respuesta a las preguntas de investigación planteadas

P1: ¿Qué tecnología en plataforma de hardware se usa para laboratorios remotos?

Un número considerable de artículos reflejan que utilizan computadores, esto se debe a que en la mayoría de trabajos enfocados con los laboratorios remotos se usa una computadora como simulador en realidad virtual (Matsas & Vosniakos, 2017; Maoudj, Bouzouia, Hentout, Kouider, & Toumi, 2019; Yu & Han, 2021), para los casos de no contar con un modelo físico con el cual trabajar. Para los trabajos que poseen su modelo físico remoto para llevar a cabo el laboratorio, adicionalmente a la PC se utiliza un microcontrolador, encargado de recibir las órdenes remotas del usuario y ejecutarlas con los actuadores correspondientes (Jiménez, Avilés, & Mauledoux, 2018; Macleod, Dobie, Pierce, Summan, & Morozov, 2018).

P2: ¿Qué tecnología en plataforma de hardware se usa para celdas de manufactura remotas?

Gran cantidad de trabajos presentan tecnologías que permiten la realización de tareas previamente programadas, y no cuentan con un paradigma que permita programar y controlar fuera de lo predeterminado. Dentro de este enfoque, los trabajos se ven más orientados al entrenamiento de personal nuevo en operaciones conocidas (Manou, Vosniakos, & Matsas, 2019; Matsas & Vosniakos, 2017; Yu & Han, 2021).

P3: ¿Qué protocolo se usa para la comunicación remota de manipuladores?

No se especifica un estándar de protocolo de comunicación remota, ya que algunos artículos únicamente contemplaban la simulación de manipuladores. Los protocolos mencionados son: TCP-IP para aplicaciones en red local (Bedaka, Vidal, & Lin, 2019; Kamath, Balachandra, & Prabhu, 2019) y MQTT para aplicaciones remotas (Kumar, et al., 2020).

P4: ¿Qué técnicas se utilizan para la realización de laboratorios remotos enfocados en celdas de manufactura?

Existe una tendencia hacia los laboratorios remotos enfocados únicamente en desarrollar y/o mejorar la experiencia de un simulador, agregando gemelos digitales en realidad aumentada (Runji & Lin, 2020) y realidad virtual (Manou, Vosniakos, & Matsas, 2019; Matsas & Vosniakos, 2017; Yu & Han, 2021), sin la inclusión de un modelo físico remoto.

Para los trabajos que incluyen un modelo físico remoto, Jiménez implementó un sistema con comandos fijos, mismos que no pueden ser cambiados de forma remota ni sus conexiones (Jiménez, Avilés, & Mauledoux, 2018). En contraparte, Mar-Cornelio, presenta un proyecto más dinámico, permitiendo un sistema completo para cualquier tipo de laboratorio remoto, mismo que es construido sobre la base del proyecto y su programación se realiza de forma remota (Mar-Cornelio, Santana-Ching, & Gonzalez-Gulin, 2019).

Algunos de los trabajos a destacar después de realizar el SLR son los siguientes:

- Celdas de manufactura remotas (Macleod, Dobie, Pierce, Summan, & Morozov, 2018; Maoudj, Bouzouia, Hentout, Kouider, & Toumi, 2019; Matsas & Vosniakos, 2017)
- Manipuladores industriales remotos (Manou, Vosniakos, & Matsas, 2019; Safeea, Neto, & Bearee, 2019)
- Laboratorios remotos (Jiménez, Avilés, & Mauledoux, 2018; Jaziri, Charaabi, & Jelassi, 2018; Mar-Cornelio, Santana-Ching, & Gonzalez-Gulin, 2019)

Otras conclusiones

Adicionalmente, se llegó a una serie de conclusiones adicionales gracias a la profundización de los temas de interés:

- Existe interés de la comunidad científica en las celdas de manufactura remotas y los laboratorios remotos, temas que son el foco del SLR.

- Pese al interés que existe, se puede notar que el enfoque no es hacia trabajos que involucren un dispositivo físico cuyo comportamiento pueda ser modificable. Sin embargo, gracias a estos trabajos se puede implementar sistemas mejor integrados entre la parte ciber y la parte física.
- Algunos de los trabajos utilizan sistemas de visión artificial de bajo costo para poder implementar sistemas de control más complejos e incluso son capaces de recrear entornos a partir de esta cámara.
- Muchos de los trabajos que se revisaron no definen específicamente la tecnología que está ocupando, únicamente define de forma esquemática las interfaces que está utilizando para la parte de hardware y software.

Glosario de términos

- SCARA: Brazo robótico articulado de cumplimiento selectivo
- IoT: Internet de las Cosas
- MQTT: Transporte de telemetría de cola de mensajes
- HMI: Interfaz Humano Máquina
- HPHMI: Interfaz Humano Máquina de Alto Rendimiento
- GDL: Grados De Libertad

Capítulo 2: Marco teórico

El capítulo comprende toda la información y conceptos que se requieren para la realización del proyecto; abarca la información relacionada con la parte física (robótica aplicada) y la parte de software (arquitectura de software, IoT, visión artificial, entre otras).

Robots manipuladores

En 1942, Isaac Asimov acuñó el término robótica haciendo referencia a una máquina que sea capaz de realizar operaciones de utilidad para el ser humano. Desde ese entonces se ha ido desarrollando más sobre el significado de la palabra robot, que en palabras de Miranda (2016), “Un robot es un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos variables programados para desempeñar una amplia variedad de tareas”.

Los robots manipuladores han evolucionado desde las primeras implementaciones de Devol y Engelberger, con el robot industrial llamado Unimate, pasando por otros hitos en la historia de los robots manipuladores como la Máquina Programable Universal para Ensamble o PUMA por sus siglas en inglés (Programmable Universal Machine for Assembly) en 1980, que se puede apreciar en la Figura 14, o el robot brazo robótico de montaje conforme selectivo o SCARA por sus siglas en inglés (Selective Compliant Assembly Robot Arm) presentado como un concepto en el año 1981.

Figura 14

Robot PUMA



Nota. Tomado de *Cinemática de Robot PUMA*, por J. Cortés, 2008.

El enfoque del desarrollo y construcción de robots no es exclusivo para la industria, se ha conseguido avances para el desarrollo de prótesis para extremidades faltantes como brazos o piernas, como se observa en la Figura 15; o vehículos aéreos no tripulados (Unmanned Aerial Vehicle [UAV]) que permite llevar suministros o tomar videos a grandes alturas y lugares difíciles de alcanzar.

Figura 15

Prótesis robótica de brazo



Nota. Tomado de *Fisioterapia, un paso de la igualdad*, por Universidad Anáhuac Cancún, 2019.

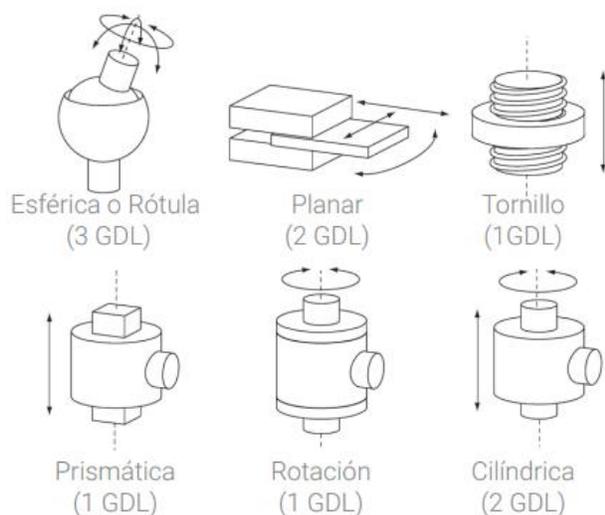
Para comprender de forma esquemática a los robots, se los define como un conjunto de eslabones unidos entre sí por medio de articulaciones. Bajo este concepto se puede dividir en dos los tipos de robots: los que cuentan con cadena cinemática abierta, en la que el último eslabón no se conecta con nada, y el segundo tipo cuenta con cadena cinemática cerrada en la que el último eslabón se conecta otro eslabón del robot.

Las articulaciones que unen a los eslabones de un robot permiten que exista movimiento en los denominados Grados de Libertad, que se definen como el número mínimo de parámetros para definir la posición, velocidad y número de reacciones de un cuerpo.

Dependiendo del tipo de articulación se calculan los grados de libertad de los robots, que puede interpretarse como la cantidad de motores que se requieren para que un mecanismo pueda moverse de forma completa. En la Figura 16 se puede visualizar los diferentes tipos de articulaciones que se utilizan en mecanismos.

Figura 16

Tipos de articulaciones



Nota. Tomado de *Morfología básica de un robot*, por Dirección de Educación Técnica de Buenos Aires, 2020.

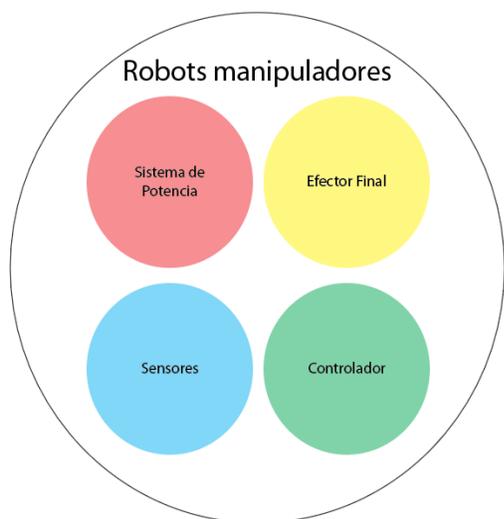
Estas articulaciones permiten que el robot pueda realizar movimientos en el espacio, dicho espacio o volumen que se genera por el movimiento de los eslabones se conoce como espacio de trabajo. Se dice que para que un robot pueda alcanzar cualquier punto dentro de su espacio de trabajo en una orientación arbitraria se requiere de 6-GDL, para un robot que cumpla con esta condición se dice que tiene un espacio de trabajo diestro, caso contrario únicamente cuenta con un espacio de trabajo alcanzable (Miranda, 2016)

Para entender el funcionamiento de los robots se debe conocer los elementos que conforman al robot que se observan en la Figura 17, que de forma general se pueden definir en cuatro grupos:

1. Sistema de potencia: este grupo comprende todos los elementos que se dedican a generar la alimentación necesaria para que los actuadores, ya sean motores eléctricos, neumáticos o hidráulicos; usualmente este grupo lo conforman las fuentes de alimentación o amplificadores de potencia.
2. Efector Final: este grupo es únicamente para los robots de cadena cinemática abierta, y hace referencia a la herramienta que se coloca en el eslabón final con el que el robot va a interactuar con el medio. El efector final depende del tipo de acción que el robot vaya a realizar, ya que puede ir desde un electroimán hasta una antorcha de soldadura.
3. Sensores: son dispositivos electrónicos o electromecánicos que permiten al robot conocer de su estado (propioceptivos) o el estado de su entorno (exteroceptivos), y pueden ser utilizados dentro de lazos para controlar el comportamiento del robot.
4. Controlador: este grupo comprende un sistema computacional que le permite realizar las acciones para las que se diseñó el robot. En la actualidad, los programas que conforman el controlador no necesariamente deben ser estáticos, ya que se puede mejorar con entrenamiento que mejoren su rendimiento original, haciendo uso de la inteligencia artificial.

Figura 17

Elementos que conforman a los robots manipuladores

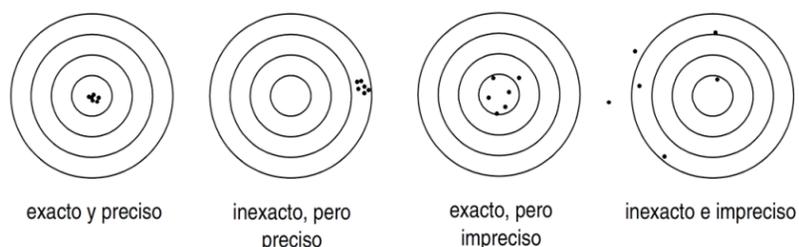


Además de los conceptos ya mencionados, al hablar de robots manipuladores también se deben tratar características relacionadas al rendimiento del robot, siendo estas la exactitud, repetibilidad y estabilidad.

La exactitud habla de la capacidad del del robot de obtener un valor igual o semejante al valor deseado, mientras que la repetibilidad es la capacidad de obtener el mismo resultado al medir la salida ante la misma entrada en diferentes ocasiones con intervalos de tiempo entre mediciones, y la precisión se relaciona con la dispersión que tienen las medidas al repetirse, entre menos dispersas, más preciso. En la Figura 18 se puede apreciar la diferencia que existe entre precisión y exactitud. La estabilidad hace referencia a la capacidad que posee de mantener la medición en la salida pese a los cambios en el ambiente, ya sea por el simple paso del tiempo o por variación de temperatura.

Figura 18

Diferencia entre exactitud y precisión



Nota. Tomada de Análisis del sistema de medición mediante estudio de repetibilidad y reproducibilidad en Schott envases farmacéuticos SAS, por E. Rodríguez, 2015.

Robot SCARA

El robot SCARA es un tipo de robot manipulador del tipo RRP, es decir, un manipulador que tiene articulaciones rotacional, rotacional y prismática. No obstante, la variante más común cuenta con 4-GDL (usando una configuración RRRP) para dar más movilidad al robot, como el que se muestra en la Figura 19.

Figura 19

Robot SCARA



Nota. Tomado de Epson LS6-B Scara Robot, por EPSON US., 2012.

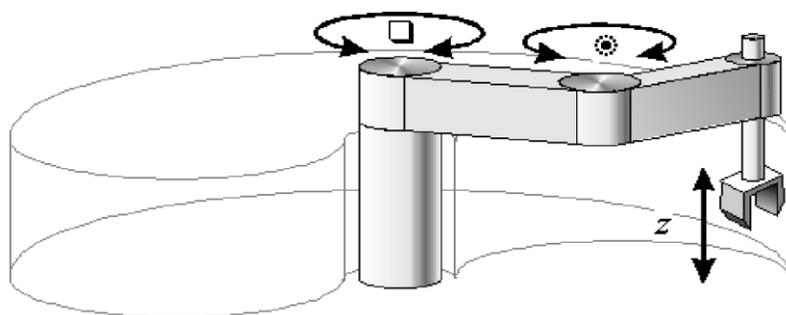
Este robot es utilizado en la actualidad de forma regular para el ensamblaje y/o clasificación de piezas en una variedad de industrias, desde la clasificación de medicamentos en líneas de producción farmacéuticas hasta ensamblar piezas mecánicas utilizadas en los vehículos. Al igual que el resto de robots manipuladores, los robots SCARA son acoplables a una variedad de efectores finales, aunque por su tamaño se los utiliza para realizar trabajo repetitivo y más delicado.

La implementación de un robot SCARA en la industria presenta como ventaja su reducido tamaño, lo que lo hace fácil de instalar en líneas de producción y por un bajo costo de acondicionamiento de lo ya implementado. Además, la mayoría de fabricantes ha conseguido desarrollar sistemas de control que permiten obtener altas precisión y exactitud en velocidades muy altas.

El espacio de trabajo de un robot SCARA es definido usando coordenadas cilíndricas; al realizar un análisis del movimiento del robot, se observa que se forman trayectorias circulares y su componente prismática, usualmente ubicada en el efector final, forma una figura similar a un cilindro como se observa en la Figura 20.

Figura 20

Espacio de trabajo de manipulador RRP



Nota. Tomado de *Robots de configuración mezclada o SCARA*, por Universidad Santiago de Chile, 2016.

Cinemática del robot

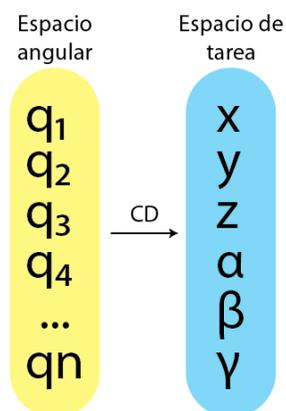
La cinemática del robot comprende la determinación de las expresiones matemáticas que permiten definir la posición de los eslabones en base a los valores que tomen los grados de libertad, o viceversa, es decir, obtener los valores que deben tomar los grados de libertad para conseguir una posición en concreto.

Cinemática directa

La cinemática directa de un manipulador define la posición y orientación del actuador cuando a los grados de libertad se asignan valores, ya sea de ángulo o de longitud, pasando del espacio angular al espacio de tarea, como se observa en la Figura 21.

Figura 21

Transformación realizada por la cinemática directa

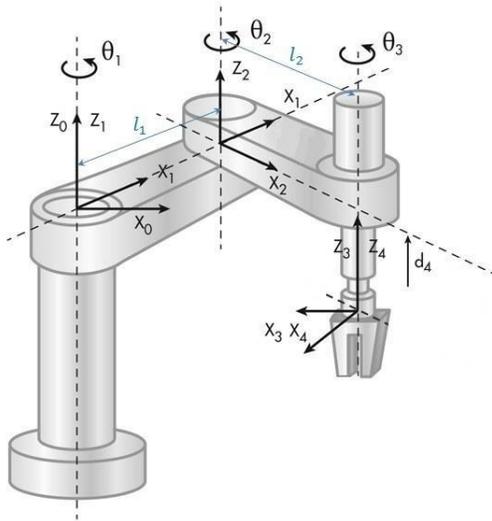


Para la determinación de las expresiones de la cinemática directa se requiere la asignación de sistemas de referencia para cada articulación, de tal manera que sea visible la transformación que sufren al cambiar la posición del robot, como se muestra en la Figura 22. Con los sistemas de referencia se espera poder describir de forma completa la posición y orientación del robot, para ello se utilizan herramientas matemáticas que permitan describir un sistema de referencia móvil en términos de un sistema fijo.

Matrices de transformación

Figura 22

Representación de los sistemas de referencia de los grados de libertad de un robot SCARA



Nota. Tomado de *Robot Manipulador SCARA*, por F. Torres, 2019.

En la Figura 22 se muestra como el sistema de referencia {0} tiene una orientación determinada, mientras que para el sistema de referencia {4} se tiene un sistema de referencia con una orientación diferente, misma que cambia por cada GDL que pasa para llegar desde el sistema de referencia inicial hasta el sistema de referencia del efector final.

Las matrices de rotación (representadas por la letra R acompañada de los números de los sistemas de referencia en los que se está realizando la rotación) de {0} a {1} se definen como se muestra en la expresión matemática y en la Figura 23.

$$R_1^0 = [\hat{x}_1^0 \quad \hat{y}_1^0 \quad \hat{z}_1^0]$$

$$p^0 = R_1^0 p^1$$

Figura 23

Estructura de las matrices de rotación

$$p^0 = \begin{bmatrix} (x_1, x_0) & (y_1, x_0) & (z_1, x_0) \\ (x_1, y_0) & (y_1, y_0) & (z_1, y_0) \\ (x_1, z_0) & (y_1, z_0) & (z_1, z_0) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Donde \hat{x} , \hat{y} y \hat{z} son vectores que representan la transformación de cada eje en términos del nuevo sistema de referencia; p es un vector de posiciones en coordenadas rectangulares. La obtención de posición y orientación del efector final o de los eslabones utilizando este tipo de matrices ocasionalmente es largo y complicado. En base a las matrices de rotación se construyen las matrices homogéneas, que define la rotación y traslación de un sistema de referencia a otro, y se representa utilizando la letra H con superíndice y subíndice los números de los sistemas de referencia involucrados, como las que se observan en la Figura 24.

Figura 24

Estructura de las matrices homogéneas

$$H_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix}$$

$$H_1^0 = \begin{bmatrix} (x_1, x_0) & (y_1, x_0) & (z_1, x_0) & a \\ (x_1, y_0) & (y_1, y_0) & (z_1, y_0) & b \\ (x_1, z_0) & (y_1, z_0) & (z_1, z_0) & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

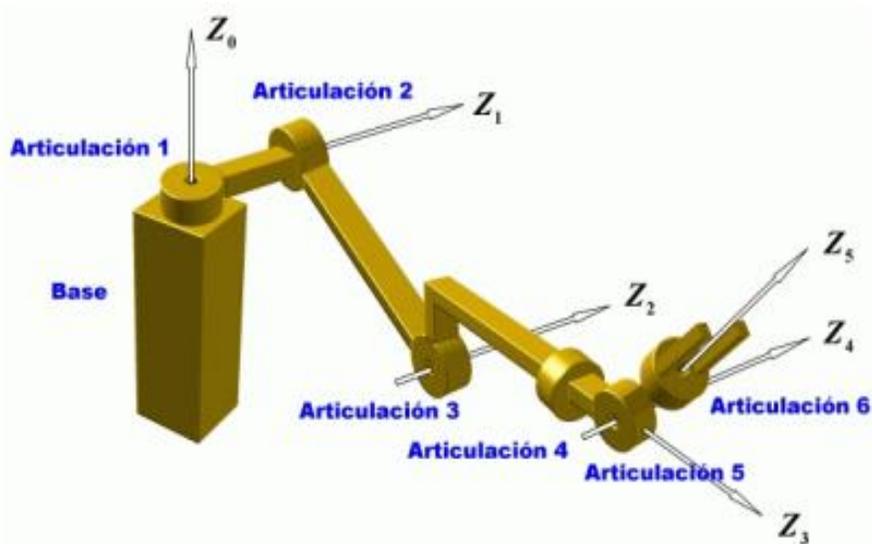
Utilizando las matrices homogéneas se simplifica el proceso de realizar las transformaciones de un sistema de referencia desde el origen de coordenadas del robot hasta el sistema de referencia del efector final, al únicamente necesitar las transformaciones por cada GDL que pasa hasta llegar al efector final y se realiza una multiplicación de matrices.

Parámetros Denavit-Hartenberg

La obtención de las matrices homogéneas se realiza en base a transformaciones de posición y rotación de los sistemas referenciales. Este proceso se simplifica al utilizar los parámetros de Denavit-Hartenberg (DH). Los parámetros DH se basan en representar las transformaciones de un sistema referencial en base a rotaciones y traslaciones utilizando únicamente el eje X y Z de cada sistema, por lo que el proceso de representar los sistemas referenciales sigue una convención. La primera pauta es mantener el eje Z alrededor del grado de libertad de cada articulación, como se muestra en la Figura 25.

Figura 25

Trazado del eje Z para la representación de Denavit - Hartenberg

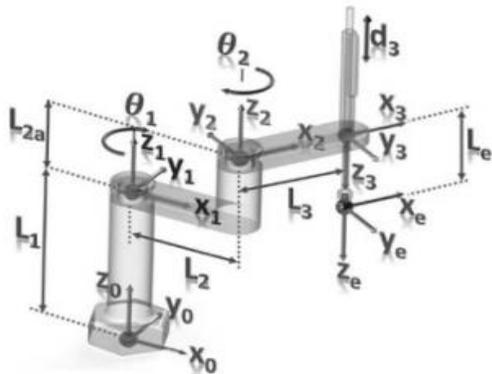


Nota. Tomado de *Representación de Denavit – Hartenberg*, por J. Cortés, 2008.

El siguiente paso de la convención es que el eje x_{i-1} sea ortogonal al z_i y z_{i-1} (en caso sean paralelos, la orientación x_{i-1} puede ser arbitraria) y la asignación del eje y va acorde con la regla de la mano derecha, como se muestra en la Figura 26.

Figura 26

Trazado del eje X para la representación de Denavit - Hartenberg



La convención determina únicamente realizar giros y desplazamientos en los ejes z y x , siguiendo esta norma:

- Giro alrededor de z (θ)
- Desplazamiento en z (d)
- Desplazamiento en x (a)
- Giro alrededor de x (α)

En la Figura 27 se puede observar el resultado de la determinación de los parámetros de Denavit – Hartenberg para el esquema de la Figura 26, de tal manera que se representa los giros y desplazamientos requeridos para pasar de una posición a otra.

Figura 27

Parámetros de Denavit - Hartenberg

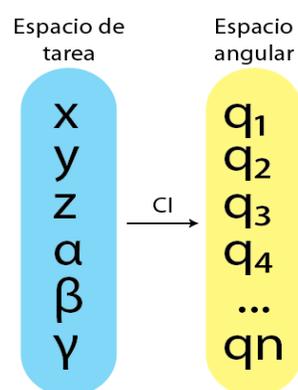
i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	L_1	θ_1
2	0	L_2	L_{2a}	θ_2
3	180	L_3	d_3	0
e	0	0	L_e	0

Cinemática inversa

La cinemática inversa del robot manipulador comprende las expresiones que permiten determinar los valores requeridos en el espacio angular para llegar a un punto en el espacio de tarea, siendo la entrada la posición y orientación en el espacio de tarea, como se indica en la Figura 28. El problema de la cinemática inversa suele ser un problema complejo y que usualmente no tiene una única respuesta por la naturaleza de sus cálculos.

Figura 28

Transformación realizada por la cinemática inversa



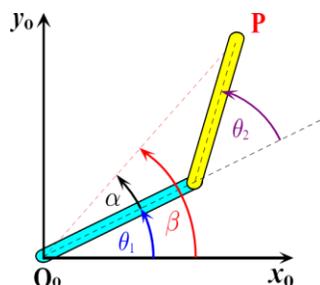
Para realizar esta transformación se puede utilizar varios métodos, que pueden ser el algebraico, iterativo, geométrico, entre otros. El método que presenta más simplicidad para realizar cálculos e implementar las expresiones es el método geométrico, que únicamente utiliza relaciones trigonométricas para hallar la respuesta en el espacio angular.

Método geométrico

Este método gráfico permite obtener la cinemática inversa del manipulador, necesitando representar de forma esquemática el manipulador en el o los planos donde se pueda obtener información relevante de los ángulos del espacio articular, como se muestra en la Figura 29.

Figura 29

Representación esquemática de los eslabones de un manipulador planar RR (primera solución)

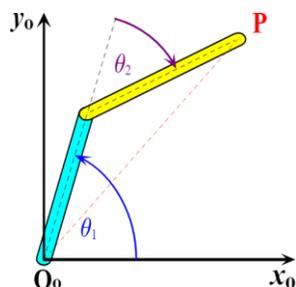


Nota. Tomado de *Modelación cinemática del brazo del manipulador*, por Universidad Santiago de Chile, 2016.

Para encontrar los ángulos correspondientes que dan solución al problema, teniendo en cuenta que se conoce las coordenadas del punto P y las dimensiones de cada eslabón, en base al esquema el proceso de cálculo se reduce a un problema de trigonometría donde es necesario utilizar las relaciones trigonométricas fundamentales y la ley de cosenos. Una característica a considerar del proceso de cálculo de la cinemática inversa es que es posible obtener un conjunto de soluciones que permiten alcanzar la posición deseada, es decir, no existe una única solución, como se puede observar en la Figura 30.

Figura 30

Representación esquemática de los eslabones de un manipulador planar RR (solución 2)



Nota. Tomado de *Modelación cinemática del brazo del manipulador*, por Universidad Santiago de Chile, 2016.

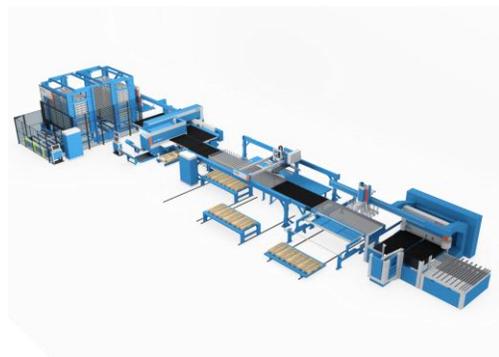
Celdas de manufactura inteligente

Las aplicaciones más comunes de los robots manipuladores son las celdas de manufactura, mismas que con el paso del tiempo y la llegada de la industria 4.0 fueron dejando de utilizar al ser humano como fuente principal de mano de obra y se dio paso a robots que mejoren y optimicen el proceso que se esté realizando, minimizando las fallas que se pueden generar por el cansancio y fatiga del ser humano.

Una celda de manufactura se puede definir como el conjunto de robots, maquinaria CNC, dispositivos e incluso humanos que se reúne en un espacio para lograr la fabricación de un elemento de forma parcial o integral (Cardona, 2015). El papel del humano en las celdas de manufactura recientemente ha sido reducido únicamente a la monitorización y control de las máquinas, dejando que la interacción sea únicamente máquina-máquina. En la Figura 31 se muestra un ejemplo de celda de manufactura que cuenta con varias estaciones de trabajo que se comunican entre sí.

Figura 31

Ejemplo de celda de manufactura



Nota. Tomado de *Sistemas flexibles de manufactura*, por Prima Power, 2022.

Pese a que, en español, el término que se utiliza es Celda de manufactura inteligente, en inglés existe una diferencia entre dos términos similares, Intelligent Manufacturing (IM) y

Smart Manufacturing (SM) (Wang, 2020). Estos términos difieren en la cantidad y propósito de la tecnología que se utiliza en la celda.

En su trabajo, Wang muestra que hay un paralelismo entre IM y SM, aunque se puede considerar que IM, o la terminología adoptada desde el año 2020, New Generation Intelligent Manufacturing (NGIM), integra más las nuevas tecnologías como big data, redes neuronales o visión por computadora con el objetivo de mejorar el proceso aprendiendo y entrenando el proceso de mejor manera entre máquinas y no basados en el trabajo humano, cosa en la que SM se basa un poco más.

Con la evolución de la tecnología, las industrias mejoran sus procesos integrando cada vez más máquinas para obtener procesos completamente automatizados y reducir la cantidad de operarios requeridos, aun cuando únicamente se dediquen a la monitorización del proceso, como se puede observar en la Figura 32, que es un ejemplo de un proceso de producción completamente automatizado.

Figura 32

Proceso de producción automatizado



Nota. Tomado de *Maximo Application Suite*, por IBM, 2021.

Internet de las cosas

Con el nacimiento de las redes de comunicación como el internet, surgió la curiosidad en los investigadores de conectar más dispositivos que no sean solamente computadoras, con este precedente, en 1999 Kevin Ashton utiliza el término internet de las cosas, o mejor conocido como Internet of Things (IoT), por primera vez, aunque el significado que se le dio en su momento ha sido superado.

Al principio se pretendía utilizar únicamente sensores con conexión a internet para tener información del mundo real e interconectarse con la internet, pero gracias al protocolo de internet (IP) se ha logrado conseguir comunicación máquina a máquina (Machine to Machine [M2M]) que realiza operaciones más complejas en las que no solamente se recibe información, pero se puede interactuar con actuadores conectados al internet y ubicados en lugares lejanos.

En la actualidad, existe una gran cantidad de dispositivos conectados al internet, Huawei ha anticipado que para el año 2025 existirán más de 100 mil millones de dispositivos conectados, es por tal motivo que se deben plantear soluciones para lograr hacer una escalabilidad de los dispositivos conectados a internet y hacerlo de forma segura. Alcanzar este objetivo permitirá implementar más ciudades inteligentes (Smart Cities) y mejorar la industria.

Con la industria 4.0 se pretende utilizar las tecnologías del internet de las cosas para mejorar los procesos, ya que anteriormente el desarrollo de la industria no permitía una conexión entre las máquinas eficientemente, por lo que la escalabilidad y productividad se veían drásticamente afectadas. Con la implementación del IoT en la industria, ahora se tienen procesos mucho más eficientes, como se observa en la Figura 33, que refleja las actividades que permite desarrollar la implementación de IoT, con una facilidad para escalar los procesos ya que la conectividad de los dispositivos es mucho más sencilla y sin tanta infraestructura.

Figura 33*Implementación de IoT en la industria*

Nota. Tomado de *What is Industrial Internet of Things?*, por TIBCO, 2021.

Otra ventaja que trajo el IoT a la industria es la capacidad de realizar trabajos remotos y no depender de la presencialidad de un operador, permitiendo la tele operabilidad de las máquinas, y obteniendo reportes en tiempo real de las acciones realizadas por la máquina.

Protocolos de comunicación

Para lograr la comunicación M2M que requiere el Internet de las cosas, es necesario que las máquinas hablen, por hacer una analogía, el mismo “idioma”. De esto se encargan los protocolos de comunicación, que no son más que un conjunto de estándares que permite que se haga un intercambio de información seguro y con la capacidad de ser entendido por los dispositivos que se están comunicando.

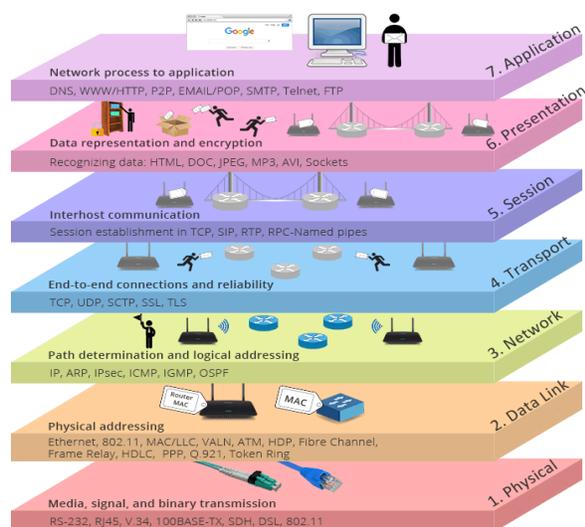
En internet, los protocolos más utilizados son el Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP), pero no son los únicos que existen o incluso hay protocolos que trabajan en conjunto como puede ser el TCP/IP. Gracias a estos protocolos es que los dispositivos son capaces de comunicarse a diario, pero se debe tener un control

estricto para que los mensajes que se envían por medio de estos protocolos no causen problemas con el ancho de banda que se tiene en la actualidad en la infraestructura de red.

El sistema de interconexión abierta (OSI) que se observa en la Figura 34 organiza las familias de los protocolos de red en capas más específicas en la que se pretende optimizar las comunicaciones de red de los diferentes sistemas (Kio Networks, 2021).

Figura 34

Estructura del modelo OSI



Nota. Tomado de *¿Cuál es la diferencia entre modelo OSI y modelo TCP/IP?*, por FS Global, 2021.

Message Queuing Telemetry Transport (MQTT)

Este es un protocolo de comunicación que ha crecido con la aparición del internet de las cosas, ya que es un protocolo enfocado a la comunicación M2M basada en TCP/IP. Este protocolo fue desarrollado con el objetivo de conectar remotamente dispositivos en la industria petrolera (Llamas, 2019). Una de sus características que la hacen ideal para trabajar en IoT es sus mensajes ligeros y estructura simplificada como se observa en la Figura 35, que permiten que más dispositivos se conecten a la red sin afectar el ancho de banda.

Figura 35

Estructura de los mensajes en el protocolo MQTT

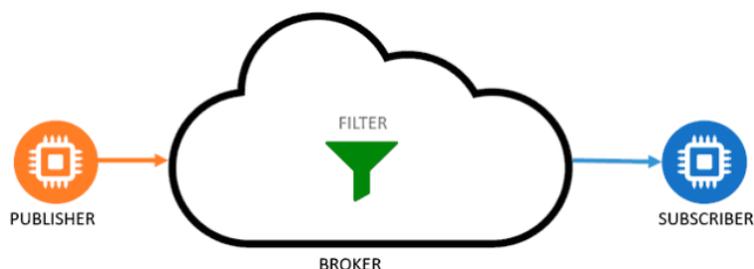


Nota. Tomado de *¿Qué es MQTT? Su importancia como protocolo IoT*, por L. Llamas, 2019.

Llamas (2019) define al MQTT como “un servicio de mensajería push con patrón publicador/suscriptor”. En otras palabras, cuando un publicador (pub) desea publicar un mensaje, lo hará en un determinado tópico y se envía al broker como se observa en la Figura 36, desde ahí se filtra los mensajes que se enviaron y el suscriptor de dicho tópico recibirá el mensaje.

Figura 36

Proceso de publicación de mensajes por MQTT



Nota. Tomado de *¿Qué es MQTT? Su importancia como protocolo IoT*, por L. Llamas, 2019.

Otro mecanismo que tiene MQTT para mejorar asegurar el envío y recepción de los mensajes es la calidad de servicio (Quality of Service [QoS]), con esto se puede determinar si el mensaje fue recibido o no. MQTT tiene 3 niveles de QoS que van de 0 a 2:

- QoS 0: el mensaje es enviado, pero no se asegura que se reciba, también se conoce como “a lo mucho uno”.

- QoS 1: el mensaje es enviado y se asegura la entrega, pero cabe la posibilidad de repetir el mensaje recibido, también se lo conoce como “al menos uno”.
- QoS 2: el mensaje es enviado y se garantiza que el suscriptor lo reciba una sola vez, también se lo conoce como “exactamente uno”

Cada nivel tiene sus ventajas y desventajas, por lo que es responsabilidad del ingeniero seleccionar el nivel adecuado dependiendo del mensaje que se esté enviando y recibiendo.

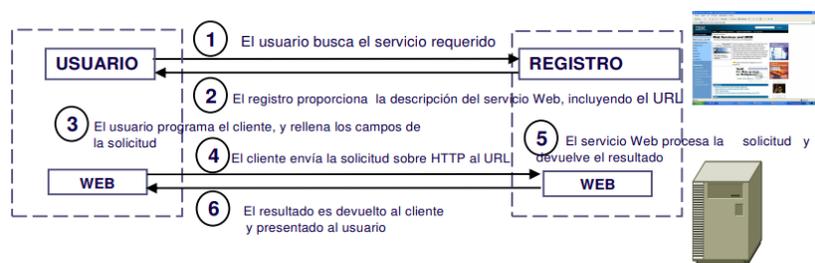
Servicios WEB

Un servicio se define como un procedimiento, método u objeto que cuenta con una interfaz estable y que puede ser invocado por un cliente. En base a esto, se define como servicio web al conjunto de aplicaciones con conexión a Internet, que permiten la comunicación con otras aplicaciones con la finalidad de publicarse, describirse, invocarse o localizarse dentro de la web (IBM, 2022).

Estos servicios permiten que dos instancias o máquinas se conecten entre sí como se observa en la Figura 37; esta tecnología cuenta con características como ser multiplataforma para evitar incompatibilidades entre las configuraciones que posean los elementos incluidos en la comunicación, y es distribuible para más de un cliente a la vez, por lo que debe soportar el acceso múltiple a un mismo servicio (IONOS Digital Guide, 2020).

Figura 37

Funcionamiento de los servicios web



Nota. Tomado de *Introducción a los servicios web*, por M. Gallegos, 2005.

Para realizar la comunicación los servicios web recurren al protocolo SOAP (Simple Object Access Protocol), que es una combinación del protocolo HTTP y el formato XML, permitiendo enviar archivos XML por protocolo HTTP. La utilización de esta tecnología permite la compatibilidad entre plataformas con diferentes configuraciones, dándole a los servicios web la característica de multiplataforma, y permitiendo que las aplicaciones que invocan al servicio web puedan estar desarrolladas en cualquier lenguaje de programación sin restricciones (Universidad de Vigo, 2004).

Amazon Web Services

Amazon Web Services, por sus siglas AWS, es una plataforma que ofrece recursos para el levantamiento de infraestructuras de comunicación de negocios o corporaciones, en forma de servicios web. Algunos de los servicios destacables que se encuentran en esta plataforma se muestran en la Figura 38.

Figura 38

Servicios ofrecidos por Amazon Web Services



Nota. Tomado de What is AWS? Introduction to Amazon Web Services, por B. Tale, 2021.

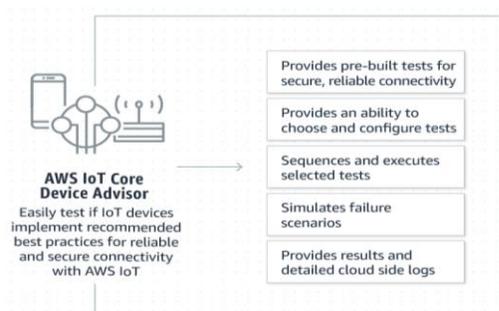
Dentro de los servicios que ofrece AWS se encuentran aquellos que permiten realizar computación en la nube (cloud computing), manejo de bases de datos, creación de redes virtuales, almacenamiento de archivos, utilización de Internet de las Cosas (IoT), transmisión de video, aplicaciones para inteligencias artificiales, entre otros.

Para cloud computing, AWS cuenta con el servicio Elastic Compute Cloud (EC2), que permite la creación de instancias a modo de máquinas virtuales, permitiendo la elección de su configuración como el sistema operativo, métodos de seguridad, características de procesador, memoria, cantidad de almacenamiento, entre otros parámetros. Estas instancias cuentan con direcciones IP públicas, de tal manera que es accesible desde cualquier localidad si se cuenta con conexión a internet (Amazon, 2022).

Para la comunicación al utilizar IoT, AWS ofrece el servicio IoT Core, que permite el envío y recepción de información mediante protocolos como MQTT, HTTPS (Hypertext Transfer Protocol Secure) y MQTT sobre WSS (Websockets). Como se muestra en la Figura 39, el uso de este servicio permite configurar los dispositivos que deban comunicarse entre sí de manera sencilla y realizar pruebas de comunicación mediante herramientas como MQTT test client, que permite comprobar la correcta comunicación entre los suscriptores de los tópicos que manipula el broker.

Figura 39

Funcionamiento del servicio Amazon IoT Core

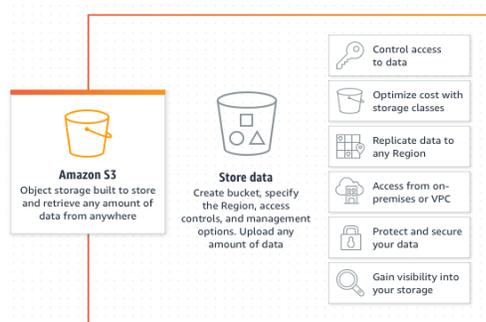


Nota. Tomado de *Amazon IoT Core Advisor*, por Amazon, 2022.

Para el almacenamiento de archivos, AWS ofrece el servicio Simple Storage Service (S3), el cual permite el alojamiento de archivos, información de bases de datos, aplicaciones móviles, entre otros, como se observa en la Figura 40. Este servicio ofrece métodos de seguridad para los archivos almacenados, acceso desde cualquier localidad, facilidad de manejo de datos; guardando la información en cubos o "buckets" que el usuario puede crear y personalizar dependiendo de sus necesidades.

Figura 40

Funcionamiento del servicio Amazon S3



Nota. Tomado de *Amazon S3*, por Amazon, 2022.

Visión artificial

García en su libro define la visión artificial como "la ciencia y la tecnología que permite a las máquinas ver" (García, 2012), con esta capacidad las máquinas ahora pueden interactuar de diferentes maneras con el entorno ya que el objetivo no es únicamente observar en escena que lo rodea, la máquina debe entender lo que está viendo.

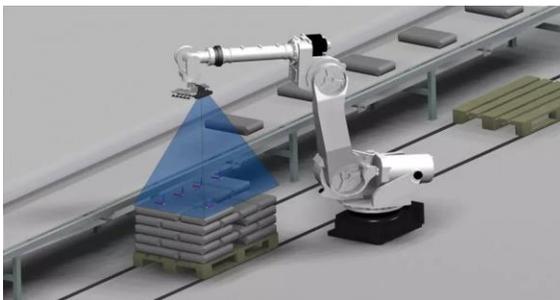
Actualmente se han implementado algoritmos de visión artificial o por computadora, en la industria, medicina, ciberseguridad, entre otras, siendo más rápidas para detectar y tomar acción (de ser implementada) que humanos con años de experiencia. En la Figura 41 se observa un ejemplo de aplicación de visión en la industrial, al implementar sistemas de visión para brazos robóticos.

Para que la máquina pueda “ver”, se utilizan algunas herramientas básicas y que se deben encontrar en las implementaciones de visión artificial, estas son:

- **Cámara:** Es el dispositivo encargado de emular la función del ojo humano, es decir que tiene un sensor que es capaz de capturar la luz que pase por una apertura. Dependiendo de la calidad del sensor, la imagen tendrá una mayor resolución y por lo tanto más fácil de hacer una detección, pero requiere de más recursos para procesar la imagen.
- **Iluminación:** esta es más una característica que una herramienta, sin embargo, cada escena que vaya a trabajar con visión artificial debe ser bien iluminada para que el sensor reciba una cantidad apropiada de luz y se facilite la detección.
- **Entornos de programación:** esta es la encargada del procesamiento de las imágenes y ejecutar el algoritmo de visión artificial. En la actualidad existen varios entornos para trabajar con visión artificial, pero de forma básica se puede utilizar lenguajes como C + + o Python, haciendo uso de la librería de código abierto OpenCV.

Figura 41

Brazo robótico con sistema de visión artificial



Nota. Tomado de *Visión artificial para mejorar la calidad*, por C. Juárez, 2020.

Open Computer Vision (OpenCV)

Desde el año 1999 aparece la primera versión de una librería que facilita el manejo y operación de imágenes por computador hasta la actualidad, OpenCV ha sido una herramienta

que permite de forma sencilla hacer detección de una gran variedad de objetos implementados por la propia librería o desarrollar una propia.

Las ventajas que han llevado a OpenCV a ser una librería tan utilizada es su licencia libre, la capacidad de ser multiplataforma y ser capaz de funcionar en una gran variedad de dispositivos y una buena documentación al respecto. En general, el proceso que se sigue para la detección de objetos es el siguiente:

- Adquisición de imagen: se puede obtener de dos tipos: foto o video, aunque su análisis se mantiene igual que el de una foto.
- Procesamiento: son operaciones simples que se aplican a la imagen para mejorar la detección, como un filtro, pasar a escala de grises o hacer una umbralización.
- Extracción de datos: dependiendo el tipo de detección que se esté utilizando se utilizarán diferentes métodos para dar solución a la extracción de datos.
- Extracción de información: consiste en convertir los datos en información útil para el usuario, por ejemplo, realizar una conversión de píxeles a centímetros.

Evaluación de los resultados

Una forma práctica de evaluar el desempeño de la visión artificial es utilizar las matrices de confusión como la que se observa en la Figura 42, ya que permiten determinar qué tan bueno es un algoritmo para predecir de manera correcta un determinado objeto de prueba.

Figura 42

Matriz de confusión para sistemas de visión artificial

Valores de predicción	Verdaderos positivos	Falsos positivos
	Falsos negativos	Verdaderos negativos
	Valores reales	

Nota: Verdaderos Positivos (TP), Falsos Positivos (FP), Falsos Negativos (FN) y Verdaderos Negativos (TN)

En la matriz de confusión se puede encontrar una terminología que permite hacer las evaluaciones pertinentes, en donde los verdaderos positivos y negativos corresponden a la capacidad del algoritmo de detectar y rechazar un objeto de prueba, mientras que los falsos positivos y negativos son los errores que comete el algoritmo para detectar o rechazar un objeto de prueba (XLSTAT, 2022).

Los resultados se evalúan en base a 4 parámetros obtenidos al operar los datos de la matriz de confusión, estos parámetros son:

- Sensibilidad: la tasa de casos positivos bien detectados

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

- Especificidad: la tasa de los casos negativos bien detectados

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

- Exactitud: la tasa de los datos clasificados correctamente

$$\text{Exactitud} = \frac{VP + VN}{\text{Total}}$$

- Precisión: la tasa de clasificado correcto cuando se predice positivamente

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Capítulo 3: Diseño

Desarrollo de conceptos

Casa de la calidad

Para proceder con el desarrollo de conceptos, fue necesario considerar los requerimientos que presenta el usuario con respecto a la celda de manufactura remota, de tal manera que la experiencia de utilizarla sea satisfactoria. Estos requerimientos fueron obtenidos al entrevistar a estudiantes de la asignatura de robótica industrial.

- Fácil de utilizar
- Accesibilidad a la plataforma desde diferentes dispositivos
- Visualización del funcionamiento del robot en tiempo real
- El comportamiento del robot sea modificable
- Vida útil prolongada

En base a estos requerimientos del usuario (Voice of Customer [VOC]) se plantean las características técnicas para el diseño.

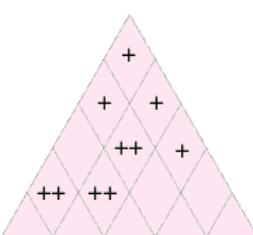
- Interfaz humano máquina (HMI) intuitiva
- Compatibilidad con Node-RED y Python
- Detección e identificación de piezas en el espacio de trabajo por visión artificial.
- Acceso a servidor virtual privado (VPS) mediante IP pública
- Diseño y construcción de ingeniería

Considerando los requerimientos del usuario y las características técnicas para el diseño se despliega el siguiente diagrama QFD (Quality Function Deployment), que es una herramienta que permite determinar la relación existente entre los requerimientos presentados por el usuario y los planteados como características técnicas para el diseño, reflejando si existe una relación fuerte, moderada o débil entre criterios.

Se utiliza esta metodología para mantener los requerimientos del cliente durante el proceso de desarrollo del proyecto, de tal manera que las soluciones que se implementen estén enfocadas hacia estas necesidades; para transformar los requerimientos del usuario en características técnicas que permitan solventar estas necesidades (Valenzuela, 2015). Los resultados de la evaluación por medio de la matriz QFD se reflejan en la Figura 43.

Figura 43

Matriz QFD



Dirección de mejora			X	▲	▲	X	X	
Requerimientos funcionales	Requerimientos del usuario	Importancia	Importancia relativa	Interfaz humano máquina (HMI) intuitiva	Integración de Node-RED y Python	Identificación de materiales por visión artificial.	Acceso a servidor virtual privado (VPS) mediante IP pública	Diseño y construcción de ingeniería
Fácil de utilizar	4	0.21	●	△	△		○	
Accesibilidad a la plataforma desde diferentes dispositivos	5	0.26	○	●		●	○	
Visualización del funcionamiento del robot en tiempo real	4	0.21	○	●	○	△		
Comportamiento del robot modificable	3	0.16		○	○		●	
Vida útil prolongada	3	0.16				●	○	
Importancia técnica			15	22	7	19	18	
Peso relativo (%)			18.5	27.1	8.64	23.5	22.2	

En las tablas 5 y 6 se detallan las simbologías utilizadas para el desarrollo de la casa de la calidad (QFD).

Tabla 5*Simbología para las relaciones de la matriz QFD*

Símbolo	Descripción	Valor
	Relación Fuerte	9
	Relación Moderada	3
	Relación Débil	1
	Relación nula	0

Tabla 6*Simbología utilizada en la matriz QFD*

Símbolo	Descripción
++	Correlación fuerte positiva
+	Correlación positiva
-	Correlación negativa
	Ninguna

Conclusiones de la casa de la calidad

Una vez concluido el análisis de los requerimientos del cliente (VOC) y los requerimientos técnicos (la voz del ingeniero), en la Tabla 7 se muestran los resultados obtenidos en base a la importancia técnica de cada uno de los requisitos.

Tabla 7

Resultados de la evaluación de la matriz QFD

Requerimientos técnicos	Peso relativo (%)
Interfaz humano máquina (HMI) intuitiva	18.5
Compatibilidad con Node-RED y Python	27.1
Detección e identificación de piezas por visión artificial.	8.64
Acceso a servidor virtual privado (VPS) mediante IP pública	23.5
Diseño y construcción de ingeniería	22.2

Conociendo los requerimientos tanto del usuario como los técnicos, en la Tabla 8 se desglosan las funciones que debe cumplir la celda de manufactura para poder plantear subsistemas que cubran cada una de estas.

Tabla 8

Subsistemas que conforman la celda de manufactura

Subsistema	Función
Subsistema mecánico	Movimiento del robot, recolección de piezas
Subsistema electrónico	Sensado del robot, procesamiento de instrucciones enviadas por el usuario
Subsistema de visión artificial	Detección e identificación de piezas en el espacio de trabajo del robot
Subsistema de control	Control de posición de los grados de libertad del robot
Subsistema de comunicación	Comunicación para el envío de información de los sensores y de las instrucciones del usuario
Subsistema de interfaz gráfica	Monitorización en tiempo real, control manual y automático del robot.

Diseño del subsistema mecánico

Dentro de la celda de manufactura remota, el subsistema mecánico comprende la estructura del robot SCARA, que es la encargada de realizar las tareas de movimiento y recolección de piezas en el espacio de trabajo, de acuerdo a las instrucciones enviadas por el usuario. Para el diseño de este subsistema se tomaron en cuenta las siguientes consideraciones

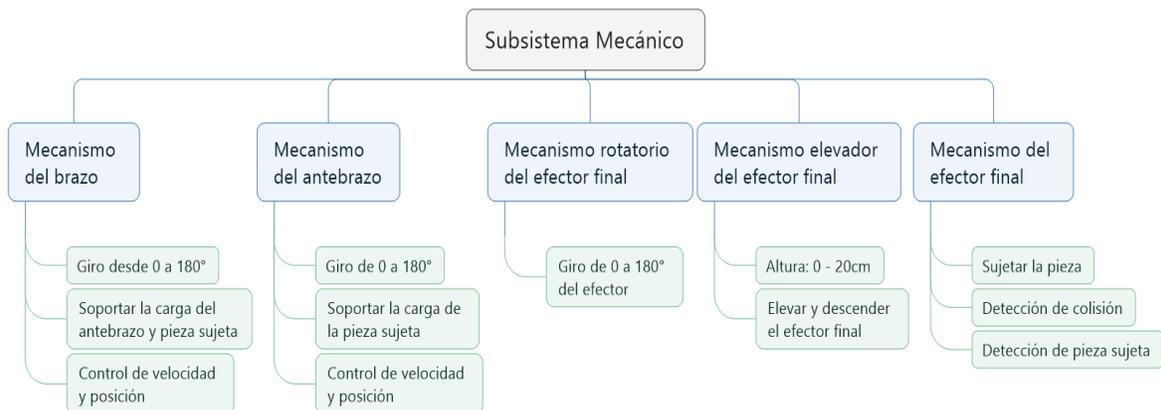
- Tamaño adecuado con respecto a la celda de manufactura
- Material adecuado en las piezas (Considerando la resistencia y peso)
- Costo de fabricación de las piezas
- Facilidad de manufactura
- Disponibilidad de los materiales dentro del mercado nacional

Esta estructura del robot SCARA puede desglosarse en cinco partes, las cuales permiten que el robot funcione de forma adecuada en la celda de manufactura.

- Mecanismo del brazo (Grado de libertad Q1)
- Mecanismo del antebrazo (Grado de libertad Q2)
- Mecanismo rotatorio del efector final (Grado de libertad Q3)
- Mecanismo elevador del efector final (Grado de libertad Q4)
- Mecanismo del efector final

En la Figura 44 se muestra en forma de diagrama los requerimientos que presentan cada uno de estos mecanismos que conforman la estructura del robot SCARA

Figura 44

Requerimientos del subsistema mecánico

La instrucción que envía el usuario contiene información de los cuatro grados de libertad y si se requiere que se active o desactive el efector final, por lo que el funcionamiento del robot se basa en: mover los grados de libertad Q1 y Q2 de forma simultánea, una vez posicionado sobre el plano, rotar el efector en base al valor del grado de libertad Q3; si la instrucción lo indica, realizar el movimiento de bajar, recoger pieza con el efector final y finalmente elevar el efector final.

Mecanismo de brazo

El mecanismo del brazo es el que permite posicionar al efector final en base al valor del grado de libertad Q1. Esta estructura al contener al primer eslabón, debe soportar con las cargas que producen el peso del resto del robot SCARA, adicionalmente del peso de la pieza sujeta. Al ser un robot SCARA de tipo RRRP, el primer grado de libertad es rotacional, por lo que su mecanismo se basa en un motor que rota un eje, el mismo que provoca que el primer eslabón se movيلة a la posición indicada por la instrucción. Para la construcción de este mecanismo se plantearon varias alternativas, mostradas a continuación:

Alternativa 1: Estructura con piezas impresas en PLA

Se planteó construir las piezas que comprenden el mecanismo del primer grado de libertad en manufactura aditiva en PLA como se muestra en la Figura 45, de tal manera que el motor se encuentre en la parte baja de la estructura, y por medio de un mecanismo de piñón - corona realizar el movimiento rotatorio para ubicar el eslabón en la posición correcta de acuerdo al valor del grado de libertad Q_1 . El eslabón, que es un tubo de hierro cuadrado se une al eje por medio de una pieza impresa en PLA, la cual provoca que gire.

Figura 45

Alternativa 1 para el mecanismo de brazo

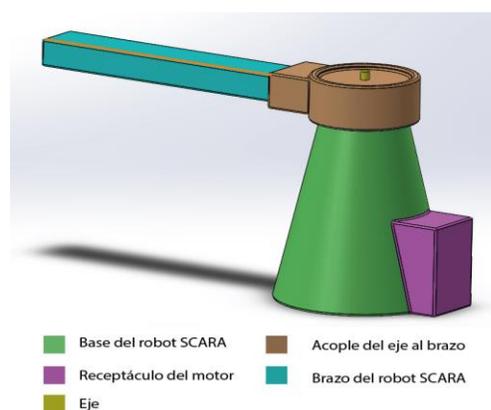


Tabla 9

Ventajas y desventajas de la alternativa 1 del brazo

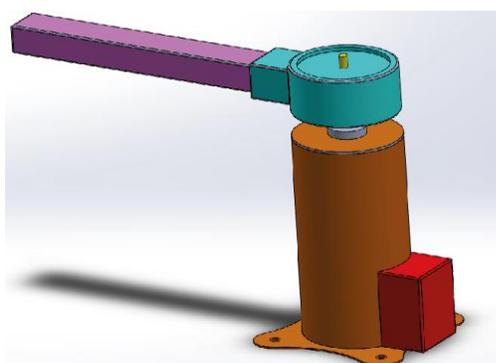
Ventajas	Desventajas
Piezas replicables	Alto costo de manufactura
	Tiempo de manufactura considerable
	Baja resistencia mecánica
	Inestabilidad de la estructura

Alternativa 2: Estructura metálica con acoples plásticos

La base del robot es un tubo metálico de hierro redondo, que por medio de chumaceras y rodamientos mantienen un eje vertical, el cual gira por un mecanismo de piñón - corona ubicado en la parte inferior, como se observa en la Figura 46. El brazo, que es un tubo cuadrado de hierro se une al eje por medio de una pieza impresa en 3D, la cual provoca que este gire.

Figura 46

Alternativa 2 para el mecanismo de brazo



- Base del robot SCARA
- Acople del eje al brazo
- Receptáculo del motor
- Brazo del robot SCARA
- Eje

Tabla 10

Ventajas y desventajas de la alternativa 2 del brazo

Ventajas	Desventajas
Piezas replicables	Tiempo de manufactura considerable por las piezas impresas
Se reduce el costo de manufactura	Baja resistencia mecánica
	Inestabilidad de la estructura

Alternativa 3: Mecanismo con estructura metálica y acoples metálicos

La base del robot es un tubo metálico de hierro redondo, que por medio de chumaceras y rodamientos mantienen un eje vertical, el cual se acopla directamente a un motor DC ubicado en la parte superior de la estructura, como se muestra en la Figura 47. El brazo, que es un tubo rectangular de aluminio, se une al eje por medio de acoples metálicos que permiten que este gire cuando el motor se encuentra activo.

Figura 47

Alternativa 3 para el mecanismo de brazo

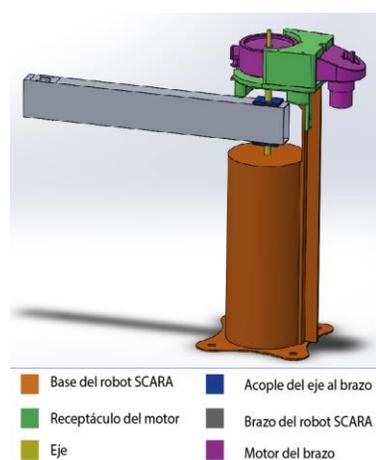


Tabla 11

Ventajas y desventajas de la alternativa 3 del brazo

Ventajas	Desventajas
Piezas replicables	Aumenta el peso de la estructura
Se reduce el costo de manufactura	
Alta resistencia mecánica	
Alta estabilidad de la estructura	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para el mecanismo del grado de libertad Q1.

Tabla 12

Matriz de evaluación para las alternativas de solución al mecanismo del brazo

Criterio	Ponderación	Estructura		Base metálica con acoples plásticos		Base metálica con acople metálico	
		CA	EP	CA	EP	CA	EP
		CA	EP	CA	EP	CA	EP
Costo	20%	2	0.4	3	0.6	4	0.8
Estabilidad	30%	2	0.6	3	0.9	5	1.5
Resistencia	30%	2	0.6	3	0.9	5	1.5
Facilidad de manufactura	20%	3	0.6	3	0.6	4	0.8
Total		2.2		3.0		4.6	

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la tercera alternativa, que incluye base metálica con acoples metálicos para unir el eje con el brazo del robot, puesto que ofrece mejor desempeño en lo que corresponde a resistencia y facilidad de manufactura.

Mecanismo del antebrazo

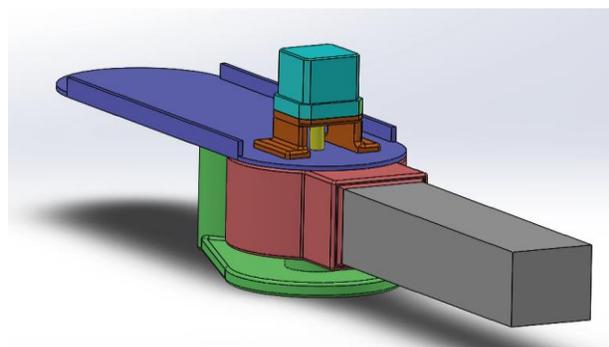
El mecanismo del antebrazo es el que permite posicionar al efector final en base al valor del grado de libertad Q2. Esta estructura debe poder soportar el peso de los elementos contenidos en el antebrazo (motores, mecanismos de elevación, circuitos electrónicos), adicionalmente de la pieza sujeta. Esta articulación es de tipo rotacional, por lo que se plantean tres alternativas que pueden cubrir esta necesidad.

Alternativa 1: Motor a pasos acoplado al eje

El mecanismo cuenta con una pieza impresa en PLA, la cual soporta a una platina que será el antebrazo cuya función es contener el mecanismo elevador y su respectivo circuito. Esta platina está soportada por una pieza que sirve de apoyo en la parte inferior, de tal manera que se mantenga horizontal, tal como se muestra en la Figura 48. El movimiento es producido por un motor a pasos acoplado al eje de tal manera que gire la platina al ser fijo el eje al que se sujeta.

Figura 48

Alternativa 1 para el mecanismo de antebrazo



■ Soporte de la platina	■ Platina	■ Brazo
■ Pieza de apoyo inferior	■ Soporte del motor	
■ Eje	■ Motor	

Tabla 13

Ventajas y desventajas de la alternativa 1 del antebrazo

Ventajas	Desventajas
Piezas replicables	Elevado costo de manufactura
Alta precisión por motor a pasos	Tiempo de manufactura considerable
	Baja resistencia mecánica

Alternativa 2: Motor DC con caja reductora de engranajes

Similar a la alternativa 1, el mecanismo cuenta con una pieza impresa en PLA, la cual soporta a una platina que será el antebrazo cuya función es contener el mecanismo elevador y su respectivo circuito, como se observa en la Figura 49. Esta platina está soportada por una pieza que sirve de apoyo en la parte inferior, de tal manera que se mantenga horizontal. El movimiento se produce por un motor DC que cuenta con caja reductora, que se acopla al eje para provocar que el antebrazo gire.

Figura 49

Alternativa 2 para el mecanismo de antebrazo

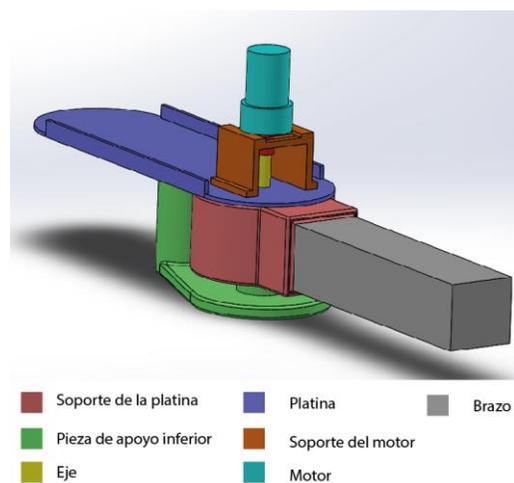


Tabla 14

Ventajas y desventajas de la alternativa 2 del antebrazo

Ventajas	Desventajas
Piezas replicables	Elevado costo de manufactura
	Tiempo de manufactura considerable
	El espacio requerido para alojar el motor es mayor

Ventajas**Desventajas**

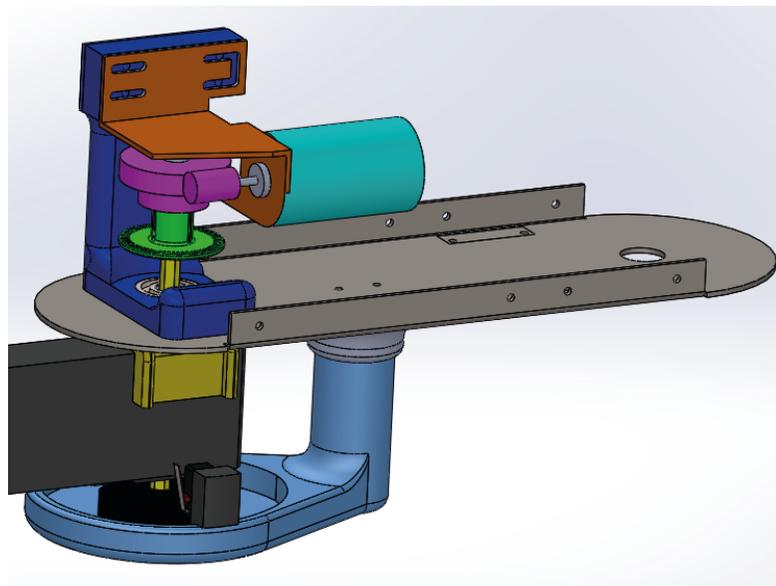
Precisión reducida en comparación al motor a pasos

Alternativa 3: Motor DC acoplado al mecanismo sinfín - corona

El mecanismo conserva la pieza que sirve de soporte en la parte inferior del antebrazo, con la modificación que se utiliza un motor DC que se acopla a un mecanismo sinfín - corona, el cual transmite el movimiento al antebrazo por medio de un acople que une la corona del mecanismo con el eje, como se muestra en la Figura 50. Esta estructura es soportada por una pieza impresa en PLA que sostiene al soporte del motor y a un rodamiento.

Figura 50

Alternativa 3 para el mecanismo de antebrazo



- | | |
|---|---|
|  Brazo |  Motor DC |
|  Pieza de apoyo inferior |  Mecanismo sinfin - corona |
|  Eje |  Soporte del motor DC |
|  Platina (Antebrazo) |  Acople de ejes |
|  Soporte de estructura | |

Tabla 15

Ventajas y desventajas de la alternativa 3 del antebrazo

Ventajas	Desventajas
Piezas replicables	Precisión reducida contrastado con el motor a pasos
Tiempo de manufactura reducido	
El espacio necesario es reducido	
Facilidad de repuestos	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para el mecanismo del grado de libertad Q2.

Tabla 16

Matriz de evaluación para las alternativas de solución al mecanismo del antebrazo

Criterio	Ponderación	Mecanismo de motor a pasos		Mecanismo motor DC con caja reductora		Mecanismo motor DC con sinfín-corona	
		CA	EP	CA	EP	CA	EP
Costo	20%	1	0.2	3	0.6	4	0.8
Estabilidad	30%	3	0.9	3	0.9	5	1.5
Resistencia	30%	3	0.9	4	1.2	5	1.5
Facilidad de manufactura	20%	3	0.6	3	0.6	4	0.8
Total			2.6		3.3		4.6

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la tercera alternativa, que incluye el motor DC con el mecanismo sinfín - corona, puesto que es el que ofrece mejor desempeño en base a los criterios establecidos. Adicionalmente cabe destacar que es el mecanismo que mejor optimiza el espacio a utilizar dentro del antebrazo, en comparación a las alternativas que utilizan motor a pasos y motor con caja reductora.

Mecanismo rotatorio del efector final

El mecanismo del antebrazo es el que permite posicionar al efector final en base al valor del grado de libertad Q3. Este mecanismo permite rotar a Q3 en un ángulo que puede encontrarse entre 0 y 180°. Para cubrir esta necesidad se plantearon tres alternativas de solución.

Alternativa 1: Motor DC con juego de engranajes

El mecanismo cuenta con una base que soporta a las piezas que sujetan al motor DC y a los engranajes que permiten el giro del eje del efector, como se muestra en la Figura 51. Este motor recibe una señal y realiza el giro respectivo.

Figura 51

Alternativa 1 para el mecanismo rotatorio del efector

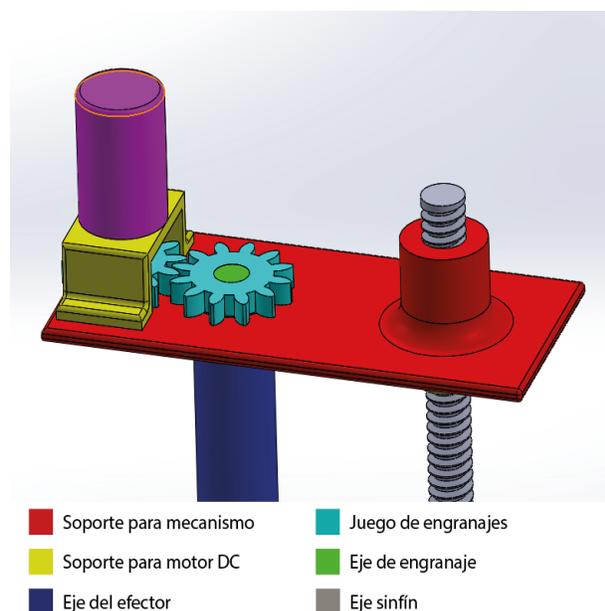


Tabla 17

Ventajas y desventajas de la alternativa 1 del mecanismo rotatorio del efector

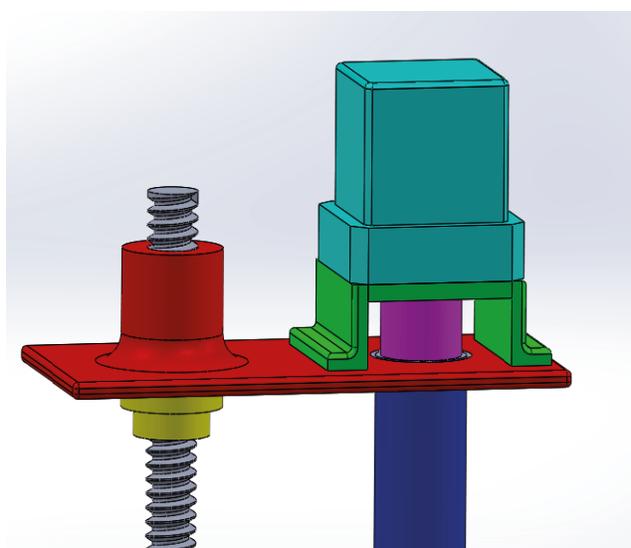
Ventajas	Desventajas
Mecanismo sencillo	El mecanismo ocupa espacio considerable
Bajo costo	Se requiere realizar un sistema de control de posición
	Imprecisión en el ángulo medido por las tolerancias del juego de engranaje

Alternativa 2: Mecanismo con motor a pasos

El mecanismo cuenta con una base que soporta a las piezas que sujetan al motor a pasos que, con un acople al eje del efector final, permite el giro del eje del efector como se muestra en la Figura 52. Este motor recibe una señal y realiza el giro respectivo.

Figura 52

Alternativa 2 para el mecanismo rotatorio del efector



- | | |
|---|--|
| ■ Soporte para mecanismo | ■ Motor a pasos |
| ■ Soporte para motor a pasos | ■ Acople motor - eje del efector |
| ■ Acople para eje sinfín | ■ Eje del efector |
| ■ Eje sinfín | |

Tabla 18

Ventajas y desventajas de la alternativa 2 del mecanismo rotatorio del efector

Ventajas	Desventajas
Mecanismo sencillo	El mecanismo ocupa espacio considerable
Alta precisión	Costo considerable

Alternativa 3: Servomotor

El mecanismo cuenta con una base que soporta a un servomotor, el cual dependiendo de la instrucción recibida permite que el eje del efector final realice el giro respectivo. El servomotor se une al eje del efector por medio de un acople que encaja en la salida del motor como se observa en la Figura 53.

Figura 53

Alternativa 3 para el mecanismo rotatorio del efector

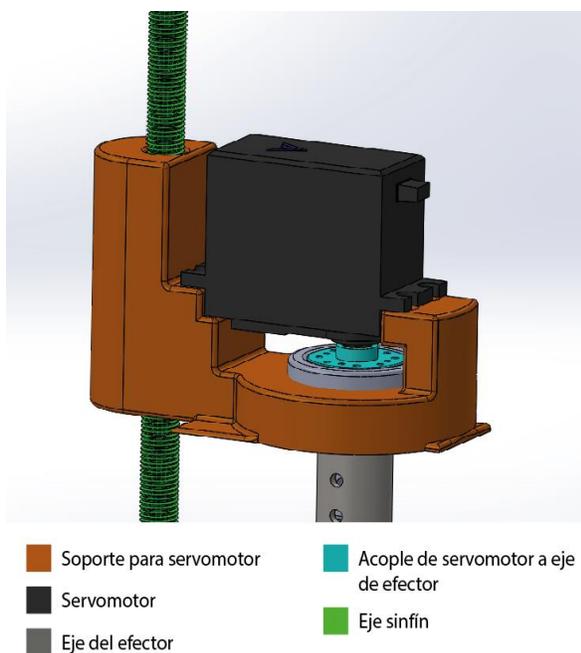


Tabla 19

Ventajas y desventajas de la alternativa 3 del mecanismo rotatorio del efector

Ventajas	Desventajas
Mecanismo sencillo	Costo considerable dependiendo del motor utilizado
No requiere sistema de control	
Alta precisión	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para el mecanismo del grado de libertad Q3.

Tabla 20

Matriz de evaluación para las alternativas de solución al mecanismo rotatorio del efector

Criterio	Ponderación	Motor DC con engranajes		Motor a pasos		Servomotor	
		CA	EP	CA	EP	CA	EP
Costo	20%	3	0.6	2	0.4	3	0.6
Precisión	30%	3	0.9	4	1.2	4	0.9
Espacio que ocupa	25%	2	0.5	3	0.75	4	1
Facilidad de manufactura	25%	3	0.75	4	1	5	1.25
Total			2.75		3.35		3.75

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la tercera alternativa, que incluye el servomotor que se acopla al eje del efector final, puesto que es el que ofrece mejor desempeño en base a los criterios establecidos.

Mecanismo elevador del efector final

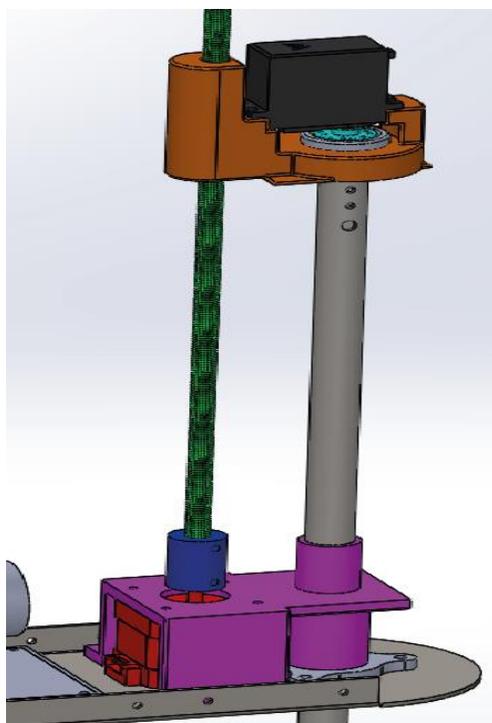
El mecanismo elevador es el encargado de realizar el movimiento de subida y bajada el efector final para que se pueda tomar un objeto del espacio de trabajo del robot SCARA. Para cubrir esta necesidad se plantean dos alternativas.

Alternativa 1: Eje sinfín con motor a pasos.

Se plantea un mecanismo que cuente con un motor a pasos, el cual por medio de un acople esté unido al eje sinfín, el cual provoque que el mecanismo rotatorio del efector final se eleve y descienda, como se muestra en la Figura 54.

Figura 54

Alternativa 1 para el mecanismo elevador del efector final



- Eje sinfín
- Soporte de estructura
- Motor
- Acople de motor a pasos a eje

Tabla 21

Ventajas y desventajas de la alternativa 2 del mecanismo rotatorio del efector

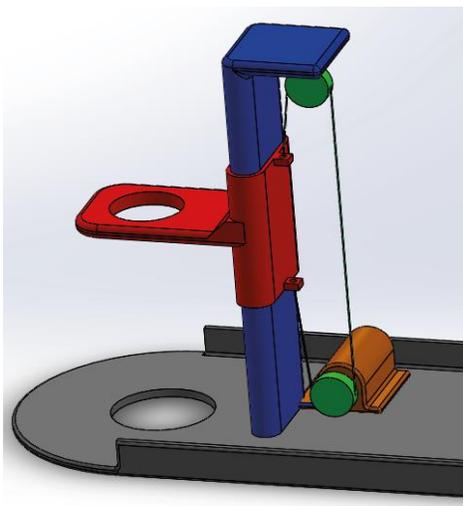
Ventajas	Desventajas
Mecanismo sencillo	Velocidad baja de subida y bajada
Piezas estándar	
Fácil ensamble	

Alternativa 2: Juego de poleas con motor DC

Se plantea un mecanismo que incluya un juego de poleas, las cuales mueven una banda que se encuentra fija contra una estructura corrediza que se eleva y desciende dependiendo del sentido del motor DC, el mismo que está acoplado a una de las poleas. Esta estructura se muestra en la Figura 55.

Figura 55

Alternativa 3 para el mecanismo elevador del efector final



- Soporte de la estructura
- Soporte del motor
- Estructura corrediza
- Juego de poleas

Tabla 22

Ventajas y desventajas de la alternativa 3 del mecanismo rotatorio del efector

Ventajas	Desventajas
Mecanismo sencillo	Posible inestabilidad del mecanismo por utilización de bandas
Piezas estándar	Ocupa espacio considerable dentro de la estructura del antebrazo Requiere de un sistema de freno para evitar que se desplace la estructura corrediza cuando el motor no esté activo

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para el mecanismo del grado de libertad Q4.

Tabla 23

Matriz de evaluación para las alternativas de solución al mecanismo rotatorio del efector

Criterio	Ponderación	Eje sinfín		Juego de poleas	
		CA	EP	CA	EP
Costo	20%	3	0.6	5	1
Estabilidad	40%	5	2	3	1.2
Espacio que ocupa	20%	3	0.6	3	0.6
Facilidad de manufactura	20%	4	0.8	3	0.6
Total			4		3.2

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la primera alternativa, que incluye el motor a pasos que se acopla al eje sinfín, puesto que es el que ofrece mejor desempeño en base a los criterios establecidos. Adicionalmente, ocupa menos espacio dentro de la estructura del antebrazo, y no requiere un sistema de frenado para mantener la posición, como lo requeriría la alternativa que ocupa el juego de poleas.

Mecanismo del efector final

El mecanismo del efector final es el encargado de sujetar los objetos que se encuentren en el espacio de trabajo del robot SCARA. Las piezas que se planea utilizar en el espacio de trabajo son metálicas de hierro, por lo que se utiliza un electroimán para sujetarlas. El mecanismo del efector final debe ser capaz de albergar al electroimán y sus respectivos sensores finales de carrera que detecten colisiones y que el objeto se encuentra sujeto. Para cubrir estos requerimientos se proponen dos alternativas.

Alternativa 1: Mecanismo con finales de carrera externos

El mecanismo cuenta con un soporte que aloja al electroimán en la parte inferior, y de manera externa sirve de apoyo para los fines de carrera superior e inferior, como se muestra en la Figura 56. Este soporte puede deslizarse en el eje del efector, de tal manera que cuando el efector se eleva, el soporte desciende, y cuando el efector desciende hasta encontrar una superficie, el soporte sube. El fin de carrera superior indica que el efector ha descendido hasta encontrarse con un objeto o superficie, mientras que el fin de carrera inferior tiene la función de detectar si la pieza se encuentra sujeta al momento de elevar el efector.

Figura 56

Alternativa 1 para el mecanismo del efector final

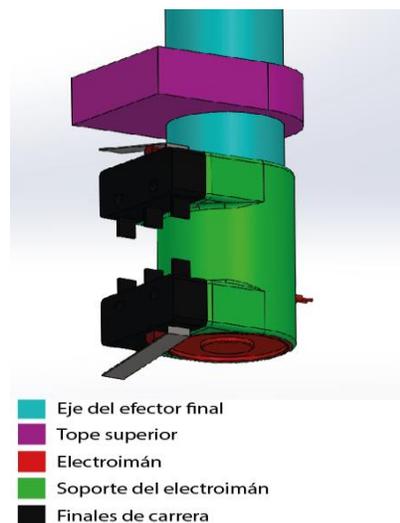


Tabla 24

Ventajas y desventajas de la alternativa 1 del mecanismo del efector final

Ventajas	Desventajas
Mecanismo sencillo	Propenso a dar lecturas erróneas al no deslizarse correctamente el soporte en el eje del efector
Fácil de ensamblar	Propenso a descalibraciones si el tope superior se desplaza

Alternativa 2: Fin de carrera interno

El mecanismo cuenta con una estructura soporte de dos partes que aloja al electroimán en la parte inferior, y de manera externa sirve de apoyo para el fin de carrera inferior. Este soporte se une al eje del efector final por medio de un acople que se estira cuando el efector se eleva, y se retrae cuando este encuentra una superficie al descender, como se muestra en la Figura 57. El fin de carrera interior indica que el efector se encuentra chocando contra una superficie, mientras que el fin de carrera exterior cumple la función de indicar si existe una pieza sujeta al momento de elevar el efector.

Figura 57

Alternativa 2 para el mecanismo del efector final

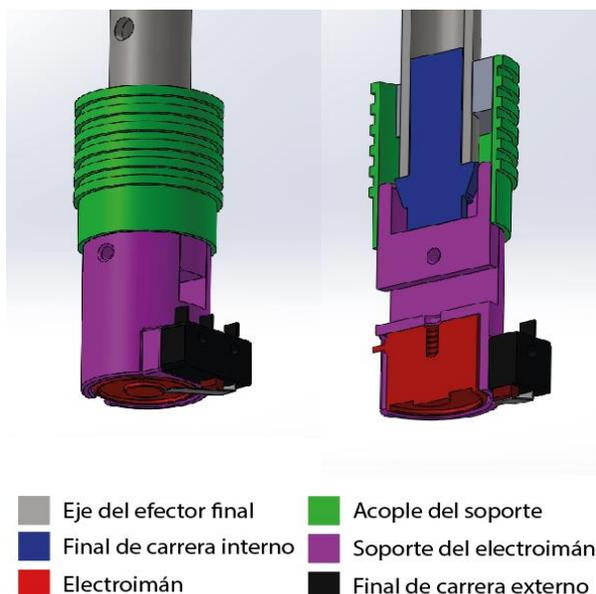


Tabla 25

Ventajas y desventajas de la alternativa 2 del mecanismo del efector final

Ventajas	Desventajas
Mecanismo sencillo	Requiere de más fuerza para activar el fin de carrera interno en comparación con la alternativa 1
No existe riesgo de atascamiento del mecanismo	
Fácil de ensamblar	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para el mecanismo del efector final.

Tabla 26

Matriz de evaluación para las alternativas de solución al mecanismo del efector final

Criterio	Ponderación	Fines de carrera		Fin de carrera interno y externo	
		externos			
		CA	EP	CA	EP
Costo	20%	4	0.8	4	0.8
Confiabilidad	40%	3	1.2	5	2.0
Facilidad de manufactura	20%	4	0.8	3	0.6
Facilidad de ensamble	20%	3	0.6	4	0.8
Total			3.4		4.2

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la segunda alternativa, que incluye un interruptor final de carrera interno y uno externo, puesto que es el

que ofrece mejor desempeño en base a los criterios establecidos. Adicionalmente, su desmontaje es más sencillo que el de la alternativa de los fines de carrera externos.

Modelamiento de las selecciones

Los apartados anteriores se utilizaron para determinar cuáles fueron las mejores alternativas para cada una de las partes que conforman el subsistema mecánico. En este apartado se modelan en software CAD 3D, de tal manera que se visualice los esfuerzos, desplazamientos y factores de seguridad que presentan cada uno de los mecanismos seleccionados por medio de simulaciones aplicando las cargas que soporta cada parte.

Mecanismo del brazo. Posterior a la selección se modeló el mecanismo del brazo como se muestra en la Figura 58 con las dimensiones que se muestran en la Figura 59 y 60. Para que la simulación de esfuerzos se vea de manera más detallada se divide la estructura en dos partes, separando la estructura que soporta el motor DC, como se muestran en las Figuras 61, 62 y 63.

Figura 58

Modelado 3D del mecanismo del brazo

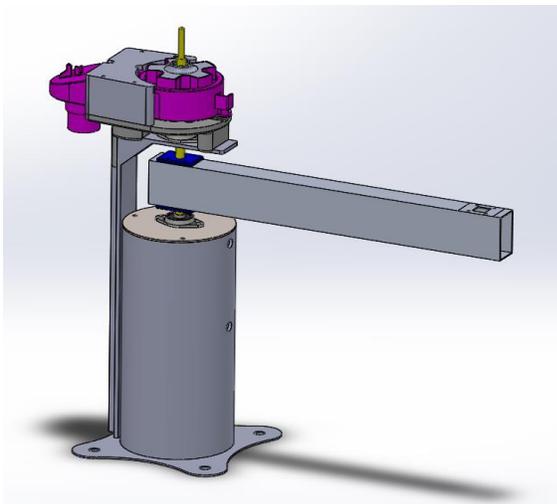


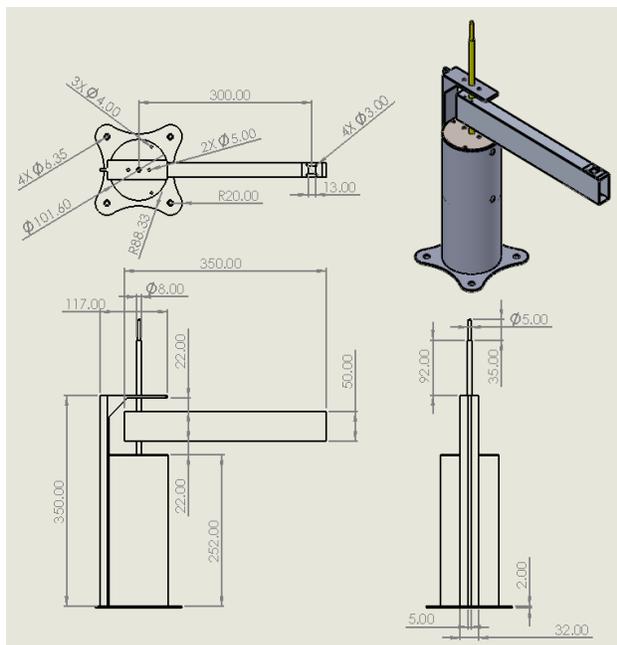
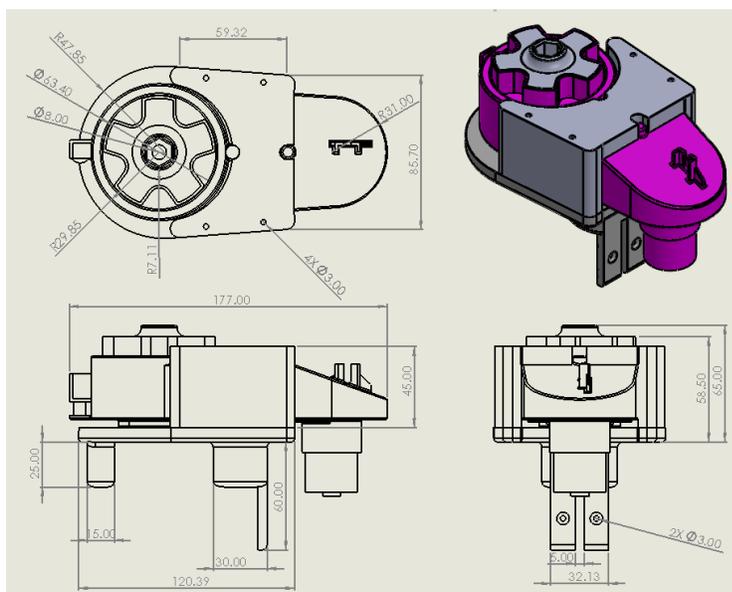
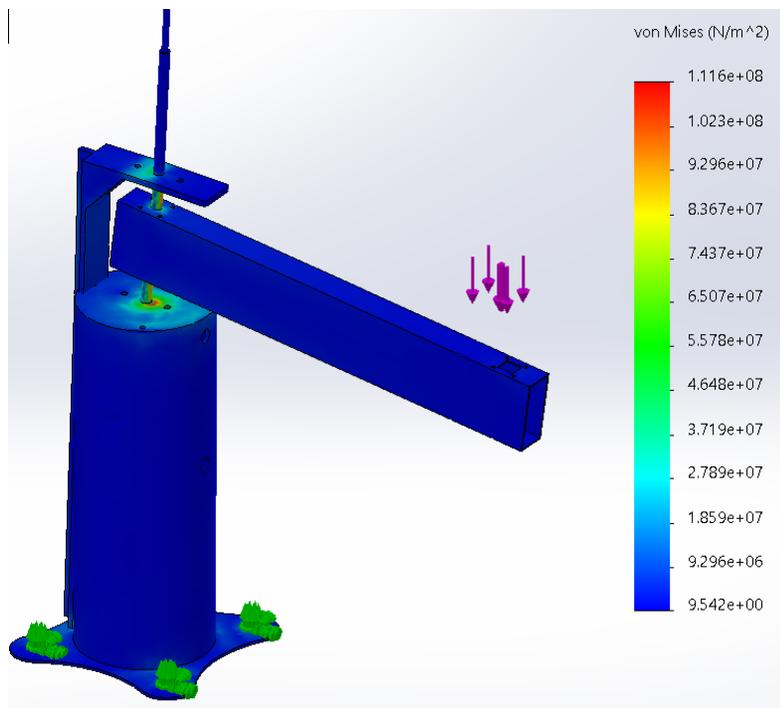
Figura 59*Dimensiones del mecanismo del brazo***Figura 60***Dimensiones del mecanismo del brazo (Soporte del motor)*

Figura 61

Simulación de esfuerzos de von Mises para el mecanismo del brazo

**Figura 62**

Simulación de desplazamientos para el mecanismo del brazo

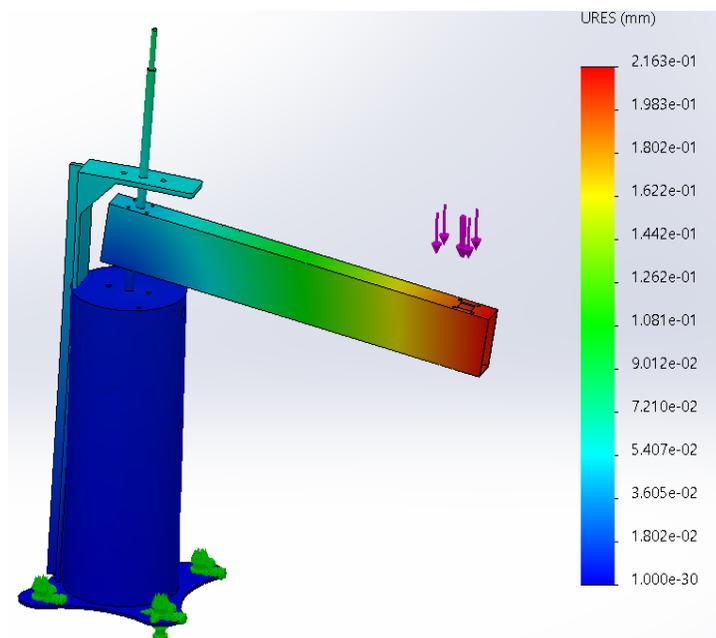
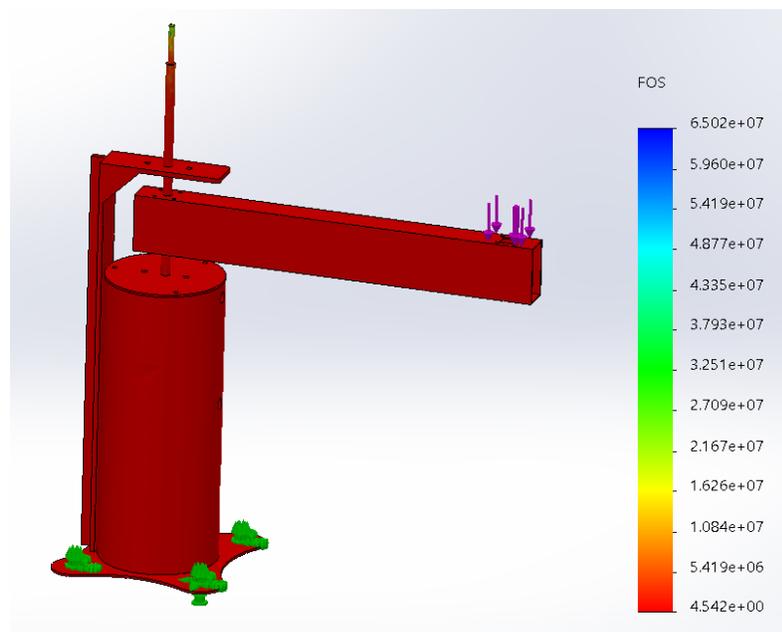


Figura 63

Simulación de factores de seguridad para el mecanismo del brazo



En la estructura que soporta el motor DC se hace énfasis en el análisis del acople que permite unir el eje con el motor, puesto que es la pieza crítica en el mecanismo del brazo. En las Figuras 64, 65 y 66 se muestran los resultados de las simulaciones.

Figura 64

Simulación de esfuerzos de von Mises para el acople del motor del brazo

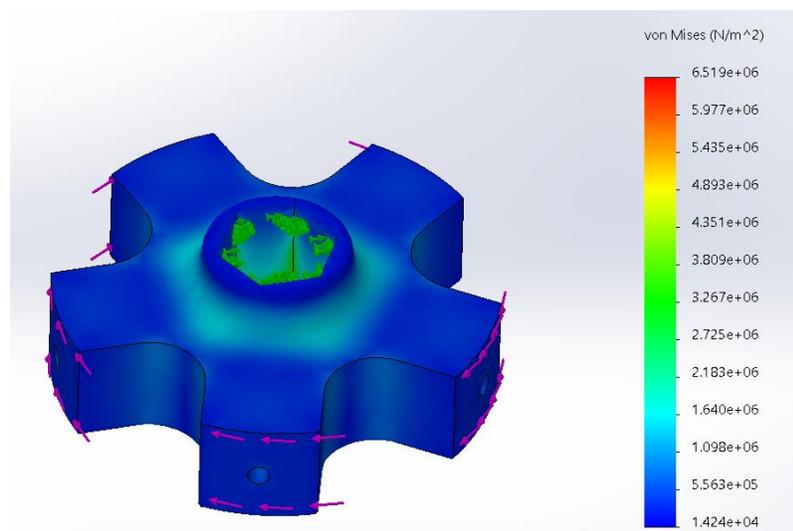
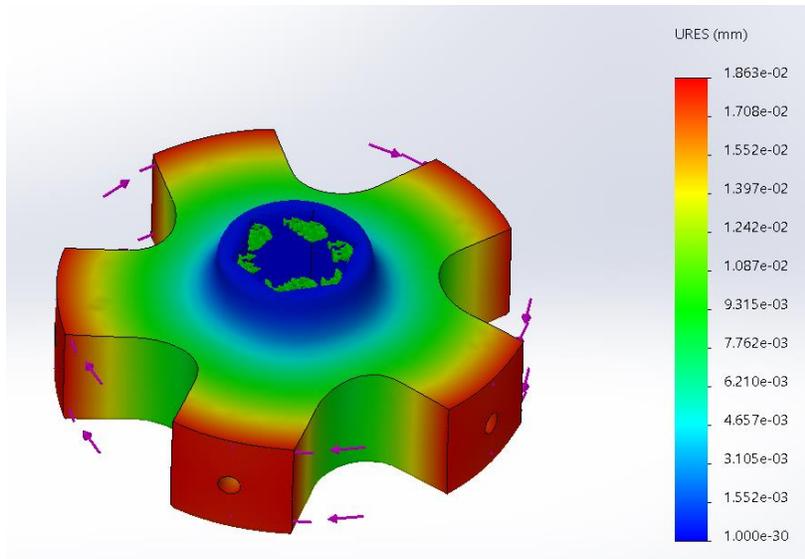
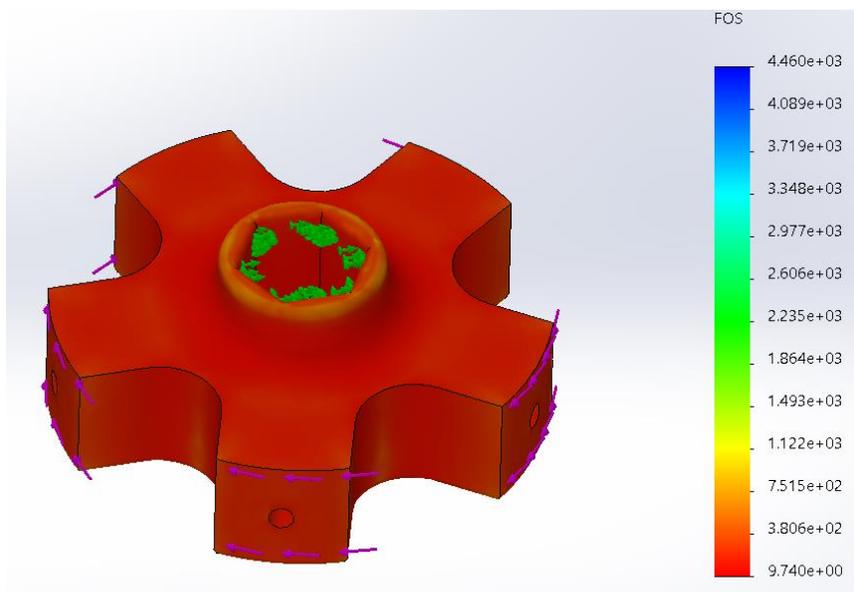


Figura 65

Simulación de desplazamiento para el acople del motor del brazo

**Figura 66**

Simulación de factores de seguridad para el acople del motor del brazo



Mecanismo del antebrazo. Se modeló el mecanismo del antebrazo como se muestra en la Figura 67 con las dimensiones que se muestran. En las figuras 68, 69 y 70 se observan los resultados de la simulación de esfuerzos, desplazamientos y factor de seguridad.

Figura 67

Mecanismo del antebrazo

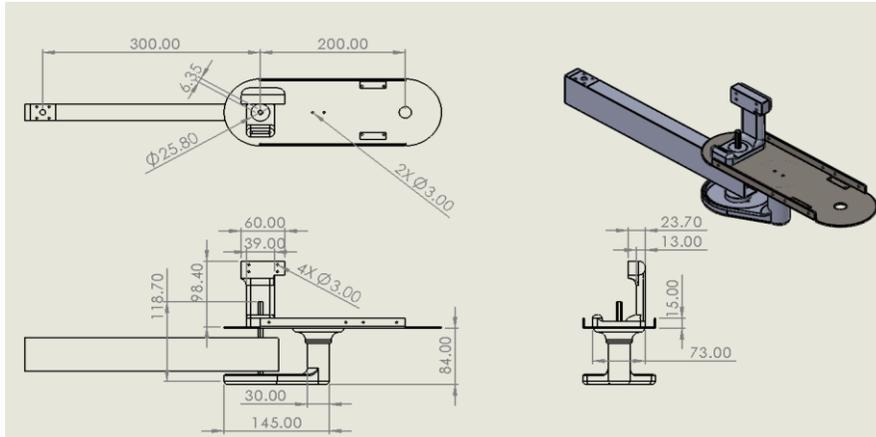


Figura 68

Simulación de esfuerzos de von Mises para el mecanismo del antebrazo

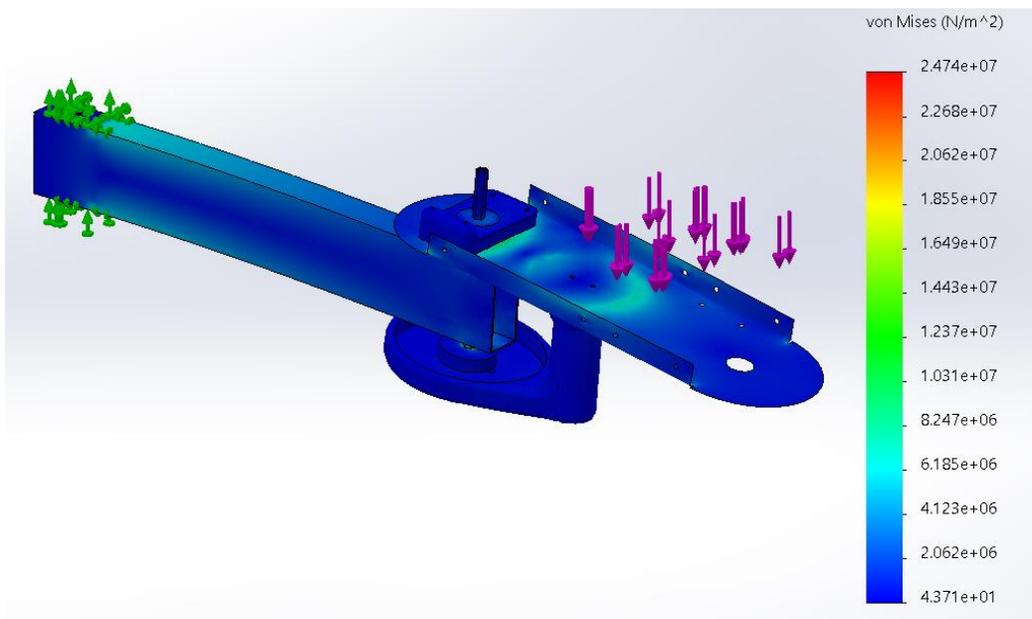
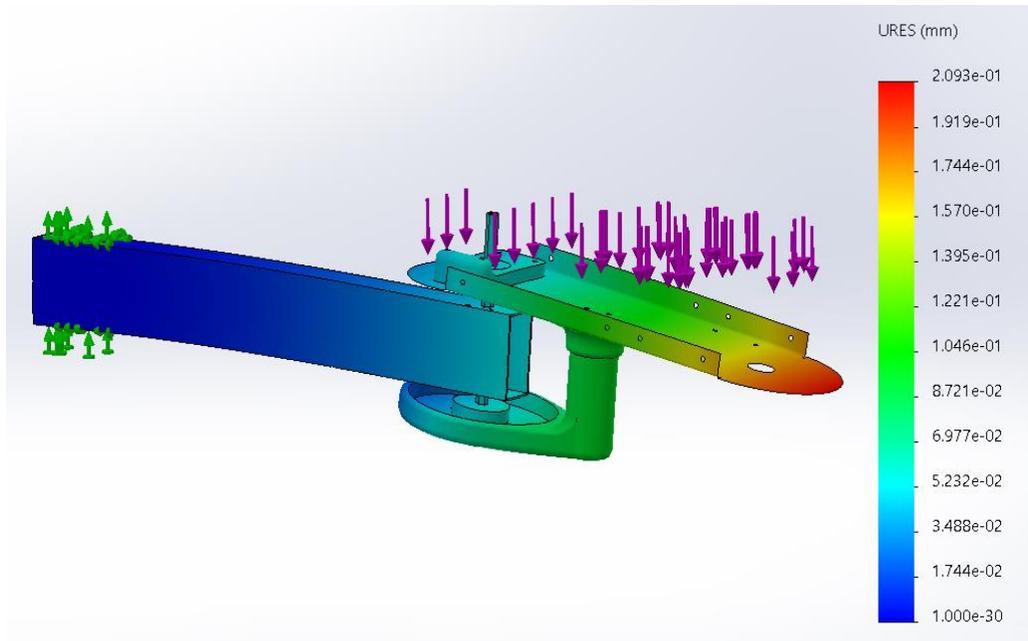
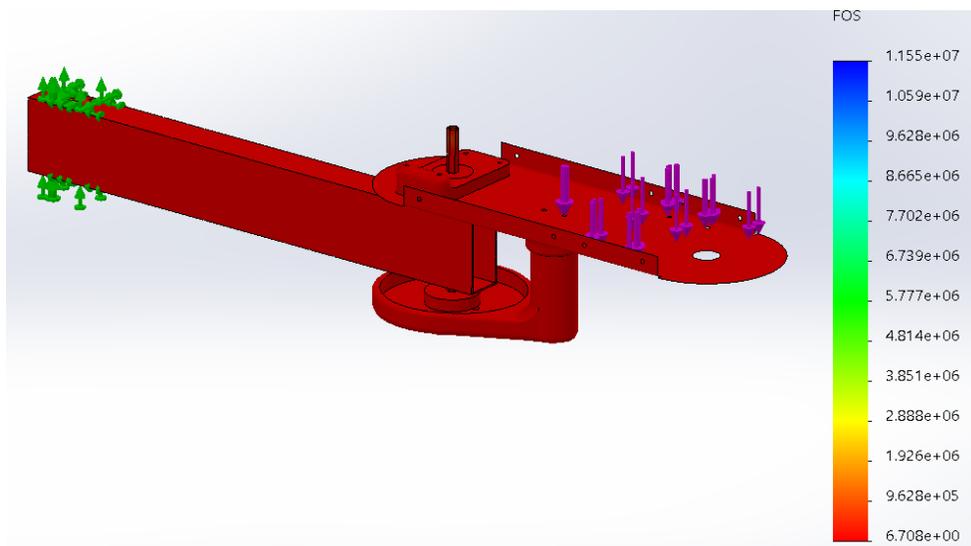


Figura 69

Simulación de desplazamientos para el mecanismo del antebrazo

**Figura 70**

Simulación de factores de seguridad para el mecanismo del antebrazo



Las simulaciones previas fueron realizadas para el mecanismo completo, pero es necesario simular de forma individual las piezas críticas. En este caso, se simula los esfuerzos

y deformaciones del acople del motor hacia el eje del antebrazo, cuyas dimensiones se muestran en la figura 71 y los resultados de las simulaciones en las figuras 72, 73 y 74.

Figura 71

Acople del motor del antebrazo

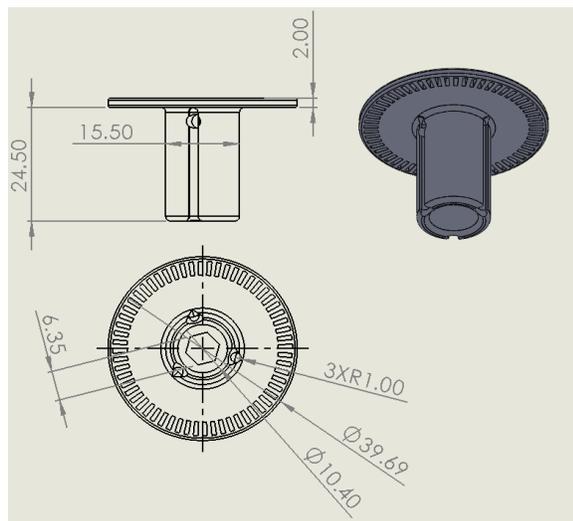


Figura 72

Simulación de esfuerzos de von Mises para el acople del motor del antebrazo

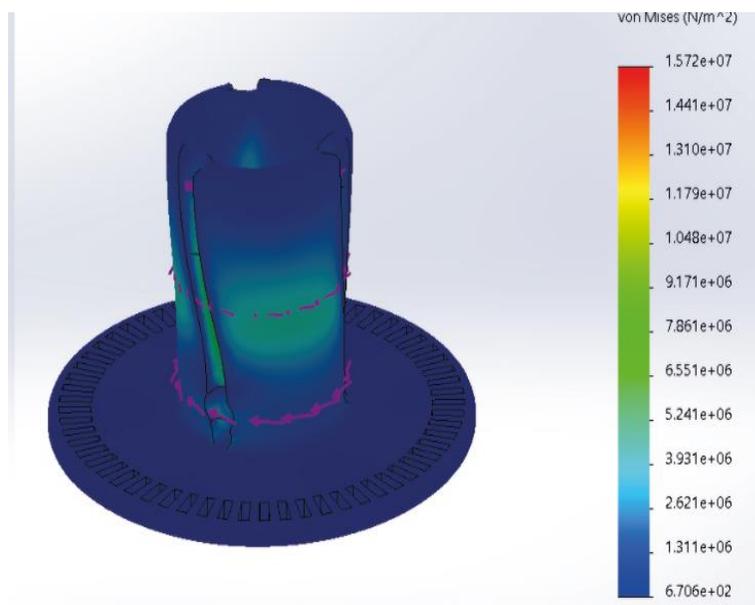
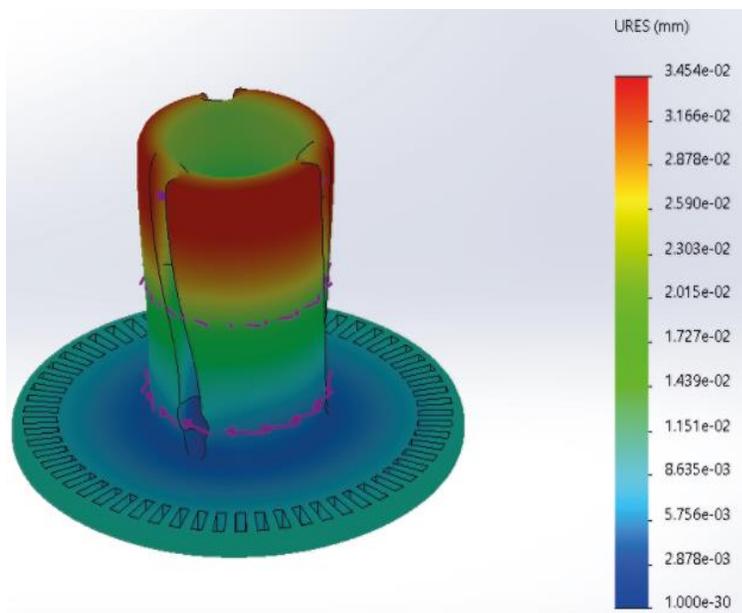
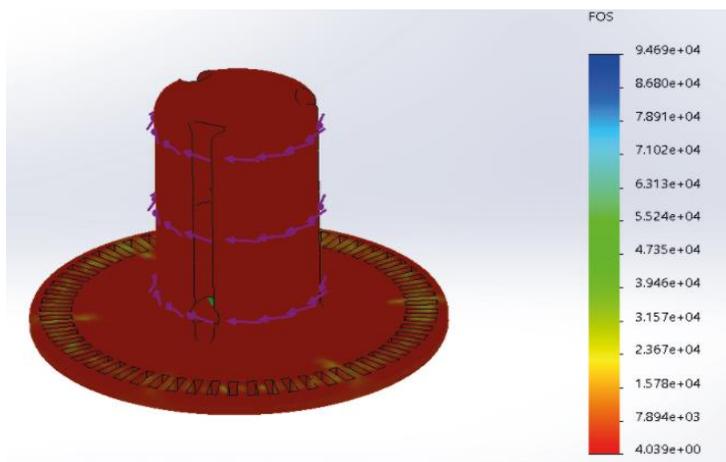


Figura 73

Simulación de desplazamientos para el acople del motor del antebrazo

**Figura 74**

Simulación de factores de seguridad para el acople del motor del antebrazo



Mecanismo elevador y rotatorio del efector final. Se modeló el mecanismo elevador y rotatorio en conjunto, como se muestra en la Figura 75 con las dimensiones que se muestran. En las figuras 76, 77 y 78 se observan los resultados de la simulación de esfuerzos, desplazamientos y factor de seguridad.

Figura 75

Mecanismo elevador y rotatorio del efector

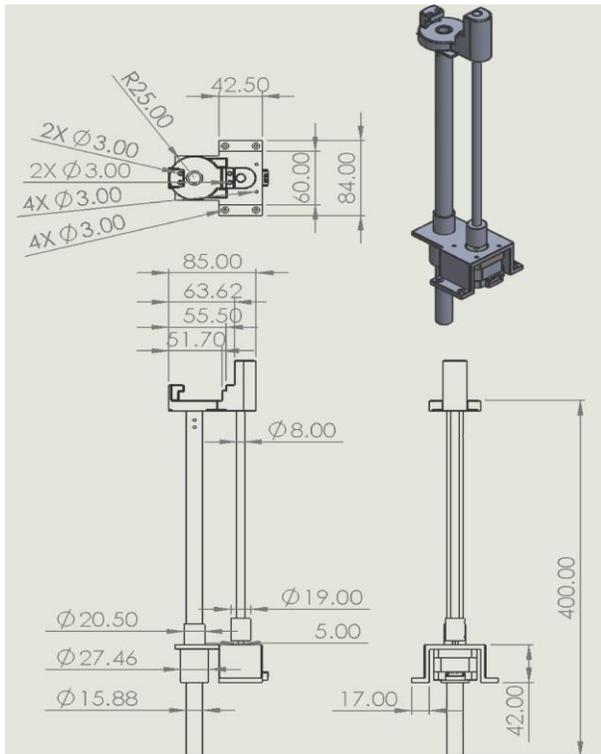
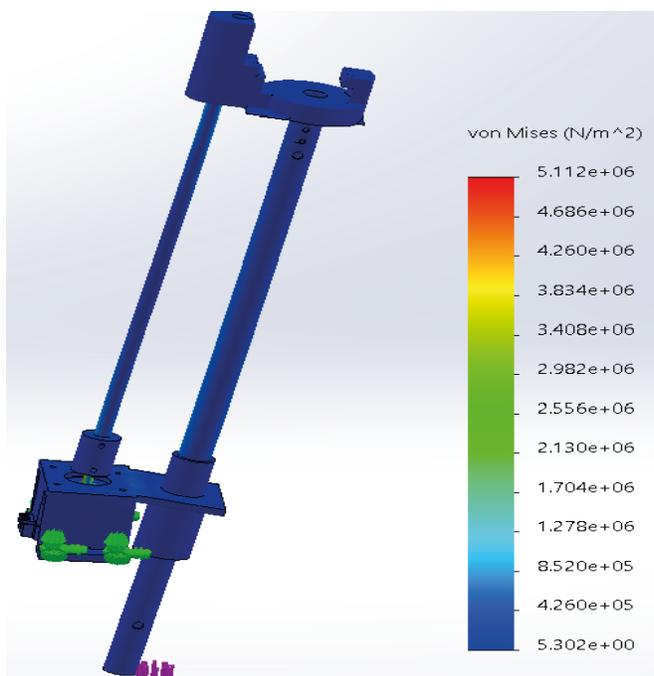


Figura 76

Simulación de esfuerzos de von Mises para el mecanismo elevador rotatorio

**Figura 77**

Simulación de desplazamientos para el mecanismo elevador rotatorio

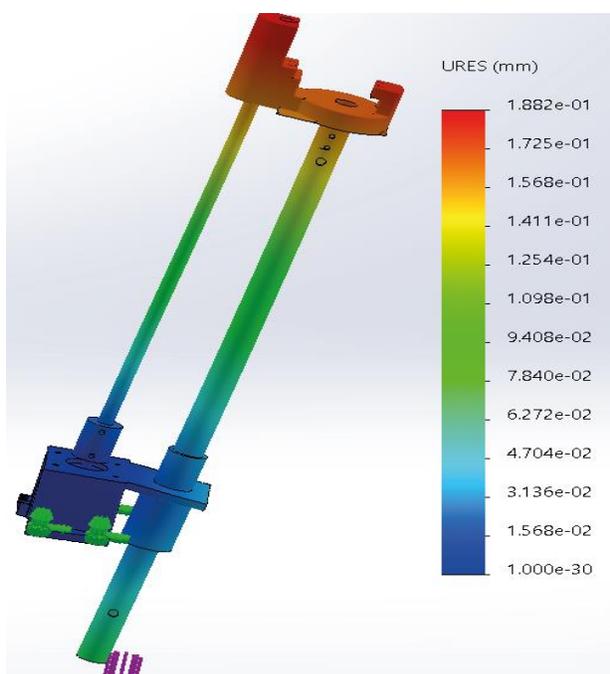
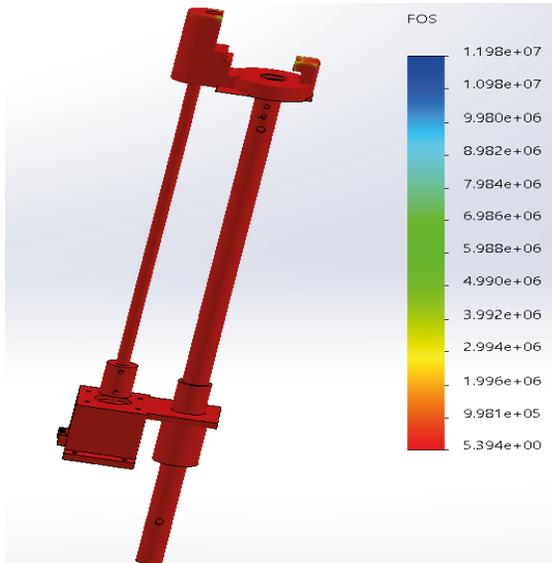


Figura 78

Simulación de factores de seguridad para el mecanismo elevador rotatorio



Mecanismo del efector final. Se modeló el mecanismo elevador y rotatorio en conjunto, como se muestra en la Figura 79 con las dimensiones que se muestran. En las figuras 80, 81 y 82 se observan los resultados de la simulación de esfuerzos, desplazamientos y factor de seguridad.

Figura 79

Mecanismo del efector final

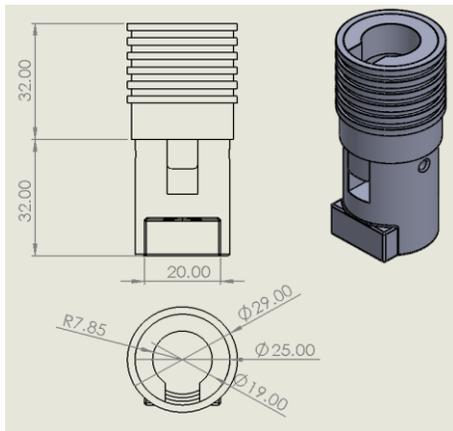
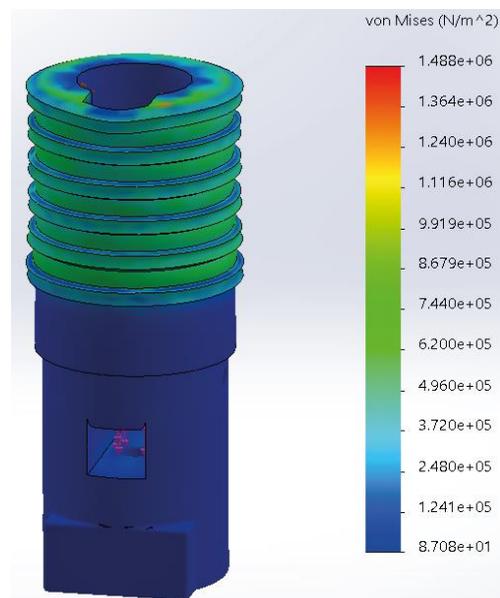


Figura 80

Simulación de esfuerzos de von Mises para el mecanismo del efector final

**Figura 81**

Simulación de desplazamientos para el mecanismo del efector final

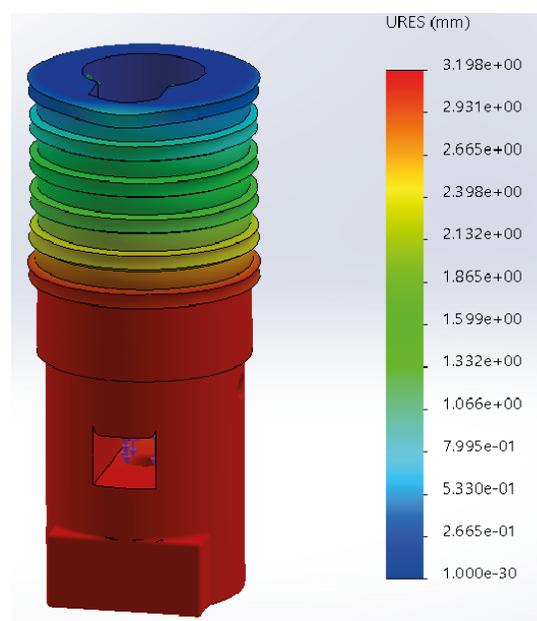
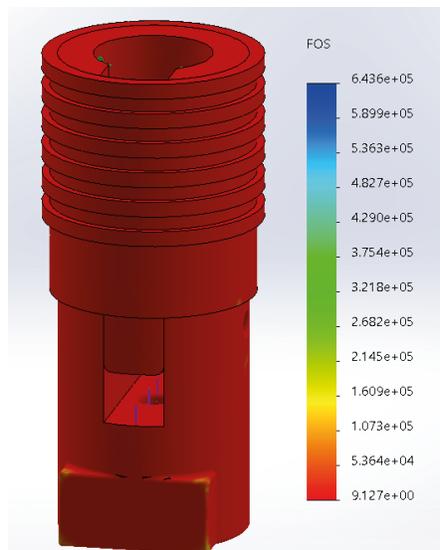


Figura 82

Simulación de factores de seguridad para el mecanismo del efector final



Validación del diseño

Para realizar la validación del diseño planteado, se toma de referencia la especificación número 70 de la Asociación de Fabricantes de Grúas de América (Crane Manufacturers Association of America [CMAA]), que especifica que la deflexión vertical máxima permitida para una viga es de $L/888$, donde L es la longitud del elemento.

Para elementos que no se comportan como vigas, la validación es realizada por medio de software, al utilizar un método adaptativo de mallado en las simulaciones de tipo h, que cuenta con 3 iteraciones que consiguen el 95% de precisión.

En la Tabla 27 se puede visualizar los mecanismos que tienen elementos que pueden comportarse como vigas, y que han sido simulados con sus cargas respectivas, y las deflexiones máximas tomadas de las gráficas previas del apartado de modelamiento de las selecciones.

Tabla 27

Resumen de deflexiones obtenidas en brazo y antebrazo

Mecanismo	Deflexión máxima obtenida [mm]	Longitud de la viga [mm]	Deflexión máxima permitida (L/888) [mm]
Brazo	0.2163	300	0.3378
Antebrazo	0.2093	200	0.2252

En ambos casos se puede apreciar que la deflexión máxima obtenida de las simulaciones es menor que la deflexión máxima permitida por el criterio tomado como referencia, por lo que se determina que el diseño es válido para las cargas que debe soportar.

Las simulaciones realizadas en los mecanismos que conforman al subsistema mecánico se validan al utilizar un tipo de mallado adaptativo de tipo h, con una precisión del 95%. Las Figuras desde la 83 hasta la 94 muestran el resultado de los mallados para cada uno de los casos, y sus gráficas de convergencia al ejecutar las simulaciones.

Figura 83

Mallado adaptativo del mecanismo del brazo

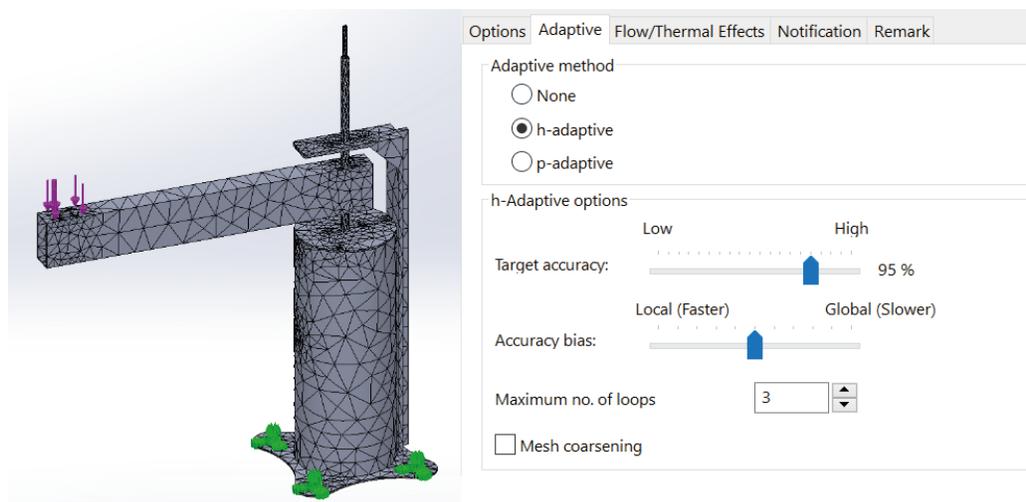


Figura 84

Curva de convergencia de la simulación para el mecanismo del brazo

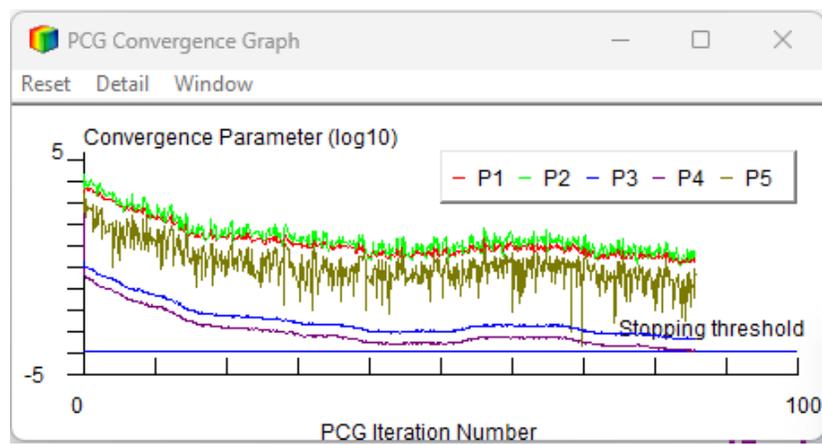


Figura 85

Mallado adaptativo del acople del motor del brazo

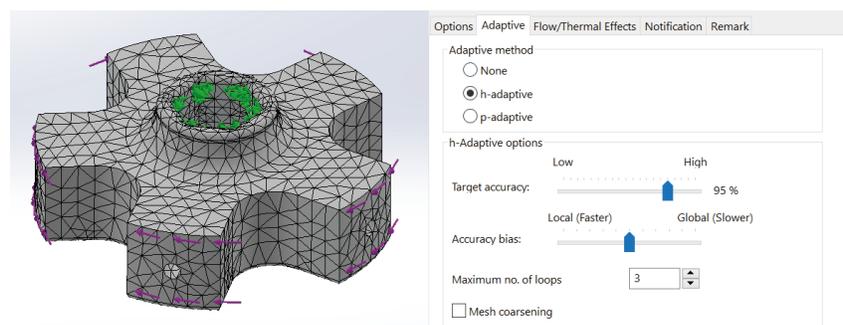


Figura 86

Curva de convergencia de la simulación para el acople del motor del brazo

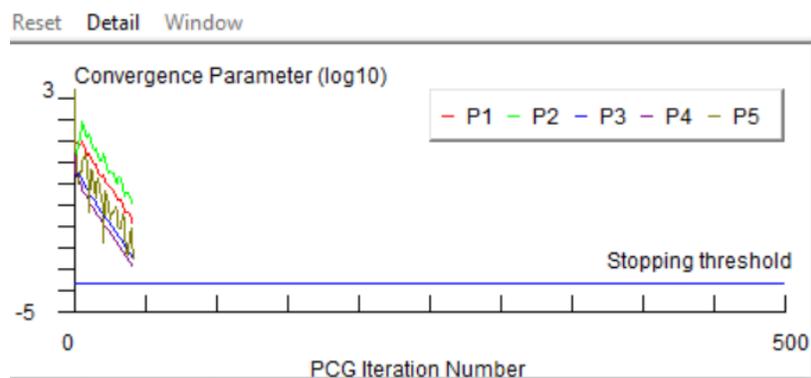


Figura 87

Mallado adaptativo del mecanismo del antebrazo

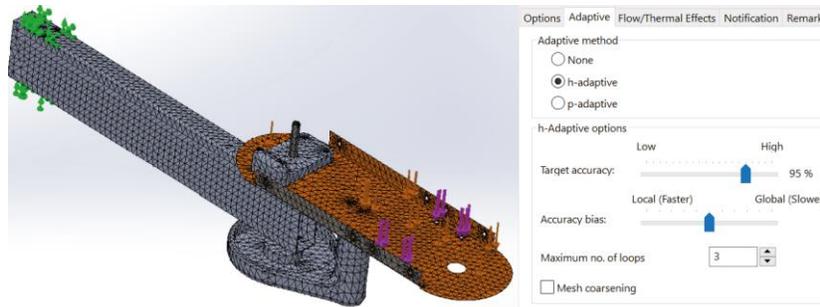


Figura 88

Curva de convergencia de la simulación para el mecanismo del antebrazo

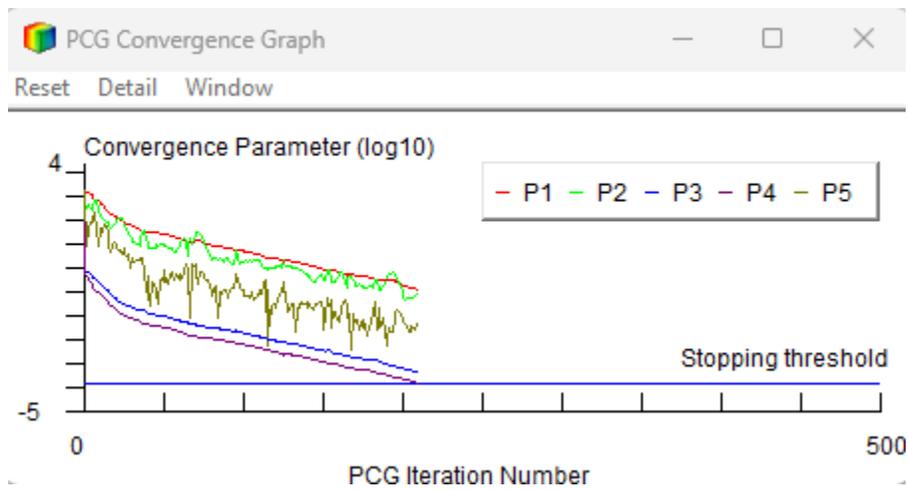


Figura 89

Mallado adaptativo del acople del motor del antebrazo

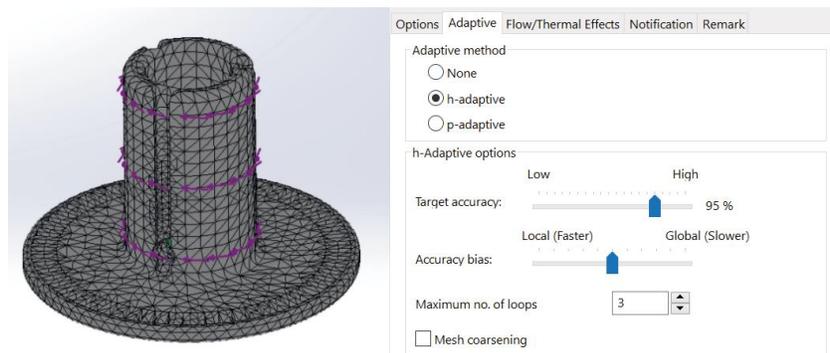
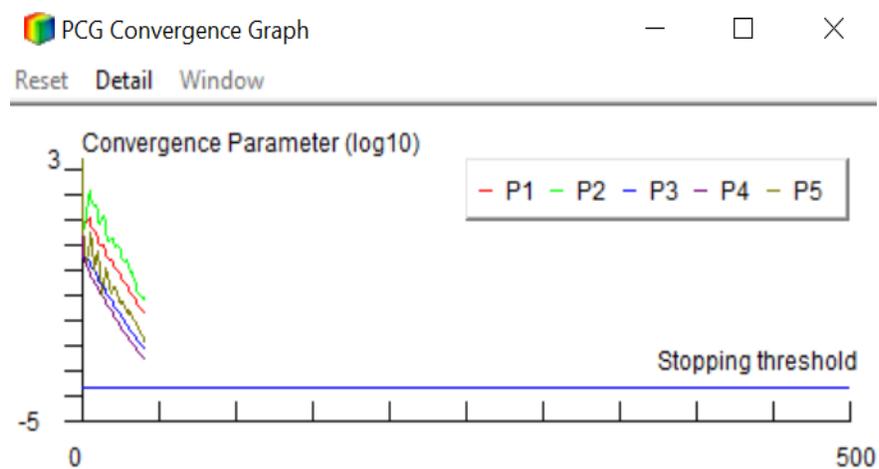


Figura 90

Curva de convergencia de la simulación para el acople del motor del antebrazo

**Figura 91**

Mallado adaptativo del mecanismo elevador rotatorio

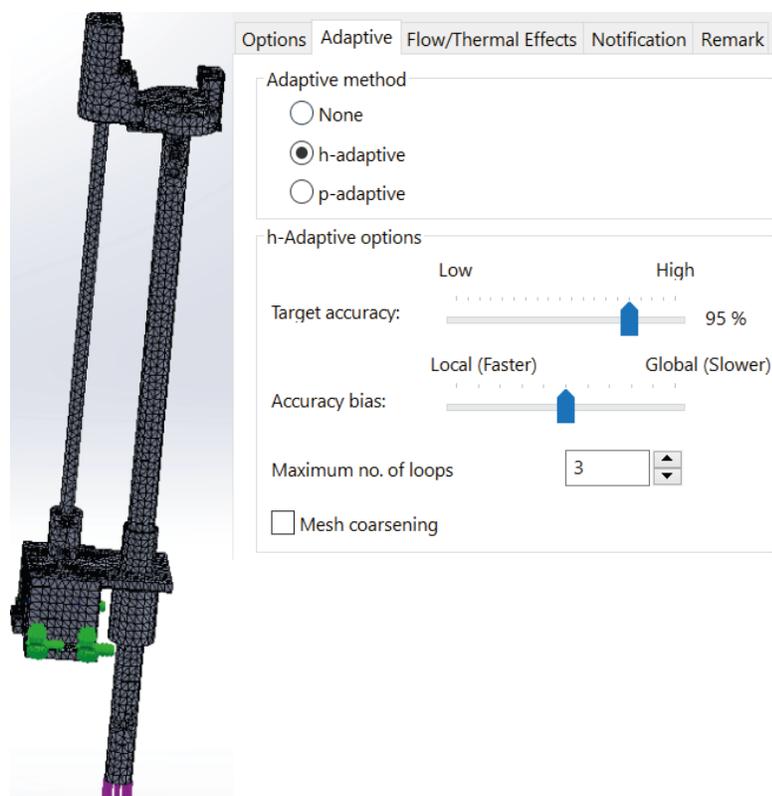


Figura 92

Curva de convergencia de la simulación para el mecanismo elevador rotatorio

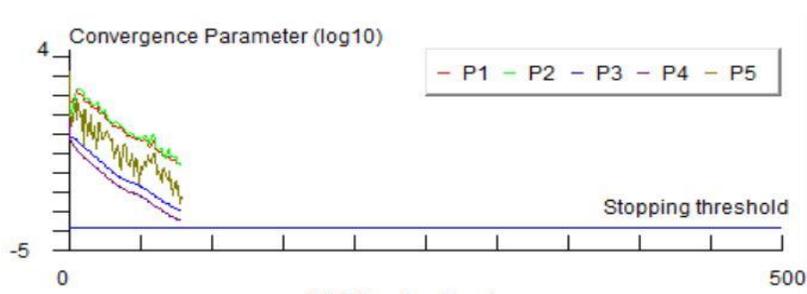


Figura 93

Mallado adaptativo del mecanismo del efector final

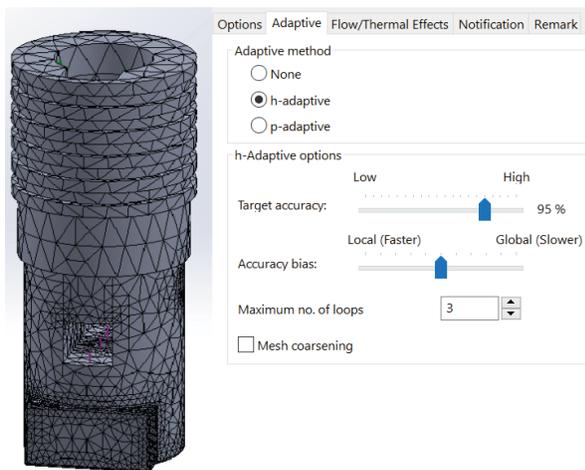
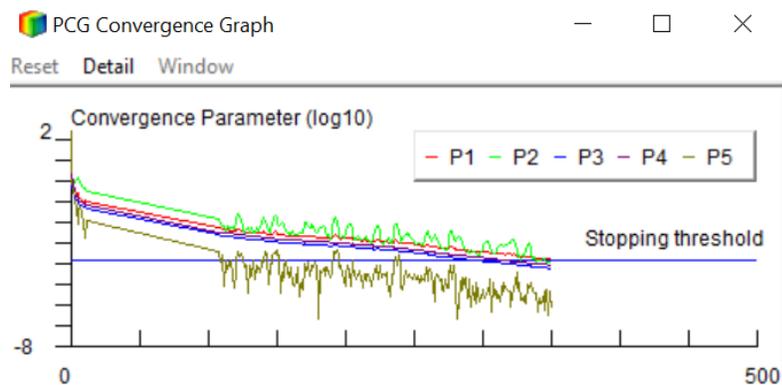


Figura 94

Curva de convergencia de la simulación para el mecanismo del efector final



Dimensionamiento de motores

En base al modelo 3D que se ha desarrollado del robot SCARA, se puede obtener las propiedades de masa del conjunto para proceder con la determinación de las características que deben poseer los motores requeridos para realizar los movimientos de cada uno de los grados de libertad.

El par necesario para realizar movimientos en los grados de libertad se calcula con la ecuación.

$$E = \frac{1}{2} I \omega^2 + \mu N r$$

Donde:

- E es el torque necesario para mover el brazo
- I es la inercia del cuerpo
- ω es la velocidad angular
- $\mu N r$ es el torque producido por la fuerza de fricción

Se calcula el momento de inercia de la masa que debe mover el motor del grado de libertad Q1 por medio del software CAD para su posterior dimensionamiento. Las propiedades de masa obtenidas se muestran en la Figura 95.

Figura 95

Propiedades de masa del mecanismo del brazo

Mass = 3441.70 grams		
Volume = 1632585.59 cubic millimeters		
Surface area = 930647.00 square millimeters		
Center of mass: (millimeters)		
X = -2.93		
Y = 980.62		
Z = 253.15		
Principal axes of inertia and principal moments of inertia: (grams * square millimeters)		
Taken at the center of mass.		
lx = (0.01, 0.81, 0.59)	Px = 26011054.76	
ly = (0.01, -0.59, 0.81)	Py = 53343875.39	
lz = (1.00, 0.00, -0.01)	Pz = 74164246.86	
Moments of inertia: (grams * square millimeters)		
Taken at the center of mass and aligned with the output coordinate system.		
Lxx = 74159383.04	Lxy = 119810.86	Lxz = 381265.80
Lyx = 119810.86	Lyy = 35445048.29	Lyz = 12994030.51
Lzx = 381265.80	Lzy = 12994030.51	Lzz = 43914745.68
Moments of inertia: (grams * square millimeters)		
Taken at the output coordinate system.		
lxx = 3604300771.02	lxy = -9781559.02	lxz = -2174793.47
lyx = -9781559.02	lyy = 256033520.04	lyz = 867369660.52
lzx = -2174793.47	lzy = 867369660.52	lzz = 3353526906.33

Como parámetro de diseño se establece que el brazo del robot debe girar a 20 RPM, con este dato se calcula el par que se requiere para mover el eslabón.

$$E = \frac{1}{2}I\omega^2 + \mu Nr = 0.5(335352690.33 \text{ g} * \text{mm})(20 \text{ RPM})^2 + (0.2)(33.72 \text{ N})(0.05\text{m})$$

$$E = 0.5(0.335 \text{ kg} * \text{m})(2.094 \text{ rad/s})^2 + (0.2)(33.72 \text{ N})(0.05\text{m}) = 1.06\text{Nm}$$

Se requiere de un torque de 1.06 Nm para que el brazo pueda desplazarse; si se considera que el motor DC supera las 10 RPM que se desea, se necesita una caja reductora. La potencia del motor requerido es

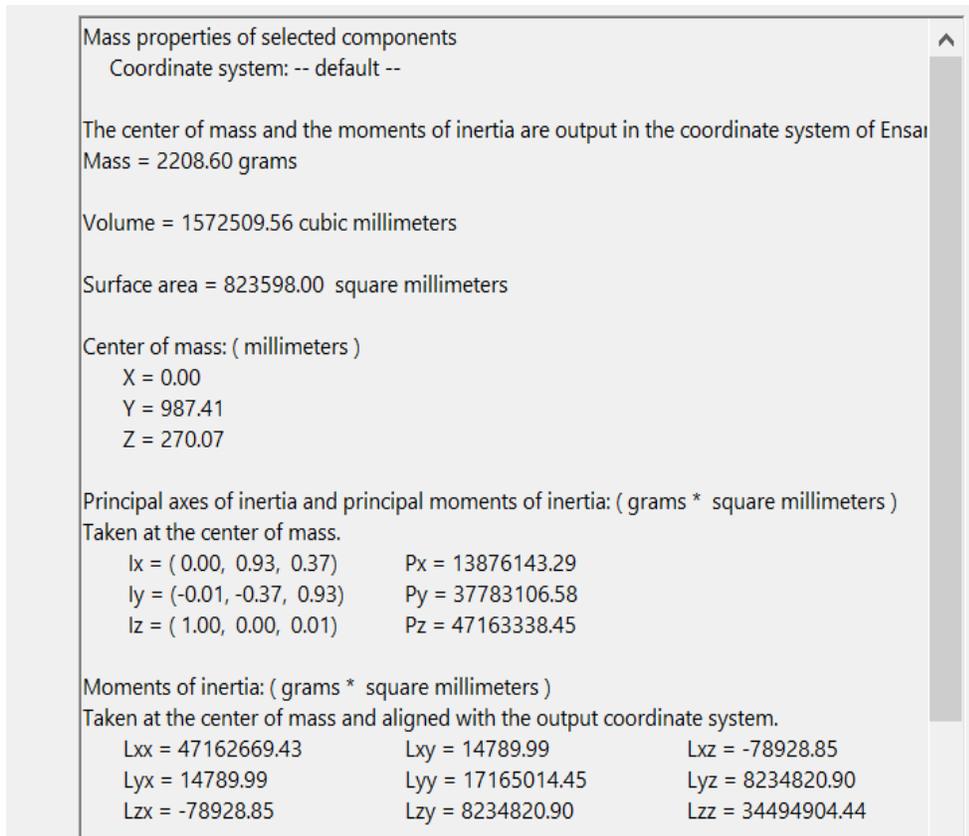
$$P = T * \omega = 1.06\text{Nm} * 20 \text{ RPM} = 21.2\text{W}$$

Tabla 28*Características del motor seleccionado*

Marca	Akozon
Modelo	9NUGQVH3MD-02
Potencia	48W
Voltaje Nominal	12V
Rango de voltaje	3 - 12V
Par de salida	8Nm

Figura 96*Motor seleccionado para el brazo*

Se calcula el momento de inercia de la masa que debe mover el motor del grado de libertad Q2 por medio del software CAD para su posterior dimensionamiento. Las propiedades de masa de esta estructura se observan en la Figura 97.

Figura 97*Propiedades de masa para la estructura del antebrazo*

Como parámetro de diseño se establece que el antebrazo del robot debe girar a 20 RPM, con este dato se calcula el par que se requiere para mover el eslabón.

$$E = \frac{1}{2}I\omega^2 + \mu Nr = 0.5(34494904.44 \text{ g} * \text{mm})(20 \text{ RPM})^2 + (0.2)(33.72 \text{ N})(0.05\text{m})$$

$$E = 0.5(0.34 \text{ kg} * \text{m})(2.094 \text{ rad/s})^2 + (0.2)(33.72 \text{ N})(0.05\text{m}) = 0.76\text{Nm}$$

Se requiere de un torque de 1.06 Nm para que el brazo pueda desplazarse; si se considera que el motor DC supera las 10 RPM que se desea, se necesita una caja reductora.

La potencia del motor requerido es

$$P = T * \omega = 0.76\text{Nm} * 20 \text{ RPM} = 15.2\text{W}$$

Tabla 29*Características del motor seleccionado*

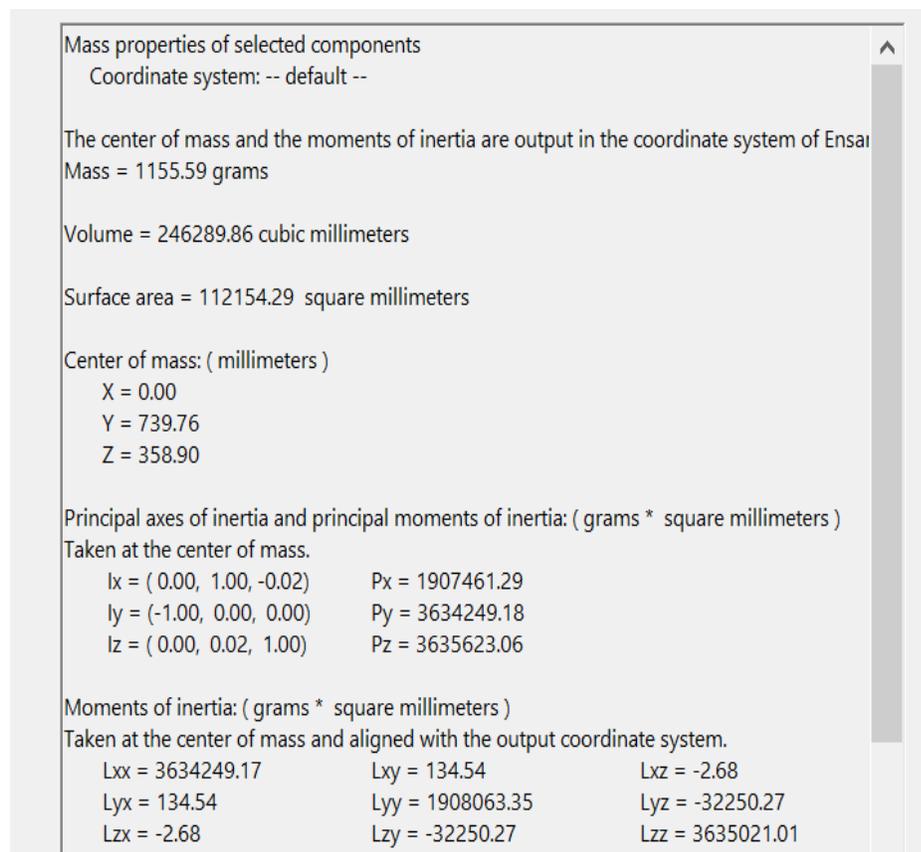
Marca	Xerox
Modelo	127K53630
Potencia	24W
Voltaje Nominal	12V
Rango de voltaje	3 - 12V
Par de salida	10Nm

Figura 98*Motor seleccionado para el antebrazo*

Se obtiene la masa total de los cuerpos que debe elevar el motor del grado de libertad Q3 por medio del software CAD, y se muestran los resultados en la Figura 99.

Figura 99

Propiedades de masa de la estructura del mecanismo rotatorio



La velocidad nominal de los servomotores oscila entre 0.16 y 0.17 segundos por cada 60 grados. Considerando esta especificación, se calcula el torque que debe poseer el servomotor.

$$E = \frac{1}{2} I \omega^2 = 0.5(3635021.01 \text{ g} * \text{mm})(59 \text{ RPM})^2$$

$$E = 0.5(0.0036 \text{ kg} * \text{m})(6.18 \text{ rad/s})^2 = 0.06 \text{ Nm}$$

En base al valor obtenido, se dimensiona el motor requerido, y se muestra su detalle en la Tabla 30.

Tabla 30

Características del motor seleccionado para el mecanismo rotatorio

Marca	TowerPro
Modelo	MG996R
Corriente	1.2A
Ángulo por paso	1.8°
Par de salida	0.7Nm

Figura 100

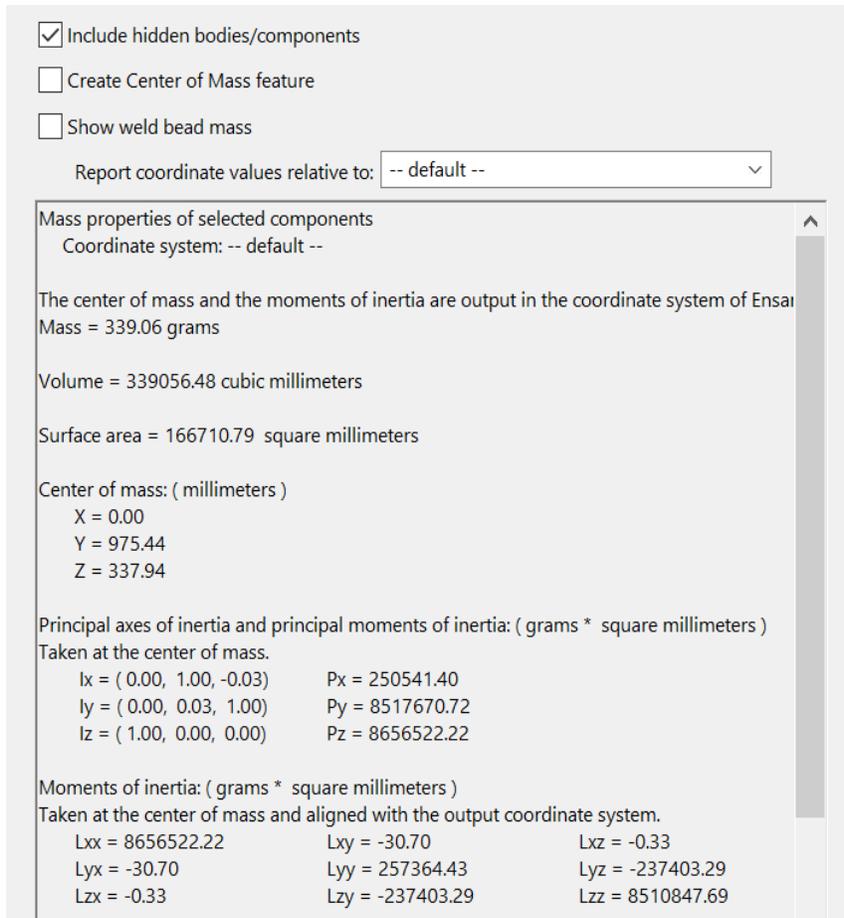
Motor seleccionado para el mecanismo rotatorio



Se calcula la masa total de los cuerpos que debe elevar el motor del grado de libertad Q4 por medio del software CAD, y se obtienen los resultados mostrados en la Figura 101.

Figura 101

Propiedades de masa del mecanismo rotatorio



El mecanismo que se utiliza para elevar y descender estas masas es con un tornillo sinfín, por lo que la ecuación que determina la fuerza axial del eje sinfín es

$$F_a = \frac{T}{r_p} * \frac{1}{\tan(\lambda)}$$

Donde:

- F_a : la fuerza axial del tornillo sinfín
- T : el torque de entrada del tornillo sinfín
- r_p : radio primitivo del eje sinfín
- λ : ángulo de la hélice

La carga que se debe elevar es el peso de la estructura del mecanismo, adicional el peso del objeto sujeto por el efector final.

$$F_a \approx 15N$$

De la ecuación se obtiene que el torque necesario para elevar la carga es de

$$T \approx 0.2Nm$$

En base al valor obtenido, se dimensiona el motor requerido, y se escoge el motor a pasos Nema 17.

Tabla 31

Características del motor seleccionado para el mecanismo elevador

Marca	Minebea
Modelo	17PM-K301-11V
Tipo	Monopolar
Corriente nominal	1.2A
Ángulo por paso	1.8°
Par de salida	0.25Nm

Figura 102

Motor seleccionado para el mecanismo elevador



Diseño del circuito electrónico

Dentro de la celda de manufactura remota, el subsistema electrónico comprende el cableado del robot SCARA, que es el encargado de la activación de actuadores en base a las instrucciones recibidas, y la lectura de los sensores incluidos. Para el diseño de este subsistema se tomaron en cuenta las siguientes consideraciones

- Sensores requeridos para el correcto funcionamiento
- Controlador y módulos incluidos pueden ser removibles
- Dimensionamiento del ancho de pistas de placas impresas (PCB) adecuado
- Costo de los componentes
- Disponibilidad de los materiales dentro del mercado nacional

Esta estructura del robot SCARA puede desglosarse en cuatro partes, las cuales manejan las conexiones eléctricas del robot.

- Placa principal (Alojada en caja de control)
- Placa secundaria (Alojada en el antebrazo)
- Placas auxiliares
- Módulos auxiliares

Selección de sensores

Para el diseño de los circuitos, se requiere determinar qué sensores serán implementados dentro del robot SCARA de acuerdo a las variables a controlar.

- Posición angular de brazo y antebrazo
- Velocidad angular de brazo y antebrazo
- Altura de efector final

La posición y velocidad angular pueden ser medidas en conjunto por el mismo sensor, por lo que los criterios de selección se aplican a un solo tipo de sensor. Para la lectura de estas variables se proponen dos alternativas.

Alternativa 1: Sensor tipo Encoder Incremental

Se plantea utilizar sensores de tipo encoder incremental para la medición de posición y velocidad angular, como el que se muestra en la Figura 103, puesto que por su construcción mecánica permiten la determinación del sentido de giro que realizan los eslabones. El modelo seleccionado para esta alternativa es el encoder KY-040, que presenta las siguientes características:

- Voltaje de operación: 5V
- Resolución: 18°/Paso (20 pasos)
- Salida: 2 bits
- Ángulo: 360° continuos
- Compatible con microcontroladores/Raspberry Pi

Figura 103

Sensor tipo encoder incremental



Tabla 32

Ventajas y desventajas de la alternativa 1 del sensor de posición angular

Ventajas	Desventajas
Bajo costo	Requiere de una estructura o de juego de poleas o engranajes para su acople

Ventajas	Desventajas
Detección de sentido de giro	Sensible a vibraciones
	Resolución baja

Alternativa 2: Sensor tipo Encoder óptico

Se plantea utilizar sensores de tipo encoder incremental para la medición de posición y velocidad angular como el indicado en la Figura 104, puesto que por su construcción mecánica permiten la determinación del sentido de giro que realizan los eslabones. El modelo seleccionado para esta alternativa es el módulo FC-03 que cuenta con el opto interruptor MOCH 22A, que presenta las siguientes características:

- Voltaje de operación: 3.3 - 5V
- Ángulo de visión: Dependiente del número de muescas del disco utilizado
- Salida: Analógica y digital TTL
- Ángulo: 360° continuos
- Compatible con microcontroladores/Raspberry Pi

Figura 104

Sensor tipo encoder óptico

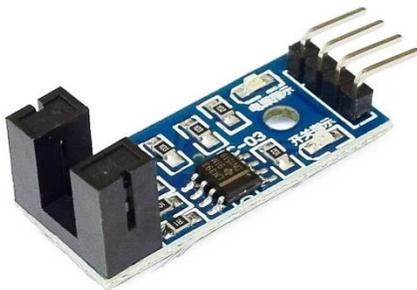


Tabla 33

Ventajas y desventajas de la alternativa 2 del sensor de posición angular

Ventajas	Desventajas
Bajo costo	No permite la detección de sentido de giro
Acoplamiento sencillo	
Resolución modificable dentro del rango de detección de luz del optointerruptor	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para la lectura de posición y velocidad angular.

Tabla 34

Matriz de evaluación para las alternativas de sensores para medición de posición angular

Criterio	Ponderación	Encoder incremental		Encoder óptico	
		CA	EP	CA	EP
		Costo	20%	4	0.8
Confiabilidad	40%	3	1.2	4	1.6
Facilidad de ensamble	40%	3	1.2	5	2
Total			3.2		4.6

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la segunda alternativa, que corresponde al uso de encoder óptico, puesto que es el que ofrece mejor desempeño en base a los criterios establecidos.

Para la lectura de la altura en la que se encuentre el efector final se plantean dos alternativas.

Alternativa 1: Sensor ultrasónico

Se plantea utilizar un sensor ultrasónico que permita la lectura de la altura del mecanismo del efector final, midiendo la altura a la que se encuentra el mecanismo del giro del efector final y realizando las compensaciones adecuadas. Para esta alternativa se selecciona el sensor ultrasónico HC-SR04, que se muestra en la Figura 105, que presenta las siguientes características.

- Voltaje de operación: 5V
- Máximo rango de detección: 4m
- Mínimo rango de detección: 2cm
- Ángulo de medición: 15°

Figura 105

Sensor ultrasónico HC-SR04



Tabla 35

Ventajas y desventajas de la alternativa 1 del sensor de distancia

Ventajas	Desventajas
Bajo costo	Velocidad de respuesta baja
Acoplamiento sencillo	
Amplio rango de detección	

Alternativa 2: Sensor infrarrojo

Se plantea utilizar un sensor infrarrojo que permita la lectura de la altura del mecanismo del efector final, midiendo la altura a la que se encuentra el mecanismo del giro del efector final y realizando las compensaciones adecuadas. Para esta alternativa se selecciona el sensor SHARP 0A41SKF93, mostrado en la Figura 106, que presenta las siguientes características.

- Voltaje de operación: 5V
- Máximo rango de detección: 30cm
- Mínimo rango de detección: 3cm

Figura 106

Sensor infrarrojo tipo SHARP



Tabla 36

Ventajas y desventajas de la alternativa 2 del sensor de distancia

Ventajas	Desventajas
Acoplamiento sencillo	Bajo rango de detección
Velocidad de respuesta alta	Costo mayor al ultrasónico

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para la lectura de altura.

Tabla 37

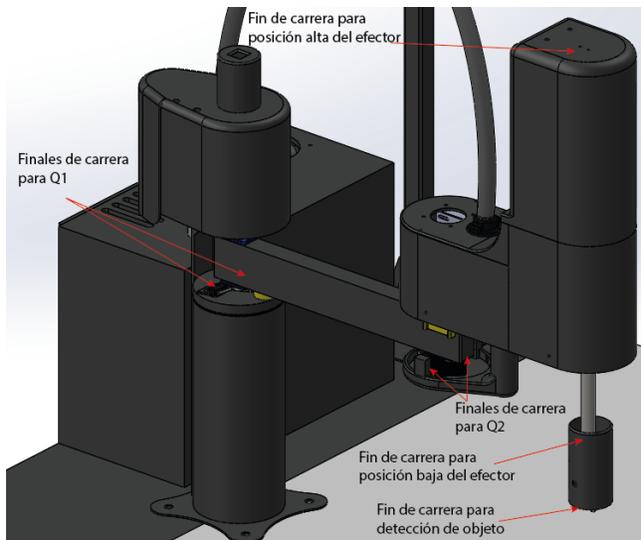
Matriz de evaluación para las alternativas de sensores para medición de distancia

Criterio	Ponderación	Sensor ultrasónico		Sensor infrarrojo	
		CA	EP	CA	EP
		Costo	20%	5	1
Confiabilidad	40%	4	1.6	4	1.6
Facilidad de ensamble	40%	5	2	5	2
Total		4.6		4.2	

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la primera alternativa, que corresponde al uso de sensor ultrasónico, puesto que es el que ofrece mejor desempeño en base a los criterios establecidos.

Adicional a estos sensores escogidos, se planea implementar interruptores finales de carrera para la determinación de las posiciones máxima y mínima que pueden tomar los eslabones del robot SCARA. En la Figura 107 se detallan los interruptores necesarios para la determinación de estas posiciones máximas.

Figura 107*Interruptores finales de carrera del robot SCARA*

Los finales de carrera para los grados de libertad Q1 y Q2 se encuentran para evitar que el brazo robótico se exceda de su rango de trabajo, que es de 0° a 180°. El final de carrera de la posición baja y alta evitan que el efector final se exceda en su rango de trabajo, que es de 0 a 24 cm, finalmente el final de carrera de detección de objetos cumple la función de detectar si un objeto está siendo sujetado o no cuando el electroimán se encuentre encendido.

Selección de microcontrolador y microcomputador

Las lecturas de los sensores y la activación de actuadores será realizada por un microcontrolador que procese las instrucciones recibidas del usuario y permita al usuario conocer el estado del robot SCARA. Para cubrir esta necesidad se plantean dos alternativas.

Alternativa 1: Placa ESP-WROOM-32

Se plantea utilizar una o varias placas ESP-WROOM-32, como la que se muestra en la Figura 108, conectadas entre sí de tal forma que permitan la lectura de los sensores del robot SCARA, y permitir la activación de los actuadores para posicionar el robot. Se considera la opción de utilizar varias placas puesto que el número de pines que pueden ser utilizados como entradas y/o salidas digitales es de 23.

Figura 108*Placa ESP-WROOM-32***Tabla 38***Ventajas y desventajas de la alternativa 1 para microcontrolador*

Ventajas	Desventajas
Alta capacidad de procesamiento	Número de entradas y salidas limitado
Módulo WiFi por defecto	Voltaje de entradas y salidas de 3.3V
	Costo mayor al requerir más de una

Alternativa 2: Placa Arduino MEGA 2560

Se plantea utilizar una placa Arduino MEGA 2560, como la mostrada en la Figura 109, que permita la lectura de los sensores del robot SCARA, y permitir la activación de los actuadores para posicionar el robot. En esta alternativa se contempla la opción de utilizar únicamente una placa, puesto que el número de pines que pueden ser utilizados de entradas y salidas digitales es de 54.

Figura 109

Placa Arduino MEGA 2560

**Tabla 39**

Ventajas y desventajas de la alternativa 2 para microcontrolador

Ventajas	Desventajas
Número de pines utilizables como I/O digitales mayor.	Capacidad de procesamiento menor a la de la alternativa 1
Entradas y salidas de nivel lógico (5V)	
Costo bajo	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta una matriz de evaluación para determinar la mejor opción para la lectura de altura.

Tabla 40

Matriz de evaluación para las alternativas de placa microcontrolador

Criterio	Ponderación	ESP-WROOM-32		Arduino MEGA 2560	
		CA	EP	CA	EP
Costo	20%	3	0.6	4	0.8
Capacidad de procesamiento	40%	5	2	4	1.6

Criterio	Ponderación	ESP-WROOM-32		Arduino MEGA 2560	
		CA	EP	CA	EP
Nivel de voltaje de entradas/salidas	40%	3	1.2	5	2
Total			3.8		4.4

Nota. CA: Calificación, EP: Evaluación ponderada.

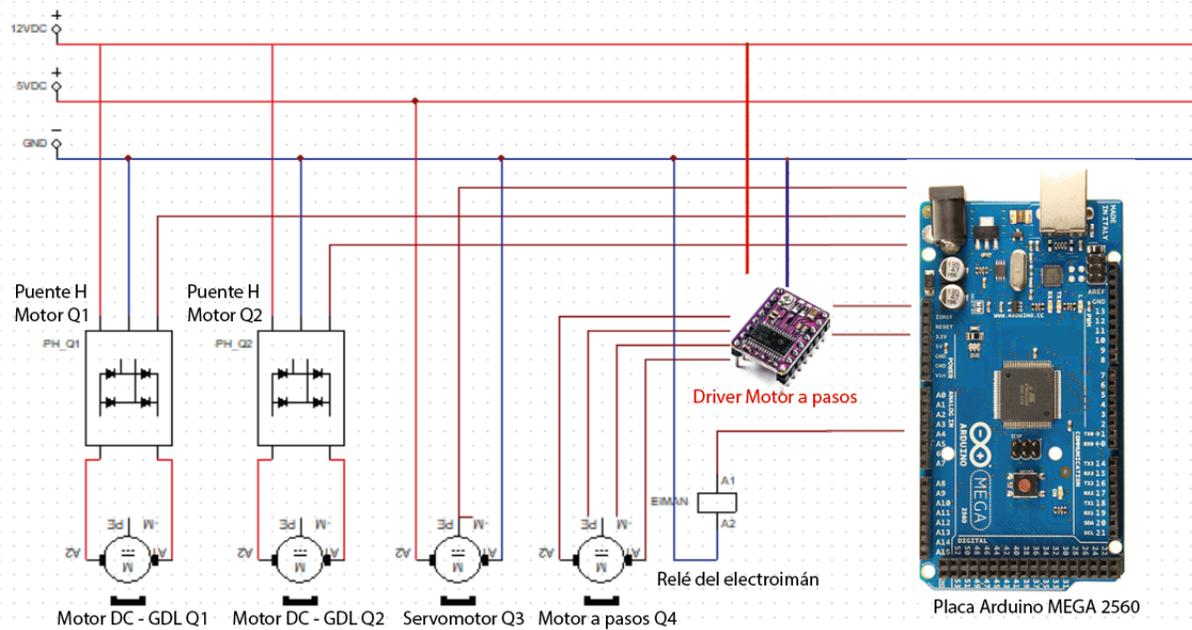
En base a la tabla de evaluación, se determinó que la mejor opción es la segunda alternativa, que corresponde al uso de una placa Arduino MEGA 2560 en base a los criterios de selección. Se considera al nivel de voltaje como un factor de evaluación puesto que elementos de control como drivers de motores, relés y adicionalmente sensores trabajan con un nivel de voltaje de 5V, haciendo que trabajar con una placa que maneje un nivel de voltaje de 3.3 V pueda requerir acondicionamiento de señales.

Circuito electrónico

La Figura 110 es la representación gráfica de las conexiones de los actuadores del robot SCARA y la placa Arduino MEGA 2560. Existe una fuente de voltaje de 12V que alimenta a los motores DC de los grados de libertad Q1, Q2 y al driver del motor a pasos Q4. Para el servomotor y el relé que activa el electroimán llega una alimentación de 5V. La placa Arduino envía las señales a los puentes H de los motores Q1 y Q2 con información de sentido de giro y valor PWM, al servomotor el grado que debe rotar, y al driver del motor a pasos envía un tren de pulsos para que el motor gire en sentido horario o antihorario.

Figura 110

Diagrama de conexiones eléctricas



Tomando en cuenta las corrientes nominales de los motores seleccionados en el apartado de Diseño del subsistema mecánico, se calculan los anchos de pista requeridos para cada uno de las conexiones en base a la norma IPC-2221, que especifica la siguiente ecuación para la determinación del ancho

$$A = \frac{I}{1.378(K_1 \Delta T^{K_2})^{1/K_3} L}$$

Donde:

- A: Ancho de pista
- I: Corriente máxima
- K1, K2, K3: Constantes de capa interna y externa
- L: Grosor de pista
- ΔT : Variación de temperatura

Considerando las corrientes nominales de los motores elegidos, se calcula el ancho de pista requerido para los conectores de cada uno.

Tabla 41

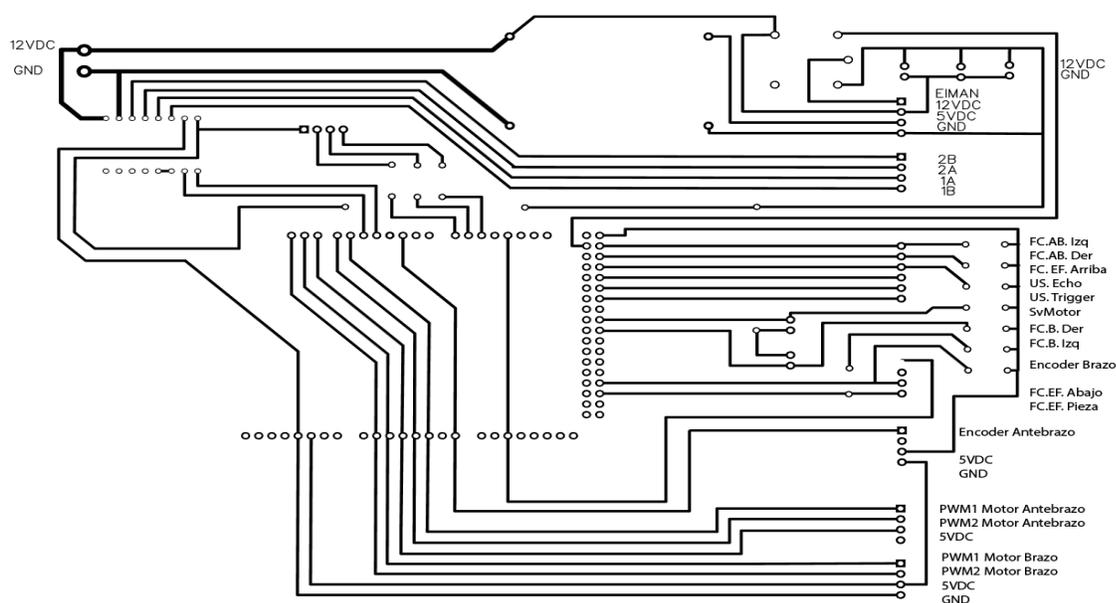
Anchos de pista requeridos en base a la corriente nominal obtenida de hojas de datos

Motor	Corriente [A]	Ancho de pista [mils]	Ancho de pista [mm]
Motor Q1	5	53.51	1.35
Motor Q2	2	13.72	0.35
Servomotor	1.2	6.42	0.16
Motor a pasos	1.2	6.42	0.16

En base a los anchos de pista calculados y las conexiones que requieren los actuadores como se mostró en la tabla 41, y adicionalmente las conexiones de los sensores que requiere el motor, se diseña la placa PCB a ser implementada, que se muestra en la Figura 111.

Figura 111

Diseño de la placa de circuitos



Diseño del módulo de visión artificial

El sistema de visión artificial permitirá al robot interactuar con objetos de su entorno de forma automática, sin embargo, su desempeño se ve afectado por el poder de procesamiento del hardware donde se esté ejecutando, por lo que se plantean dos alternativas de obtención de imagen.

Alternativa 1: Obtención de imágenes en tiempo real

Se plantea una obtención de imágenes que sea procesada a un ritmo de aproximadamente 30 imágenes por segundo, es decir, un sistema de visión artificial por video.

En la Figura 112 se muestra la alternativa de forma esquemática.

Figura 112

Esquema de la alternativa 1 para el sistema de visión artificial

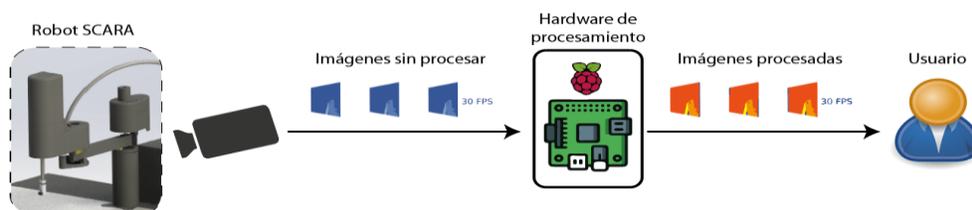


Tabla 42

Ventajas y desventajas de la alternativa 1 para el sistema de visión artificial

Ventajas	Desventajas
Detección de objetos en tiempo real	Reducida resolución de imagen
	Altos niveles de ruido en cada imagen
	Requiere de un elevado poder de procesamiento
	Imágenes que no aportan información útil

Alternativa 2: Obtención de una sola imagen

Se plantea obtener solo una imagen a la cual se realizan las operaciones para obtener los resultados del algoritmo. En la Figura 113 se muestra la alternativa de forma esquemática.

Figura 113

Esquema de la alternativa 1 para el sistema de visión artificial

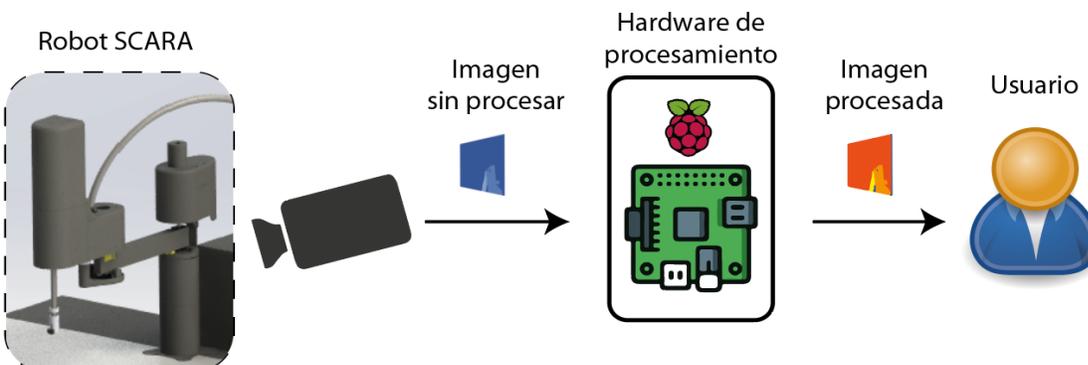


Tabla 43

Ventajas y desventajas de la alternativa 1 para el sistema de visión artificial

Ventajas	Desventajas
Alta resolución de imagen	No hay retroalimentación en tiempo real de la detección
Imagen con poco ruido	
Requiere un bajo poder de procesamiento	
La imagen muestra toda la información necesaria	

Una vez determinadas las ventajas y desventajas de las alternativas para la obtención de imágenes para el módulo de visión artificial, se procede con una matriz de evaluación para determinar a partir de criterios cuál es la mejor opción a ser implementada.

Tabla 44

Matriz de evaluación para las alternativas del sistema de visión artificial

Criterio	Ponderación	Obtención de imágenes en tiempo real		Obtención de una sola imagen	
		CA	EP	CA	EP
Calidad	20%	3	0.6	4	0.8
Poder de procesamiento	50%	2	1.0	4	2.0
Tiempo de respuesta	30%	3	0.9	4	1.2
Total			2.5		4.0

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se obtuvo que realizar la obtención de una sola imagen por vez es la más conveniente de implementar para el robot, principalmente por su bajo requerimiento del poder de procesamiento.

Adicionalmente se considera la implementación de un sistema de iluminación propio, dado a que el desempeño y calidad de imagen mejora considerablemente al tener un buen nivel de iluminación, por lo que se pretende colocar una lámpara LED que ofrezca un ambiente con una temperatura de color de aproximadamente 3500K.

Debido a que pueden existir elementos que se encuentren fuera del alcance del área de interés (espacio de trabajo), se diseña un algoritmo que soluciona dos problemas dentro de la detección de objetos para el robot, que son:

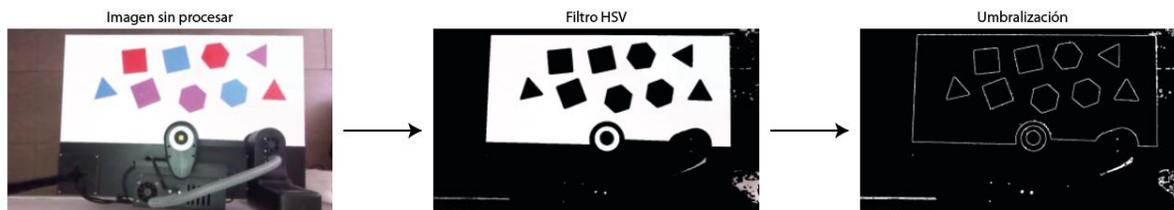
- Elimina objetos que pueden estar presentes en la imagen pero que el robot no pueda alcanzar por limitaciones físicas, es decir, limitar al espacio de trabajo la detección de objetos.

- Corrige las distorsiones inherentes a la cámara, haciendo una transformación que permite normalizar la estimación de la distancia del centro del robot a los objetos.

Para realizar este preprocesamiento de la imagen se utiliza un filtro HSV, limitando considerablemente los valores y la saturación, permitiendo separar fácilmente el espacio de trabajo de otros elementos que tengan color. Una vez filtrada la imagen se realiza una umbralización y detección de bordes de la imagen usando el método de Canny. La Figura 114 ejemplifica el proceso que realiza el algoritmo para umbralizar una imagen.

Figura 114

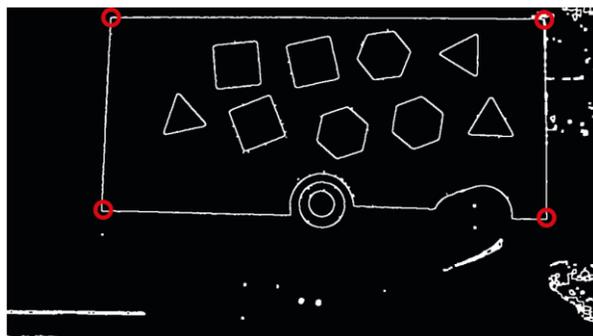
Proceso de umbralización de las imágenes



Posteriormente se procede, a determinar los puntos que conforman las esquinas del espacio de trabajo, como se muestra en la Figura 115, con la aclaración que se utilizan umbrales de longitud de arco altos para detectar cambios en ángulos grandes y descartar el cruce del robot en el área de interés. ordenando dichos puntos en base a su posición.

Figura 115

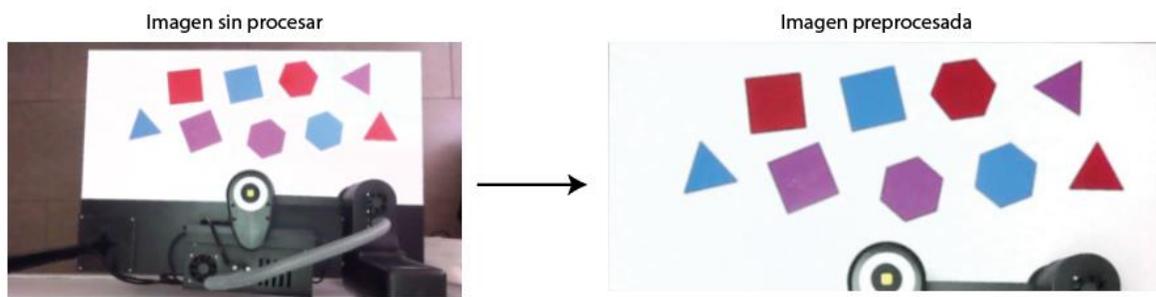
Detección de vértices del espacio de trabajo



El último paso del preprocesamiento de la imagen es realizar una transformación de perspectiva usando los puntos encontrados anteriormente y una medida con la misma proporción en píxeles que en unidades de longitud del espacio de trabajo. El resultado es una imagen más fácil de procesar y con más información útil para el algoritmo. La Figura 116 muestra el antes y después de aplicar la corrección de perspectiva a la imagen.

Figura 116

Corrección de perspectiva de la imagen



Detección de color y forma

El sistema integrado por defecto en el robot cuenta con una detección de colores en la que usa un filtro HSV para poder separar, de forma sencilla y ahorrar poder de procesamiento, las figuras de colores del fondo, es decir, del espacio de trabajo a partir de la imagen preprocesada. En la Figura 117 se muestra una interfaz que permite determinar los valores adecuados para un filtro HSV, y en la Figura 118 se puede visualizar la imagen con el filtro aplicado.

Figura 117

Determinación de valores del filtro HSV

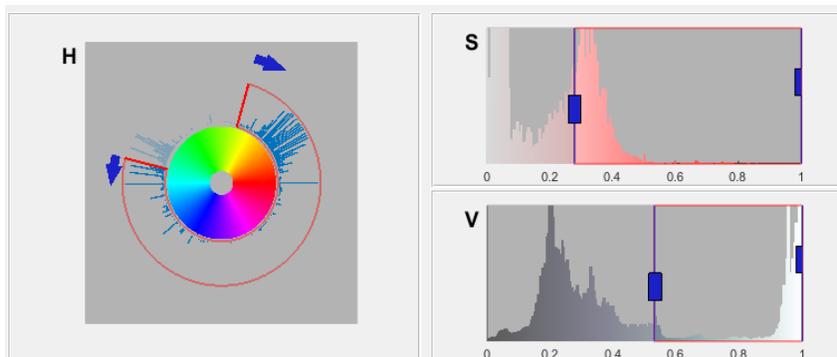
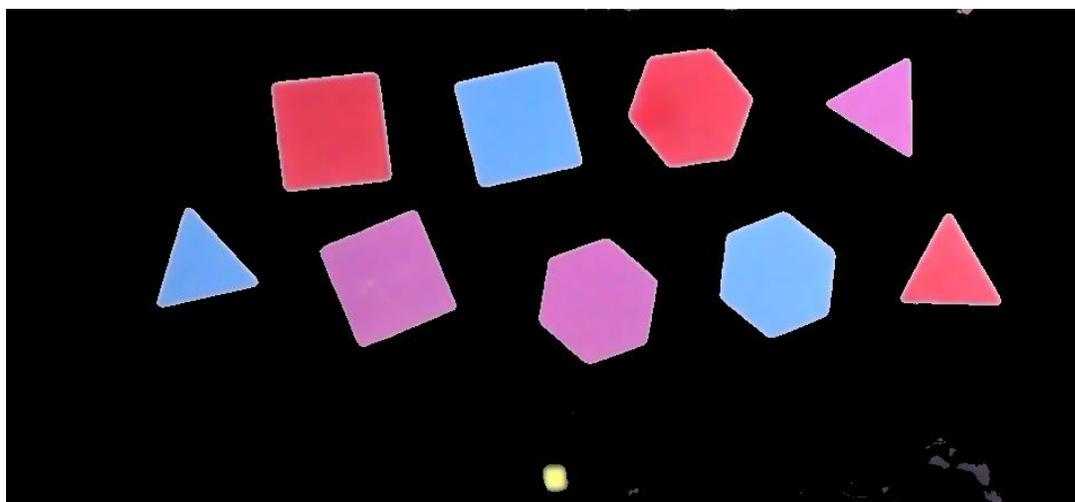
**Figura 118**

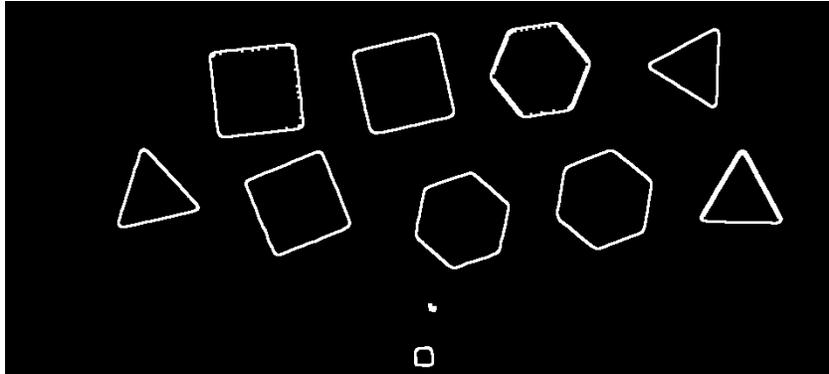
Imagen con filtro HSV aplicado



Una vez filtrada la imagen, se procede a realizar la umbralización y detección de bordes, haciendo uso del método de Canny, gracias a la implementación de OpenCV de este método se puede obtener los puntos pertenecientes a cada contorno de la figura y analizarlos de forma independiente. La figura 119 muestra la imagen con los contornos de los objetos detectados.

Figura 119

Contornos detectados dentro de la imagen



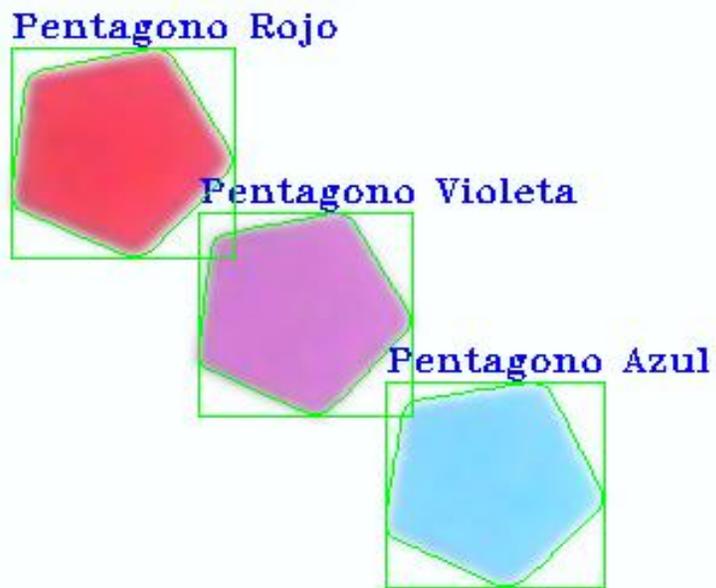
Posterior a este procedimiento, se continúa descartando contornos a partir ciertos criterios para mejorar el desempeño del algoritmo, los criterios que se tomaron en cuenta son:

- Basado en el área que forma el contorno: se descarta ya que áreas muy pequeñas o excesivamente grandes (entre 150 y 15000 pixeles²) son consideradas falla de la detección y por ende debe descartarse.
- Colores fuera de los umbrales especificados: se descartan para evitar la detección de piezas que no se realizaron para el funcionamiento por defecto del robot y que por lo tanto el robot no pueda interactuar.

Estos criterios pueden ser modificables por el usuario para hacer una aplicación diferente a la implementada por defecto. Finalmente, en la detección se realiza una comparación del color con umbrales definidos en el espacio HSV determinando el color y la forma a partir del número de ángulos mayores a 20° que tenga cada figura. La Figura 120 muestra un ejemplo de objetos detectados e identificados por forma y color.

Figura 120

Objetos detectados e identificados por forma y color



Adicionalmente, se proporciona una retroalimentación visual al usuario del resultado que produjo la detección, dibujando el contorno encontrado, un rectángulo contenedor y un texto indicando la forma y color que se detectó.

Estimación de la distancia

La estimación de la distancia de los objetos de prueba desde el origen de coordenadas del robot se realiza haciendo una diferenciación del color seleccionado para este punto, siendo un color diferente al de la gama de los objetos de prueba. En la Figura 121 se diferencia los objetos detectados con respecto al origen de coordenadas.

Figura 121

Objetos detectados dentro de la imagen



En el algoritmo de detección de forma y color se obtiene un vector con coordenadas de cada figura, mismo que se puede usar para calcular el centro de la figura y su posición en píxeles, por lo que se procede al cálculo de la distancia en píxeles desde el origen de coordenadas (forma color amarillo) hacia los objetos de prueba.

El último paso en este proceso es realizar la conversión de píxeles a centímetros utilizando un simple cálculo:

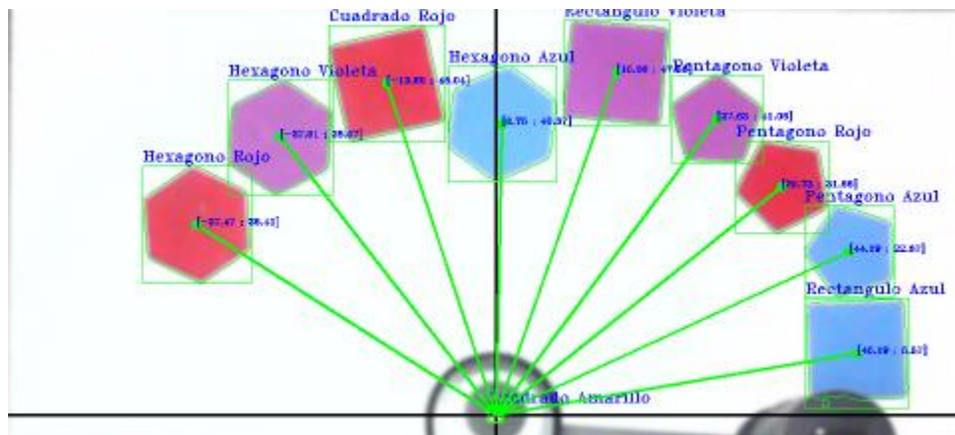
$$\text{coordenada } X \text{ [cm]} = \frac{\text{coordenada } X \text{ [píxeles]} \cdot \text{ancho del WS [cm]}}{\text{ancho de la imagen [píxeles]}}$$

$$\text{coordenada } Y \text{ [cm]} = \frac{\text{coordenada } Y \text{ [píxeles]} \cdot \text{alto del WS [cm]}}{\text{alto de la imagen [píxeles]}}$$

Usando a la transformación de la imagen que se realizó en el preprocesado de la imagen, todos los datos de la ecuación son conocidos y por lo tanto se puede resolver para hallar tanto la coordenada en el eje X como en el eje Y en centímetros. La figura 122 muestra el resultado del algoritmo.

Figura 122

Obtención de forma, color coordenadas de los centros de los objetos en la imagen

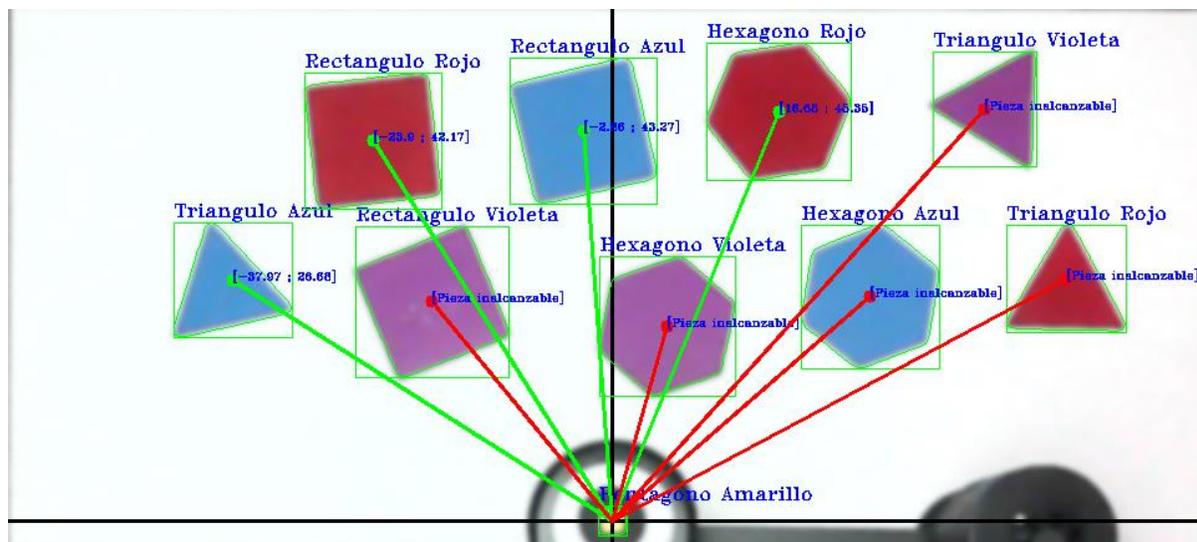


Para ofrecer retroalimentación visual para el usuario, se dibujan líneas que representan un vector indicando la estimación de la distancia y las coordenadas en centímetros al centro de cada objeto de prueba.

Además, se realiza un cálculo de la distancia en centímetros y se verifica si es alcanzable físicamente por el robot, siendo las condiciones establecidas que debe ser mayor a 36 centímetros como límite inferior y 50 centímetros como límite superior. En caso de no cumplir con las condiciones, se marca con líneas rojas el vector de la estimación de la distancia, como se observa en la Figura 123.

Figura 123

Determinación de objetos alcanzables e inalcanzables por el robot SCARA



Modelado de las plantas

La precisión del robot SCARA al momento de cumplir una instrucción depende de la posición angular que tomen los motores que controlan el brazo y antebrazo. Este comportamiento puede ser visible si se conoce la función de transferencia que describe los valores que toma la planta durante el transcurso de tiempo que le toma llegar a su valor deseado. Las funciones de transferencia de los motores que controlan la posición del robot SCARA pueden ser obtenidas por diferentes métodos.

Uno de los métodos es el modelamiento matemático a partir de la determinación de masas, inercias y constantes que se requieren para describir los mecanismos involucrados, las masas que son desplazadas y el funcionamiento de los motores en base a voltajes y corrientes. Al ser un sistema complejo por el número considerable de cuerpos que se encuentran involucrados en cada uno de los mecanismos, adicionalmente de las piezas que sirven de cubiertas, y teniendo en cuenta el proceso de determinación de constantes de los motores, el proceso de modelado matemático toma un nivel de complejidad demasiado alto.

El método que se utiliza en este apartado es la identificación de plantas, que consiste en considerar a la planta como una caja negra a la cual se le asignan entradas y se miden las salidas bajo una tasa de muestreo. Este proceso evita la necesidad de cálculos de constantes, masas e inercias del método de modelado. Las medidas obtenidas se las ingresa a un software matemático que determina una función de transferencia que se aproxima al comportamiento real de la planta, con la cual se puede obtener la respuesta de la planta en su régimen transitorio y estacionario.

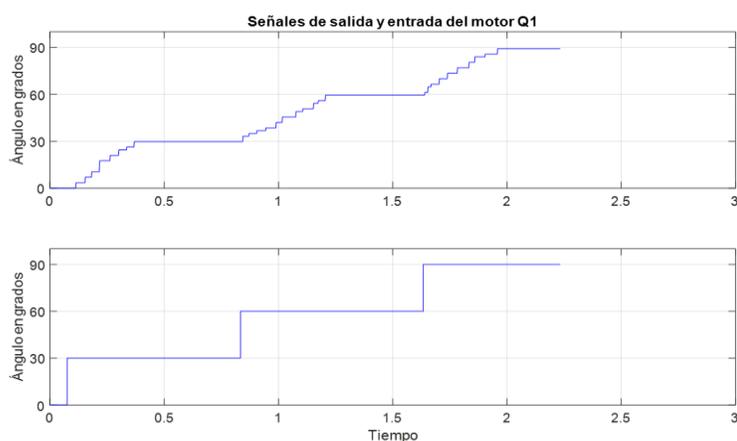
Modelamiento de las plantas

Para realizar la identificación de las plantas que requieren control, se asignan entradas de tipo escalón a la planta para tomar los datos en el transcurso del tiempo. Los valores de la entrada de escalón que se utilizaron son de 30° , 60° y 90° . Este proceso se realiza de forma individual para cada uno de los motores que permiten el movimiento en los grados de libertad Q1 y Q2, de tal manera que de este proceso se obtienen dos funciones de transferencia.

Del proceso de toma de datos del motor del grado de libertad Q1 se obtiene la Figura 124 que contrasta la salida con las entradas asignadas.

Figura 124

Señales de salida y entrada del brazo del robot



Los datos obtenidos se ingresan a la herramienta de identificación, que en base a los valores de entrada, valores de salida, y la tasa de muestreo empleada para la toma de estos datos, arroja una función de transferencia que se aproxima al comportamiento real de la planta. En este proceso de identificación se considera que la función de transferencia es confiable cuando el porcentaje de aproximación es igual o superior al 60%. Para los valores del motor del grado de libertad Q1, con un 92.7% de aproximación, como se muestra en la Figura 125, se obtiene la función de transferencia.

$$G(s) = \frac{3900}{s^3 + 175.1s^2 + 1464s + 3881}$$

Figura 125

Características de la función de transferencia del brazo obtenida por modelamiento

```
Motor1 =

  From input "u1" to output "y1":
      3900
  -----
  s^3 + 175.1 s^2 + 1464 s + 3881

Name: tf3
Continuous-time identified transfer function.

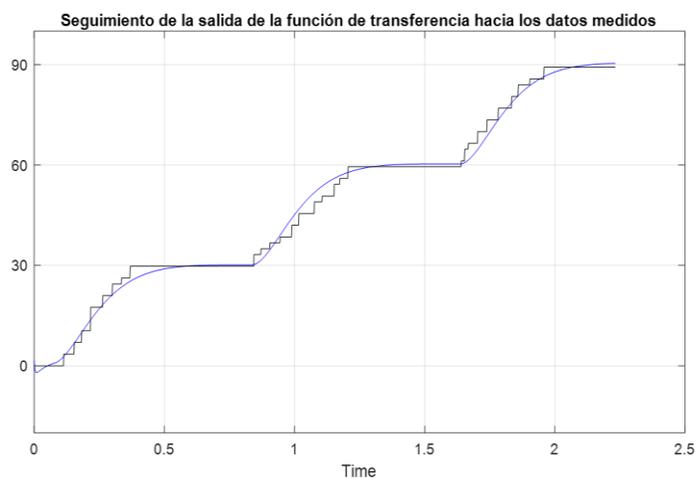
Parameterization:
  Number of poles: 3   Number of zeros: 0
  Number of free coefficients: 4
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 92.7% (stability enforced)
FPE: 3.661, MSE: 3.626
```

Mediante el uso de la herramienta de "tracking" del software matemático se puede observar en la Figura 126 cómo la respuesta de la función de transferencia modelada sigue a la curva de salida de los datos medidos.

Figura 126

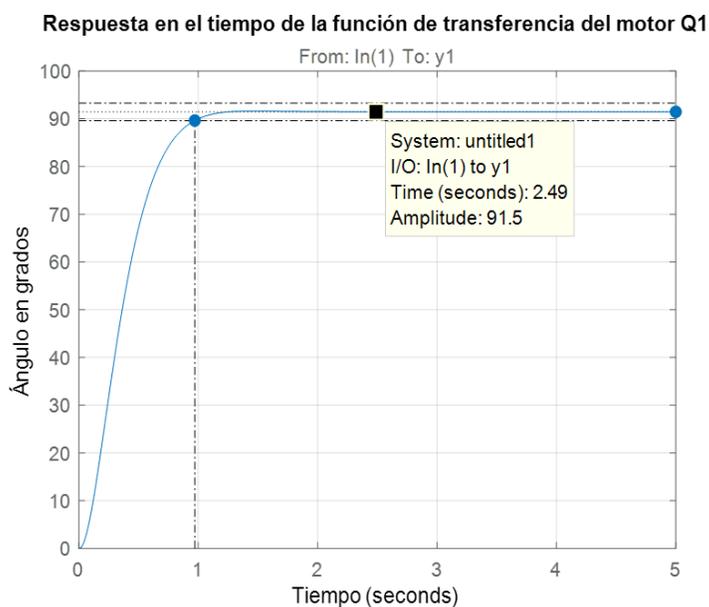
Seguimiento de la salida de la función de transferencia del brazo hacia los datos medidos



En base a la función de transferencia que arroja el software, se le asigna una entrada de tipo escalón para la determinación de las características de la planta en estado estacionario. La respuesta en el tiempo de la función de transferencia del brazo se muestra en la Figura 127.

Figura 127

Respuesta en el tiempo de la función de transferencia del brazo



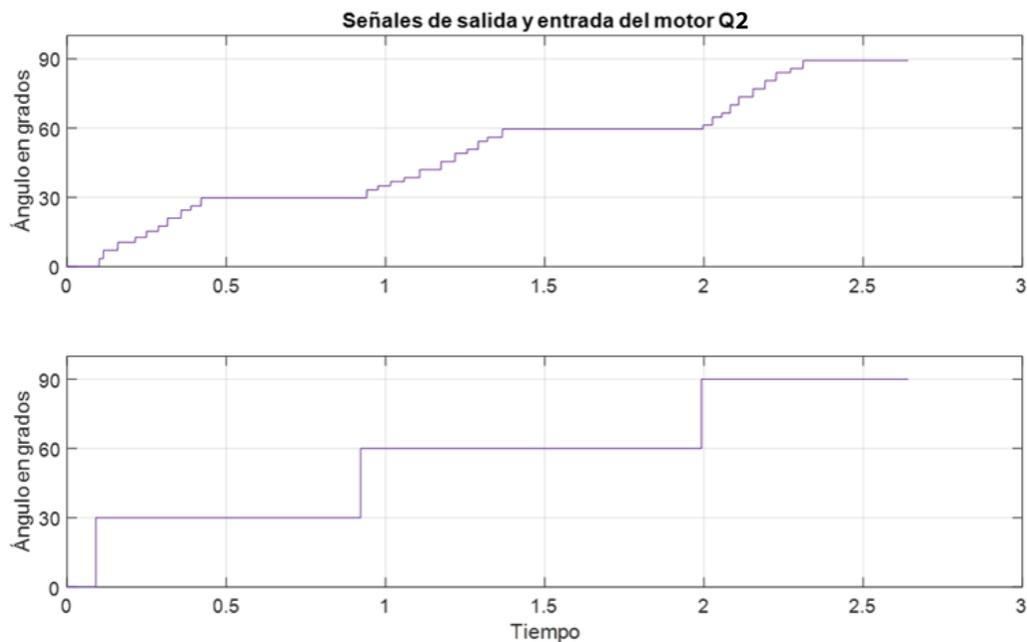
De la gráfica se puede determinar que el tiempo de establecimiento de la planta es de aproximadamente 1s, y que para una entrada de tipo escalón de 90° se obtiene una salida de 91.5° en estado estacionario, que implica un error en estado estable de 1.6%.

$$e_{ss} = \frac{91.5 - 90}{90} * 100\% = 1.6\%$$

Del proceso de toma de datos del motor del grado de libertad Q2 se obtiene la Figura 128 que contrasta la salida con las entradas asignadas.

Figura 128

Señales de salida y entrada del antebrazo



Los datos obtenidos se ingresan a la herramienta de identificación, que en base a los valores de entrada, valores de salida, y la tasa de muestreo empleada para la toma de estos datos, arroja una función de transferencia que se aproxima al comportamiento real de la planta. Para los valores del motor del grado de libertad Q2, con un 91.26% de aproximación, como se muestra en la Figura 129, se obtiene la función de transferencia

$$G(s) = \frac{10210}{s^3 + 162s^2 + 2356s + 10060}$$

Figura 129

Características de la función de transferencia del antebrazo obtenida por modelamiento

```
>> Motor2

Motor2 =

  From input "u1" to output "y1":
      1.021e04
  -----
  s^3 + 162 s^2 + 2356 s + 1.006e04

Name: tf7
Continuous-time identified transfer function.

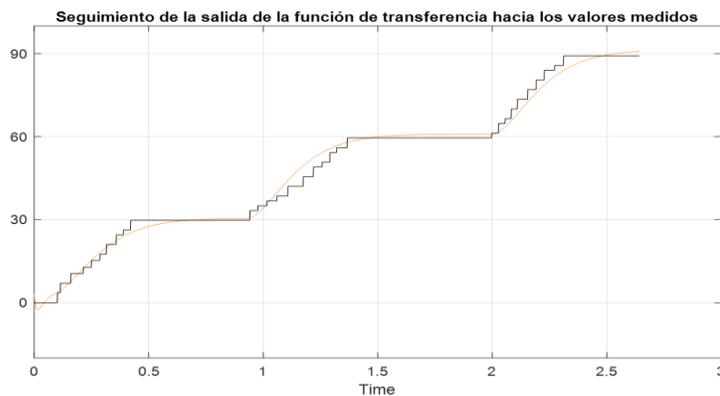
Parameterization:
  Number of poles: 3   Number of zeros: 0
  Number of free coefficients: 4
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 91.26% (stability enforced)
FPE: 4.885, MSE: 4.857
```

Mediante el uso de la herramienta de "tracking" del software matemático se puede observar en la Figura 130 cómo la respuesta de la función de transferencia modelada sigue a la curva de salida de los datos medidos.

Figura 130

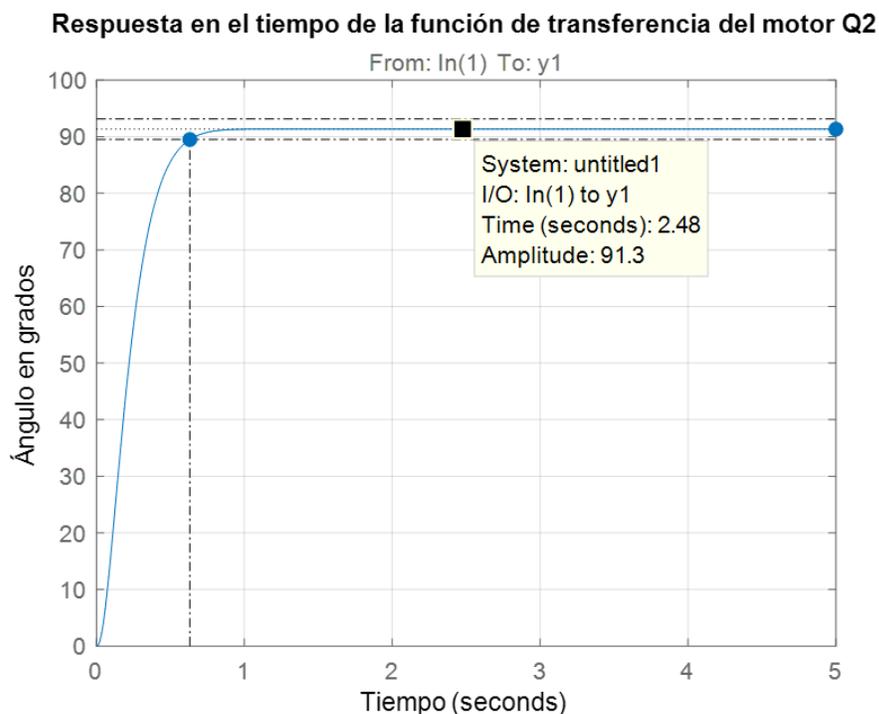
Seguimiento de la salida de la función de transferencia del brazo hacia los datos medidos



En base a la función de transferencia que arroja el software, se le asigna una entrada de tipo escalón para la determinación de las características de la planta en estado estacionario. La respuesta en el tiempo de la función de transferencia del brazo se muestra en la Figura 131.

Figura 131

Respuesta en el tiempo de la función de transferencia del antebrazo



De la gráfica se puede determinar que el tiempo de establecimiento de la planta es de aproximadamente 1s, y que para una entrada de tipo escalón de 90° se obtiene una salida de 91.3° en estado estacionario, que implica un error en estado estable de 1.4%.

$$e_{ss} = \frac{91.3 - 90}{90} * 100\% = 1.4\%$$

La Tabla 45 muestra el resumen de las funciones de transferencia obtenidas y sus características de la respuesta en el tiempo.

Tabla 45

Resumen de la identificación de las funciones de transferencia

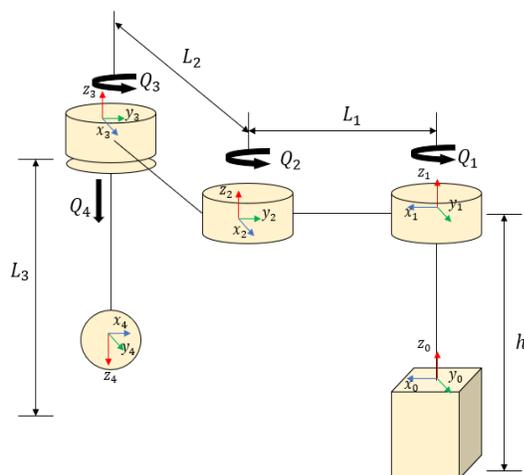
Motor	Función de transferencia	Error en estado estable	Tiempo de establecimiento [s]
Grado de libertad Q1 (Brazo)	$G(s) = \frac{3900}{s^3 + 175.1s^2 + 1464s + 38}$	1.6%	0.66
Grado de libertad Q2 (Antebrazo)	$G(s) = \frac{10210}{s^3 + 162s^2 + 2356s + 1000}$	1.4%	0.971

Cálculo de cinemática directa e inversa

El cálculo de la cinemática directa e inversa permite obtener la coordenada donde se ubica el brazo robótico a partir de los ángulos dados en los grados de libertad, mientras que la inversa es la obtención de los grados a partir de las coordenadas. El cálculo de la cinemática directa se realiza por el método de Denavit - Hartenberg en base al esquema que se muestra en la Figura 132.

Figura 132

Esquema del robot SCARA basado en el método Denavit - Hartenberg



Para la programación del microcontrolador que es el que ejecuta las instrucciones para posicionar adecuadamente el robot en el plano XY, para los cálculos se considera únicamente las transformaciones que ocurren desde la posición 0 hasta la 2, tomando en cuenta únicamente Q1 y Q2.

Se plantea la Tabla 46 donde se muestran las transformaciones que deben realizarse para pasar desde la posición 0 a la 1, de 1 a 2 y de 2 a 3, en base a los desplazamientos y giros permitidos por el método utilizado.

Tabla 46

Parámetros de Denavit - Hartenberg

Transformación	Giro en Z: θ	Desplazamiento en Z: d	Desplazamiento en X: a	Giro en X: α
H_1^0	0	h	0	0
H_2^1	Q_1	0	L_1	0
H_3^2	$Q_2 + 90^\circ$	0	$-L_2$	0

De la tabla de los parámetros de Denavit - Hartenberg se obtienen las matrices de transformación homogénea para cada transformación y la general que va desde la posición 0 a la posición 2, como se muestran en las Figuras 133 y 134.

Figura 133

Matrices homogéneas individuales del robot SCARA

```
#Cinemática directa del SCARA

#Matrices homogéneas individuales
h, l1, l2, q1, q2 = symbols('h L_1 L_2 q_1 q_2')
H01 = denavit(0,h,0,0)
H12 = denavit(q1,0,l1,0)
H23 = denavit(q2+pi/2, 0, -l2,0)

[H01, H12, H23]
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & L_1 \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & L_1 \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -\sin(q_2) & -\cos(q_2) & 0 & L_2 \sin(q_2) \\ \cos(q_2) & -\sin(q_2) & 0 & -L_2 \cos(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 134

Matriz homogénea total del robot SCARA

```
#Matriz homogénea general
H03 = H01*H12*H23
simplify(H03)
```

$$\begin{bmatrix} -\sin(q_1 + q_2) & -\cos(q_1 + q_2) & 0 & L_1 \cos(q_1) + L_2 \sin(q_1 + q_2) \\ \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & L_1 \sin(q_1) - L_2 \cos(q_1 + q_2) \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En base a esta matriz general, se puede determinar la posición en X y Y del efector final cuando se les ha asignado un valor a los grados de libertad Q1 y Q2. Considerando las medidas de $L_1 = 300\text{mm}$, $L_2 = 200\text{mm}$ y $h = 250\text{mm}$, la matriz general queda como se muestra en la Figura 135.

Figura 135

Matriz homogénea del robot SCARA con valores de longitud

$$\begin{bmatrix} -\sin(q_1 + q_2) & -\cos(q_1 + q_2) & 0 & 200 \sin(q_1 + q_2) + 300 \cos(q_1) \\ \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & 300 \sin(q_1) - 200 \cos(q_1 + q_2) \\ 0 & 0 & 1 & 250 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En base a esta matriz se puede determinar la posición en X y Y del efector final, que está dada por las ecuaciones que se muestran. Se desprecia la coordenada en Z, puesto que el efector final solo puede estar arriba en $Z = 250$ o abajo en $Z = 0$.

$$X = 200\sin(q_1 + q_2) + 300\cos(q_1)$$

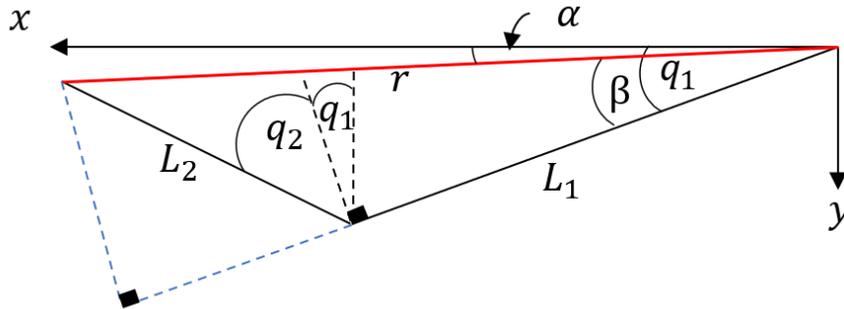
$$Y = 300\sin(q_1) - 200\cos(q_1 + q_2)$$

La cinemática inversa es realizar el proceso inverso, a partir de coordenadas en X y Y, obtener los ángulos que deben tomar cada grado de libertad. En el presente proyecto estos cálculos de cinemática inversa son los que se van a utilizar para posicionar el robot cuando se reciban coordenadas obtenidas por el algoritmo de identificación de objetos por visión artificial.

El método que se utiliza es el geométrico, que consiste en considerar al robot como un manipulador planar de tipo RR. En la Figura 136 se muestra de forma esquemática la geometría del robot para los cálculos posteriores.

Figura 136

Esquema de los eslabones del robot SCARA



Se calcula el valor del ángulo Q_2 , que se puede obtener por ley de cosenos, tomando las coordenadas en XY y la longitud de los eslabones.

$$r^2 = x^2 + y^2$$

$$r^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(90 + q_2) = L_1^2 + L_2^2 + 2L_1L_2\sin(q_2)$$

$$\sin(q_2) = \frac{x^2 + y^2 - (L_1^2 + L_2^2)}{2L_1L_2}$$

$$\cos(q_2) = \sqrt{1 - \sin^2(q_2)}$$

$$q_2 = \text{atan2}(\sin(q_2), \cos(q_2))$$

El valor de Q_1 se calcula a partir de la expresión

$$q_1 = \alpha + \beta$$

$$\alpha = \text{atan2}(y, x)$$

$$\beta = \text{atan2}(L_2\sin(90 - q_2), L_1 + L_2\cos(90 - q_2))$$

Considerando las medidas de $L_1 = 300\text{mm}$, $L_2 = 200\text{mm}$, las expresiones para calcular el valor de Q_1 y Q_2 quedan de la forma

$$\sin(q_2) = \frac{x^2 + y^2 - (300^2 + 200^2)}{2(300)(200)}$$

$$\cos(q_2) = \sqrt{1 - \sin^2(q_2)}$$

$$\beta = \text{atan2}(200 \sin(90 - q_2), 300 + 200 \cos(90 - q_2))$$

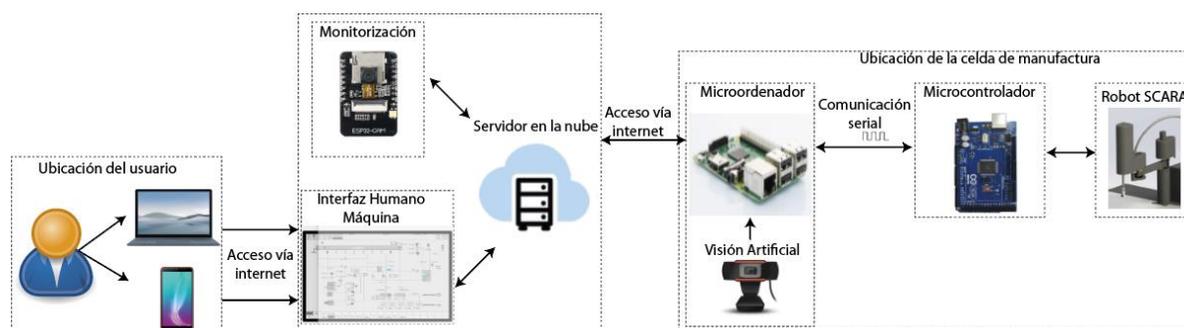
Diseño de la infraestructura de comunicación

Uno de los requerimientos de diseño es que el robot SCARA sea manipulable de manera remota, es decir, el robot se encuentra en una ubicación, y el usuario desde otra ubicación diferente puede encenderlo y realizar movimientos dentro del espacio de trabajo, y monitorizarlo en tiempo real, ya sea por medio de los indicadores de variables que muestran las mediciones de los sensores del robot, o por medio de la monitorización por vídeo en tiempo real.

De forma esquemática, se plantea el diagrama de comunicación que se muestra en la Figura 137, en el cual las instrucciones del usuario son enviadas al robot y la información del robot es enviada al usuario.

Figura 137

Esquema de la arquitectura de comunicación



Como se puede observar en la figura, la estructura se basa en que el usuario desde su ubicación, mediante conexión a internet puede acceder a una interfaz humano máquina HMI que le permite recibir información del robot y a su vez enviar instrucciones. Estas instrucciones son enviadas a través de un protocolo de comunicación hacia el microordenador, el cual envía la instrucción por comunicación serial hacia el microcontrolador, que realiza la activación de los actuadores correspondientes a una instrucción.

La información correspondiente al estado de los sensores realiza el mismo trayecto que las instrucciones, pero, en sentido opuesto, es decir, el microcontrolador envía por comunicación serial los valores de los sensores hacia el microordenador, información que es enviada mediante el protocolo de comunicación hacia el servidor en la nube, el mismo que presenta de forma ordenada la información en la interfaz gráfica con la que interactúa el usuario.

Para lograr construir la infraestructura planteada se determinan ciertos requerimientos del diseño de la estructura de comunicación.

- Alta velocidad de respuesta entre usuario y equipo
- Envío y recepción de información desde y hacia el usuario desde cualquier localidad
- Utilización de un servidor en la nube
- Envío de video en tiempo real hacia el usuario

Del esquema de la infraestructura de se puede mencionar que existen dos componentes que son el núcleo para que exista comunicación entre el usuario y el robot SCARA, que son el microordenador y el servidor alojado en la nube. Para cubrir la necesidad del microordenador se plantean dos alternativas.

Alternativa 1: Raspberry Pi Zero

Se plantea utilizar una placa Raspberry Pi Zero, como la mostrada en la Figura 138, para que realice la conexión del robot SCARA con el servidor en la nube vía internet, permitiendo el envío y recepción de información desde el HMI. Algunas de las características que esta placa ofrece son

- Procesador 1GHz de un núcleo
- 512MB de memoria RAM
- Almacenamiento mediante tarjeta MicroSD
- Conexión WiFi mediante uso de adaptador
- Puerto MicroUSB y MiniHDMI.

- GPIO de 40 pines

Figura 138

Placa Raspberry Pi Zero



Tabla 47

Ventajas y desventajas de la alternativa 1 para el microordenador

Ventajas	Desventajas
Bajo costo	Número limitado de puertos USB
	Baja capacidad de procesamiento
	No es posible conectar a internet sin adaptador de red

Alternativa 2: Raspberry Pi 3B+

Se plantea utilizar una placa Raspberry Pi 3B+ como la mostrada en la Figura 139 para que realice la conexión del robot SCARA con el servidor en la nube vía internet, permitiendo el envío y recepción de información desde el HMI. Algunas de las características que esta placa ofrece son

- Procesador 1.4GHz de cuatro núcleos
- 2GB de memoria RAM
- Almacenamiento mediante tarjeta MicroSD
- Conexión a internet por WiFi o cable de red

- 4 puertos USB 2.0
- GPIO de 40 pines
- Alimentación de 5V - 2.5A

Figura 139

Placa Raspberry Pi 3B+

**Tabla 48**

Ventajas y desventajas de la alternativa 2 para el microordenador

Ventajas	Desventajas
Mayor capacidad de procesamiento	Alto costo
Conexión a internet por defecto	
Mayor cantidad de puertos USB	

Una vez determinadas las ventajas y desventajas de las alternativas para la obtención de imágenes para el módulo de visión artificial, se procede a realizar la Tabla 49 que es una matriz de evaluación para determinar a partir de criterios cuál es la mejor opción a ser implementada.

Tabla 49

Matriz de evaluación para las alternativas de placa microordenador

Criterio	Ponderación	Raspberry Pi Zero		Raspberry Pi 3B+	
		CA	EP	CA	EP
Capacidad de procesamiento	35%	3	1.05	5	1.75
Facilidad de conexión a internet	35%	1	0.35	5	1.75
Facilidad de conexión de periféricos	20%	2	0.4	5	1
Costo	10%	5	0.5	1	0.1
Total			2.3		4.6

Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se obtuvo que la alternativa adecuada para la selección del microordenador es la placa Raspberry Pi 3B+, que ofrece un mejor desempeño para los requisitos que se presentaron.

Para la realización de la comunicación se requiere el uso de un servidor alojado en la nube, por lo que se plantean dos alternativas de plataformas que ofrecen servicios web, como el alojamiento de servidores, espacio de almacenamiento para documentos, brokers para comunicación, entre otros servicios.

Alternativa 1: Microsoft Azure

Se plantea utilizar los servicios web que ofrece la plataforma de Microsoft Azure para el alojamiento de un servidor en la nube que permite la conexión entre usuario y la celda de manufactura. Dentro de los servicios que ofrece Microsoft Azure se mencionan algunos:

- Servicios de infraestructura, permitiendo desplegar máquinas virtuales con variedad de sistemas operativos.
- Servicio de almacenamiento para archivos y bases de datos.
- Soporte para Windows Server, SQL server, Oracle, IBM.
- Servicio destinado a la utilización de IoT.

Tabla 50

Ventajas y desventajas de la alternativa 1 de plataforma de servicios web

Ventajas	Desventajas
Soporta varios sistemas operativos en sus máquinas virtuales	Complejidad en la documentación de los servicios web
Ofrece alta seguridad a las entidades creadas	Sistema complejo de cobros del servicio
Servicio para la utilización de IoT	
Es escalable y personalizable de acuerdo a las necesidades	

Alternativa 2: Amazon Web Services

Se plantea utilizar los servicios web que ofrece la plataforma de Amazon Web Services para el alojamiento de un servidor en la nube que permite la conexión entre usuario y la celda de manufactura. Dentro de los servicios que ofrece esta plataforma se mencionan algunos:

- Servicios de alojamiento de instancias con soporte de una variedad de sistemas operativos de distribución Linux, Windows Server y MacOS.
- Servicio de almacenamiento para archivos y de respaldo de archivos e instancias creadas.
- Alojamiento para bases de datos y herramientas de análisis.
- Variedad de servicios destinados a la utilización de IoT.

Tabla 51

Ventajas y desventajas de la alternativa 2 de plataforma de servicios web

Ventajas	Desventajas
Soporta varios sistemas operativos en sus instancias	Sistema complejo de cobros del servicio
Ofrece sistema de seguridad a las instancias creadas	
Cuenta con varios servicios para la utilización de IoT	
Es escalable y personalizable de acuerdo a las necesidades	
Documentación de los servicios detallada	

En base a las ventajas y desventajas que presenta cada una de las alternativas planteadas, se presenta la Tabla 52 con una matriz de evaluación para determinar la mejor opción para la lectura de altura.

Tabla 52

Matriz de evaluación para las alternativas de placa microordenador

Criterio	Ponderación	Microsoft Azure		Amazon Web Services	
		CA	EP	CA	EP
Costo	20%	4	0.8	4	0.8
Facilidad de implementación	30%	4	1.2	5	1.5
Compatibilidad con IoT	30%	4	1.2	5	1.5

Criterio	Ponderación	Microsoft Azure		Amazon Web Services	
		CA	EP	CA	EP
Compatibilidad con distribuciones Linux	20%	4	0.8	5	1
Total			4		4.8

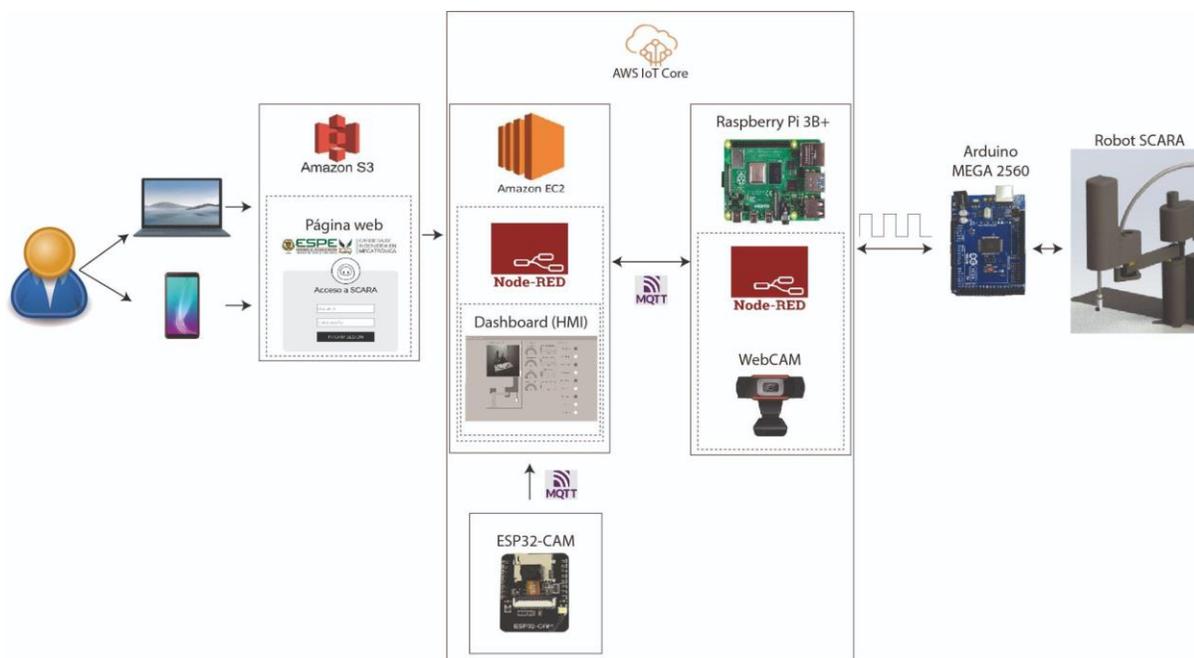
Nota. CA: Calificación, EP: Evaluación ponderada.

En base a la tabla de evaluación, se determinó que la mejor opción es la segunda alternativa, que corresponde al uso de los servicios web que ofrece Amazon Web Services, en base a los criterios seleccionados, a pesar de que la alternativa de Microsoft Azure no obtuvo una baja calificación en la evaluación.

Una vez realizada la selección de los componentes a utilizar en la estructura de comunicación se detalla en la Figura 140 la arquitectura que se utiliza para lograr la comunicación remota del usuario con la celda de manufactura en la siguiente imagen.

Figura 140

Arquitectura de comunicación completa



El usuario debe contar con una conexión a internet para acceder a una página de inicio de sesión alojada en el servicio Amazon S3 (Simple Storage Service), de tal manera que esta página redirecciona hacia las páginas web del flow y del dashboard de Node-RED, el cual se encuentra funcionando como servicio desde que se inicia la instancia con sistema operativo de distribución Linux Ubuntu, alojada en el servicio Amazon EC2 (Elastic Compute Cloud).

El usuario puede interactuar con la HMI desarrollada en el dashboard de Node-RED para enviar instrucciones hacia la celda de manufactura, así como puede interactuar con el flow de Node-RED, de tal manera que puede realizar modificaciones en la programación del comportamiento del robot. Esta modificación del comportamiento del robot está disponible para la opción de utilizar la función de visión artificial, es decir, es alterable la forma de clasificación de objetos por visión artificial.

Las instrucciones que se generen en la HMI son enviadas hacia el microordenador utilizando el servicio de Amazon IoT Core, el cual ofrece el servicio de broker MQTT, de tal manera que tanto Node-RED alojado en la instancia de Amazon, como Node-RED alojado en el microordenador se suscriben a tópicos para el envío y recepción de información. El servicio de IoT Core permite la monitorización de los mensajes enviados por medio del plugin MQTT Test Client.

En el microordenador, de igual manera que en la instancia del servicio EC2, se encuentra alojado Node-RED de manera local que se ejecuta como servicio inmediatamente después de que se enciende y se carga el sistema operativo de distribución Linux, que, para este caso, es la distribución Pi OS (anteriormente denominada Raspbian).

El microordenador se conecta por medio de cable USB hacia el microcontrolador, de tal manera que por comunicación serial envía y recibe la información de instrucciones y valores de

los sensores incluidos en el robot SCARA, los cuales se ven reflejados en la HMI con la cual el usuario interactúa.

En la celda de manufactura que se encuentra el microordenador se encuentra una cámara web enfocando hacia el espacio de trabajo del robot SCARA, de tal manera que permite capturar una imagen de los objetos que se encuentren, esta imagen es procesada por un script desarrollado dentro del microordenador, el cual devuelve una imagen con la identificación de forma y colores de los objetos. Las imágenes de antes y después del procesamiento para la identificación de forma y colores son enviadas por MQTT hacia el HMI, además de las coordenadas en las que se estima que se encuentran los centros de cada objeto.

Adicionalmente, se programa una placa ESP-32 CAM para que transmita video hacia la HMI, de tal manera que el usuario puede monitorizar la celda de manufactura en tiempo real. Esto se realiza utilizando el servicio IoT Core, de tal manera que la placa ESP envía el video por medio de protocolo MQTT.

Diseño de la interfaz humano máquina

Para que el usuario pueda realizar operaciones con el robot SCARA dentro de la celda de manufactura y monitorizar el funcionamiento del robot, requiere un medio por el cual enviar instrucciones hacia el robot y recibir información de este, con respecto al estado de sensores, interruptores y actuadores.

El medio utilizado para esta finalidad es la interfaz humano máquina (HMI) que se implementa para que el usuario sea capaz de controlar y recibir retroalimentación del funcionamiento del robot dentro de la celda de manufactura. Esta interfaz cuenta con la información necesaria obtenida de los sensores de forma ordenada, al igual que los controles con los que se envía instrucciones para que el robot realice movimientos.

El diseño de la interfaz no debe ser confuso, no debe distraer a quien está utilizando el robot y no debe causar fatiga visual; por lo que para el diseño adecuado de la HMI se recurre a la utilización del manual “The High Performance HMI Handbook [HPHMI]”, que presenta pautas y recomendaciones para desarrollar el diseño de una HMI basadas en tres criterios: claridad, consistencia y retroalimentación, de tal manera que el usuario pueda operar la interfaz de forma sencilla e intuitiva, con la menor posibilidad de cometer errores. Algunos de las recomendaciones generales que presenta este manual son

- Los gráficos en la interfaz deben ser fáciles de leer y entender.
- El estado del proceso o máquina se presenta de forma clara.
- No existen detalles innecesarios dentro de los elementos.
- Las pantallas contienen información, no datos.
- Indicadores de anomalías son identificables de forma clara.
- Los gráficos y tipos de fuente son estandarizados para todas las pantallas desarrolladas.
- Acciones importantes deben requerir confirmación antes de realizarse.
- El diseño en general debe minimizar la fatiga del usuario

Para realizar el diseño de la interfaz humano máquina se realiza la identificación de las funciones que se pueden realizar, adicionalmente de la información que se va a presentar al usuario.

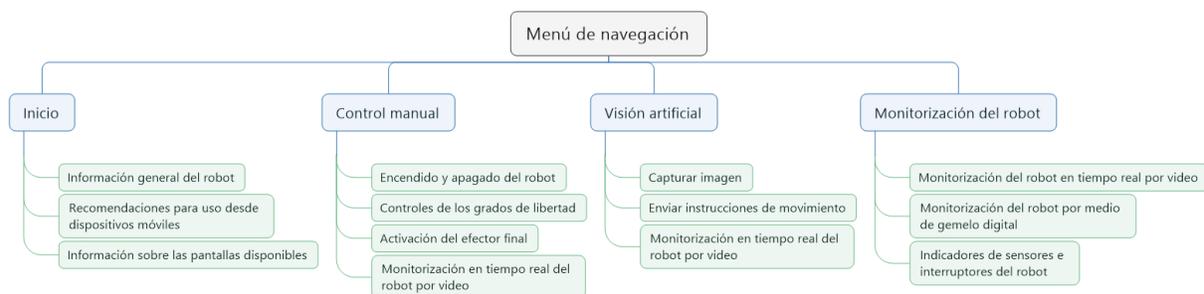
- Control manual: El usuario puede enviar instrucciones en base a parámetros de ángulos y la posible activación del efector final del robot.
- Control por visión artificial: El usuario puede hacer uso de la aplicación de visión artificial para el proceso de recolección de objetos en el espacio de trabajo del robot.
- Monitorización: El usuario puede visualizar por medio de video en tiempo real el funcionamiento del robot, gracias al uso de la cámara ESP-32 CAM.

- Pantalla de ayuda: El usuario puede acceder a esta página por información del funcionamiento de la interfaz en lo que corresponde a controles manuales y control por visión artificial.

Una vez determinadas las necesidades que se deben cubrir con la interfaz, se concluye que se requiere de cuatro pantallas que permitan las acciones mencionadas. Estas pantallas se encuentran en el mismo nivel de jerarquía, por lo que el usuario puede acceder a ellas desde cualquier pantalla mediante el uso del menú de navegación. En la Figura 141 se muestran de forma general las cuatro pantallas que se incluyen en la HMI, con su respectivo contenido.

Figura 141

Contenido de las pantallas de la HMI



El manual HPHMI contiene pautas para el diseño de HMI's para el control de varios procesos; entre estas pautas que se mencionan se incluye recomendaciones para procesos que incluyen manejo de líquidos y líneas de gas. Para el diseño de las pantallas de la HMI del presente proyecto, se toman en consideración las pautas que son de utilidad para la aplicación.

- Los gráficos indican de forma fácil de entender los valores numéricos del estado de un proceso o variable.
- No se incluyen animaciones de ninguna categoría para indicar el funcionamiento de un equipo.
- Se utiliza fondo gris para las pantallas para evitar fatiga visual

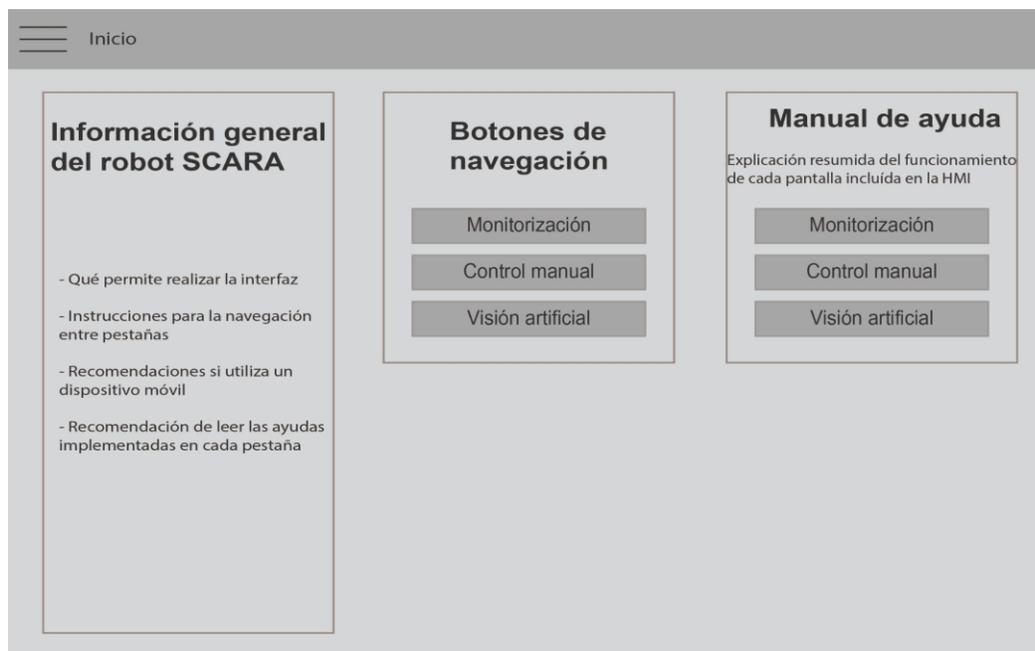
- La simbología de los actuadores se realiza en 2D de forma plana, sin perspectiva ni sombras.
- El proceso de navegación debe ser lógico y coherente con la jerarquía.
- El acceso a las pantallas debe realizarse con el mínimo número de clics.
- Solicitudes de confirmación para acciones que comprometan el funcionamiento total del equipo.
- Debe existir consistencia en el código de colores utilizado para situaciones normales y situaciones fuera de lo normal.

Como criterio adicional, cada pantalla contará con apartados de ayuda que no sean invasivos, de tal manera que el operario pueda acceder a información detallada de cada uno de los apartados con los que cuente cada una de las pantallas.

Pantalla Inicio

La pantalla Inicio es la primera que el usuario visualiza al momento de acceder a la HMI del robot SCARA. Esta pantalla sirve de introducción al funcionamiento de las pantallas incluidas, de tal manera que se explica de forma resumida cómo controlar al robot de forma manual, como controlar al robot mediante el uso de la aplicación de visión artificial, y la visualización de información del robot en la pantalla de monitorización, de tal manera que el usuario sea capaz de utilizar todos los recursos que ofrece la HMI.

Esta pantalla se enfoca en dar información a usuarios que se encuentren operando la celda de manufactura y no cuenten con la experiencia o conocimientos previos acerca del funcionamiento del robot. La Figura 142 muestra el esquema de la pantalla en base a los criterios de diseño.

Figura 142*Esquema de la pantalla Inicio****Pantalla Control manual***

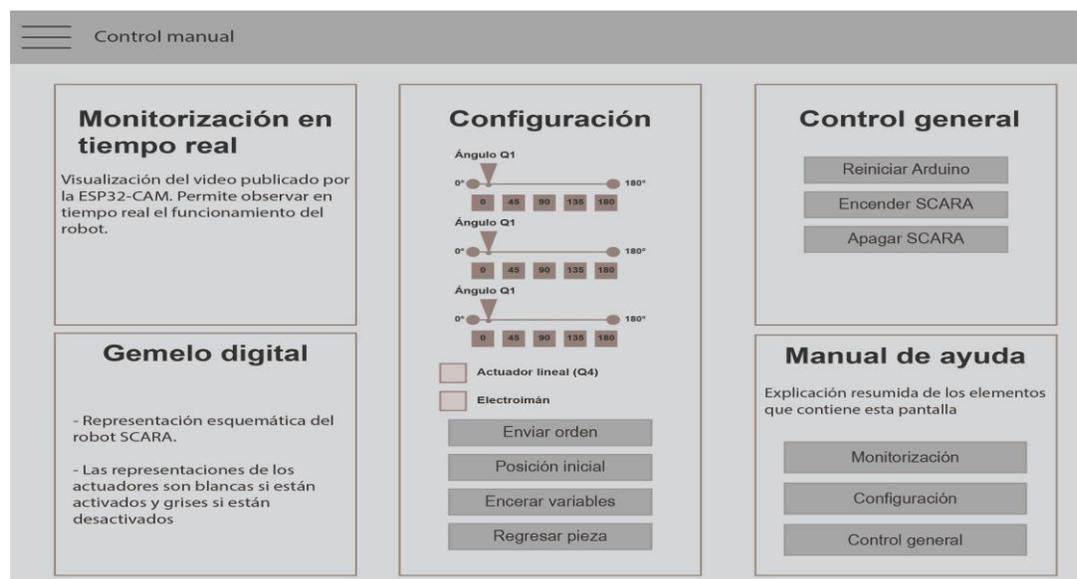
La pantalla Control manual permite al usuario encender o apagar el robot SCARA, realizar operaciones con el robot al asignarle valores de los ángulos en grados a los grados de libertad con los que cuenta el robot. Adicionalmente se puede activar o desactivar el efector final que cuenta con el electroimán que le permite sujetar objetos metálicos dentro del espacio de trabajo. Esta pantalla de control manual permite que el usuario ingrese valores de ángulos de forma manual por medio de selectores de barra (sliders), o utilizar por defecto los grados 0°, 45°, 90°, 135° y 180° por medio de botones.

Como función adicional, es posible enviar instrucciones de regresar a la posición inicial, encerrar los elementos con los que se envía los valores de los ángulos a los grados de libertad, una función de devolver el objeto que se encuentre sujeto al receptáculo de la posición inicial, y reiniciar el microcontrolador del robot si se llega a requerir.

Esta pantalla cuenta con imagen en tiempo real del funcionamiento del robot por medio de la cámara ESP32-CAM que publica el video del comportamiento del robot hacia todas las pantallas de la HMI. La Figura 143 muestra el esquema de la pantalla en base a los criterios de diseño.

Figura 143

Esquema de la pantalla Control Manual



Pantalla Visión artificial

La pantalla Visión artificial es la que permite comandar el robot SCARA en base al reconocimiento e identificación de piezas en el espacio de trabajo que se encuentran accesibles. Como se indicó en el esquema de las pantallas de la HMI, esta pantalla cuenta con imagen en tiempo real del funcionamiento del robot por medio de la cámara ESP32 CAM.

En lo que respecta a visión artificial, se tiene los botones Visión artificial, que ejecuta el script alojado en el microordenador que se encuentra en la celda de manufactura, el cual captura una imagen del espacio de trabajo del robot SCARA desde la parte superior. Con esta imagen se identifica por color y forma las piezas que se encuentren en el espacio de trabajo, y se estima el centro geométrico en coordenadas cartesianas, las mismas que son visibles en la

pantalla en una tabla, especificando para cada una su Coordenada en X, Coordenada en Y, Forma y Color.

La imagen capturada sin procesar y después de procesar se muestran en la pantalla, permitiendo verificar que la imagen sin procesar sea adecuada, o se requiere tomar otra imagen, por lo que debe presionar nuevamente el botón Visión Artificial; mientras que la imagen procesada permite visualizar los objetos detectados e identificados por color y forma, adicionalmente de si se encuentran dentro del espacio de trabajo del robot o no.

Con el botón Comenzar operaciones paso a paso se envía una instrucción al robot para que recoja la primera pieza que se muestre en la tabla de coordenadas de objetos detectados. Un clic posterior envía la instrucción de soltar el objeto tomado en el receptáculo que se encuentra en la posición inicial del robot. Si se realiza un siguiente clic, se recoge la segunda pieza que se muestre en la tabla de coordenadas.

Este proceso que se realiza paso a paso puede ser realizado de forma automática sin la necesidad de presionar el botón de Comenzar operaciones al activar la opción de Operaciones automáticas, que permitirá el envío de instrucciones cada que se termine la anterior operación. Esta opción se desactiva cuando se han terminado las operaciones pendientes por realizar, o por acción del usuario. La Figura 144 muestra el esquema de la pantalla en base a los criterios de diseño.

Figura 144

Esquema de la pantalla Visión artificial***Pantalla Monitorización***

La pantalla Monitorización es la que permite al usuario visualizar detalladamente el estado de los sensores e interruptores que incluye el robot SCARA, de tal manera que recibe retroalimentación del estado del robot constantemente. En esta pantalla se encuentra disponible la opción de visualizar el funcionamiento en tiempo real del robot por video, similar a las pantallas anteriores.

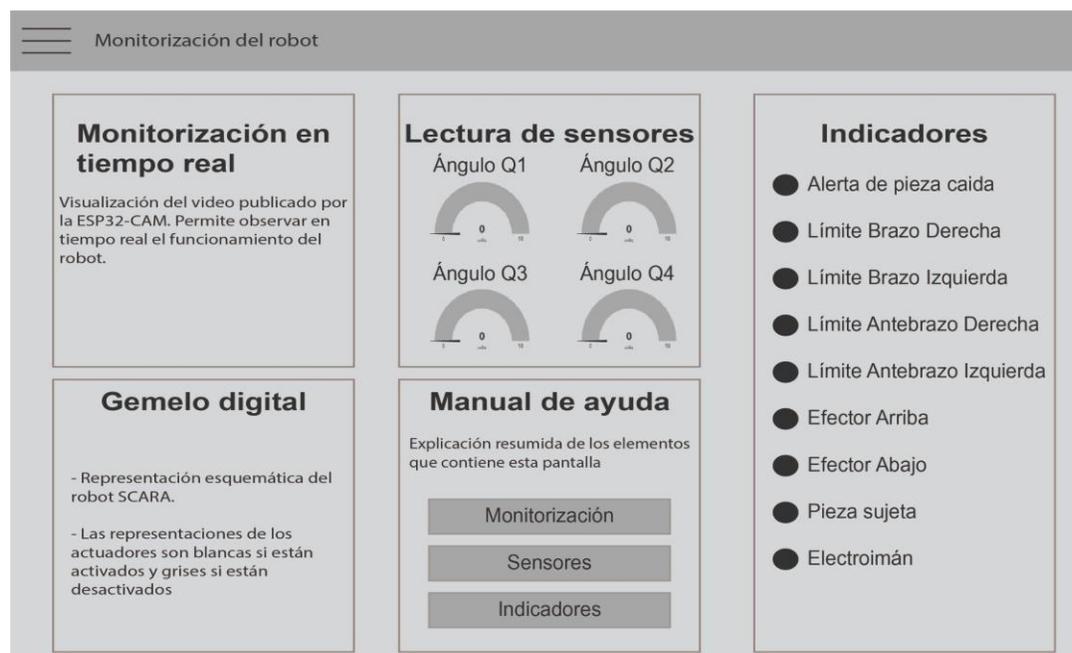
En esta pantalla se muestran de forma ordenada indicadores circulares que permiten determinar el estado de los interruptores, al encontrarse en color gris si no se encuentran presionados, y color blanco si se encuentran activados.

Adicionalmente, en forma similar a reloj analógico se muestra el valor de los ángulos en los que se encuentran los grados de libertad Q1, Q2 y Q3 del robot. La utilización de este tipo de gráfico en forma de reloj analógico está basada en la recomendación del manual de HPHMI, que determina que los indicadores analógicos reducen el proceso mental que realiza el usuario al visualizar información presentada de esta forma en comparación a si se lo realizara de forma digital únicamente en base a números. Junto a estos indicadores de ángulos es posible ver un histórico de los ángulos que ha tomado cada uno de los grados de libertad.

De forma complementaria se puede visualizar un gemelo digital, que de forma gráfica indica los actuadores que se encuentran en funcionamiento con su respectiva leyenda. De igual manera que los indicadores de los interruptores, el color gris indica que se encuentra desactivado, mientras que el color blanco indica que se encuentra activado. La Figura 145 muestra el esquema de la pantalla en base a los criterios de diseño.

Figura 145

Esquema de la pantalla Monitorización del robot



Capítulo 4: Implementación

En el capítulo 4 se detalla el proceso de construcción de los subsistemas mencionados en el capítulo 3, que trató la temática del diseño. En los siguientes apartados se evidencia el proceso de construcción de los elementos del subsistema mecánico, circuito electrónico, el proceso de programación del microcontrolador y del módulo de visión artificial, finalmente el levantamiento de la arquitectura de comunicación implementada para posibilitar la comunicación con la celda de manufactura de forma remota, considerando la implementación de la HMI dentro de la instancia creada en los servicios web del proveedor utilizado.

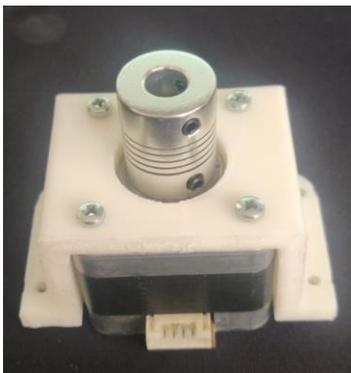
Subsistema mecánico

La estructura del robot está compuesta de una variedad de piezas mecanizadas en polímeros y metales. Las piezas metálicas se realizaron en acero estructural y aluminio, mientras que las piezas plásticas fueron confeccionadas por medio de manufactura aditiva utilizando el proceso de impresión 3D.

Piezas como los discos para los sensores de tipo encoder óptico, soportes de motores como el de la Figura 146, partes de estructuras de soporte y las piezas que sirven de cubierta del robot fueron confeccionadas por medio de impresión 3D en PLA.

Figura 146

Soporte del motor del mecanismo elevador



Las piezas que se utilizaron para el mecanismo de elevación del efector final, así como piezas utilizadas para soportar los encoders que miden los ángulos de los grados de libertad del robot fueron impresas en PLA, como se muestra en las Figuras 147 y 148. Las piezas del mecanismo rotatorio se muestran en la Figura 149, que también fue realizada en impresión 3D.

Figura 147

Acople del motor del antebrazo y soportes de encoders

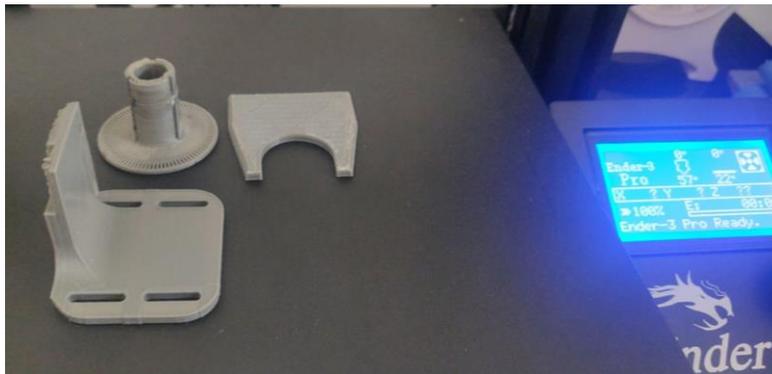


Figura 148

Piezas del mecanismo del efector final

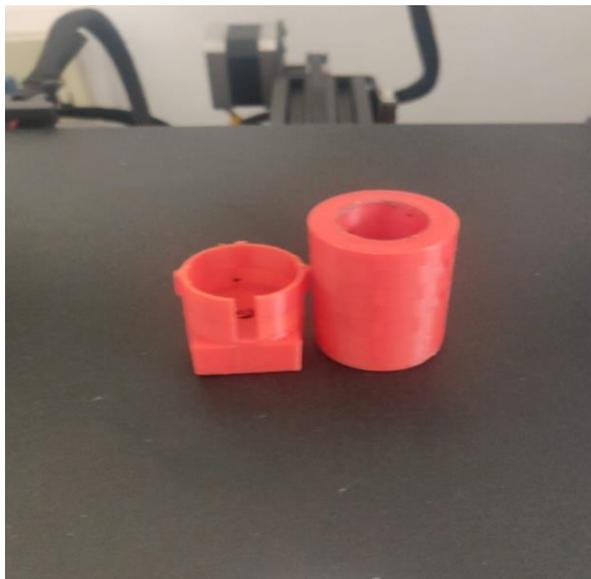


Figura 149

Piezas del mecanismo rotatorio



Para el eje del mecanismo del brazo se utilizaron chumaceras de instrumentación, que permiten que ingrese un eje de 12mm, mientras que para el eje del antebrazo se confeccionaron chumaceras en PLA, que permiten alojar rodamientos para ejes de 9mm. Estas chumaceras se muestran en la Figura 150.

Figura 150

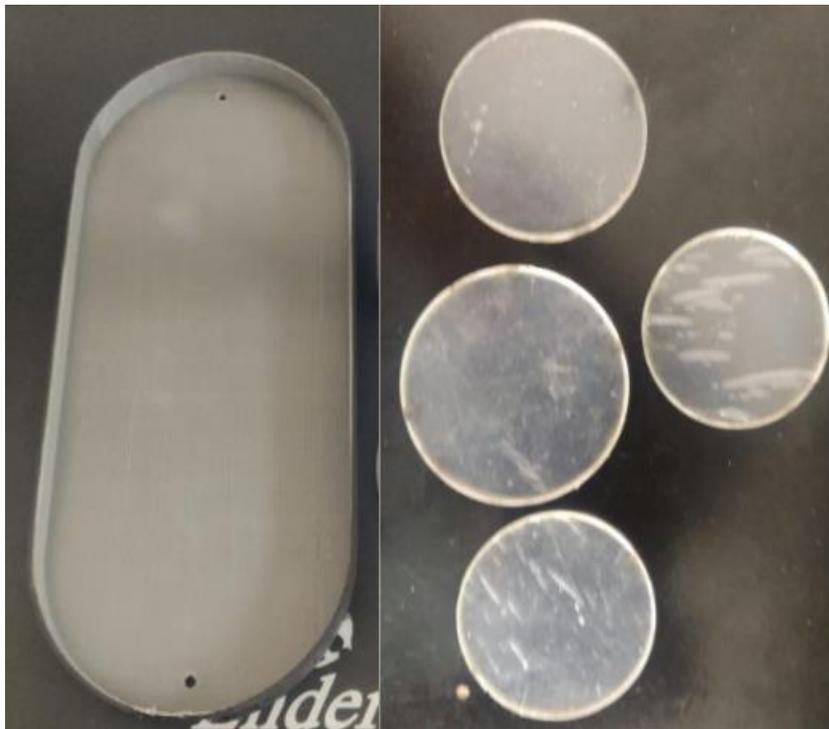
Chumaceras utilizadas en brazo y antebrazo



El robot SCARA cuenta con indicadores LED que mediante luces roja, verde y amarilla permiten informar al usuario en qué estado se encuentra, si ejecutando una orden, esperando orden, o realizando procesos de calibración. Las piezas que conforman la estructura de estos indicadores fueron impresas en PLA y cortadas por láser en acrílico transparente, como se muestra en la Figura 151.

Figura 151

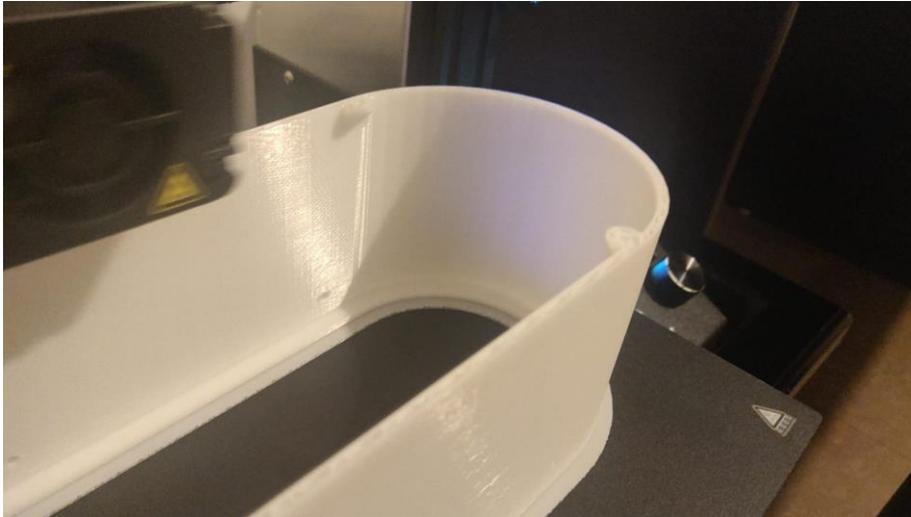
Derecha: Soporte de la estructura de los indicadores LED. Izquierda: Cubiertas de los indicadores LED.



Las piezas que sirven de cubierta para los elementos que se encuentran alojados en el antebrazo del robot se imprimieron en PLA en cuatro partes, dadas las dimensiones de las piezas requeridas y el área de trabajo de la impresora 3D utilizada. El proceso de construcción de estas piezas se muestra en las Figuras 152 y 153.

Figura 152

Proceso de impresión 3D de las piezas de cubierta

**Figura 153**

Piezas de cubierta terminadas



Una vez confeccionadas todas las piezas, se puede apreciar la apariencia de la estructura de cubierta del antebrazo, que posteriormente fue rectificada por desperfectos provocados por la impresión, y cubierta de pintura negra mate. En la Figura 154 se muestra la estructura de cubierta del antebrazo completa antes del proceso de pintura.

Figura 154

Cubierta del antebrazo del robot SCARA



La estructura del mecanismo del brazo se confeccionó utilizando un tubo de acero estructural de 4 pulgadas de diámetro, mostrado en la Figura 155, al cual se le acoplaron base y tapa que fueron cortadas en láser. Adicionalmente se muestra en la Figura 156 la pieza que es utilizada como base del antebrazo, donde se colocan los elementos contenidos en este eslabón.

Figura 155

Tubo redondo metálico utilizado para la base del robot.

**Figura 156**

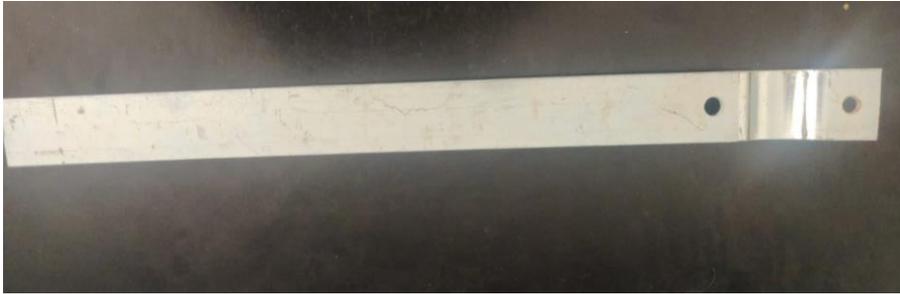
Piezas de base del robot y del antebrazo cortadas por láser



Para que el mecanismo del brazo evite deflexiones en el eje utilizado se confecciona una estructura a modo de soporte para el eje, utilizando una platina de acero estructural de 4mm de espesor y 35mm de ancho, que se muestra en la Figura 157.

Figura 157

Platina utilizada para la estructura de soporte del eje del brazo



Al ensamblar las partes del mecanismo del brazo y antebrazo, el robot tiene la apariencia que se muestra en la Figura 158. Como brazo del robot se utilizó un tubo de aluminio de 25mmx50mm.

Figura 158

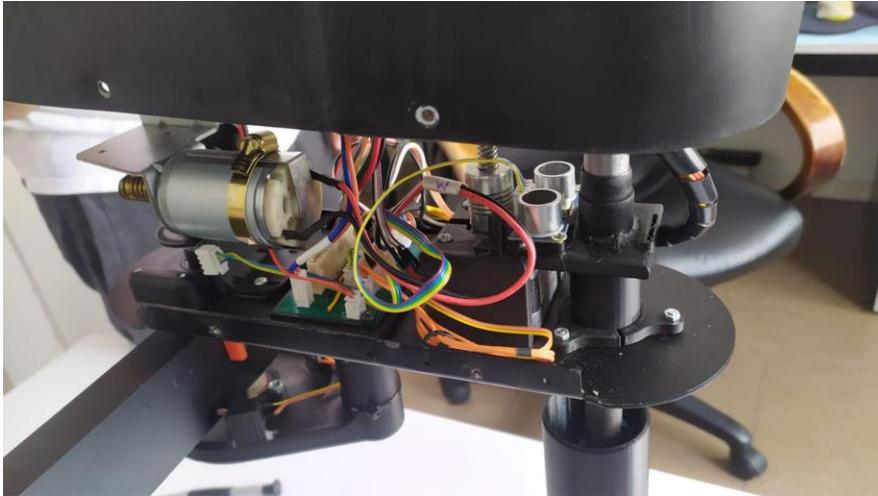
Construcción de la estructura mecánica del robot SCARA



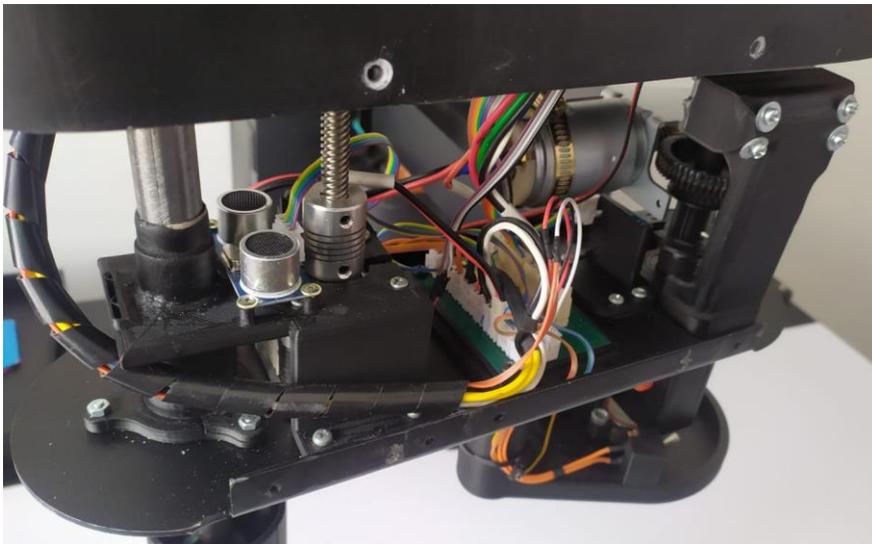
Al ensamblar las partes del mecanismo del brazo y antebrazo, la parte interna del antebrazo tiene la apariencia que se muestra en las Figuras 159 y 160. Ya cuenta con los soportes respectivos para el motor de este eslabón y los acoples necesarios para unirse al eje.

Figura 159

Parte interna del antebrazo del robot SCARA

**Figura 160**

Interior del antebrazo del robot SCARA



Una vez que el mecanismo se probó y su funcionamiento era adecuado, se le dio terminado con pintura color negro mate. Este color se escogió para que no sea invasivo para el algoritmo de visión artificial del robot. La estructura del robot finalmente toma la apariencia que se muestra en la Figura 161.

Figura 161

Robot SCARA construido



Para sostener la cámara web que se utiliza para el proceso de visión artificial se requiere de una estructura que permita apuntar hacia el espacio de trabajo del robot. Por lo que se construyó la estructura que se muestra en las Figuras 162 y 163, que permite regular en dos direcciones la posición de la cámara web. En esta misma estructura se coloca la estructura de los indicadores LED del estado del robot.

Figura 162

Estructura de soporte de la cámara y reflector



Figura 163

Soporte para la estructura de los indicadores LED



Se construyó una mesa para que soporte la carga del robot y la caja de control donde se encuentra el circuito eléctrico del robot, y la estructura para la cámara web y sistema de iluminación de esta. La mesa y el tablero que se utiliza de soporte son de color blanco, de tal manera que facilite la detección del espacio de trabajo del robot por medio de visión artificial. En la Figura 164 se muestra al robot ya situado en la mesa de trabajo.

Figura 164

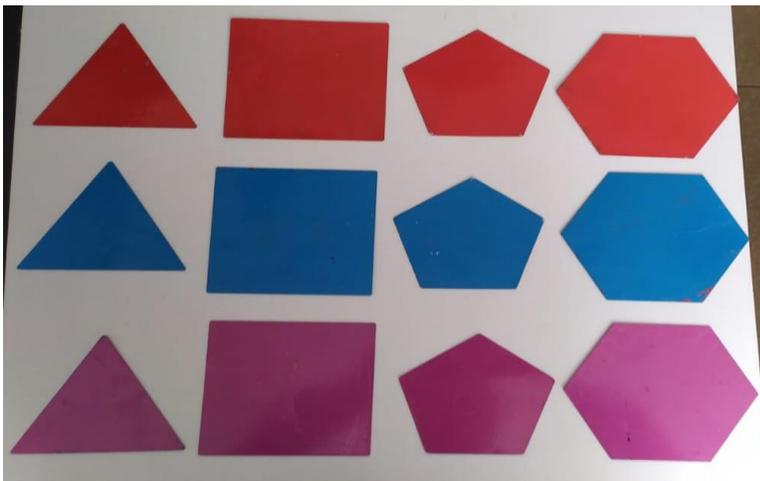
Robot SCARA situado en la mesa de trabajo.



Las piezas utilizadas para el proceso de identificación por visión artificial y recolección que se muestran en la Figura 165 se realizaron por medio de corte láser para y se las recubrió de pintura mate de diferentes colores.

Figura 165

Piezas utilizadas para el proceso de recolección de objetos



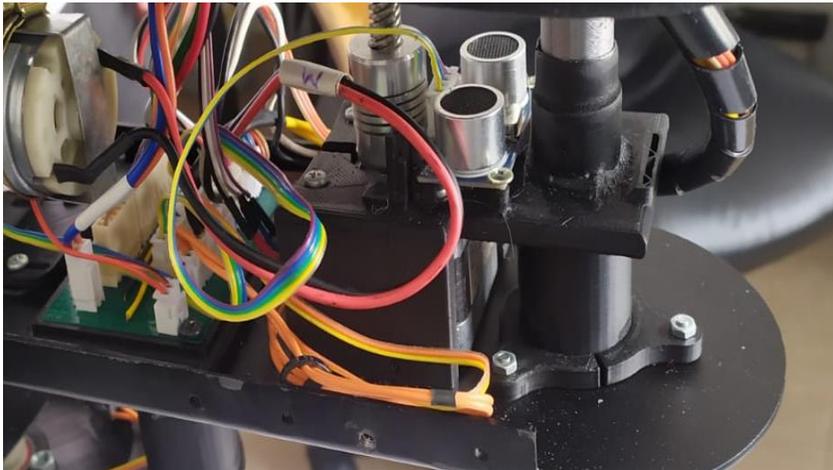
Circuito electrónico

El circuito electrónico del robot consta de dos placas principales, una alojada en el antebrazo del robot, que es la que permite repartir las señales de voltaje a los elementos que se encuentran en este eslabón, mientras que otra placa alojada en la caja de control es la que recibe la totalidad de las señales de todos los sensores y actuadores, y es la que permite las conexiones eléctricas con el microcontrolador Arduino.

La placa que se encuentra en el antebrazo cuenta con conectores de tipo Molex soldados, esto para facilitar la conexión y desconexión de cada elemento incluido en este eslabón. En la Figura 166 se muestra esta placa situada en el eslabón del antebrazo.

Figura 166

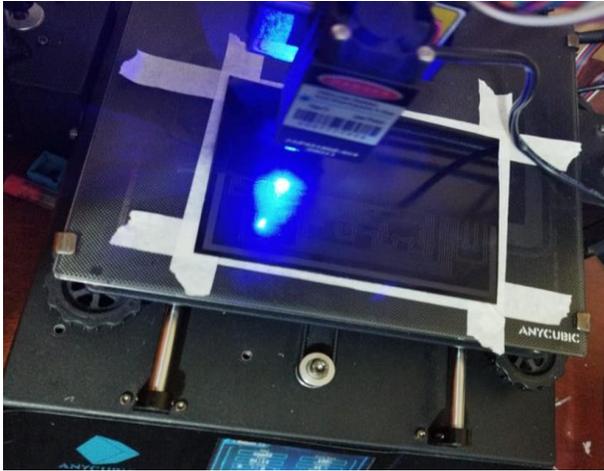
Placa electrónica del antebrazo



La placa principal que se encuentra alojada en la caja de control se diseñó y se confeccionó por medio de grabado por láser. En la Figura 167 se muestra el proceso de quemado de la placa diseñada.

Figura 167

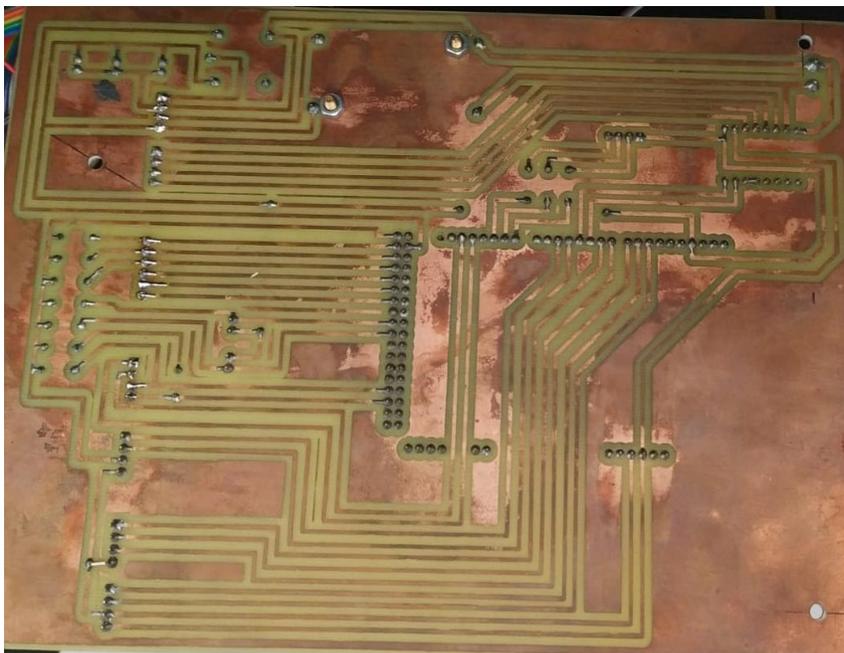
Proceso de grabado de la placa electrónica que se aloja en la caja de control



Terminado el proceso de grabar la placa, se le aplica perclorato férrico en agua caliente para que por medio de reacciones químicas el cobre que se desea retirar se desprenda de la placa de baquelita. La Figura 168 muestra la placa electrónica con el cobre excedente retirado.

Figura 168

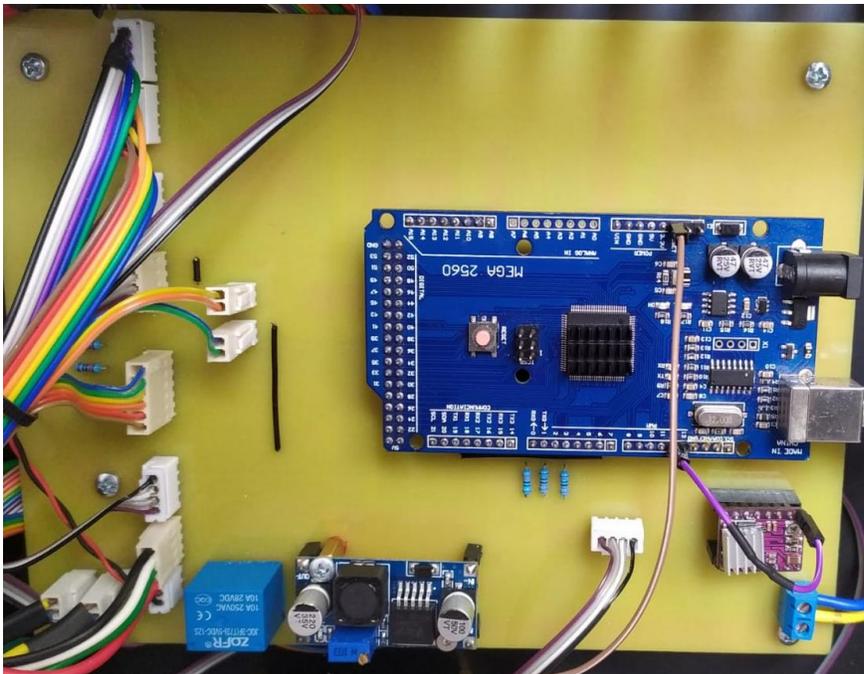
Placa electrónica después de retirar cobre excedente



Una vez que se ha terminado de retirar el cobre excedente de la placa, se realizan los agujeros para los elementos a colocar y se los une por medio de soldadura con estaño. Los elementos que se soldaron a la placa son conectores de tipo Molex, espadines tipo hembra para la conexión del microcontrolador, el relé para la activación del electroimán, un módulo step down para obtener señal de 5V para los elementos que requieren este nivel de alimentación, el driver utilizado para el manejo del motor a pasos del grado de libertad Q4 y resistencias para los circuitos pull up de los interruptores fines de carrera. La Figura 169 muestra la placa electrónica con los elementos soldados y ubicados en sus posiciones.

Figura 169

Placa electrónica ubicada en la caja de control



La activación de los motores de los grados de libertad Q1 y Q2 se realizan por medio de módulos puente H IBT-2 mostrados en la Figura 170, los cuales se conectan a la alimentación de 12V y a salidas digitales del Arduino por medio de los conectores de la placa.

Figura 170

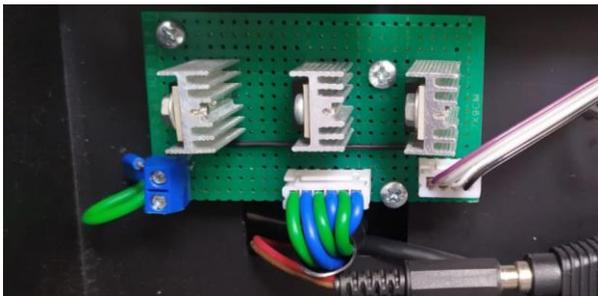
Drivers de los motores del brazo y antebrazo



Para que los indicadores LED se enciendan, requieren de una placa auxiliar que contiene un circuito que consta de tres transistores tipo MOSFET, mostrado en la Figura 171, que se activan con señales digitales que salen del Arduino.

Figura 171

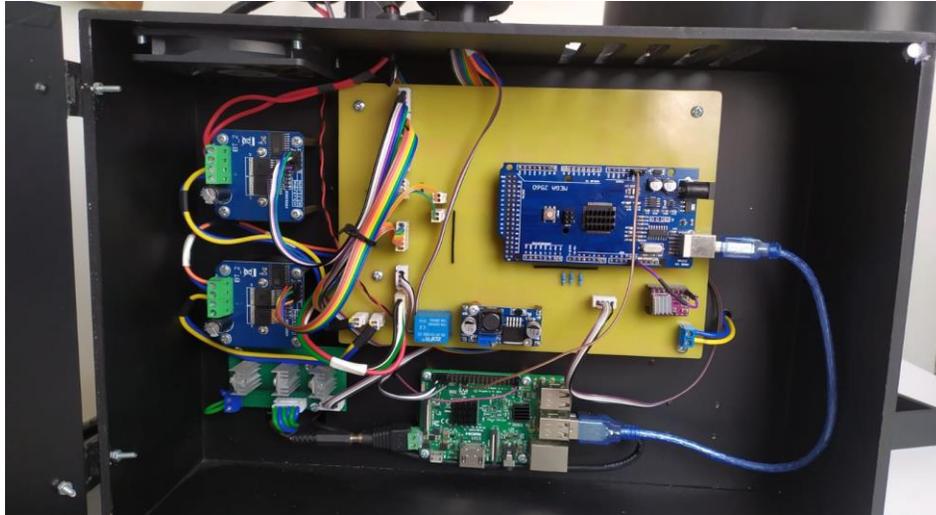
Circuito de los indicadores LED



Los elementos dentro de la caja de control se muestran en la Figura 172; como se puede apreciar, el microordenador se encuentra alojado en la caja y alimentado por la fuente de voltaje. Se puede apreciar que el microordenador alimenta al microcontrolador por medio de cable USB tipo B.

Figura 172

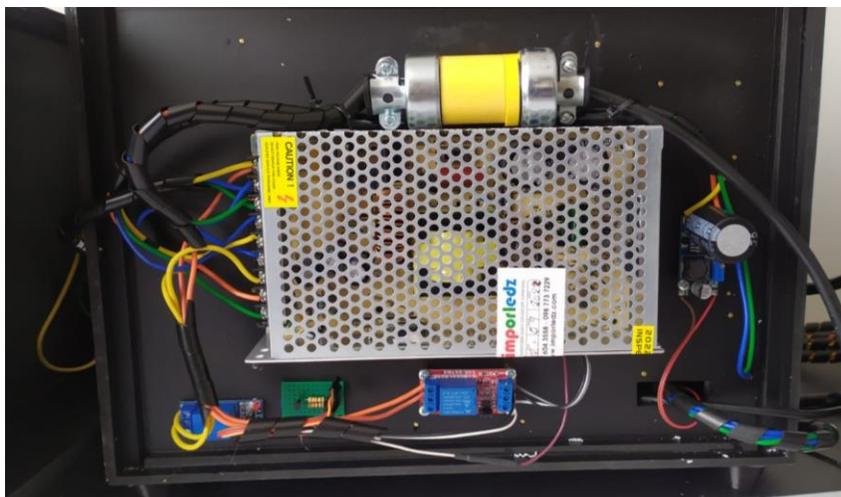
Interior de la caja de control



En la parte posterior, mostrada en la Figura 173, se encuentran las conexiones eléctricas que se reparten hacia los indicadores, hacia el microordenador y al reflector del sistema de visión artificial. La fuente de 12 V alimenta a todos los elementos que requieren corriente directa y el reflector se alimenta con corriente alterna de 120 V. Adicionalmente se observan los relés que controlan el encendido y apagado del robot y del reflector.

Figura 173

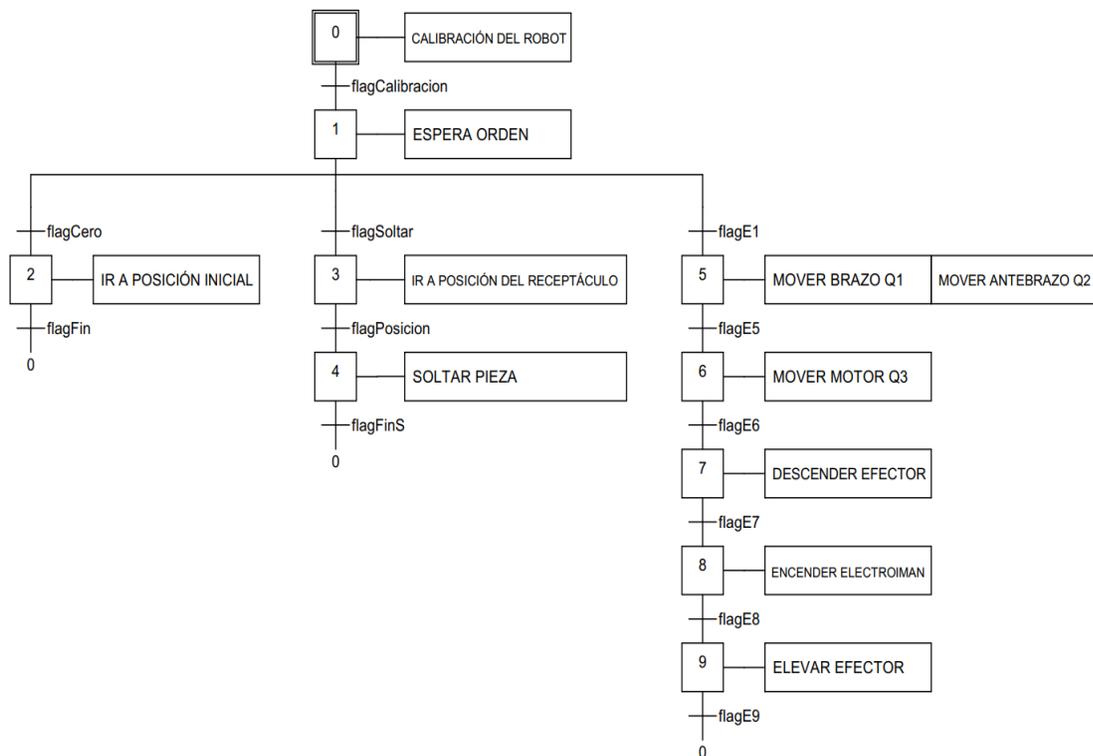
Parte posterior de la caja de control



La programación del microcontrolador se realizó en base al diagrama GRAFCET de nivel 1 que se muestra en la Figura 174, que simplifica las tareas que debe realizar el Arduino para el correcto funcionamiento del robot.

Figura 174

Diagrama GRAFCET del funcionamiento del robot SCARA



En base al GRAFCET se realiza la programación del microcontrolador Arduino; el algoritmo implementado se muestra de forma detallada en el diagrama de flujo de la Figura 175, y fragmentos de código son mostrados en las Figuras desde la 176 hasta la 184.

Figura 175

Diagrama de flujo de la programación del microcontrolador Arduino

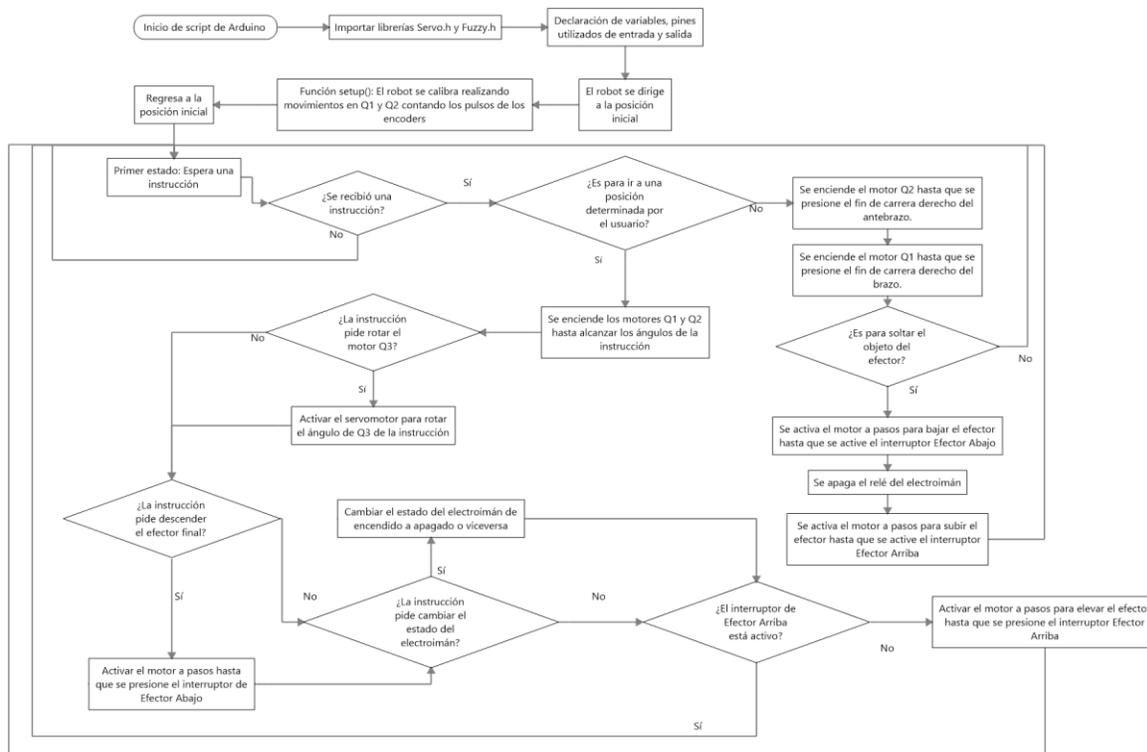


Figura 176

Estado 0 de operación del robot SCARA

```

if (E0) {
  digitalWrite(LEDv, LOW);
  digitalWrite(LEDr, LOW);
  digitalWrite(LEDa, HIGH);
  flagListo = true;
  if (Serial.available()) {
    flagListo = false;
    digitalWrite(LEDa, LOW);
    digitalWrite(LEDr, LOW);
    String mensaje = Serial.readStringUntil('\n');
    separarMensaje(mensaje);
    contA = 0;
    contB = 0;

    if (encenar == 1) { ...
    } else if (encenar == 2) { ...
    } else if (encenar == 3) { ...
    } else { ...
  }
}

```

Figura 177*Estado 1 de operación del robot SCARA*

```

if (E1) {
  if (abs(contB - abs(angBNR)) >= 2 && !flag11) {
    //Control fuzzy de velocidad
    fuzzy->setInput(1, abs(contB - abs(angBNR)));
    fuzzy->fuzzify();
    float output = fuzzy->defuzzify(1);
    int PWMBAux = 0.9*output; //NM: 0.87
    if (angBNR > 0) {...
    } else if (angBNR < 0) {...
    } else {...
  } else {...
  if (abs(contA - abs(angANR)) >= 2.5 && !flag12) {...
  } else {...
  flag1 = flag11 * flag12;
}

```

Figura 178*Estado 2 de operación del robot SCARA*

```

if (E2) {
>   if (angANR > 0) {...
>   if (angANR < 0) {...
>   if (angBNR > 0) {...
>   if (angBNR < 0) {...
  contA = 0;
  contB = 0;
>   if (angSNR != 0) {...
    svMotor.write(angSNR);
    motorServo = 0;
    flag2 = true;
  }
}

```

Figura 179*Estado 3 de operación del robot SCARA*

```

if (E3) {
  if (stepNR == 1) {
    int i = 0; // Contador de iteraciones del bucle while
    int tmp = 550; //Tiempo entre pulsos
    digitalWrite(dir, LOW);
    digitalWrite(freno, LOW);
    motorPasos = 1;
    lecturas();
  >   while (digitalRead(FCB)) {...
    motorPasos = 0;
    digitalWrite(paso, LOW);
    digitalWrite(freno, HIGH);
    flag3 = true;
  >   } else {...
    calculoDist();
    pulsos = 0;
  }
}

```

Figura 180*Estado 4 de operación del robot SCARA*

```

if (E4) {
  lecturas();
  delay(200);
  if (eimanNR == 1) {
    digitalWrite(eiman, HIGH);
  }
  if (eimanNR == 0) {
    digitalWrite(eiman, LOW);
  }
  flag4 = true;
}

```

Figura 181*Estado 5 de operación del robot SCARA*

```

if (E5) {
  if (stepNR == 1) {
    int i = 0; // Contador de iteraciones del bucle while
    int tmp = 500; //Tiempo entre pulsos
    digitalWrite(dir, HIGH);
    digitalWrite(freno, LOW);
    motorPasos = 1;
    lecturas();
    > while (digitalRead(FCA)) {...
    motorPasos = 0;
    flag5 = true;
    digitalWrite(paso, LOW);
    digitalWrite(freno, HIGH);
  } else {
    digitalWrite(freno, HIGH);
    motorPasos = 0;
    flag5 = true;
  }
  pulsos = 0;
  calculoDist(); //Función para calcular distancia con el sensor de ultrasonido
}
}

```

Figura 182

Función para poner en la posición inicial al robot

```

void cero() {
  //El electroimán sube
  digitalWrite(freno, LOW);
  digitalWrite(dir, HIGH);
  > while (digitalRead(FCA)) {...
    pulsos = 0;
    digitalWrite(freno, HIGH);
    motorPasos = 0;
    //El antebrazo recorre hacia la izquierda
  > while (digitalRead(FCAD)) {...
    motorA = 0;
    analogWrite(PHApwm1, 0);
    analogWrite(PHApwm2, 0);
    //El brazo recorre hacia la derecha
  > while (digitalRead(FCBD)) {...
    motorB = 0;
    analogWrite(PHBpwm2, 0);
    analogWrite(PHBpwm1, 0);
    //Se encera el ángulo del servomotor
    motorServo = 1;
    svMotor.write(0);
    delay(500);
    motorServo = 0;
    //Se encera todos los contadores
    delay(500);
    contA = 0;
    contB = 0;
    angA = 0; //Angulo actual del Brazo
    angB = 0; //Angulo actual del Antebrazo
    t = 0; //Timepo que demora en llegar el eco
    d = 0; //Distancia en centimetros
    digitalWrite(freno, HIGH);
  }
}

```

Figura 183

Función que separa el mensaje recibido por el puerto serial

```

void separarMensaje(String str) {
  int t1 = (str.indexOf('X'));
  angBNR = (str.substring(0, t1)).toInt();

  int t2 = (str.indexOf('X', t1 + 1));
  angANR = (str.substring(t1 + 1, t2)).toInt();

  int t3 = (str.indexOf('X', t2 + 1));
  angSNR = (str.substring(t2 + 1, t3)).toInt();

  int t4 = (str.indexOf('X', t3 + 1));
  stepNR = (str.substring(t3 + 1, t4)).toInt();

  int t5 = (str.indexOf('X', t4 + 1));
  eimanNR = (str.substring(t4 + 1, t5)).toInt();

  int t6 = (str.indexOf('X', t5 + 1));
  encerar = (str.substring(t5 + 1, t6)).toInt();
}

```

Figura 184

Función que forma el mensaje con las lecturas de los sensores y lo envía por el puerto serial

```

void lecturas() {
  Serial.print(angB); //Angulo antebrazo
  Serial.print('\t');
  Serial.print(angA); //Angulo brazo
  Serial.print('\t');
  Serial.print(d); //Distancia ultrasonido
  Serial.print('\t');
  Serial.print(svMotor.read()); //Angulo servo
  Serial.print('\t');
  Serial.print(digitalRead(eiman)); //Estado electroiman
  Serial.print('\t');
  Serial.print(digitalRead(FCAD)); //FC Antebrazo Derecha
  Serial.print('\t');
  Serial.print(digitalRead(FCAI)); //FC Antebrazo Izquierda
  Serial.print('\t');
  Serial.print(digitalRead(FCBD)); //FC Brazo Derecha
  Serial.print('\t');
  Serial.print(digitalRead(FCBI)); //FC Brazo Izquierda
  Serial.print('\t');
  Serial.print(digitalRead(FCA)); //FC Motor a pasos arriba
  Serial.print('\t');
  Serial.print(digitalRead(FCB)); //FC Motor a pasos abajo (Detecta piso)
  Serial.print('\t');
  Serial.print(digitalRead(FCEI)); //FC Electroiman (Detecta pieza)
  Serial.print('\t');
  Serial.print(motorB);
  Serial.print('\t');
  Serial.print(motorA);
  Serial.print('\t');
  Serial.print(motorServo);
  Serial.print('\t');
  Serial.print(motorPasos);
  Serial.print('\t');
  Serial.println(flagListo);
}

```

Subsistema de visión artificial

El subsistema de visión artificial se implementó de tal manera que el usuario desde la interfaz gráfica envía la orden de ejecutar el script alojado en el microordenador, el cual toma una imagen previamente capturada por medio de la cámara web, y la procesa detectando el

espacio de trabajo e identificando los objetos que se encuentren en este espacio por forma y color. Físicamente cuenta con la cámara web y el reflector para mejorar la iluminación, esto se mencionó en el apartado del subsistema mecánico.

La programación del script que identifica objetos se realizó en lenguaje Python, utilizando la librería de procesamiento de imágenes OpenCV, que está enfocada a su aplicación dentro de tareas de visión artificial, y que ofrece métodos para la implementación de filtros, máscaras, umbralización, binarización de imágenes, entre otras operaciones.

El código implementado toma una imagen previamente capturada y le aplica un filtro de tipo HSV para posteriormente realizar detección de contornos por medio del método de Canny. La primera búsqueda de contornos tiene el objetivo de encontrar el espacio de trabajo del robot, mientras que una segunda búsqueda tiene la finalidad de encontrar los objetos que se encuentren en la superficie del espacio de trabajo. Si se detecta objetos, el script los identifica en base a las formas y colores que se encuentran programadas; para formas se pueden detectar triángulos, cuadrados, pentágonos y hexágonos, mientras que en colores puede detectar rojo, azul, violeta y amarillo.

Si se detecta una figura de color amarillo, significa que se encontró el centro de coordenadas del espacio de trabajo, que físicamente es la parte superior del eje del brazo. Para las demás figuras detectadas se calcula la distancia hacia el origen de coordenadas desde sus centros y se imprime un arreglo que contiene coordenadas en X, Y, forma y color. De forma más detallada, en el diagrama de flujo que se muestra en la Figura 185 se encuentra el algoritmo para la detección de objetos por visión artificial, mientras que en las Figuras desde la 186 hasta la 190 se muestran fragmentos del código.

Figura 185

Diagrama de flujo de la programación del algoritmo de visión artificial

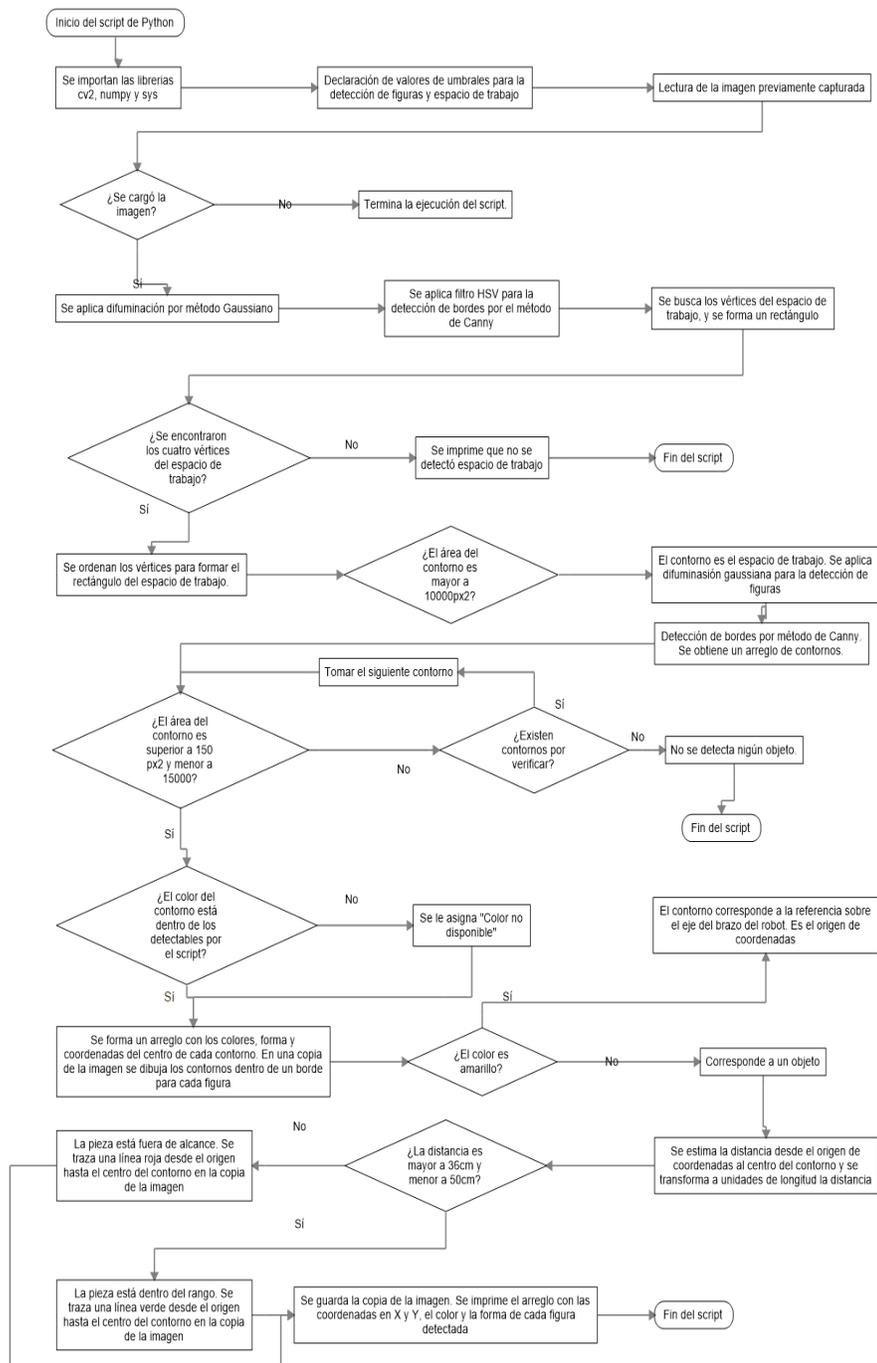


Figura 186*Umbral en espacio HSV para filtro del espacio de trabajo y figuras*

```
# Definicion de los umbrales para las mascarar que se utilizaran para la deteccion de espacio de trabajo y figuras
preUmbralBajo = (int(0 * 179), int(0 * 255), int(0.697 * 255))
preUmbralAlto = (int(1 * 179), int(0.122 * 255), int(1 * 255))

umbralBajo1 = (int(0 * 179), int(0.192 * 255), int(0.265 * 255))
umbralAlto1 = (int(0.191 * 179), int(1 * 255), int(1 * 255))
umbralBajo2 = (int(0.497 * 179), int(0.192 * 255), int(0.265 * 255))
umbralAlto2 = (int(1 * 179), int(1 * 255), int(1 * 255))
```

Figura 187*Preprocesamiento y binarización por el método de Canny*

```
# Se realiza un filtro y preprocesado de la imagen previo a la umbralizacion para detectar el espacio de trabajo
img_brd = cv2.GaussianBlur(img_brd, (9, 9), cv2.BORDER_CONSTANT)
img_brd_hsv = cv2.cvtColor(img_brd, cv2.COLOR_BGR2HSV)
maskPreUmbral = cv2.inRange(img_brd_hsv, preUmbralBajo, preUmbralAlto)
res_brd = cv2.bitwise_and(img, img, mask=maskPreUmbral)
gray_brd = cv2.cvtColor(res_brd, cv2.COLOR_BGR2GRAY)

# Se utiliza el metodo de Canny implementado en OpenCV para la deteccion de bordes
canny_brd = cv2.Canny(gray_brd, 50, 50)
canny_brd = cv2.dilate(canny_brd, None, iterations=2)
canny_brd = cv2.erode(canny_brd, None, iterations=1)
cnts_brd, _ = cv2.findContours(canny_brd, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts_brd = sorted(cnts_brd, key=cv2.contourArea, reverse=True)[:1]
```

Figura 188*Transformación de la imagen para normalizar en el espacio de trabajo*

```
if area_brd > 10000:
    pts1 = np.float32([crnSL, crnSR, crnIL, crnIR])
    pts2 = np.float32([[0, 0], [964, 0], [0, 434], [964, 434]])

    # Se realiza una transformacion de la imagen para corregir la distorsion en el espacio de trabajo
    M = cv2.getPerspectiveTransform(pts1, pts2)
    dst = cv2.warpPerspective(img_brd, M, (964, 434))
```

Figura 189

Detección y clasificación de figuras válidas en el array de trabajo

```

for c in cnts:
    count += 1
    area = cv2.contourArea(c)
    # Se descartan figuras con el area muy pequenas a manera de filtro en la deteccion
    if 150 < area < 15000:
        # Se dibuja el contorno que se encontro, junto con el rectangulo que lo contiene
        x, y, w, h = cv2.boundingRect(c)
        centroX = x + int(w / 2)
        centroY = y + int(h / 2)
        imAux = np.zeros(img.shape[:2], dtype="uint8")
        imAux = cv2.drawContours(imAux, [c], -1, 255, -1)
        maskHSV = cv2.bitwise_and(img_hsv, img_hsv, mask=imAux)
        name = figName(c, w, h)
        color = figColor(maskHSV)
        # Se aplica una condicional para descartar colores no definidos previamente
        if color != 'Color no disponible':
            # Se construye un vector que contiene las coordenadas, colores y formas encontradas en la imagen
            coords.append([color, name, centroX, centroY])
            nameColor = name + ' ' + color
            # Fisicamente, el centro del color amarillo dentro de la imagen coincide con el centro de coordenadas
            # del robot, por lo que se dibuja lineas guia
            if color == 'Amarillo':
                cv2.line(img, (centroX, 0), (centroX, height), color=[0, 0, 0], thickness=2)
                cv2.line(img, (0, centroY), (width, centroY), color=[0, 0, 0], thickness=2)
                cv2.putText(img, nameColor, (x, y - 5), font, 0.5, color=[255, 0, 0], thickness=1)
                cv2.rectangle(img, (x, y), (x + w, y + h), color=[0, 255, 0], thickness=1)
                cv2.drawContours(img, cnts, count, color=[0, 255, 0], thickness=1)

```

Figura 190

Cálculo de la distancia de los objetos de prueba al centro del robot SCARA

```

for i in range(len(coords)):
    # Se verifica si se encontro el centro (color amarillo)
    if coords[i][0] == 'Amarillo':
        # Se traza una linea representando el vector del origen de coordenadas hacia la pieza
        if coords[i][0] != 'Amarillo':
            distX = str(round((((coords[i][2] - coords[0][2]) * 120) / width), 2))
            distY = str(round((((coords[0][3] - coords[i][3]) * 60) / height), 2))
            distVect = np.sqrt(float(distX) ** 2 + float(distY) ** 2)
            # Se verifica que la distancia hacia la pieza sea alcanzable para los limites del robot
            if abs(distVect) > 50 or abs(distVect) < 36:
                else:
                    dist = '\t-\t-\t-\t-'

```

Arquitectura de comunicación

La arquitectura de comunicación se implementó de tal manera que el robot sea operable desde cualquier localidad que cuente con conexión a internet. Se utilizaron los servicios web de la plataforma Amazon Web Services para lograr este objetivo. Para la utilización de un servidor en la nube se utilizó el servicio Amazon EC2, que permitió crear una instancia, similar a una

máquina virtual, con sistema operativo Linux Ubuntu Server 22.04 LTS, como se muestra en las Figuras 191 y 192.

Figura 191

Menú principal del servicio Amazon EC2

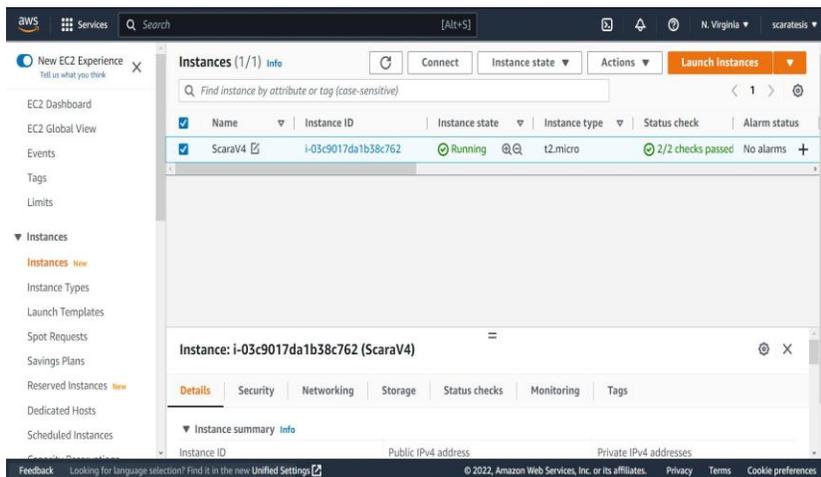
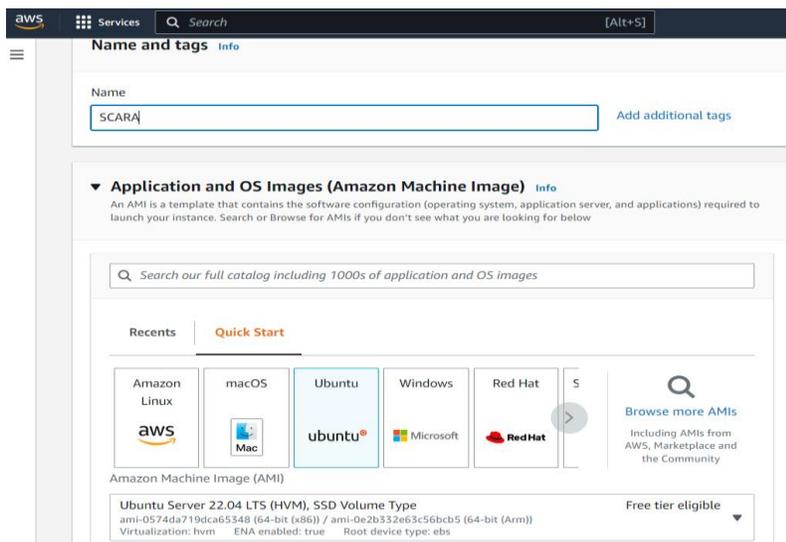


Figura 192

Menú de creación de instancias de Amazon EC2



Creada la instancia, Amazon le asigna una dirección IP pública con la cual se puede acceder por medio de software que permite comunicación por protocolo Secure Shell Protocol

(SSH), para realizar la instalación de Node - RED. Una vez instalado Node - RED, se activó la opción para que se ejecute como servicio, es decir, inicie Node - RED cuando se encienda la instancia. El proceso de instalación de Node – RED se muestra en la Figura 193.

Figura 193

Instalación del Node – RED en la instancia de Amazon EC2

```

ubuntu@ip-172-31-91-124: ~
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Fri Dec 16 15:36:41 UTC 2022

System load:  0.14697265625   Processes:           99
Usage of /:   48.5% of 7.57GB   Users logged in:    0
Memory usage: 64%             IPv4 address for eth0: 172.31.91.124
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

16 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Fri Dec 16 15:34:34 2022 from 181.199.46.142
ubuntu@ip-172-31-91-124:~$ node -v
v12.22.9
ubuntu@ip-172-31-91-124:~$ sudo systemctl enable nodered.service

```

La dirección IP pública asignada cambia cada vez que la instancia es apagada y encendida nuevamente, por lo que fue necesario utilizar la herramienta de IP Elástica como se muestra en la Figura 194, de tal manera que se asocia la instancia creada a una IP pública fija, que redirecciona a la IP pública variable de la instancia; de esta forma se puede acceder a la instancia utilizando la IP fija.

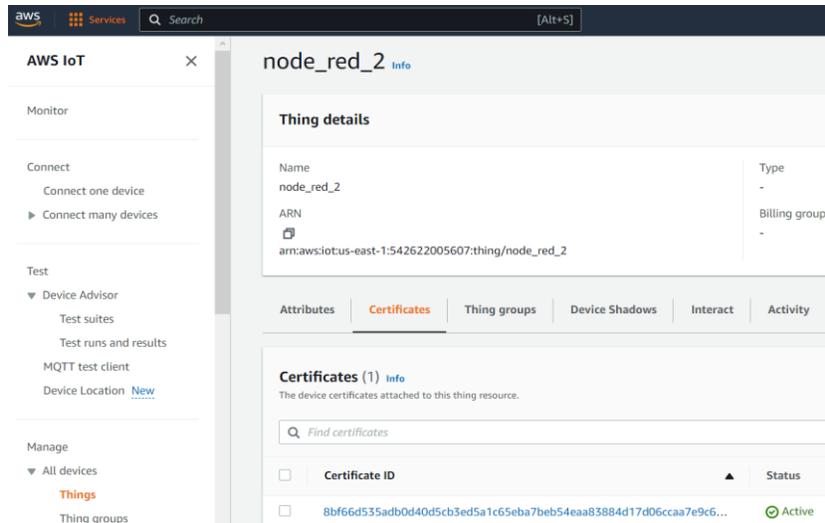
Figura 194*Asignación de IP elástica a la instancia*

44.212.228.187 Actions ▾ Associate Elastic IP address

Summary

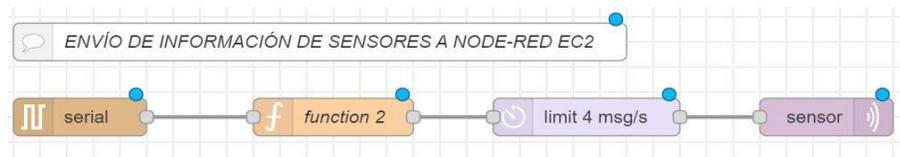
Allocated IPv4 address 44.212.228.187	Type Public IP	Allocation ID eipalloc-070ae81fbb48baf51	Reverse DNS record -
Association ID eipassoc-0c521f58a35c8c4b6	Scope VPC	Associated instance ID i-03c9017da1b38c762	Private IP address 172.31.91.124
Network interface ID eni-00897db9beae038a8	Network interface owner account ID 542622005607	Public DNS ec2-44-212-228-187.compute-1.amazonaws.com	NAT Gateway ID -
Address pool Amazon	Network Border Group us-east-1		

Una vez creada la instancia, asignada la IP pública fija e instalado Node - RED como servicio, se configura el servicio de Amazon IoT Core para enviar y recibir información por medio del protocolo MQTT, como se muestra en la Figura 195. Al realizar las configuraciones se obtienen un conjunto de certificados para su utilización en Node - RED, que le indican que broker está siendo utilizado para el manejo de la información. Estos mismos certificados son utilizados dentro de la programación de la cámara ESP32-CAM para el envío de la señal de video.

Figura 195*Configuración del servicio Amazon IoT Core*

La información de Node - RED debe enviarse desde y hacia al microordenador, por lo que se instaló el sistema operativo Linux Pi OS (antes Raspbian) en la Raspberry Pi 3B+, y se instaló Node - RED como servicio para que se ejecute cuando se encienda el dispositivo.

Con los dos servicios de Node - RED configurados, se realizó la programación en el flow de la Raspberry, que cuenta con cuatro partes: envío de información de sensores, recepción de instrucciones, visión artificial y control general. El envío de información permite tomar las lecturas que envía periódicamente el Arduino por el puerto serial y enviarlas por un tópico de MQTT, con un límite de 4 mensajes por segundo; que se muestra en la Figura 196.

Figura 196*Nodos de envío de información por MQTT*

El tópico “sensor” permite la salida del mensaje por MQTT. Este nodo fue configurado con los certificados que otorgó el servicio Amazon IoT Core, como se muestra en las Figuras

197 y 198. Se crea un perfil de configuración para el broker MQTT y es el que se utiliza en todos los nodos de entrada y salida de MQTT dentro de la Raspberry Pi y de la instancia de Amazon. Por nomenclatura, los nodos de entrada y salida de MQTT llevan el nombre del tópic que utilizan.

Figura 197

Configuración del broker MQTT en Node - RED

The screenshot shows the 'Edit mqtt out node' configuration window for an MQTT broker node. The window has a title bar 'Edit mqtt out node > Edit mqtt-broker node' and buttons for 'Delete', 'Cancel', and 'Update'. Below the title bar is a 'Properties' section with a gear icon and a document icon. The main configuration area is divided into three tabs: 'Connection', 'Security', and 'Messages'. The 'Connection' tab is active and contains the following settings:

- Name:** AWS
- Server:** a1e0eety3s6577-ats.iot.us-east-1.an
- Port:** 8883
- Enable secure (SSL/TLS) connection
- TLS Configuration:** TLS configuration (dropdown menu)
- Client ID:** Leave blank for auto generated
- Keep alive time (s): 60
- Use clean session
- Use legacy MQTT 3.1 support

Figura 198

Asignación de las credenciales de Amazon IoT Core en Node - RED

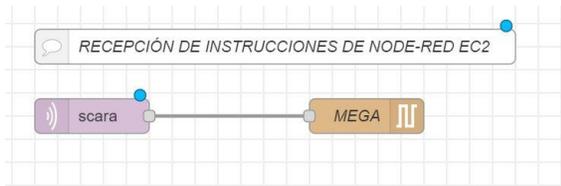
The screenshot shows the 'Edit mqtt out node > Edit mqtt-broker node > Edit tls-config node' configuration window. The window has a title bar 'Edit mqtt out node > Edit mqtt-broker node > Edit tls-config node' and buttons for 'Delete', 'Cancel', and 'Update'. Below the title bar is a 'Properties' section with a gear icon. The main configuration area contains the following settings:

- Use key and certificates from local files
- Certificate:** Upload 037bcb2d05480158d9a6a23d31db66fc8d98f20f479c7dedf59619f24640d2d4-certificate.p...
- Private Key:** Upload 037bcb2d05480158d9a6a23d31db66fc8d98f20f479c7dedf59619f24640d2d4-private.pem...
- Passphrase:** private key passphrase (optional)
- CA Certificate:** Upload AmazonRootCA1.pem
- Verify server certificate
- Server Name:** for use with SNI

Para la recepción de instrucciones mostrada en la Figura 199, la entrada es el mensaje del tópic “scara”, a través del cual se envían cadenas de caracteres con los valores para cada uno de los grados de libertad y si es requerida o no la activación del electroimán.

Figura 199

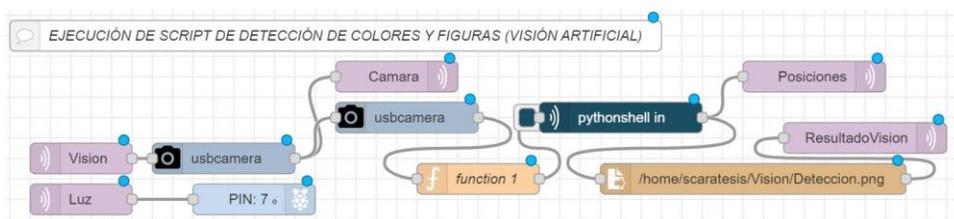
Nodos utilizados para la recepción de instrucciones



Para la visión artificial, la programación visual realizada es la que se muestra en la Figura 200; la entrada es una señal que le indica los nodos “usbcamera” que utilicen la cámara web para tomar una foto del espacio de trabajo de la celda. El primer nodo toma una imagen y la guarda en baja resolución, y la envía a través del tópic “Camara”. El segundo nodo la toma con alta calidad, se guarda y es la que se utiliza para la detección de figuras; la ejecución del script de Python se realiza con el nodo “pythonshell in”, que, al terminar de ejecutarse, guarda la copia de la imagen que se menciona en la implementación de la visión artificial, y la envía a través del tópic “ResultadoVision”, mientras que el arreglo de posiciones con los colores y formas identificadas, se envía por el nodo “Posiciones”. El nodo “Luz” permite activar una de las salidas digitales de la Raspberry Pi para encender o apagar el reflector.

Figura 200

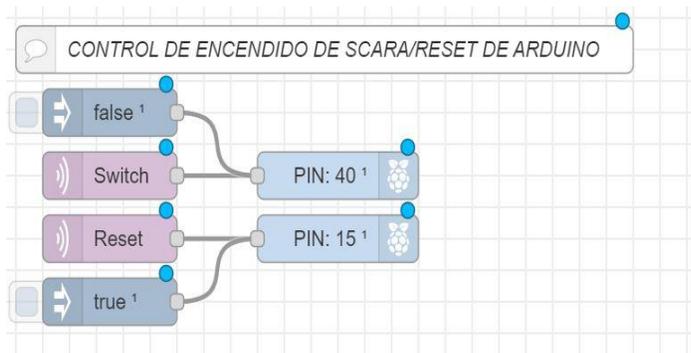
Nodos utilizados para el uso de la visión artificial



El control general mostrado en la Figura 201 permite encender o apagar el robot SCARA, y adicionalmente reiniciar el Arduino. El nodo “Switch” es el que recibe la señal de si debe encenderse o apagarse el robot, y Reset, la señal de reinicio del Arduino enviada por el usuario.

Figura 201

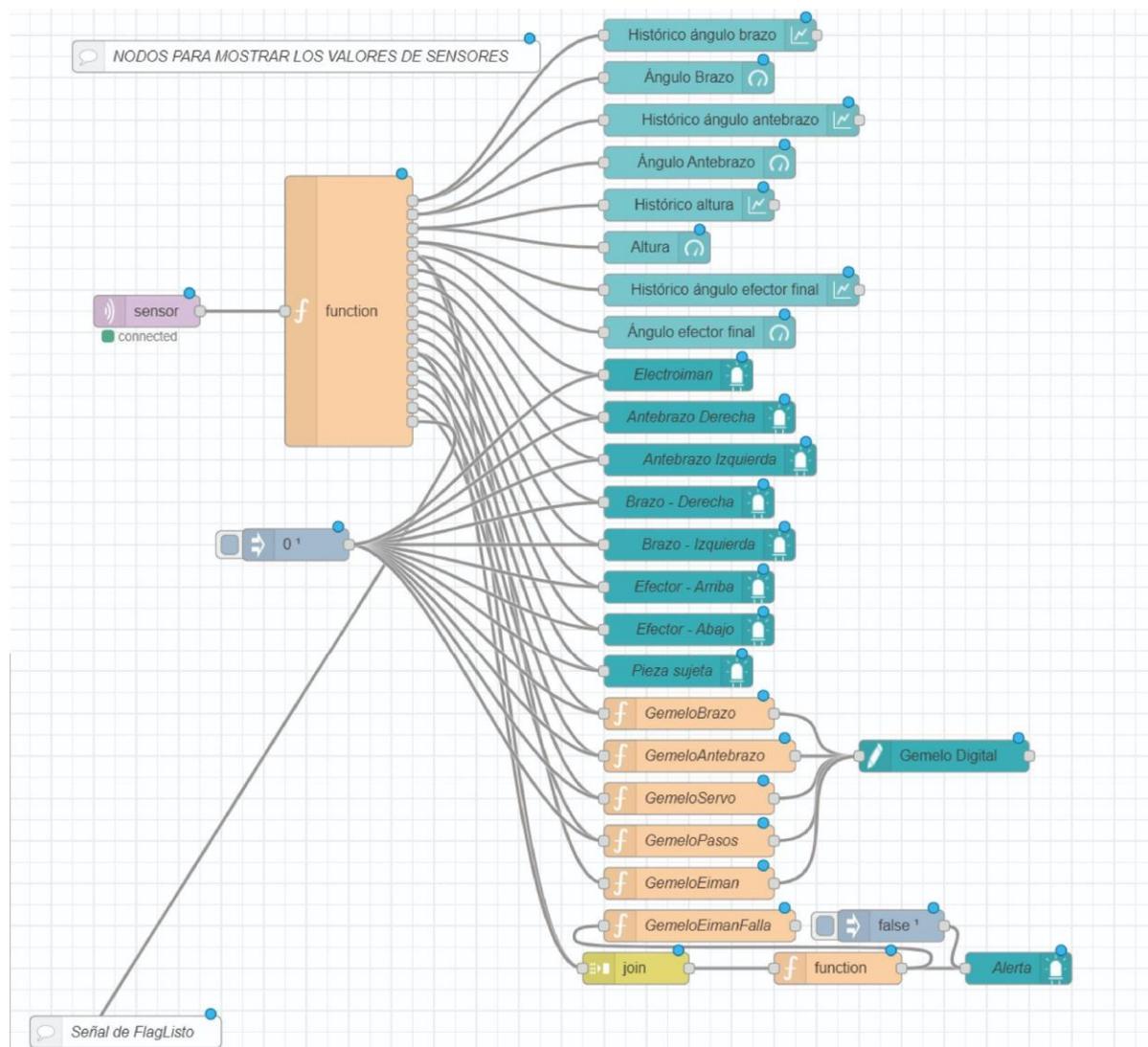
Nodos utilizados para el control general



En Node - RED de la instancia de Amazon se realizó la programación por bloques; el flujo construido se divide en cinco partes: recepción de información, envío de instrucciones, control manual, control general, y cámaras. La información llega por el nodo “sensor” como una cadena de caracteres, la cual es separada para obtener de forma individual los valores de cada sensor e interruptor y mostrarlo en la HMI, como se observa en la Figura 202. De igual manera, se toma las señales de si se encuentran encendidos o apagados los actuadores, para reflejar sus estados en el gemelo digital.

Figura 202

Nodos utilizados para la recepción de información

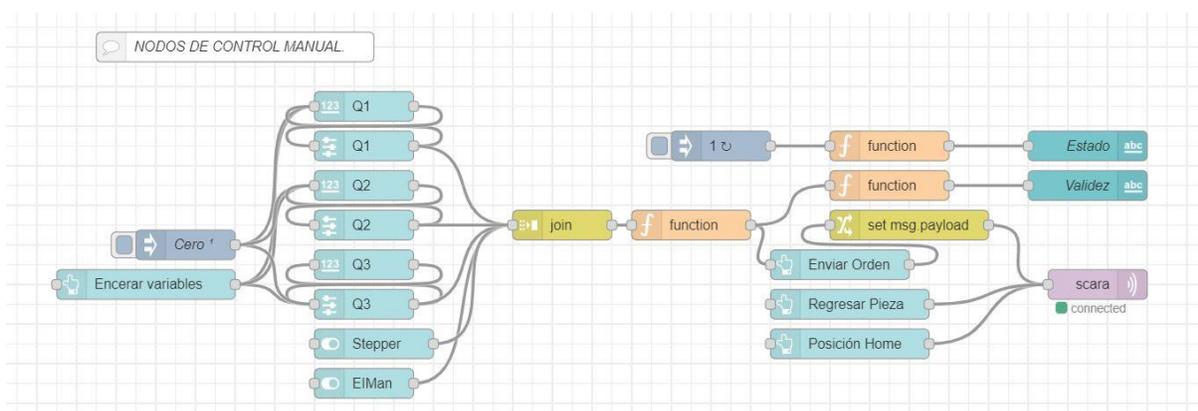


La parte de control manual mostrada en la Figura 203 permite enviar las instrucciones de movimiento hacia el microordenador. Los nodos Q1, Q2, Q3, Stepper y EIman son los utilizados para determinar cada valor de cada grado de libertad. Estos valores se unen en una cadena de valores y son enviados por el nodo “scara”.

Adicionalmente se pueden enviar instrucciones de Regresar pieza para soltar el objeto tomado en el recipiente de la posición inicial, y Posición Home que regresa el brazo a la posición inicial sin modificar el estado del electroimán. El nodo “Estado” refleja si el robot se encuentra con una instrucción pendiente, o es posible enviar una instrucción, y el nodo “Validez” indica si los valores que se desea enviar como instrucciones se encuentran en el espacio de trabajo válido del robot.

Figura 203

Nodos utilizados para el control manual



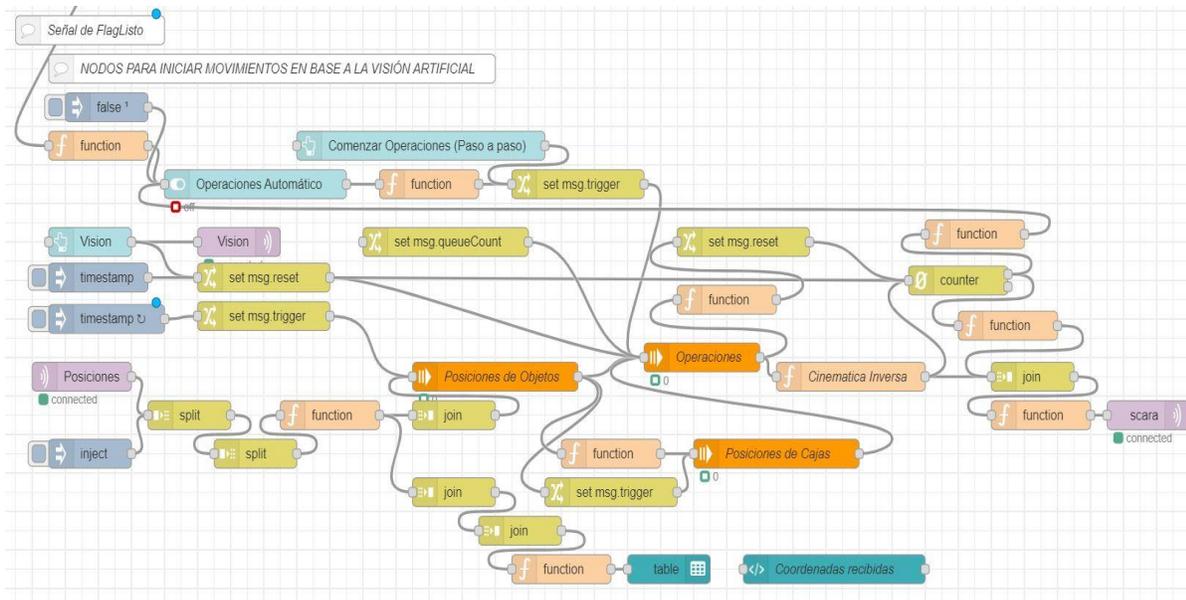
La parte de visión artificial mostrada en la Figura 204 envía la señal de ejecución del script de Python del microordenador por medio del nodo “Vision”. Se ejecuta el proceso en la Raspberry Pi y se reciben las posiciones en el nodo “Posiciones” a través del tópico “Posiciones”. Estas posiciones se deben colocar en una cola de espera, puesto que cada una representa una instrucción que el robot debe ejecutar; además que entre cada instrucción de posición recibida, debe ejecutarse una instrucción de soltar el objeto en el recipiente correspondiente. El comportamiento por defecto es que recoja las piezas en orden de colores, y las coloque en el recipiente ubicado en la posición inicial. Este comportamiento puede ser modificado al cambiar el código del nodo “function” que es anterior al nodo “scara”.

En base a las coordenadas recibidas de la visión artificial se debe calcular los ángulos que deben rotar los actuadores Q1 y Q2, por lo que se requiere las ecuaciones de la cinemática

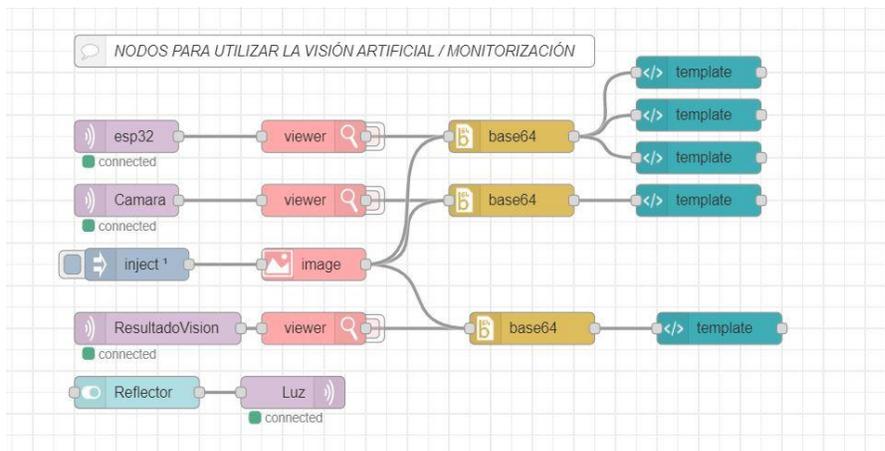
inversa del robot SCARA. Se implementan en el nodo “Cinemática Inversa”, el cual devuelve los valores de los ángulos en grados. La función implementa los cálculos desarrollados en el cálculo de cinemática inversa del capítulo 3.

Figura 204

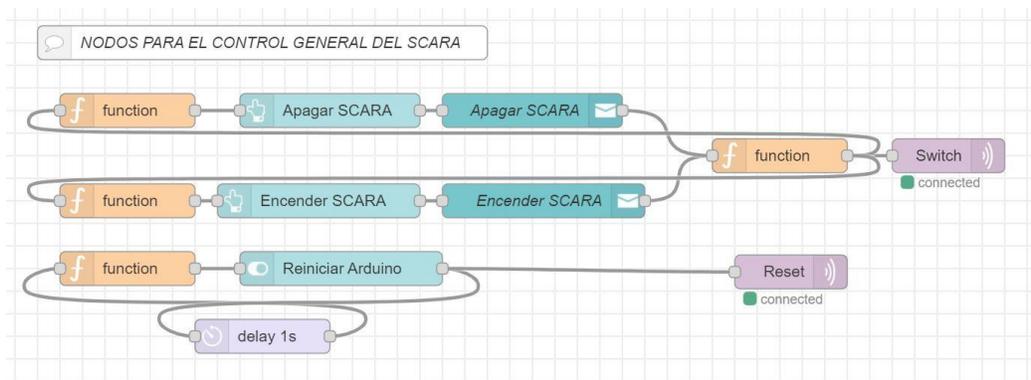
Nodos utilizados para el manejo de la visión artificial



La parte de cámaras mostrada en la Figura 205 contiene la programación por bloques que permite visualizar en tiempo real el funcionamiento del robot y poder mostrar la imagen capturada del proceso de visión artificial. El nodo “esp32” a través del tópico “esp32” recibe el video transmitido por la ESP32-CAM, y es publicado en las pantallas de la HMI. El nodo “Camara” recibe la foto que tomó el microordenador y la muestra en la pantalla de visión artificial de la HMI. El nodo “ResultadoVision” recibe la copia de la foto que creó el script de Python con los detalles de la identificación de figuras y la estimación de su distancia; esto se muestra en la pantalla de visión artificial de la HMI.

Figura 205*Nodos de cámaras de visión artificial y monitorización*

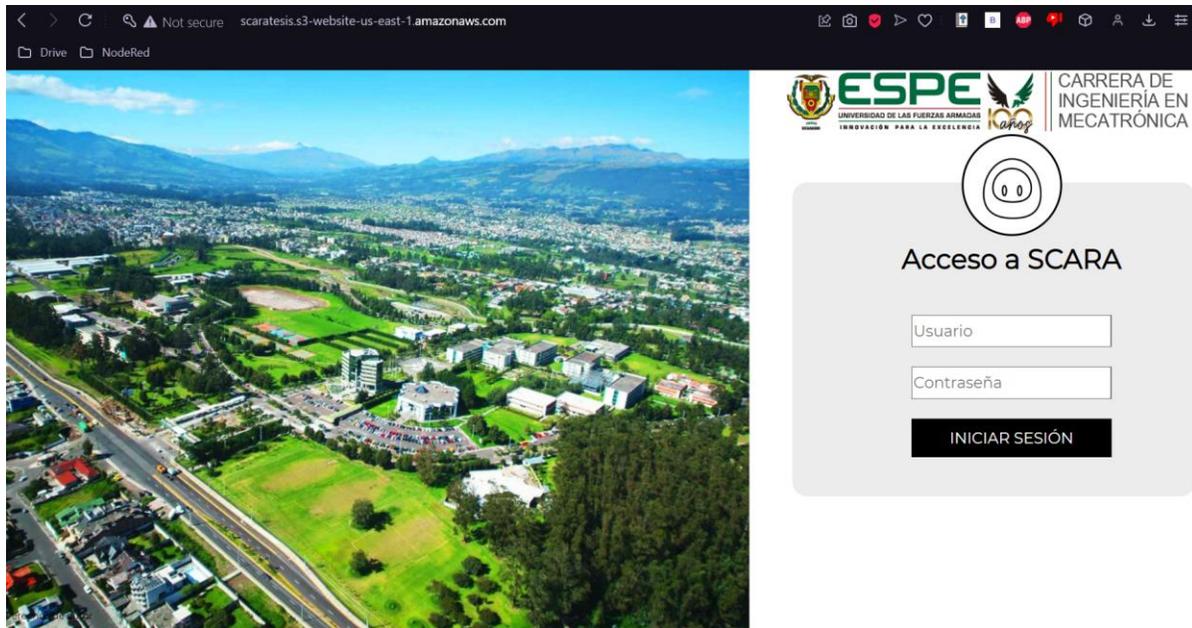
La parte de control general mostrada en la Figura 206 permite enviar la orden de encender el robot, apagar el robot o reiniciar el Arduino. La señal de encendido o apagado se envía por el nodo Switch, mientras que la de reiniciar el Arduino por medio del nodo Reset.

Figura 206*Nodos utilizados para el control general*

Para que el usuario pueda acceder a la HMI, debe utilizar una página de inicio de sesión mostrada en la Figura 207. Esta página está programada en HTML y cargada en el servicio web S3 de AWS, de tal manera que accediendo a la dirección web de la página e ingresando las credenciales, se redirecciona hacia Node - RED.

Figura 207

Página de inicio de sesión



Interfaz humano máquina

La interfaz humano-máquina (HMI) se implementó en Node - RED, utilizando el dashboard que ofrece para elementos visuales como indicadores, relojes analógicos, líneas para históricos, entre otros elementos. La interfaz cuenta con cuatro pantallas: Inicio, Monitorización del robot, Visión Artificial y Control manual.

La pantalla Inicio que se muestra en la Figura 208 sirve de introducción a la HMI, ofreciendo una descripción breve del funcionamiento del robot, las pantallas de las que se dispone y una breve explicación de las operaciones que se puede realizar en cada una de ellas.

Figura 208

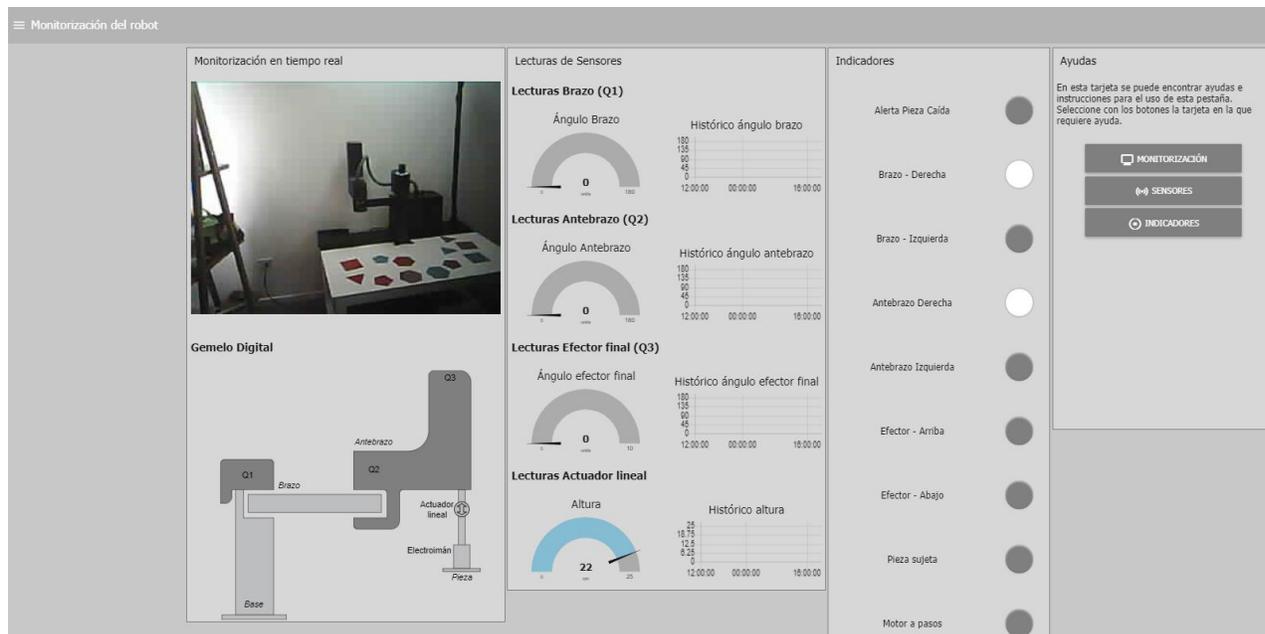
Pantalla de Inicio



La pantalla Monitorización del robot que se muestra en la Figura 209 permite visualizar los indicadores de los interruptores del robot, así como los valores en los que se encuentra cada uno de los grados de libertad, un gemelo digital que indica los actuadores que se encuentran activos y el video en tiempo real que transmite la ESP32-CAM. De forma auxiliar se colocan botones de ayuda para cada uno de los apartados de la pantalla.

Figura 209

Pantalla de Monitorización del robot

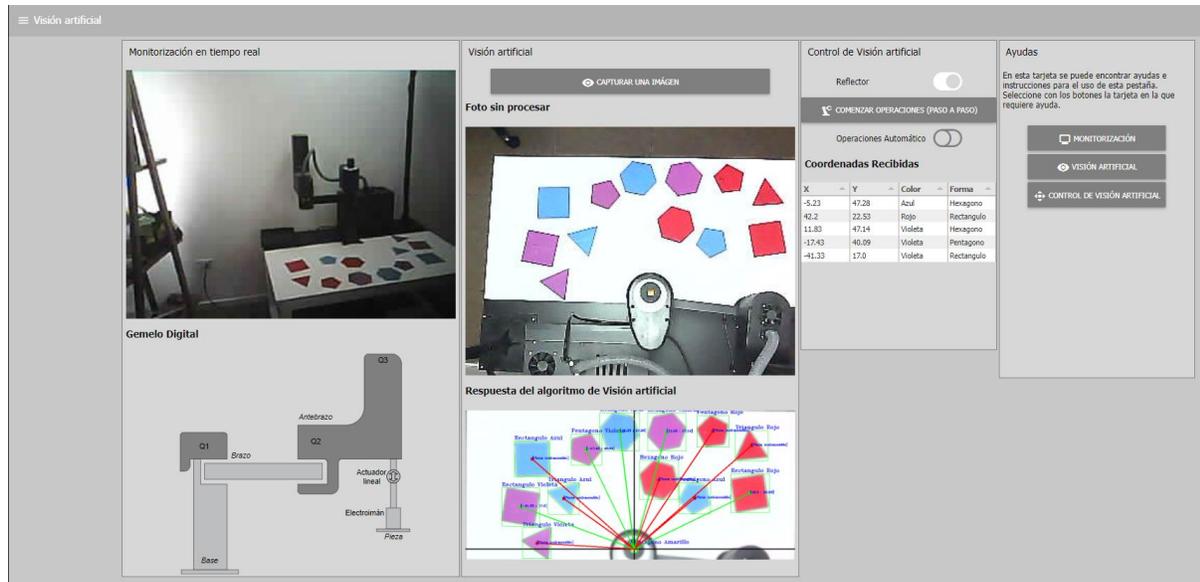


La pantalla de visión artificial mostrada en la Figura 210 permite monitorizar por medio del video transmitido en tiempo real, permite dar la orden de tomar la foto del espacio de trabajo y mostrar los resultados antes y después de la detección de objetos. Los objetos alcanzables se unen con una línea verde desde el centro de origen mientras que los que están fuera de rango, con una línea roja. En la tabla de coordenadas recibidas se muestran los valores en X y Y del centro de cada figura detectada y alcanzable junto a su forma y color.

Se puede ejecutar el algoritmo de recolección de objetos de forma manual, haciendo que por cada clic se ejecute una operación, o de forma automática. Esta pantalla también cuenta con la presencia del gemelo digital y el apartado de ayudas.

Figura 210

Pantalla de visión artificial



La pantalla de control manual que se muestra en la Figura 211 es la que permite enviar valores de forma manual al robot para posicionarlo. Se puede ingresar valores utilizando los sliders para los grados de libertad Q1, Q2 y Q3, y decidir si se va a activar el mecanismo elevador o el electroimán. Una vez establecidos los valores de cada grado de libertad, al presionar Enviar Orden la instrucción se envía al microordenador, y este se la pasa al microcontrolador vía serial. Si se presiona Posición Inicial, el robot se dirige a la posición Inicial con los ángulos Q1 y Q2 igual a 0.

Al presionar el botón Soltar pieza, si el robot se encuentra con un objeto en su efector, este regresa a la posición inicial y desciende el efector para dejar el objeto, posteriormente vuelve a elevarse el efector. El botón de Encerar variables tiene la función de reiniciar los elementos de la pantalla de control manual, asignando 0 a todos los valores.

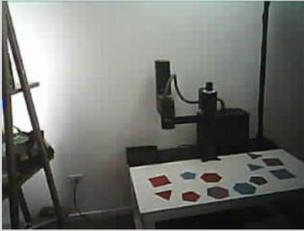
Desde esta pantalla se puede encender o apagar el robot, y provocar el reinicio del microcontrolador si es necesario. Adicionalmente cuenta con el apartado auxiliar de ayudas de las operaciones que se pueden realizar en esta pantalla.

Figura 211

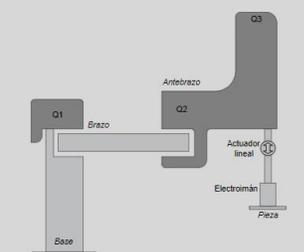
Pantalla de Control manual

Control Manual

Monitorización en tiempo real



Gemelo Digital



Configuración

Estado del brazo **Listo para recibir orden**

Ángulo para el Brazo (Q1)

Q1 0 45 90 135 180

Ángulo para el Antebrazo (Q2)

Q2 0 45 90 135 180

Ángulo para el Eector final (Q3)

Q3 0 45 90 135 180

Actuador Lineal

Electroimán

Recuerde enviar todos los comandos que requiere.

ENVIAR ORDEN Coordenada válida

ENCERRAR VARIABLES

POSICIÓN DE INICIO

REGRESAR PIEZA

Control general

Aviso

Utilizar en caso que el microcontrolador se encuentre atorado en un proceso o no responda apropiadamente. Una vez presionado el interruptor, no lo presione nuevamente hasta haberlo probado, de lo contrario **PUEDE CAUSAR DAÑO AL MICROCONTROLADOR.** Recuerde que al reiniciar el microcontrolador, inicia en un estado de calibración que puede tomar un tiempo.

Reiniciar Microcontrolador

Circuito de potencia del SCARA

APAGAR SCARA

ENCENDER SCARA

Ayudas

En esta tarjeta se puede encontrar ayudas e instrucciones para el uso de esta pestaña. Seleccione con los botones la tarjeta en la que requiere ayuda.

MONITORIZACIÓN

CONFIGURACIÓN

CONTROL GENERAL

Capítulo 5: Verificaciones

Pruebas de funcionamiento

Una vez concluida la implementación del sistema de forma integral, se realizaron pruebas que permiten validar el correcto funcionamiento del robot, siendo una de las principales características a ser probadas, el sistema de visión por computador.

Pruebas del sistema de visión por computador

El robot cuenta con una cámara que le permite obtener una imagen del espacio de trabajo, misma que permitirá programar comportamientos personalizados por el usuario. Por defecto, el robot detecta e identifica figuras geométricas de diferentes colores dentro del espacio de trabajo útil del robot.

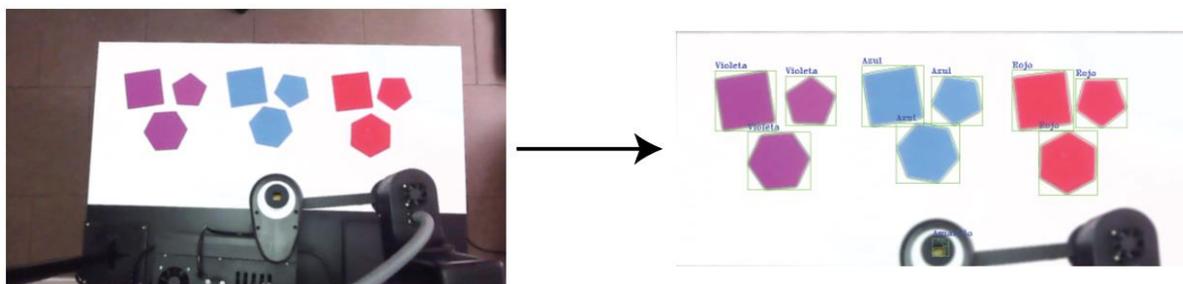
Para validar su funcionamiento, se realizaron pruebas en las que se evalúa el desempeño del sistema de visión por computador al variar posición de los objetos, diferentes colores y no hubo variación en niveles de iluminación.

Pruebas de color. Para evaluar el desempeño del sistema de visión por computador para la detección de colores mostrada en la Figura 212, se realizó 30 observaciones y se anotó los resultados en la matriz de observación, cada observación cuenta con 10 piezas, de las cuales:

- 3 piezas son de color rojo
- 3 piezas son de color azul
- 3 piezas son de color violeta
- 1 pieza es color amarillo

Figura 212

Proceso de detección de colores

**Tabla 53**

Matriz de observación de color

Clase	Medidas				Resultados				
	TP	TN	FP	FN	Sens.	Esp.	Exac.	Pre.	P.G
Rojo	86	86	1	3	96.63%	98.85%	97.73%	98.85%	95.56%
Azul	88	87	1	3	96.70%	98.86%	97.77%	98.88%	97.78%
Violeta	84	84	4	2	97.67%	95.45%	96.55%	95.45%	93.33%
Amarillo	27	27	1	2	93.10%	96.43%	94.74%	96.43%	90.00%

Nota: Para las abreviaturas usadas en la tabla se especifica su significado: Sensibilidad, Especificidad, Exactitud, Precisión y Precisión General respectivamente.

Para la detección de color, el algoritmo es capaz de detectar colores de la gama establecida con una precisión general superior al 90%, siendo el color amarillo el más difícil de ser detectado. Además, en términos de sensibilidad, especificidad, exactitud y precisión se observa aún mejores resultados con una media 96.02%, 97.40%, 96.69% y 97.40% respectivamente.

Pruebas de forma. De igual manera, se realizó 30 observaciones más para poder evaluar el rendimiento del algoritmo para la detección de figuras mostrado en la Figura 213; cada observación cuenta con 12 piezas, de las cuales:

- 3 piezas son triángulos
- 3 piezas son rectángulos
- 3 piezas son pentágonos
- 3 piezas son hexágonos

Figura 213

Proceso de detección de formas

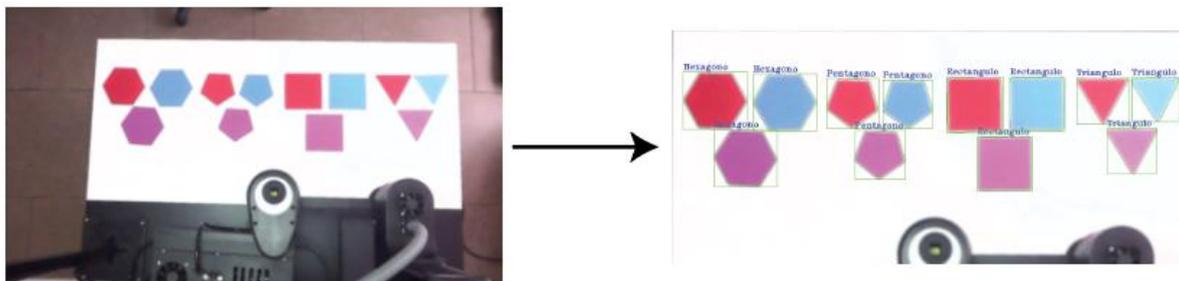


Tabla 54

Matriz de observación de formas

Clase	Medidas				Resultados				
	TP	TN	FP	FN	Sens.	Esp.	Exac.	Pre.	P.G
Triángulo	88	88	1	1	98.88%	98.88%	98.88%	98.88%	97.78%
Rectángulo	87	87	2	1	98.86%	97.75%	98.31%	97.75%	96.67%
Pentágono	86	86	1	3	96.63%	98.85%	97.73%	98.85%	95.56%
Hexágono	83	83	3	4	95.40%	96.51%	95.95%	96.51%	92.22%

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: Sensibilidad, Especificidad, Exactitud, Precisión y Precisión General respectivamente.

Para la detección de forma, el algoritmo es capaz de detectar los polígonos establecidos con una precisión de más del 92%, siendo el hexágono la figura que con menor precisión del grupo. En términos de sensibilidad, especificidad, exactitud y precisión se tiene valores con una media de 97.44%, 97.99%, 97.71% y 97,99% respectivamente.

Prueba de estimación de distancia. La última prueba realizada de forma individual para el desempeño de la visión por computador es la estimación de distancia mostrado en la Figura 214, para la cual se realizó 30 observaciones con 9 piezas ubicadas en el rango alcanzable y 9 piezas ubicadas en el rango inalcanzable, los resultados se muestran en la matriz de observación.

Figura 214

Proceso de estimación de distancia

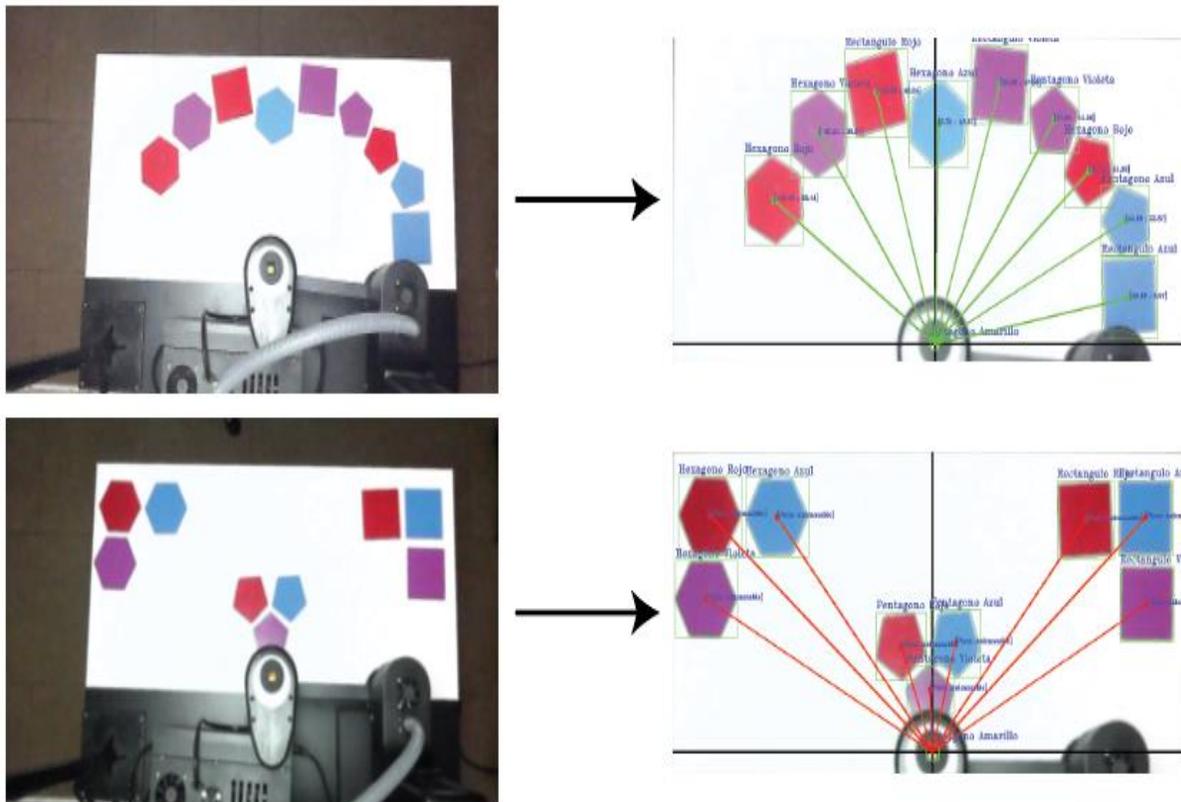


Tabla 55

Matriz de observación de estimación de distancia

Clase	Medidas				Resultados				
	TP	TN	FP	FN	Sens.	Esp.	Exac.	Pre.	P.G
Dentro del espacio de trabajo	258	258	3	9	96.63%	98.85%	97.73%	98.85%	95.56%
Fuera del espacio de trabajo	255	255	7	8	96.96%	97.33%	97.14%	97.33%	94.44%

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: Sensibilidad, Especificidad, Exactitud, Precisión y Precisión General respectivamente.

Para la estimación de la distancia, el algoritmo tiene una precisión general del 94.44% para determinar que una pieza está fuera del espacio de trabajo y un 95.56% que la pieza está dentro del espacio de trabajo. En términos de sensibilidad, especificidad, exactitud y precisión se tiene una media de 96.79%, 98.08%, 97.43% y 98.08% respectivamente.

Los resultados de cada prueba son agrupados por clases y se muestran en la Tabla 56, que es un resumen que combina el desempeño del algoritmo para poder identificar tanto, color como forma y estimación de la distancia.

Tabla 56

Matriz de observación combinada

Clase	Medidas				Resultados				
	TP	TN	FP	FN	Sens.	Esp.	Exac.	Pre.	P.G
Color	285	284	7	10	96.61%	97.59%	97.10%	97.60%	96.33%
Forma	344	344	7	9	97.45%	98.01%	97.73%	98.01%	94.33%

Clase	Medidas				Resultados				
	TP	TN	FP	FN	Sens.	Esp.	Exac.	Pre.	P.G
Distancia	513	513	10	17	96.79%	98.09%	97.44%	98.09%	95.02%

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: Sensibilidad, Especificidad, Exactitud, Precisión y Precisión General respectivamente.

Los resultados de las pruebas realizadas para cada clase de forma independiente muestran un resultado positivo a favor del algoritmo de visión artificial para colores, forma y estimación de distancia.

Precisión de clasificación. Finalmente se realiza un análisis de la precisión que tiene el robot en respuesta a las coordenadas que se obtienen de realizar la visión artificial y detectar 9 piezas por prueba, haciendo un total de 30 pruebas mostradas en la Tabla 57.

Tabla 57

Resultados de las pruebas de precisión realizadas

Prueba	Cantidad	Piezas	Precisión	Error	Error porcentual
	de piezas	clasificadas	n [%]	Absoluto	[%]
1	9	8	88.89	1.00	11.11
2	9	8	88.89	1.00	11.11
3	9	8	88.89	1.00	11.11
4	9	7	77.78	2.00	22.22
5	9	8	88.89	1.00	11.11
6	9	9	100.00	0.00	0.00
7	9	7	77.78	2.00	22.22
8	9	8	88.89	1.00	11.11
9	9	8	88.89	1.00	11.11
10	9	8	88.89	1.00	11.11
11	9	7	77.78	2.00	22.22

Prueba	Cantidad de piezas	Piezas clasificadas	Precisió n [%]	Error Absoluto	Error porcentual [%]
12	9	9	100.00	0.00	0.00
13	9	9	100.00	0.00	0.00
14	9	7	77.78	2.00	22.22
15	9	8	88.89	1.00	11.11
16	9	8	88.89	1.00	11.11
17	9	8	88.89	1.00	11.11
18	9	8	88.89	1.00	11.11
19	9	9	100.00	0.00	0.00
20	9	9	100.00	0.00	0.00
21	9	7	77.78	2.00	22.22
22	9	8	88.89	1.00	11.11
23	9	8	88.89	1.00	11.11
24	9	8	88.89	1.00	11.11
25	9	9	100.00	0.00	0.00
26	9	8	88.89	1.00	11.11
27	9	8	88.89	1.00	11.11
28	9	9	100.00	0.00	0.00
29	9	9	100.00	0.00	0.00
30	9	8	88.89	1.00	11.11

Tabla 58.

Estadísticas descriptivas de la precisión de clasificación

Variable	Conteo	M.	Desv.	Coef.	Mín.	Med.	Máx.	Rango
	Total		Est.	Var.				
Precisión	270	90	7.53	8.17	77.78	88.89	100	22.22

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: media, desviación estándar, coeficiente de variación, mínimo, primer cuartil, mediana, tercer cuartil, máximo y rango.

Para representar de forma visual y validar los datos de mejor manera, se grafican los datos obtenidos en un histograma, diagrama de caja y gráfica de prueba de probabilidad que se muestran en las Figuras 215, 216 y 217.

Figura 215

Histograma de precisión

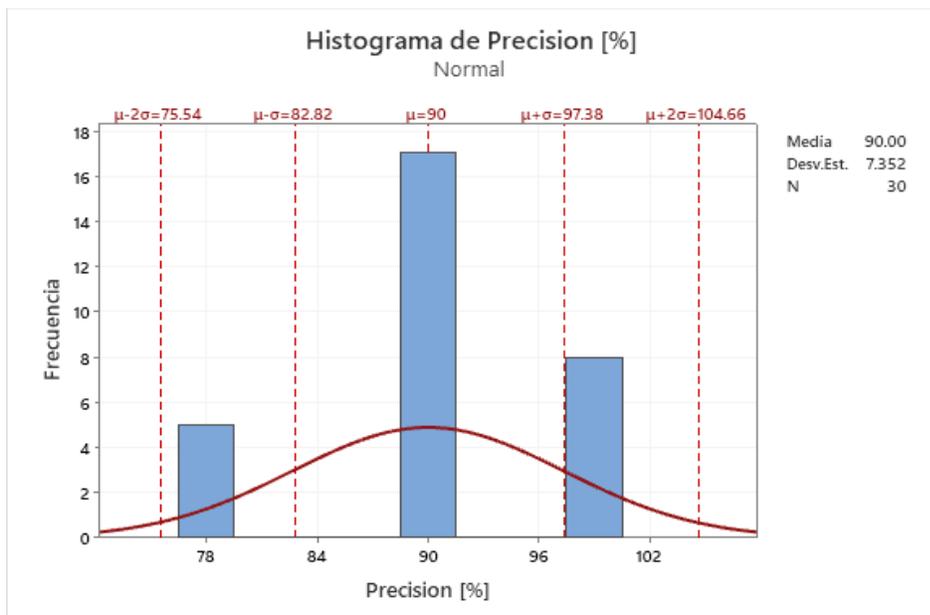
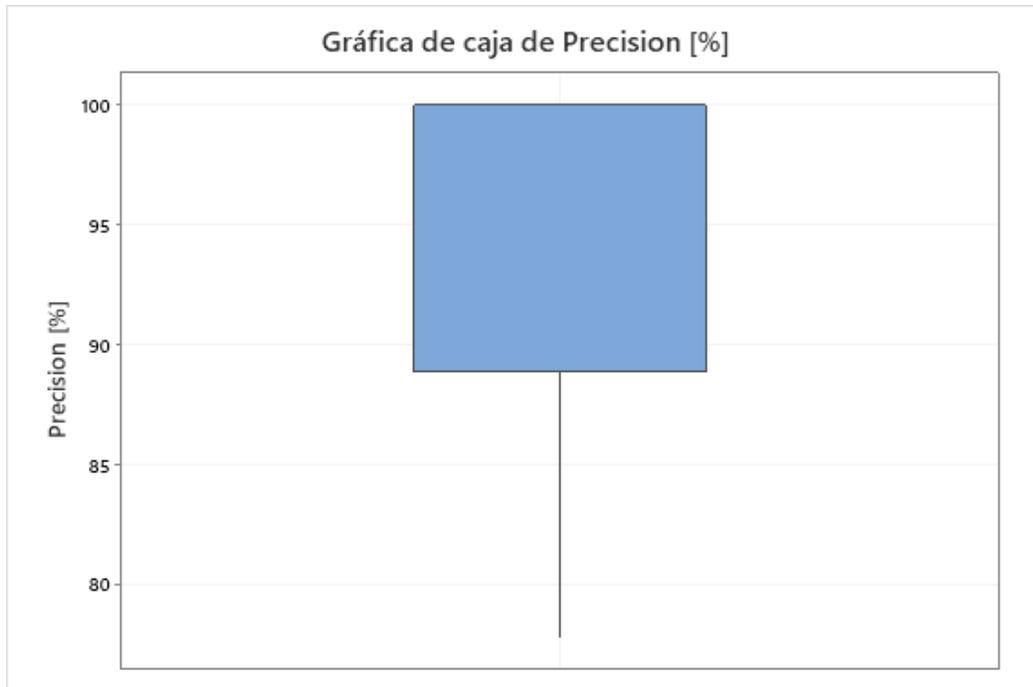
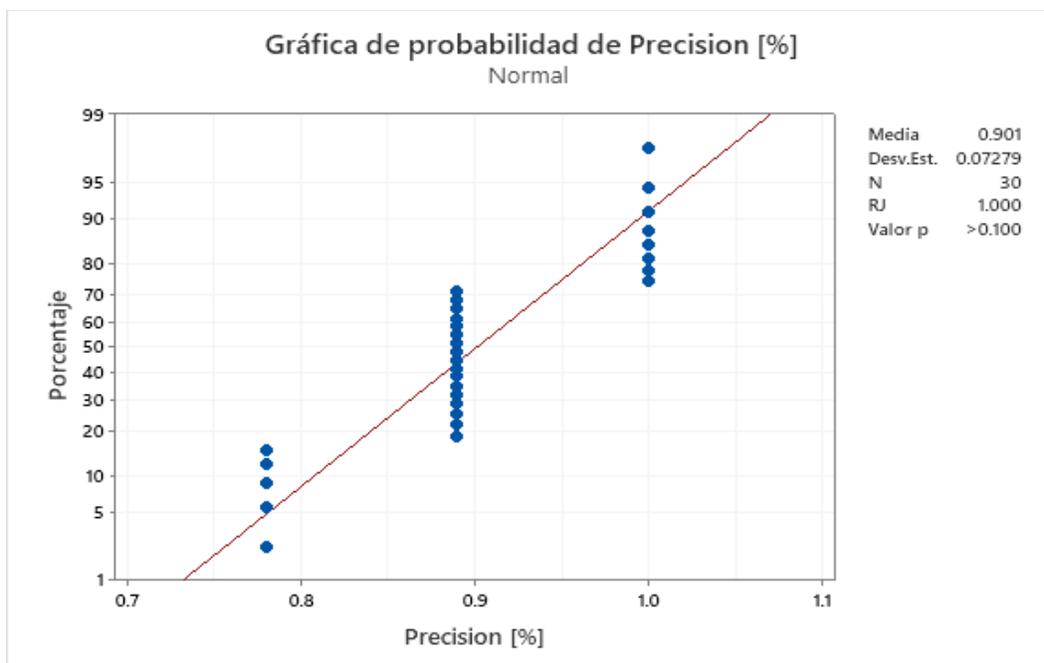


Figura 216*Diagrama de caja de la precisión***Figura 217***Gráfica de normalidad de los valores de precisión*

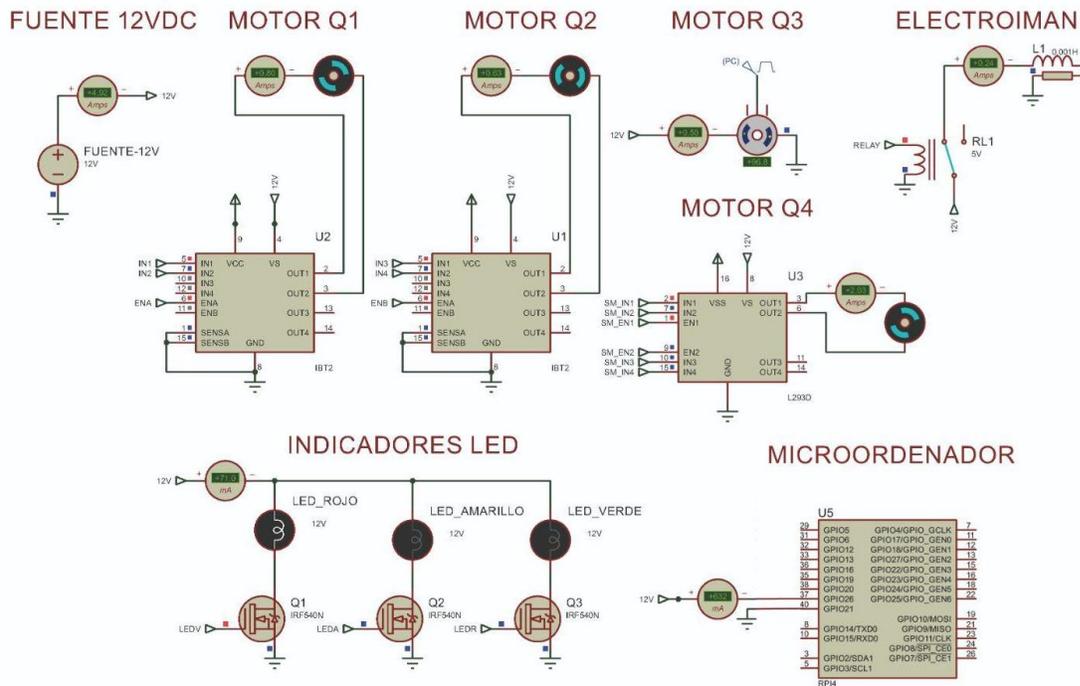
Gracias a las estadísticas y gráficos realizados, se puede observar cómo los datos siguen una distribución normal, teniendo una media del 90% de precisión al clasificar piezas en el espacio de trabajo con una desviación estándar 7.35%. En la Figura 214 se puede corroborar los datos obtenidos en el histograma y su distribución normal, aunque en la prueba de normalidad se muestra un coeficiente Ryan-Joiner (RJ) que verifica la normalidad de la precisión de las pruebas realizadas, teniendo un valor de 1, por lo que se puede concluir que existe una alta correlación de los datos con las puntuaciones normales de los datos (Vidal, 2019).

Pruebas de conexiones eléctricas

Para validar las conexiones eléctricas realizadas en el circuito del robot SCARA, se realiza una simulación de los elementos incluidos mostrada en la Figura 218, de tal manera que permitan determinar la corriente que idealmente consumen los actuadores, que son los motores de los grados de libertad Q1, Q2, Q3 y Q4, el efector final que es el electroimán, los indicadores LED que permiten visualizar el estado en el que se encuentra el robot y el microordenador que recibe las instrucciones del servidor en la nube.

Figura 218

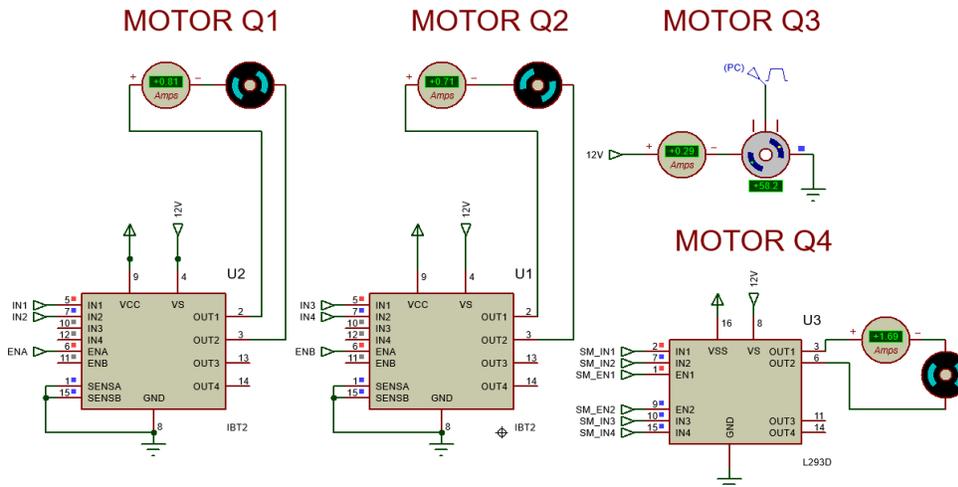
Simulación del circuito electrónico



Desglosando la simulación, se tiene que el motor del grado de libertad Q1 al ser activado, en estado estable consume 0.8A de corriente en la simulación, el motor del grado de libertad Q2 en estado estable consume 0.63A, el motor del grado de libertad Q3 consume 0.29A, y el motor del grado de libertad Q4 consume 1.69A. Estos valores se observan en los amperímetros de la simulación de la Figura 219.

Figura 219

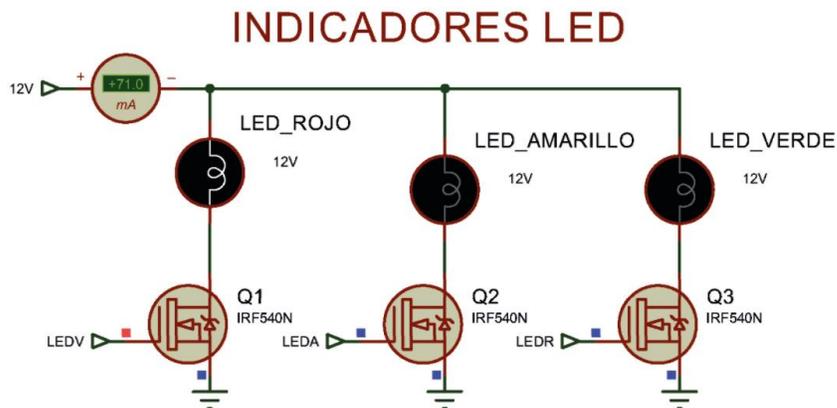
Simulación del consumo de corriente de los motores de los grados de libertad



Adicionalmente se encuentra el circuito de los LED's indicadores de estado mostrado en la Figura 220, que es un circuito compuesto de transistores tipo MOSFET que permiten que se encienda uno de los indicadores a la vez, siendo rojo, verde o amarillo. Cuando uno de estos se encuentra encendido, el circuito consume 0.071A.

Figura 220

Simulación del circuito de los indicadores LED

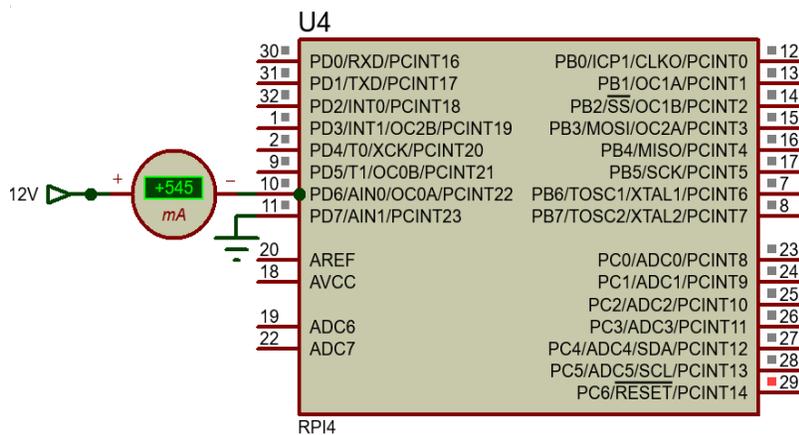


La fuente de voltaje adicionalmente alimenta al microordenador que se utiliza para la recepción y envío de información. En la descripción de la hoja de datos de esta placa, se especifica que consume 2.5A, al realizar la simulación mostrada en la Figura 221 se obtuvo que en estado estable consume 0.545A.

Figura 221

Simulación del consumo de corriente del microordenador

MICROORDENADOR



Para comprobar que los valores obtenidos de la simulación son aproximados a los reales con el robot en funcionamiento, se realiza el proceso de medición de corriente para contrastar datos.

La primera medición realizada es para la parte del circuito que alimenta a los motores de los grados de libertad Q1, Q2 y Q3, que en conjunto arrojan un valor de corriente de 1.77 A, mostrados en la Figura 222.

Figura 222

Medición conjunta de corriente de los motores de los grados de libertad Q1, Q2 y Q3



La siguiente medición realizada es para el motor del grado de libertad Q4, que corresponde al motor a pasos, mostrada en la Figura 223. Una vez medido con el amperímetro se obtiene un valor de corriente de 1.616A.

Figura 223

Medición de corriente del motor a pasos



La medición realizada al circuito que alimenta al microordenador mostrada en la Figura 224 arroja un valor de corriente de 0.575A.

Figura 224

Medición de consumo de corriente del microordenador



La medición realizada al circuito que se utiliza para los LED's indicadores de estado del robot arroja una medición de 0.069A, mostrado en la Figura 225.

Figura 225

Medición del consumo de corriente del circuito de los LED's indicadores



Se contrastaron las medidas obtenidas de la simulación con las obtenidas por medición mediante amperímetro en la Tabla 59, y se obtienen los resultados de error porcentual. Al reflejar errores menores al 5% se puede determinar que las conexiones realizadas son acordes al diagrama planteado, y no existen fugas de corriente en el circuito electrónico implementado.

Tabla 59

Comparación de valores de corriente reales y simulados

Circuito	Corriente en simulación [A]	Corriente medida [A]	Error [%]
Motor Q1, Q2 y Q3 simultáneos	1.779	1.81	1.7
Motor Q4	1.69	1.61	4.7
Indicadores LED	0.071	0.069	2.8
Microordenador	0.545	0.575	5.5

Pruebas del subsistema mecánico

Para verificar el funcionamiento del subsistema mecánico del robot, se realizaron pruebas con las que se puede cuantificar la precisión del robot para alcanzar los ángulos ingresados por el usuario o del algoritmo de visión por computador.

Para realizar esta prueba, se seleccionó 4 valores de ángulos específicos que pueden ser alcanzados por el robot (45°, 90°, 135° y 180°), mismos que serán enviados al robot en dos fases:

- Durante la primera fase de evaluación se realizará un total de 16 pruebas por cada grado de libertad a analizar, para el caso de interés únicamente se hará el estudio de Q1 y Q2. En cada prueba se enviará un valor de ángulo distribuido de forma aleatoria y se anotará el ángulo que arroja el sensor.

- Durante la segunda fase de evaluación se realizará un total de 32 pruebas, en las cuales, los grados de libertad a analizarse se mueven en conjunto, de igual manera, únicamente se toma en cuenta Q1 y Q2 y se mantiene la metodología de obtención de datos.

Fase 1

Para realizar esta fase se realizó la Tabla 60 con los valores a ser probados para el GDL Q1, siendo estos 45°, 90°, 135° y 180°, y repartidos de forma aleatoria en una tabla un total de cuatro veces cada valor, posteriormente se ingresó los valores al robot y se anotó los resultados.

Tabla 60

Valores obtenidos de la lectura de ángulos en el brazo (GDL Q1)

Prueba	Valor	Valor	Error	Error	Precisión [%]
	Teórico	Medido	Absoluto	Porcentual [%]	
1	45	44	1	2.22	97.78
2	90	89	1	1.11	98.89
3	45	44	1	2.22	97.78
4	135	133	2	1.48	98.52
5	180	178	2	1.11	98.89
6	90	88	2	2.22	97.78
7	90	89	1	1.11	98.89
8	45	43	2	4.44	95.56
9	180	178	2	1.11	98.89
10	135	133	2	1.48	98.52
11	135	133	2	1.48	98.52
12	180	173	7	3.89	96.11

Prueba	Valor	Valor	Error	Error	Precisión [%]
	Teórico	Medido	Absoluto	Porcentual [%]	
13	90	89	1	1.11	98.89
14	180	179	1	0.56	99.44
15	135	133	2	1.48	98.52
16	45	43	2	4.44	95.56

Para estos datos se realizó un análisis estadístico para evaluar el desempeño del grado de libertad Q1. Los resultados se muestran en la Tabla 61.

Tabla 61

Resultado de las estadísticas descriptivas del GDL Q1

Variable	Conteo	M.	Desv.	Coef.	Mín.	Med.	Máx.	Rango
	Total		Est.	Var.				
Precisión	16	98.03	1.23	1.26	95.56	98.52	99.44	3.88

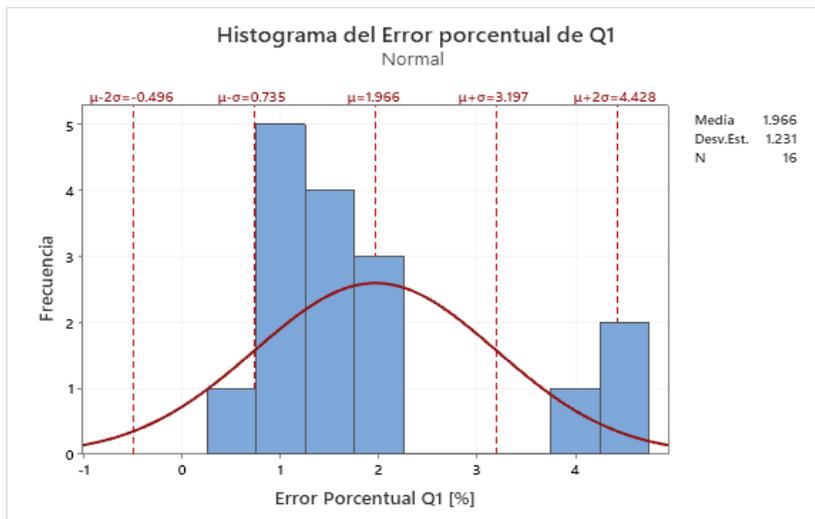
de Q1

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: media, desviación estándar, coeficiente de variación, mínimo, primer cuartil, mediana, tercer cuartil, máximo y rango.

La tabla resumen presenta las estadísticas de la precisión del robot ante las 16 pruebas, mostrando valores con una media de 98.03%. Además, se realizó un análisis de la distribución de los datos para el error porcentual, error absoluto y precisión para poder validar los resultados. El histograma del error porcentual de la Figura 226 muestra una distribución normal, teniendo una media de 1.966% con una desviación estándar de 1.231%.

Figura 226

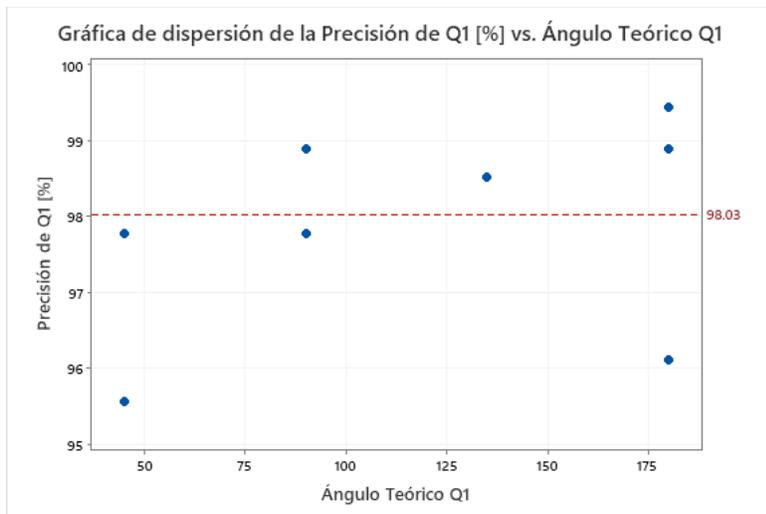
Histograma de los valores de error porcentual del GDL Q1



La gráfica de dispersión de la precisión para el grado de libertad Q1 de la Figura 227 da como resultado una dispersión con un factor de $r = 0.0126$, que indica una baja dispersión para la precisión.

Figura 227

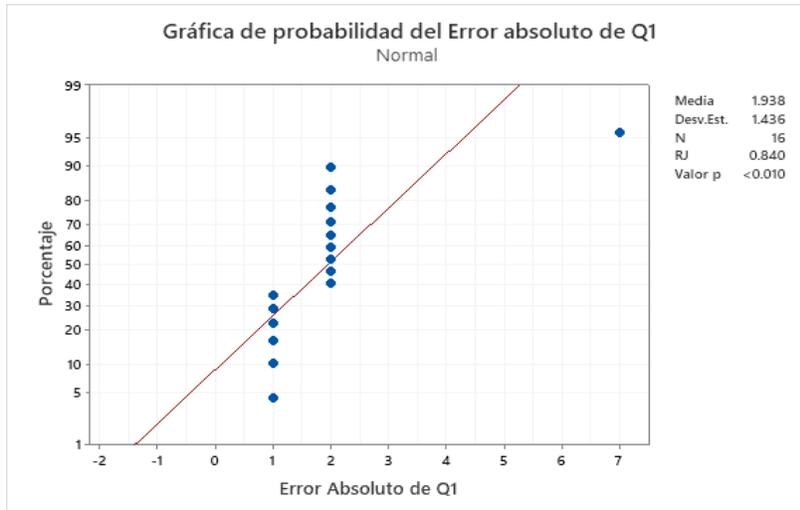
Gráfica de dispersión de la precisión del GDL Q1



La prueba de normalidad del error absoluto de la Figura 228 arroja un factor $RJ = 0.84$ bastante alto debido a su cercanía a 1, por lo que se puede concluir que los datos que se obtuvieron cuentan con una distribución normal.

Figura 228

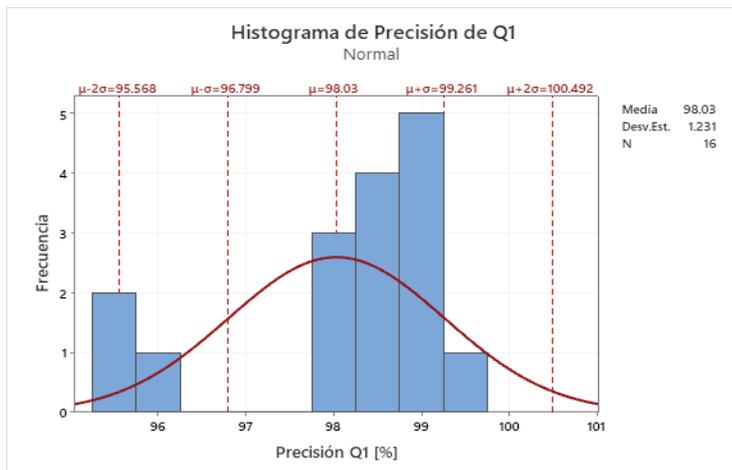
Gráfica de normalidad de la precisión del GDL Q1



El histograma de la precisión de la Figura 229 muestra la distribución normal de los datos obtenidos con una media bastante alta, de 98.03%, y una desviación estándar de 1.23%.

Figura 229

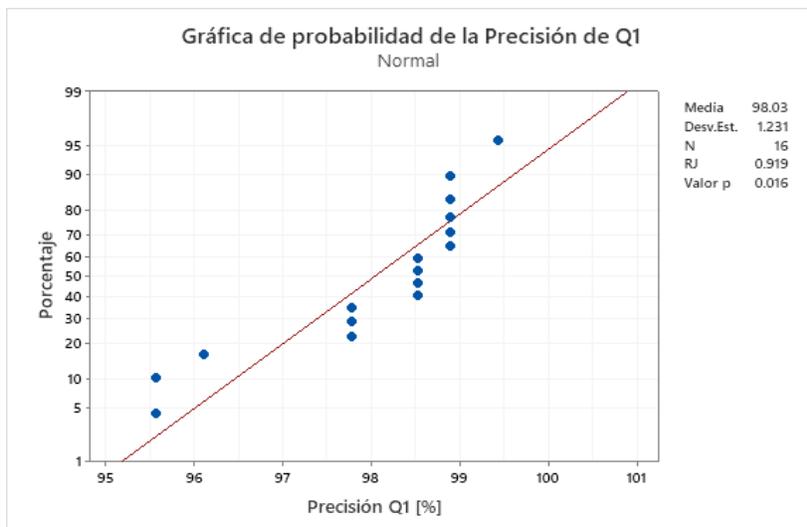
Histograma de valores de precisión del GDL Q1



La prueba de normalidad que se obtuvo de la precisión de Q1 de la Figura 230 muestra el coeficiente $RJ = 0.919$ que se trata de una distribución de datos normal. De igual manera se realizará el análisis para el grado de libertad Q2.

Figura 230

Prueba de normalidad de los valores de precisión de Q1



El proceso se realiza de forma similar para el grado de libertad Q2, que arroja los valores que se muestran en la Tabla 62.

Tabla 62

Valores obtenidos de la lectura de ángulos en el antebrazo (GDL Q2)

Prueba	Valor	Valor	Error	Error	Precisión
	Teórico	Medido	Absoluto	Porcentual [%]	[%]
17	135	133	2	1.48	98.52
18	45	43	2	4.44	95.56
19	135	134	1	0.74	99.26
20	135	134	1	0.74	99.26
21	135	134	1	0.74	99.26

Prueba	Valor	Valor	Error	Error	Precisión
	Teórico	Medido	Absoluto	Porcentual [%]	[%]
22	180	161	19	10.56	89.44
23	45	43	2	4.44	95.56
24	180	167	13	7.22	92.78
25	45	45	0	0.00	100.00
26	90	89	1	1.11	98.89
27	90	89	1	1.11	98.89
28	180	170	10	5.56	94.44
29	180	161	19	10.56	89.44
30	45	43	2	4.44	95.56
31	90	87	3	3.33	96.67
32	90	89	1	1.11	98.89

Para estos datos también se realizó un análisis estadístico para evaluar el desempeño del grado de libertad Q2, mostrado en la Tabla 63.

Tabla 63

Estadísticas descriptivas del GDL Q2

Variable	Conteo	M.	Desv.	Coef.	Mín.	Med.	Máx.	Rango
	Total		Est.	Var.				
Precisión	16	96.40	3.43	3.56	89.44	97.59	100.00	10.56

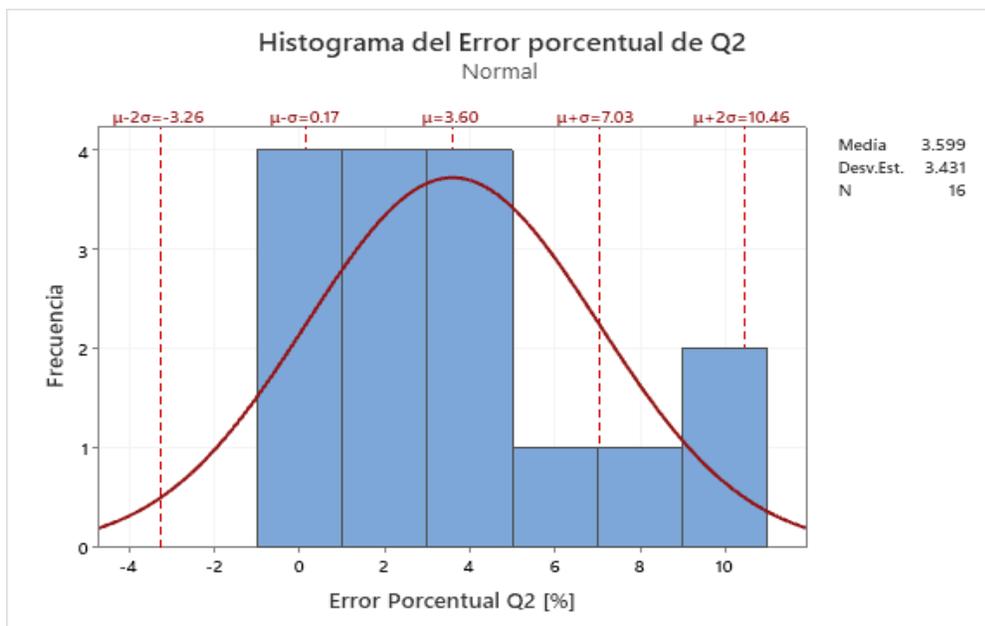
de Q2

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: media, desviación estándar, coeficiente de variación, mínimo, primer cuartil, mediana, tercer cuartil, máximo y rango.

La tabla resumen muestra las siguientes 16 pruebas, realizadas al grado de libertad Q2, mostrando nuevamente una precisión bastante alta, con una media de 96.4%. Con estos datos también se realizará un análisis de los datos que se obtuvieron para el error porcentual, error absoluto y precisión para poder validar los resultados. El histograma del error porcentual de Q2 de la Figura 231 muestra una distribución normal, con una media de 3.599% y una desviación estándar de 3.431%.

Figura 231

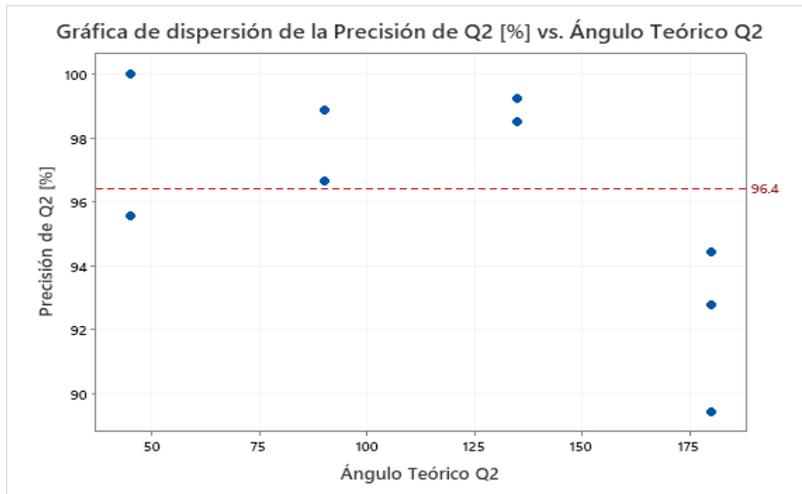
Histograma del error porcentual del GDL Q2



La gráfica de dispersión para la dispersión del grado de libertad Q2 de la Figura 232 da como resultado una dispersión con un factor de $r = 0.0356$, este factor indica una baja dispersión para la precisión de Q2.

Figura 232

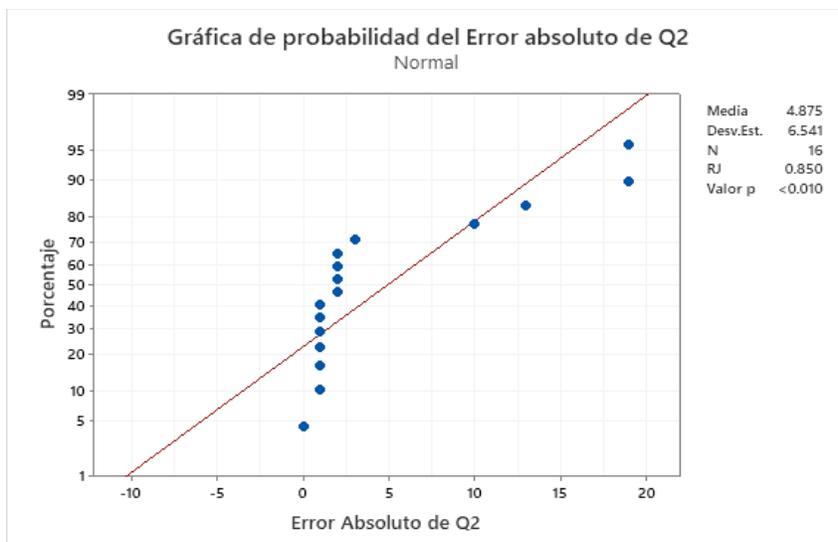
Gráfica de dispersión de la precisión del GDL Q2



La prueba de normalidad de la Figura 233 arroja un factor $RJ = 0.85$ bastante alto debido a su cercanía a 1, por lo que se puede concluir que los datos que se obtuvieron cuentan con una distribución normal.

Figura 233

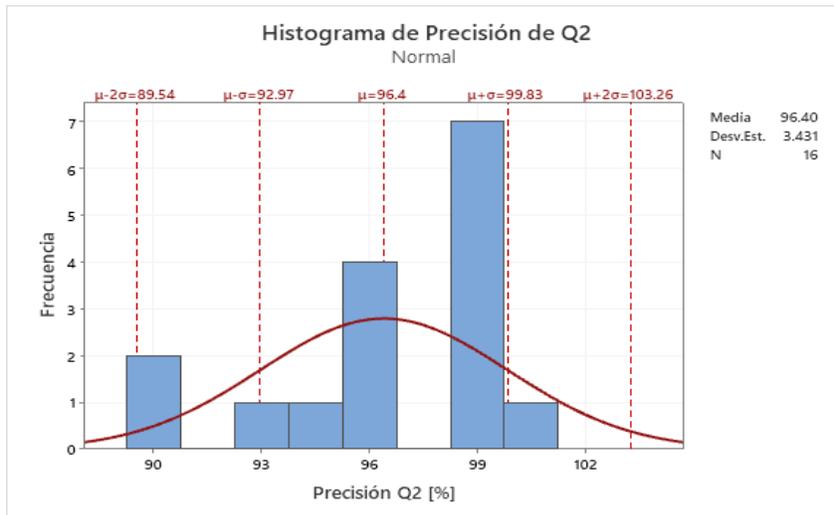
Prueba de normalidad del error absoluto del GDL Q2



El histograma de precisión para Q2 de la Figura 234 muestra una distribución normal, con una media de 96.4% y una desviación estándar de 3.431%.

Figura 234

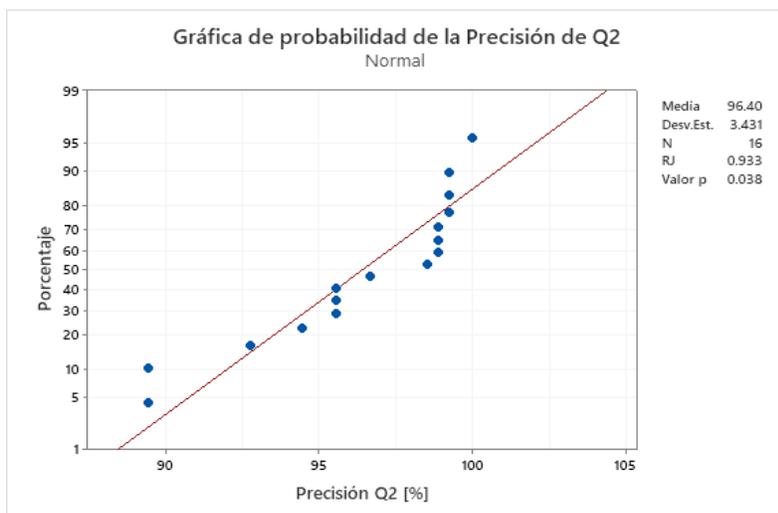
Histograma de precisión del GDL Q2



La prueba de normalidad que se obtuvo de la precisión de Q2 de la Figura 235 muestra el coeficiente $RJ = 0.933$, confirmando que se trata de una distribución de datos normal.

Figura 235

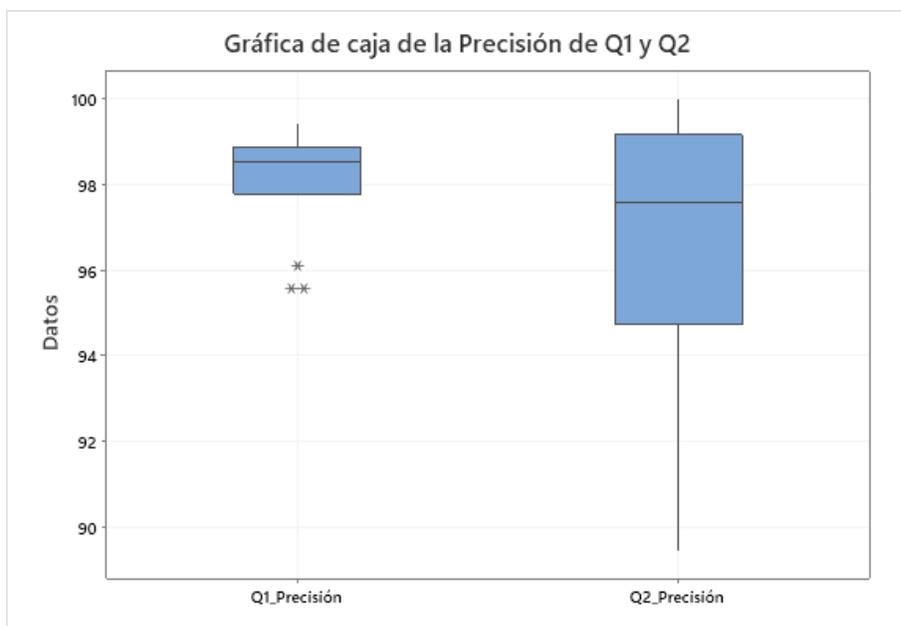
Gráfica de normalidad de la precisión del GDL Q2



Se realizó una gráfica de caja, en la que se puede visualizar como en las pruebas pertenecientes al grado de libertad Q1, se obtuvo datos atípicos, estos datos pueden afectar las mediciones, es por eso que estos valores son descartados. Por su parte, el grado de libertad Q2 no cuenta con valores atípicos, por lo que no se esperan distorsiones por parte de estos datos. Esto es visible en la Figura 236.

Figura 236

Diagrama de caja de la precisión de los GDL Q1 y Q2



Fase 2

Para la segunda fase de análisis del funcionamiento, se realizó un proceso similar al de la fase 1, con la diferencia que ahora que se harán 32 pruebas en las que tanto Q1 como Q2 se mueven al mismo tiempo a los ángulos asignados (45° , 90° , 135° y 180°), esto para determinar si el movimiento conjunto provoca una menor precisión. Los valores obtenidos se muestran en la Tabla 64.

Tabla 64

Valores de la medición de los GDL Q1 y Q2 de forma simultánea

Prueba	Valor		Valor		Error		Error		Precisión [%]	
	Esperado		Esperado		Absoluto		Porcentual [%]		Q1	Q2
	Q1	Q2	Q1	Q2	Q1	Q2	Q1	Q2		
1	135	45	134	45	1	0	0.741	0.000	99.259	100.000
2	45	90	43	89	2	1	4.444	1.111	95.556	98.889
3	45	135	44	133	1	2	2.222	1.481	97.778	98.519
4	45	180	44	159	1	21	2.222	11.667	97.778	88.333
5	135	45	134	45	1	0	0.741	0.000	99.259	100.000
6	90	90	89	87	1	3	1.111	3.333	98.889	96.667
7	135	135	134	133	1	2	0.741	1.481	99.259	98.519
8	135	180	133	169	2	11	1.481	6.111	98.519	93.889
9	90	45	88	43	2	2	2.222	4.444	97.778	95.556
10	45	90	44	89	1	1	2.222	1.111	97.778	98.889
11	180	90	177	89	3	1	1.667	1.111	98.333	98.889
12	135	180	133	175	2	5	1.481	2.778	98.519	97.222
13	45	45	44	43	1	2	2.222	4.444	97.778	95.556
14	135	90	133	87	2	3	1.481	3.333	98.519	96.667
15	180	45	180	43	0	2	0.000	4.444	100.00	95.556
16	180	90	170	87	10	3	5.556	3.333	94.444	96.667
17	180	45	179	45	1	0	0.556	0.000	99.444	100.000
18	45	90	44	87	1	3	2.222	3.333	97.778	96.667
19	180	90	175	87	5	3	2.778	3.333	97.222	96.667
20	135	180	134	169	1	11	0.741	6.111	99.259	93.889

Prueba	Valor		Valor		Error		Error		Precisión [%]	
	Esperado		Esperado		Absoluto		Porcentual [%]		Q1	Q2
	Q1	Q2	Q1	Q2	Q1	Q2	Q1	Q2		
21	180	45	177	43	3	2	1.667	4.444	98.333	95.556
22	90	90	88	89	2	1	2.222	1.111	97.778	98.889
23	90	135	88	133	2	2	2.222	1.481	97.778	98.519
24	90	180	88	170	2	10	2.222	5.556	97.778	94.444
25	45	45	43	45	2	0	4.444	0.000	95.556	100.000
26	90	90	89	87	1	3	1.111	3.333	98.889	96.667
27	90	135	88	134	2	1	2.222	0.741	97.778	99.259
28	45	180	43	169	2	11	4.444	6.111	95.556	93.889
29	90	45	88	45	2	0	2.222	0.000	97.778	100.000
30	180	90	170	89	10	1	5.556	1.111	94.444	98.889
31	135	135	134	134	1	1	0.741	0.741	99.259	99.259
32	180	45	175	45	5	0	2.778	0.000	97.222	100.000

Para estos datos se realizó un análisis estadístico para evaluar el desempeño del grado de libertad Q1 y Q2, mostrado en la Tabla 65.

Tabla 65

Estadísticas descriptivas de los GDL Q1 y Q2

Variable	Conteo	M.	Desv.	Coef.	Mín.	Med.	Máx.	Rango
	Total		Est.	Var.				
Precisión	32	97.85	1.40	1.43	94.4	97.77	100.	5.56
de Q1					4		00	
Precisión	32	97.26	2.58	2.65	88.3	97.87	100.	11.67

Variable	Conteo	M.	Desv.	Coef.	Mín.	Med.	Máx.	Rango
	Total		Est.	Var.				
de Q2					3		00	

Nota. Para las abreviaturas usadas en la tabla se especifica su significado: media, desviación estándar, coeficiente de variación, mínimo, primer cuartil, mediana, tercer cuartil, máximo y rango.

Con los resultados mostrados en la tabla resumen se observa que aún se mantiene una precisión bastante alta, de media de 97.85% para Q1 y 97.26% para Q2. Con estos resultados se podría concluir que el robot tiene una precisión dentro de los rangos aceptables ya que tiene una precisión superior al 95%, o que tiene menos del 5% de error en cada grado de libertad.

Para validar los resultados, se hicieron análisis del error porcentual y precisión para cada grado de libertad. Los histogramas del error porcentual para Q1 y Q2 trabajando en conjunto mostrados en las Figuras 237 y 238, en ambos casos, muestran una distribución normal de los datos con una media de 2.147% y 2.737% respectivamente y una desviación estándar de 1.402% y 2.581%.

Figura 237

Histograma de error porcentual del GDL Q1

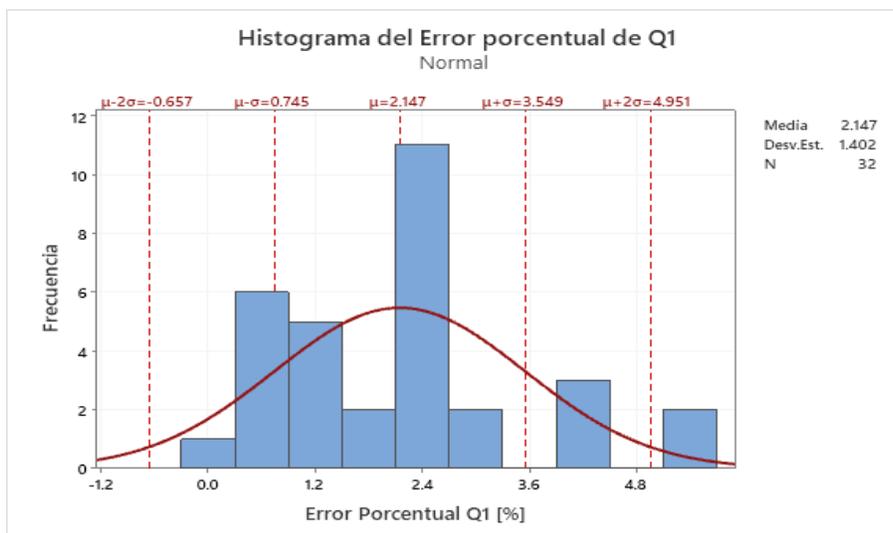
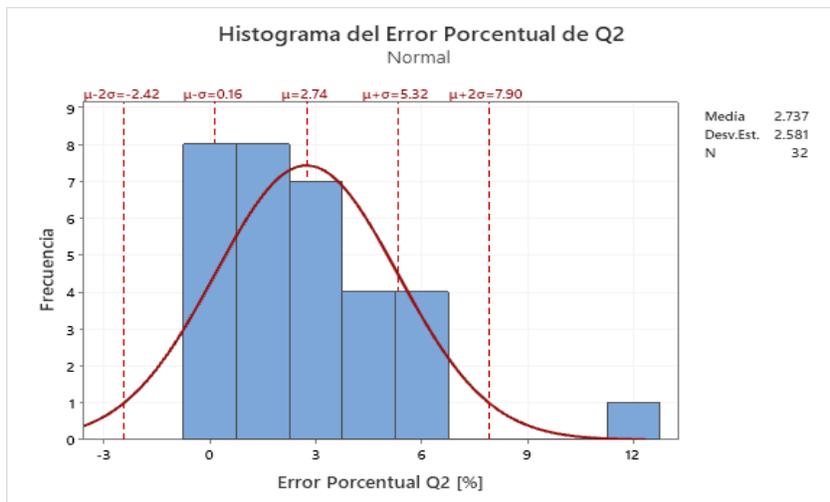


Figura 238

Histograma de error porcentual del GDL Q2



De igual manera, los histogramas de la precisión de las Figuras 239 y 240 muestran la misma tendencia normal de los datos, obteniendo una media de 97.85% con una desviación estándar de 1.402% para Q1, y una media de 97.26% con una desviación estándar de 2.581%.

Figura 239

Histograma de precisión del GDL Q1

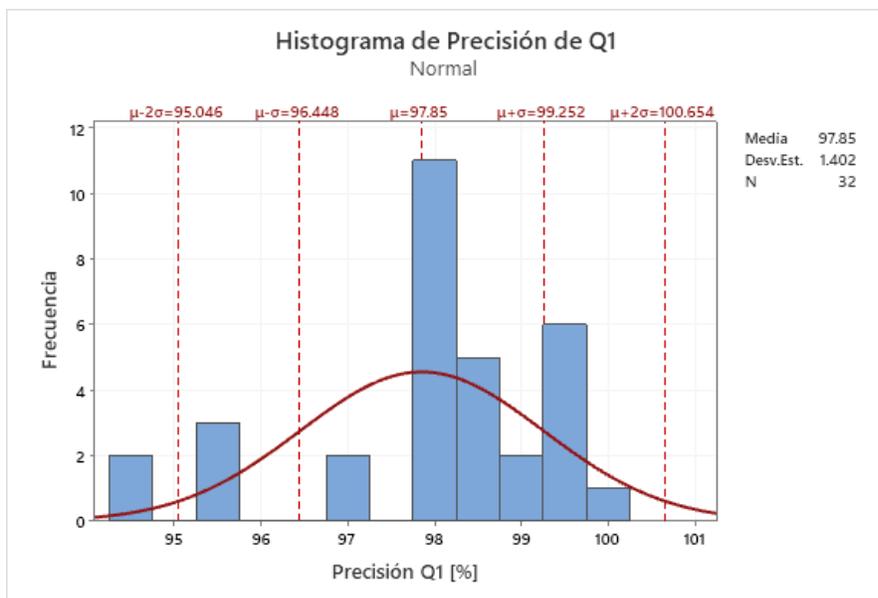
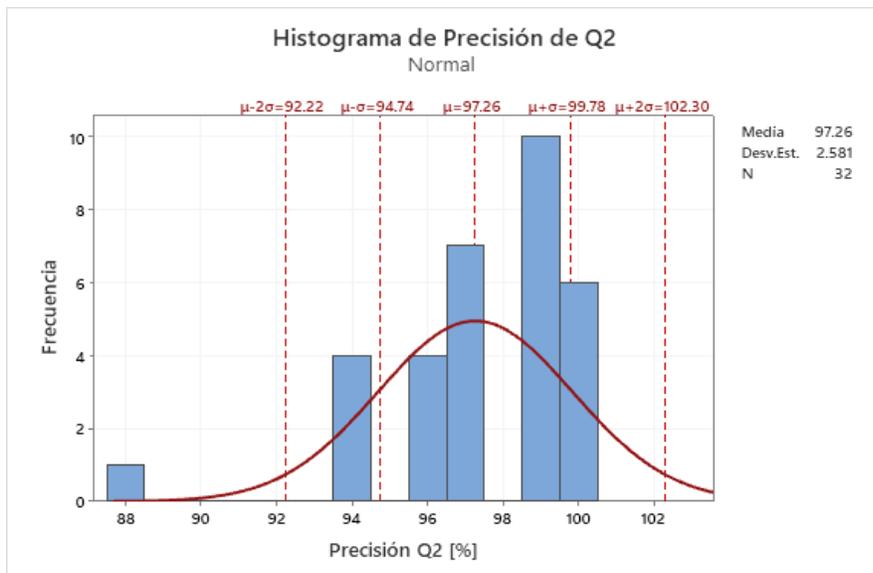


Figura 240

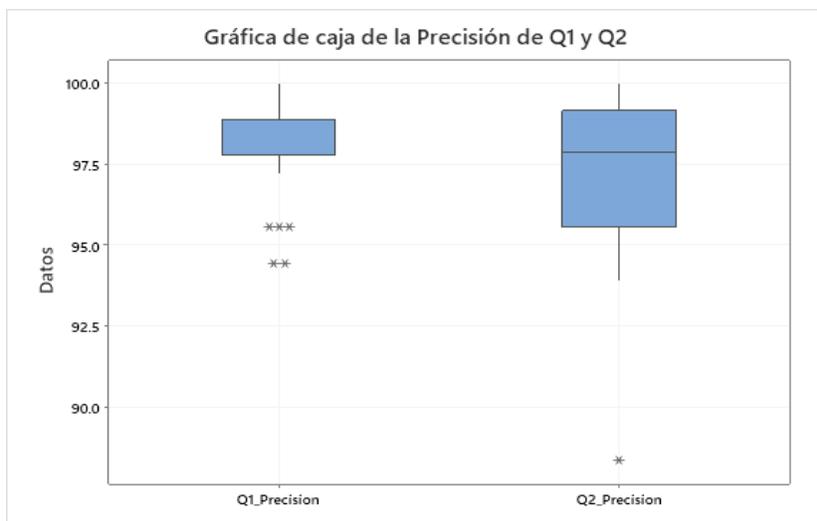
Histograma de precisión del GDL Q2



Con el diagrama de caja de la Figura 241, se puede observar que existen valores atípicos que provocan distorsión en las medidas, por lo que no deben ser tomados en cuenta en el cálculo de las estadísticas, notando que para el caso de Q2 es un valor bastante alejado de la distribución normal.

Figura 241

Gráfico de caja de la precisión de Q1 y Q2



En la gráfica de dispersión para la precisión de los grados de libertad Q1 y Q2 de las Figuras 242 y 243, se tiene un coeficiente $r_1 = 0.0143$ y $r_2 = 0.0265$, al ser valores tan bajos se puede concluir de una baja dispersión de los datos para la precisión de Q1 y Q2.

Figura 242

Gráfica de dispersión de la precisión de Q1

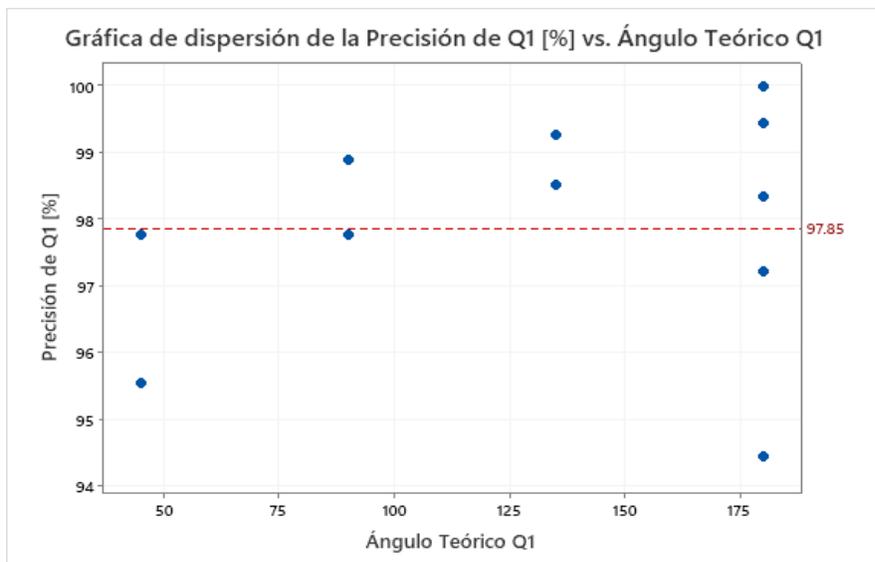
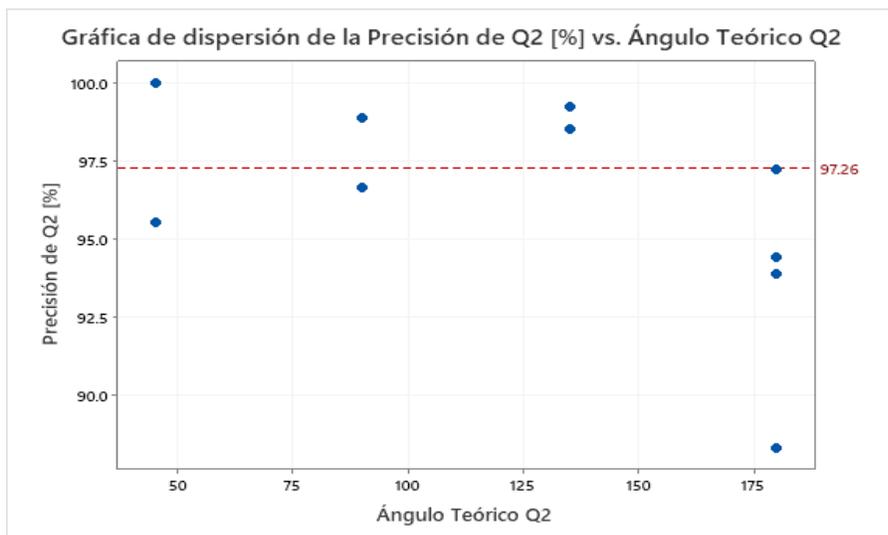


Figura 243

Gráfica de dispersión de la precisión de Q2



En las pruebas de normalidad de las Figuras 244 y 245 se obtuvieron factores de Ryan-Joiner de $RJ_1 = 0.919$ para Q1 y $RJ_2 = 0.933$ para Q2, demostrando que los datos de precisión para los dos grados de libertad que se analizó son de distribución normal. Cabe recalcar que durante el análisis de los datos se deben tener en cuenta a los valores atípicos para excluirlos del análisis para evitar que los resultados se vean alterados.

Figura 244

Gráfica de normalidad de la precisión de Q1

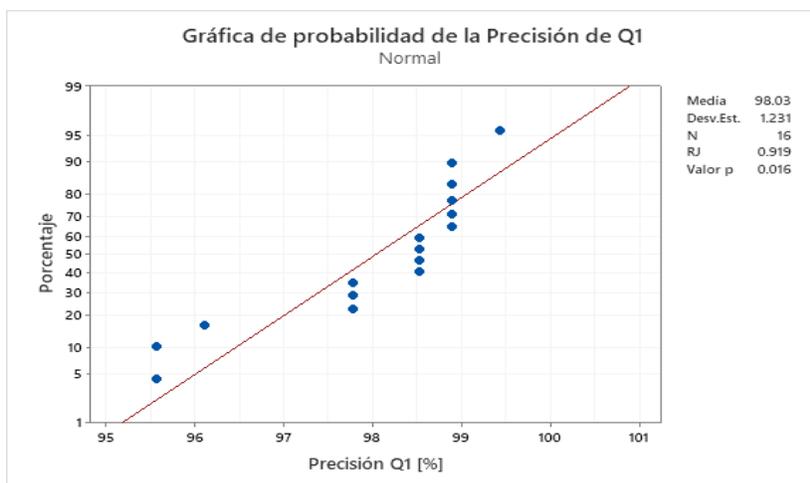
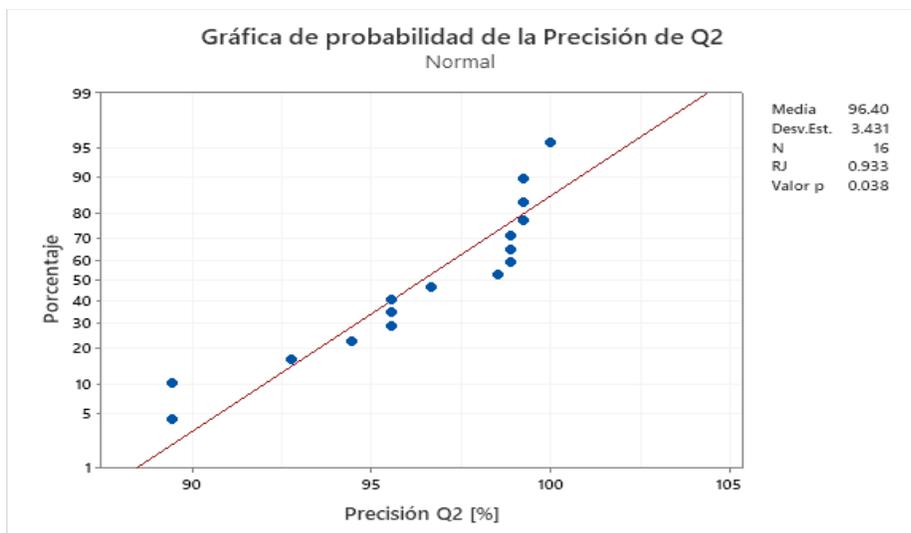


Figura 245

Gráfica de normalidad de la precisión de Q2



Pruebas de usabilidad con usuarios

Para verificar la facilidad de uso con los usuarios por medio de la interfaz humano-máquina, se realizó una prueba usando el sistema de escalas de usabilidad (System Usability Scale [SUS]). Esta prueba cuenta con 10 preguntas estandarizadas que permite determinar el nivel de usabilidad de la interfaz.

Las preguntas que se hicieron fueron las siguientes:

1. Me gustaría usar este sistema con frecuencia.
2. El sistema es innecesariamente complejo.
3. Me pareció que el sistema fue fácil de usar.
4. Creo que necesito de un experto técnico para que me ayude a usar el sistema.
5. Encontré varias funciones en el sistema que estaban bien integradas.
6. Pensé que había muchas inconsistencias en el sistema.
7. Me imagino que muchas personas aprenderán a utilizar el sistema muy rápido.
8. Encontré que el sistema es engorroso de utilizar.
9. Me sentí muy confiado al usar el sistema.
10. Necesité aprender muchas cosas antes de que pueda comenzar a utilizar el sistema.

Para responder a estas preguntas, el encuestado debía escoger en una escala del 1 al 5, siendo 1 “Para nada de acuerdo” y 5 “Completamente de acuerdo”. Para realizar la prueba se escogió a 10 usuarios, donde la mayoría de ellos tienen conocimientos técnicos en la materia, mientras que el resto no se relacionan con el área de conocimiento. Los resultados que de la prueba de funcionalidad se muestran en la Tabla 66.

Tabla 66*Resultados de la prueba de usabilidad*

Usuario	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
U1	3	3	5	2	4	3	4	2	4	1
U2	4	2	4	3	4	1	4	2	4	3
U3	4	1	4	2	4	1	4	2	4	1
U4	3	2	5	2	5	1	5	3	4	2
U5	5	3	5	1	4	1	5	1	5	2
U6	4	2	5	1	3	2	5	1	5	2
U7	4	1	5	1	5	1	5	1	5	1
U8	5	1	5	3	4	1	5	1	5	4
U9	5	3	3	3	4	2	4	3	5	2
U10	4	2	2	2	3	3	3	2	4	3

Nota. Las iniciales que se usan en la tabla representan la palabra “Pregunta” y “Usuario”

Para analizar los resultados, se debe hacer algunas operaciones a las calificaciones de los encuestados. El primer paso es restar 1 a todas las puntuaciones de las preguntas impares y restar de 5 las puntuaciones de todas las preguntas pares, posteriormente se suman los resultados y se multiplica por un factor de 2.5 (Brooke, 1995).

Al realizar las operaciones, se obtiene las puntuaciones que se muestran en la Tabla 67.

Tabla 67

Tabla de puntajes en la escala SUS

Usuario	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Puntuación SUS
U1	2	2	4	3	3	2	3	3	3	4	72.5
U2	3	3	3	2	3	4	3	3	3	2	72.5
U3	3	4	3	3	3	4	3	3	3	4	82.5
U4	2	3	4	3	4	4	4	2	3	3	80
U5	4	2	4	4	3	4	4	4	4	3	90
U6	3	3	4	4	2	3	4	4	4	3	85
U7	3	4	4	4	4	4	4	4	4	4	97.5
U8	4	4	4	2	3	4	4	4	4	1	85
U9	4	2	2	2	3	3	3	2	4	3	70
U10	3	3	1	3	2	2	2	3	3	2	60
Promedio											79.5

Nota. Las iniciales que se usan en la tabla representan la palabra "Pregunta" y "Usuario"

En base a la Figura 246 se puede establecer que el producto desarrollado e implementado es Aceptable en la escala SUS al obtener una calificación de 79.5 sobre 100.

Figura 246

Escala SUS



Nota. Tomado de *Medir la usabilidad con el Sistema de Escalas de Usabilidad*, por C.

Busquets, 2021.

La puntuación obtenida es bastante aceptable, ya que los criterios de aceptación del puntaje están por encima de 68%. Sin embargo, se tomó en consideración algunos de los criterios de los usuarios con puntuación más baja para poder mejorar los aspectos con deficiencia.

Esta prueba ayuda a incrementar la usabilidad especialmente para los usuarios que no se encuentran dentro del área de conocimiento.

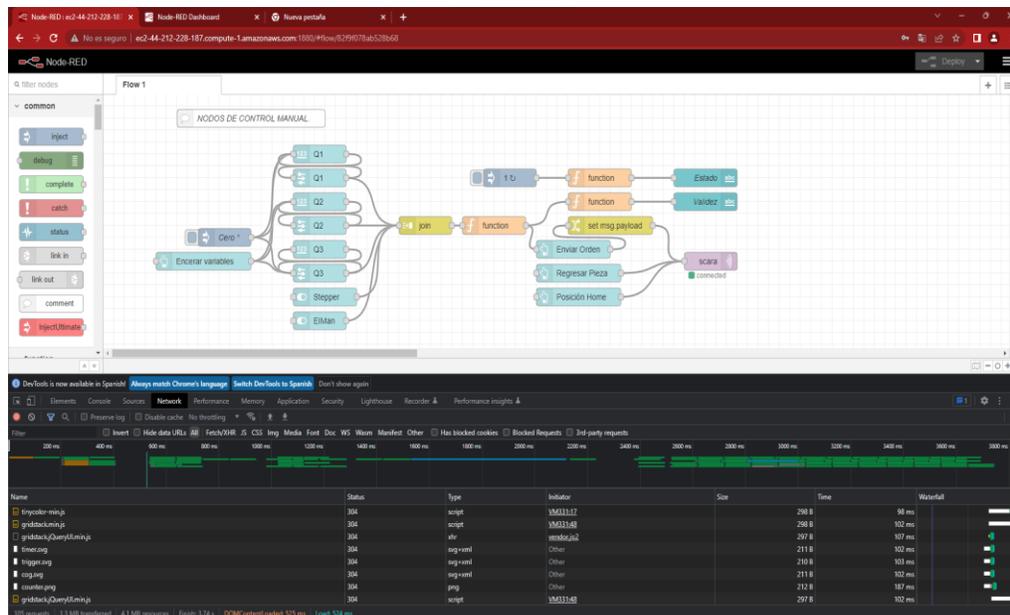
Pruebas de carga del sistema

Para evaluar la respuesta que tendrá la arquitectura de software ante cargas masivas de usuarios, se realizaron pruebas de carga del sistema usando la herramienta Gatling. Con esta herramienta es posible simular cómo responden los servicios web utilizados ante una gran cantidad de usuarios conectados y realizando peticiones.

Para esto se utilizó la herramienta para desarrolladores que proporciona Google Chrome para grabar las peticiones que se realizan en una página web como se muestra en la Figura 247, para ello se ingresó a la página web de inicio de sesión y se ejecutaron las operaciones que se realizan normalmente, como acceder con usuario y contraseña mientras la herramienta del navegador graba.

Figura 247

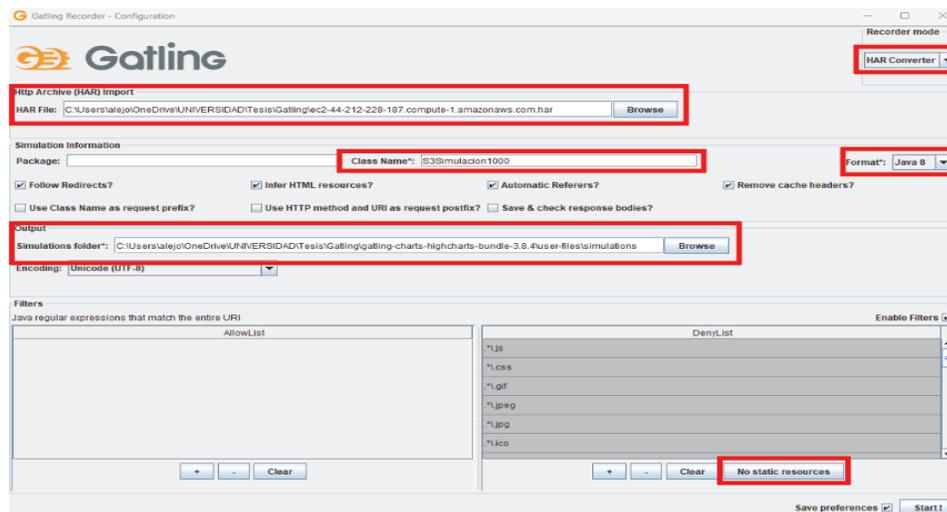
Prueba de carga de la página web de Node - RED



Al finalizar la grabación, se guardó un archivo de tipo E-Sword Harmony Resource (HAR), formato que es reconocible por la herramienta de grabación de Gatling. La configuración de esta herramienta se realizó como se muestra en la Figura 248.

Figura 248

Configuración de la herramienta Gatling



Al terminar el proceso creará un archivo en el lenguaje que se haya seleccionado en la herramienta, para el presente caso se seleccionó en lenguaje Java. En las Figuras 249, 250 y 251 se muestran las partes relevantes del código generado.

Figura 249

Código generado de la prueba de carga

```
public class S3simulacion1000 extends Simulation { ← Nombre de la clase/prueba
{
    HttpProtocolBuilder httpProtocol = http
    .baseUrl("http://ec2-44-212-228-187.compute-1.amazonaws.com:1888")
    .inferHtmlResources(AllowList(), DenyList(".*\\.js", ".*\\.css", ".*\\.gif", ".*\\.jpeg", ".*\\.jpg", ".*\\.ico", ".*\\.woff", ".*\\.woff2", ".*\\.t(jo)tf", ".*\\.png", ".*detectportal\\.firefox\\.com.*"))
    .acceptHeader("image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8")
    .acceptEncodingHeader("gzip, deflate")
    .acceptLanguageHeader("es-ES,es;q=0.9")
    .userAgentHeader("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36");
}
```

URL donde se ejecutan las pruebas

Lista de los formatos que se excluyen de la prueba

Figura 250

Código generado de la prueba de carga (Escenario de simulación)

```
ScenarioBuilder scn = escenario("S3simulacion1000") ← Creación del escenario para la simulación
    .exec(
        http("request_0")
            .get("/")
            .headers(headers_0)
            .resources(
                http("request_1")
                    .get("/theme")
                    .headers(headers_1),
                http("request_2")
                    .get("/locales/editor?lng=en-US")
                    .headers(headers_1),
                http("request_3")
                    .get("/locales/editor?lng=es-ES")
                    .headers(headers_1),
                ...
                http("request_59")
                    .get("/icons/node-red-contrib-simple-message-queue/simple-message-queue.svg"),
                http("request_60")
                    .get("/icons/node-red/file.svg"),
                http("request_61")
                    .get("/icons/node-red/feed.svg"),
                http("request_62")
                    .get("/icons/node-red/arrow-in.svg")
            )
    );
```

Todas las peticiones que se van a simular

Figura 251

Código generado de la prueba de carga (Usuarios a simular)

Número de usuarios a simular

```
setUp(scn.injectOpen(atOnceUsers(1000)).protocols(httpProtocol);
```

Al finalizar de realizar los cambios y ejecutar, se muestran los resultados arrojados por la herramienta de Gatling, nótese en la Figura 252, que con una carga de 1000 usuarios se tiene un porcentaje alto de peticiones fallidas, aunque las peticiones enviadas totales siguen representando la mayoría.

Figura 252

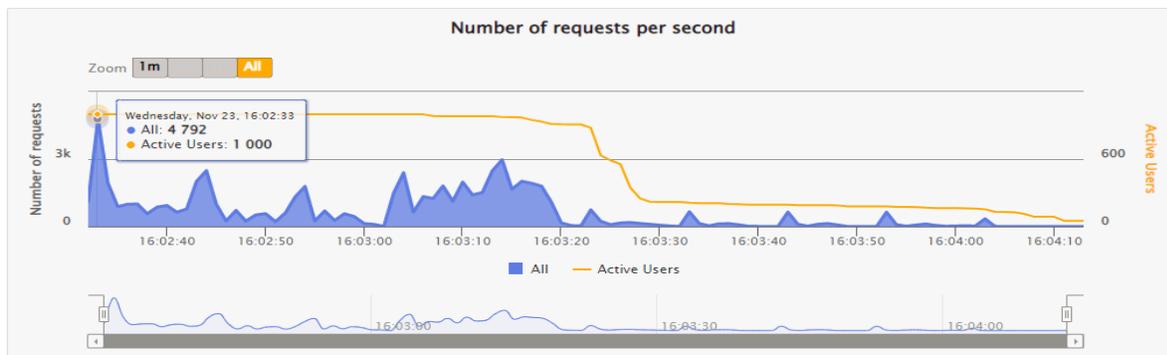
Prueba de carga con 1000 usuarios



Los resultados que arroja la herramienta proporcionan información de la respuesta ante una gran carga (1000 usuarios). En la Figura 253, el 23% de las peticiones se logró enviar en un tiempo menor a 800 milisegundos (ms), un 13% lo envió entre 800 y 1200 ms y un 38% en un tiempo mayor a 1200 ms, lo que deja un 33% de peticiones fallidas.

Figura 253

Número de peticiones por segundo para 1000 usuarios



Asimismo, se llevarán a cabo pruebas que muestran las capacidades del sistema ante otras cargas de usuarios, siendo las cantidades a probar de: 10, 50, 100, 200, 500 y 800. Estas pruebas ayudaron a determinar cuál es el punto con el que se puede asegurar un buen desempeño del sistema.

Prueba con 10 usuarios

La simulación con 10 usuarios de las Figuras 254 y 255 muestra un 98% de las peticiones siendo enviadas en un tiempo menor a 800 ms y un 1% para peticiones entre 800 y 1200 ms. En conjunto con las peticiones que tardan más de 1200 ms (1%), se obtiene un 100% de mensajes enviados con una carga de 10 usuarios.

Figura 254

Prueba de carga con 10 usuarios

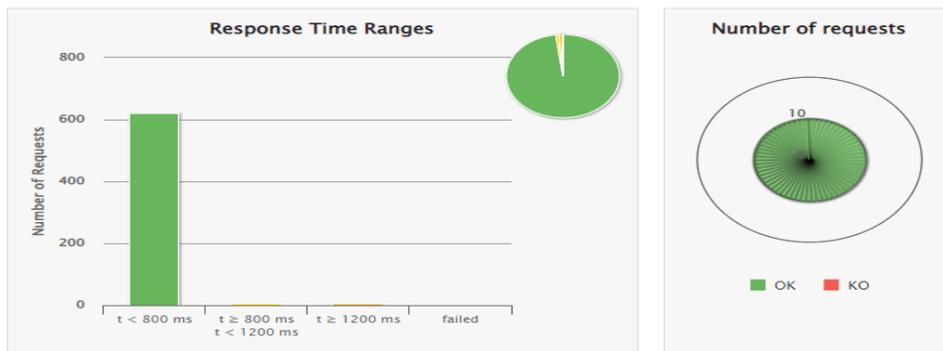
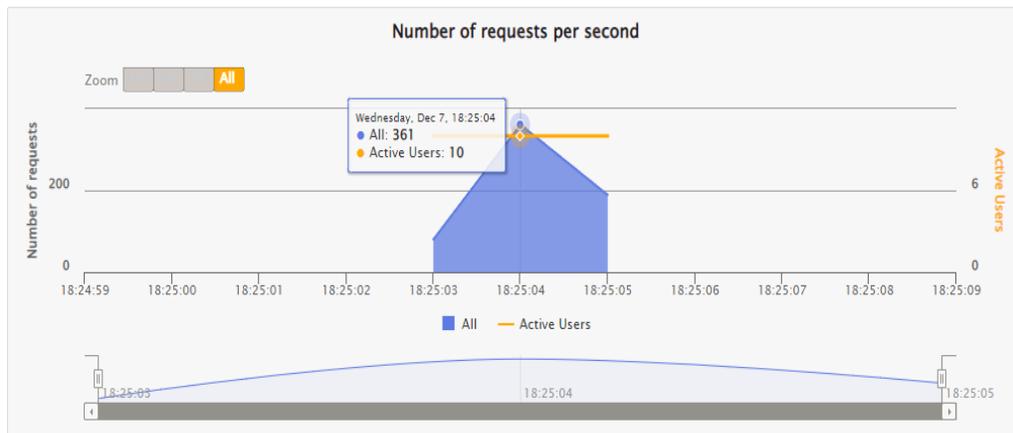


Figura 255

Número de peticiones por segundo para 10 usuarios



Prueba con 50 usuarios

La simulación con 50 usuarios responde con un 90%, 4%, y 5% de peticiones enviadas en tiempos menores a 800 ms, entre 800 y 1200 ms y mayor a 1200 ms respectivamente, mostrados en las Figuras 256 y 257, dando un 99% de peticiones enviadas con éxito y apenas un 1% de peticiones fallidas.

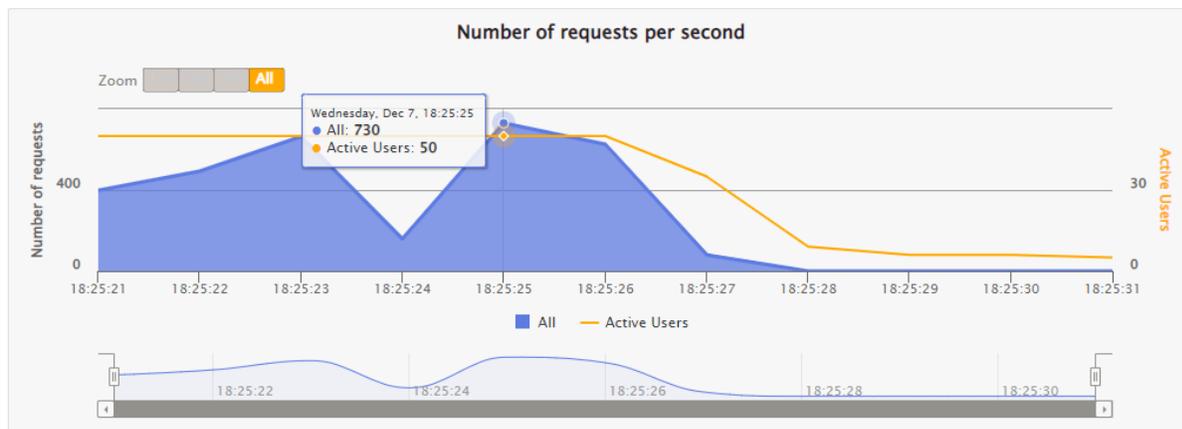
Figura 256

Prueba de carga con 50 usuarios



Figura 257

Número de peticiones por segundo para 50 usuarios



Prueba con 100 usuarios

La simulación con 100 usuarios de las Figuras 258 y 259 muestra un retraso para peticiones mayor a 800 ms, representando ahora un 38% de las peticiones enviadas y un 60% las pruebas menores a 800 ms. Las peticiones fallidas se mantienen con un 1% del total.

Figura 258

Prueba de carga con 100 usuarios

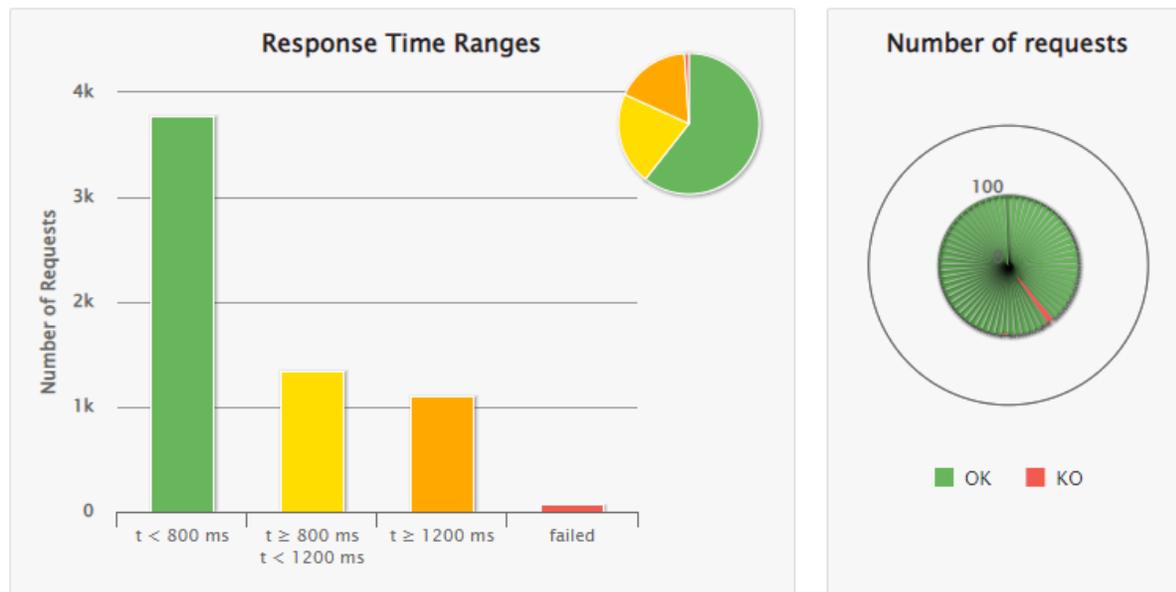
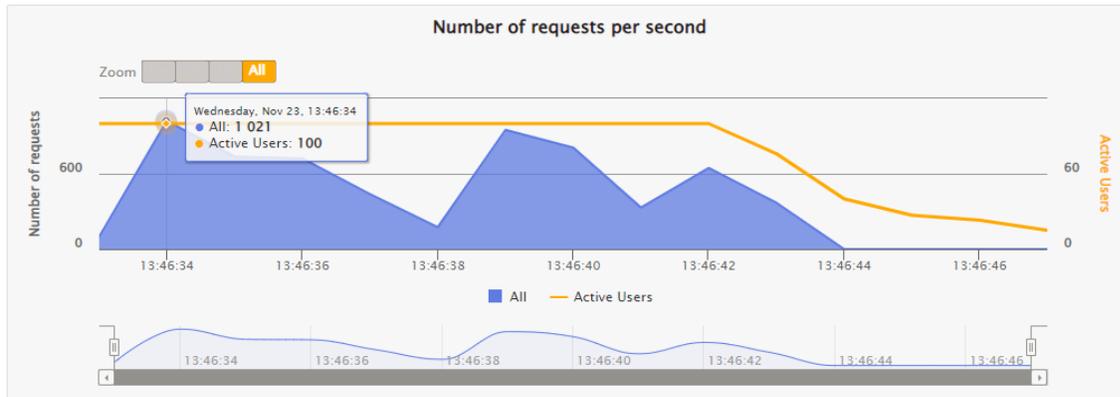


Figura 259

Número de peticiones por segundo para 100 usuarios



Prueba con 200 usuarios

La simulación con 200 usuarios de las Figuras 260 y 261 sigue la tendencia del aumento de peticiones enviadas en un tiempo mayor a 800 ms, siendo ahora un 59% del total de peticiones. Para las peticiones con un tiempo menor a 800 ms, ahora representa un 39% lo que deja un 2% de peticiones fallidas.

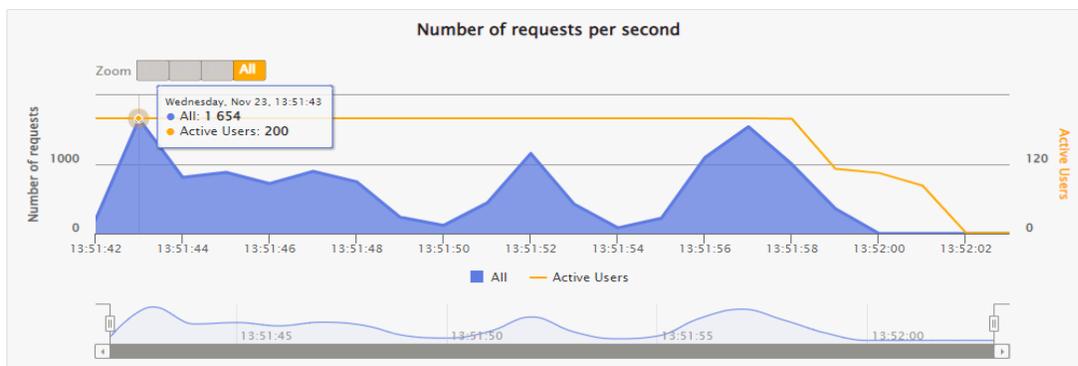
Figura 260

Prueba de carga con 200 usuarios



Figura 261

Prueba de carga con 200 usuarios

**Prueba con 500 usuarios**

La simulación con 500 usuarios de las Figuras 262 y 263 rompe con la tendencia y se puede observar nuevamente un repunte para los tiempos menores a 800 ms, siendo un 50% del total de peticiones, y los tiempos mayores a 800 ms representan un 36%. Por otro lado, las peticiones fallidas también comenzaron a elevarse, siendo un 14% del total.

Figura 262

Prueba de carga con 500 usuarios

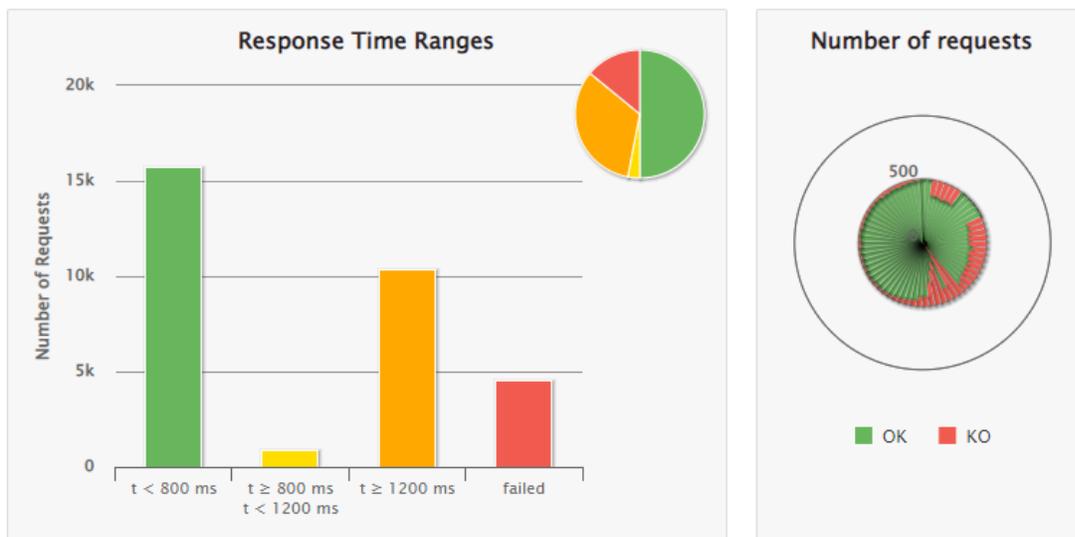
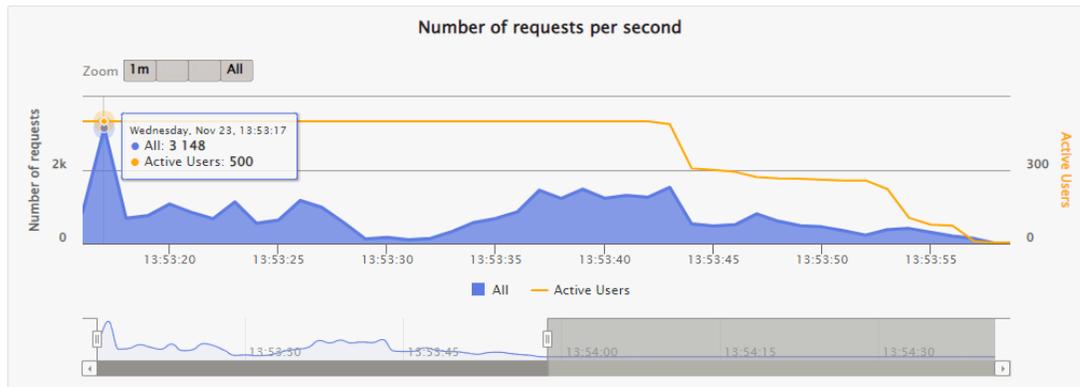


Figura 263

Número de peticiones por segundo para 500 usuarios



Prueba con 800 usuarios

La simulación con 800 usuarios de las Figuras 264 y 265 muestra que las peticiones fallidas aumentaron considerablemente, siendo un 35% del total de peticiones. Aunque el número de peticiones fallidas aumentó, la mayoría de peticiones se logró enviar, un 33% para tiempos menores a 800 ms y 32% para tiempos mayores a 800 ms.

Figura 264

Prueba de carga con 800 usuarios

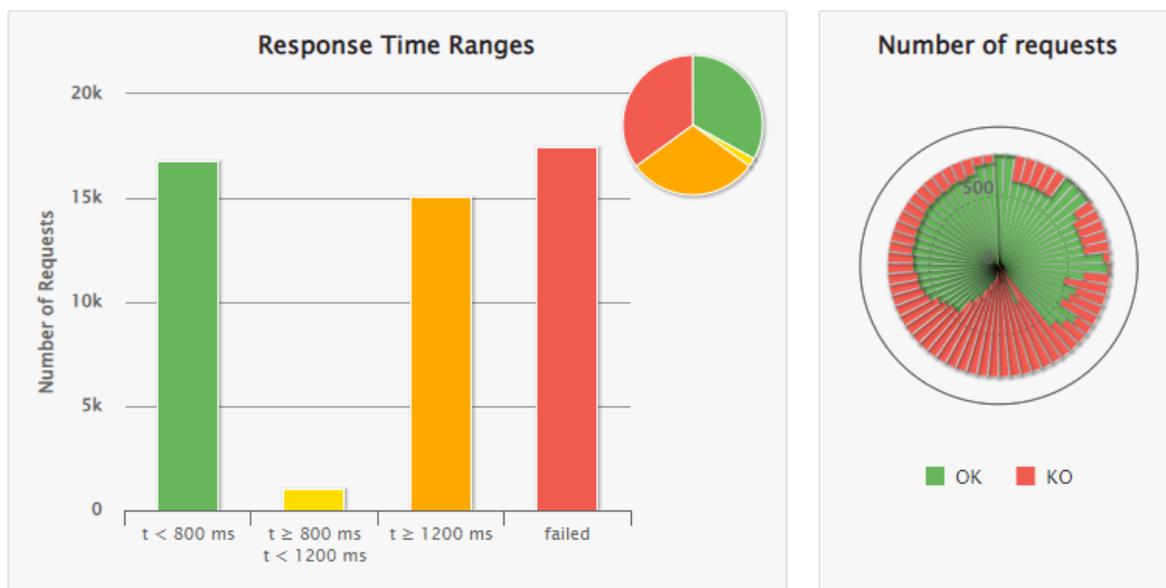
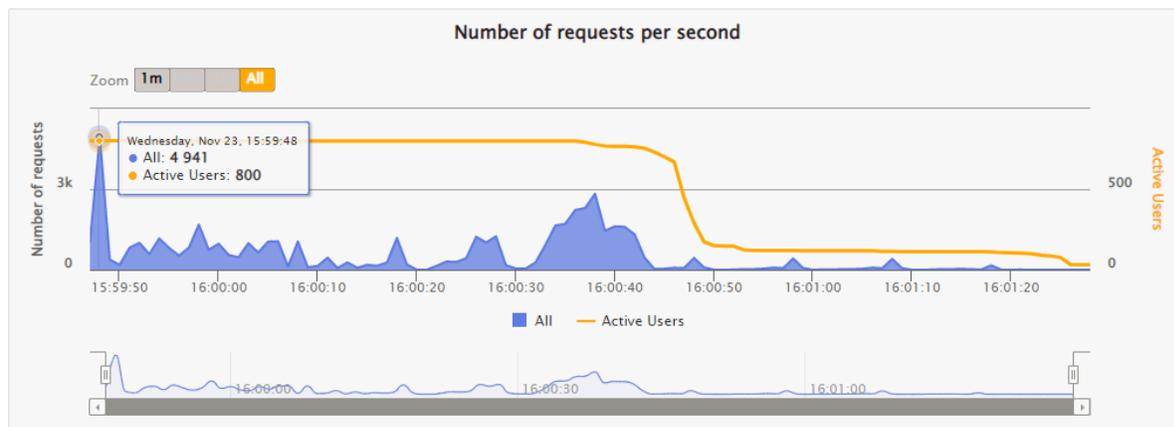


Figura 265

Número de peticiones por segundo para 800 usuarios



Se muestra la tabla 68, que es un resumen de las estadísticas obtenidas a lo largo de las simulaciones realizadas con las diferentes cargas de usuarios.

Tabla 68

Tabla resumen de las pruebas de carga

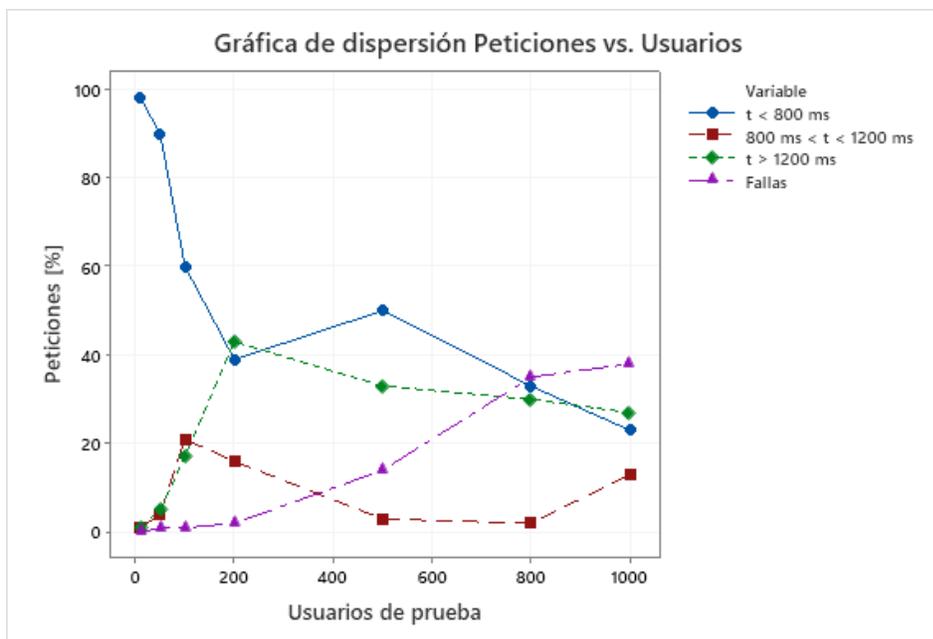
Usuarios de prueba	Éxito					Falla
	Peticiones Totales	Peticiones por segundo	t < 800 ms	800 ms < t < 1200 ms	t > 1200 ms	
10	630	210.00	98	1	1	0
50	3150	286.36	90	4	5	1
100	6300	420.00	60	21	17	1
200	12600	572.73	39	16	43	2
500	31500	366.28	50	3	33	14
800	50338	493.51	33	2	30	35
1000	63000	617.65	23	13	27	38

Los resultados obtenidos muestran datos favorables para el funcionamiento adecuado de la arquitectura de software ante cargas considerables de usuarios, sin embargo, y como es de esperarse, se nota un declive en el rendimiento a partir de los 800 usuarios, pese a que la mayoría de peticiones son enviadas y apenas un 35% falla, se nota un aumento de peticiones fallidas.

Se realizó un análisis estadístico de los resultados obtenidos en la Figura 266, esto muestra la tendencia de las peticiones recibidas en tiempos menores a 800 ms a disminuir, mientras que las peticiones para tiempos mayores e inclusive las peticiones fallidas aumentan.

Figura 266

Gráfica de dispersión de las peticiones de las pruebas de carga



Por otro lado, el sistema está diseñado para soportar una carga de un curso completo de robótica para aplicaciones de laboratorios remotos, es decir, aproximadamente unos 20 a 30 estudiantes, por lo que se puede esperar un rendimiento de al menos 90% de peticiones enviadas en menos de 800 ms, en otras palabras, al menos el 99% de peticiones enviadas con éxito.

Pruebas unitarias de Node-Red.

Las pruebas unitarias de integración se realizaron para los nodos en los que no intervienen salidas a la interfaz, para ello se utilizó el módulo de JavaScript de “mocha”, mismo en el que se implementa la lógica tras los procesos realizados en el flujo de Node-Red. En estas pruebas se analizó la respuesta que tuvieron los nodos basados en la lógica que se implementó en el flujo.

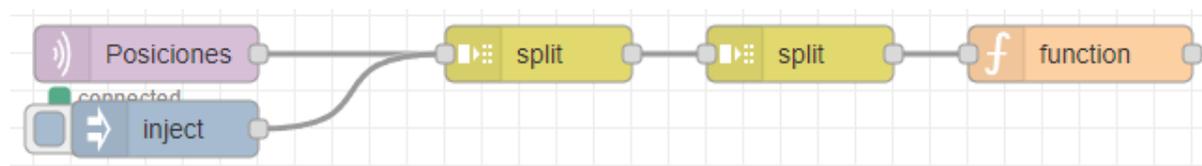
Para realizar las pruebas unitarias de forma correcta, se dividieron en función de las acciones significativas que realizan, para así tener más similaridad con la implementación de los nodos. las pruebas que se realizaron fueron las siguientes:

Prueba 1: Mensaje recibido

La primera prueba evalúa el tratamiento que se le da al mensaje de llegada desde el algoritmo de visión artificial mostrado en la Figura 267, para ello se realizaron tres evaluaciones.

Figura 267

Nodos de recepción de instrucciones de la visión artificial



- Separar el mensaje por saltos de línea
- Separar el mensaje por tabulaciones
- Eliminar espacios en blanco

Para ello se implementó un código en Javascript que permita realizar las pruebas que se mencionan anteriormente. Para comenzar se crea el mensaje con el que se va a trabajar, este es el mismo mensaje que se forma como respuesta del algoritmo de visión por

computador. Las figuras desde la 268 hasta la 271 muestran fragmentos del código implementado en Javascript para esta prueba.

Figura 268

Resultado esperado de la primera prueba unitaria

```
var result1 = "\t-41.83\t4.56\tAzul\tCuadrado\n"+
              "\t-29.13\t35.12\tAzul\tHexagono\n"+
              "\t14.94\t38.43\tAzul\tTriangulo\n"+
              "\t-4.36\t41.47\tRojo\tHexagono\n"+
              "\t-36.22\t27.79\tRojo\tRectangulo\n"+
              "\t35.6\t17.14\tRojo\tTriangulo\n"+
              "\t26.14\t42.17\tVioleta\tCuadrado\n"+
              "\t39.09\t29.45\tVioleta\tTriangulo";
```

En el grupo de Mensaje recibido, se implementa la lógica de los nodos que se implementaron y se compara con la respuesta que se debería obtener, y se compara usando una comparación de igualdad profunda (deepEqual), para verificar que los valores son los mismos.

Figura 269

Algoritmo de comparativa del resultado obtenido con el esperado de la primera función

```
describe('Mensaje recibido', function () {
  it('Separa el mensaje por saltos de línea', function () {
    result2 = result1.split('\n');
    assert.deepEqual(result2, [
      '\t-41.83\t4.56\tAzul\tCuadrado',
      '\t-29.13\t35.12\tAzul\tHexagono',
      '\t14.94\t38.43\tAzul\tTriangulo',
      '\t-4.36\t41.47\tRojo\tHexagono',
      '\t-36.22\t27.79\tRojo\tRectangulo',
      '\t35.6\t17.14\tRojo\tTriangulo',
      '\t26.14\t42.17\tVioleta\tCuadrado',
      '\t39.09\t29.45\tVioleta\tTriangulo'
    ]);
  });
});
```

Siguiendo la misma metodología, se realizan las pruebas restantes, como se muestran en las siguientes figuras

Figura 270

Algoritmo de comparativa del resultado obtenido con el esperado de la segunda función

```
it('Separa el mensaje por tabulaciones', function () {
  for (let i = 0; i < result2.length; i++) {
    result3.push(result2[i].split('\t'));
  }
  assert.deepEqual(result3, [
    ["", "-41.83", "4.56", "Azul", "Cuadrado"],
    ["", "-29.13", "35.12", "Azul", "Hexagono"],
    ["", "14.94", "38.43", "Azul", "Triangulo"],
    ["", "-4.36", "41.47", "Rojo", "Hexagono"],
    ["", "-36.22", "27.79", "Rojo", "Rectangulo"],
    ["", "35.6", "17.14", "Rojo", "Triangulo"],
    ["", "26.14", "42.17", "Violeta", "Cuadrado"],
    ["", "39.09", "29.45", "Violeta", "Triangulo"]
  ]);
});
```

Figura 271

Algoritmo de comparativa del resultado obtenido con el esperado de la tercera función

```
it('Elimina espacios en blanco', function () {
  for (let i = 0; i < result3.length; i++) {
    for (let j = 0; j < result3[i].length; j++) {
      if (result3[i][j] != ' ') {
        result4.push(result3[i][j]);
      }
    }
  }
  assert.deepEqual(result4, [
    "-41.83", "4.56", "Azul", "Cuadrado",
    "-29.13", "35.12", "Azul", "Hexagono",
    "14.94", "38.43", "Azul", "Triangulo",
    "-4.36", "41.47", "Rojo", "Hexagono",
    "-36.22", "27.79", "Rojo", "Rectangulo",
    "35.6", "17.14", "Rojo", "Triangulo",
    "26.14", "42.17", "Violeta", "Cuadrado",
    "39.09", "29.45", "Violeta", "Triangulo"
  ]);
});
```

Prueba 2: Formar mensajes de posiciones

La siguiente prueba evalúa el algoritmo para formar un mensaje que pueda ser usado para aplicaciones de visión artificial. De igual manera, se implementó una lógica en el script de prueba, similar a la que se realiza en el flujo mostrado en la Figura 272.

Figura 272

Flujo implementado en Node RED para formar mensajes



Figura 273

Algoritmo de comparativa del resultado obtenido con el esperado de la función de formar mensajes

```
describe('Formar mensaje posiciones', function () {
  it('Unir el mensaje normalizado', function () {
    let j = 0;
    for (let i = 0; i < result4.length; i += 4) {
      result5[j] = result4[i] + '\t' + result4[i + 1] + '\t' + result4[i + 2] + '\t' + result4[i + 3];
      j++;
    }
    assert.deepEqual(result5, ['-41.83\t4.56\tAzul\tCuadrado',
      "-29.13\t35.12\tAzul\tHexagono",
      "14.94\t38.43\tAzul\tTriangulo",
      "-4.36\t41.47\tRojo\tHexagono",
      "-36.22\t27.79\tRojo\tRectangulo",
      "35.6\t17.14\tRojo\tTriangulo",
      "26.14\t42.17\tVioleta\tCuadrado",
      "39.09\t29.45\tVioleta\tTriangulo"]);
  });
});
```

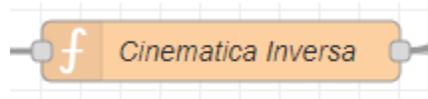
Nótese que se utilizan las respuestas de la prueba anterior para trabajar ya que esto forma parte de un proceso secuencial entre los datos.

Prueba 3: Cinemática inversa

Esta prueba toma el mensaje que se forma en la prueba anterior y debe dar como resultado los ángulos para poder efectuar el movimiento en el robot. Este proceso se consigue al calcular la cinemática inversa del robot SCARA con el nodo que se muestra en la Figura 274.

Figura 274

Nodo en Node-RED para el cálculo de la cinemática inversa

**Figura 275**

Algoritmo de comparativa del resultado obtenido con el esperado del cálculo de la cinemática inversa

```
describe('Cinematica inversa', function () {
  it('Cálculo de la cinemática inversa', function () {
    for (let i = 0; i < result5.length; i++) { ...
    }
    assert.deepEqual(result6, ["144.85X156.92",
      "107.25X139.34",
      "92.97X19.47",
      "69.49X158.56",
      "120.13X139.2",
      "55.32X12.57",
      "64.02X75.45",
      "46.61X65.88"]);
  });
});
```

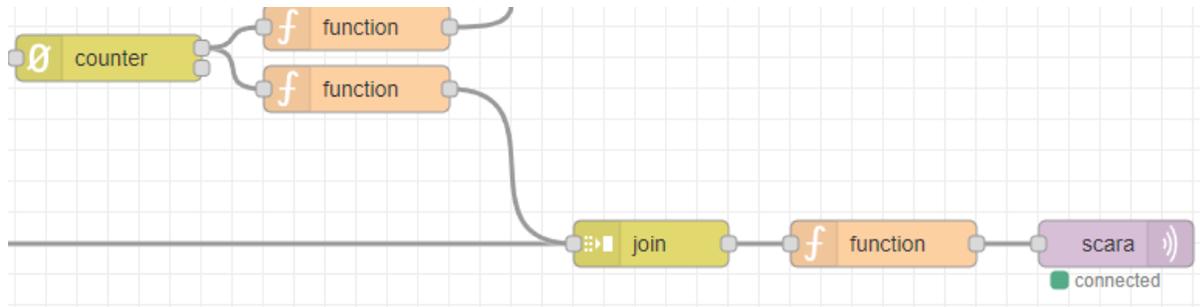
Cabe aclarar que dentro del bucle “for” se está calculando la cinemática inversa para cada coordenada, el código no se muestra por su excesiva longitud. Los ángulos con los que se compara fueron obtenidos usando la herramienta de cálculo por ordenador denominada Jupyter, que está basada en Python.

Prueba 4: Completar mensaje SCARA

Finalmente, esta prueba evalúa si el conjunto de mensajes que recibirá el SCARA cuenta con el orden correcto para realizar el algoritmo de visión por computador por defecto del robot.

Figura 276

Nodos para el envío de las instrucciones hacia el microordenador



Hay que tener en cuenta que, aunque parece que hay una gran cantidad de nodos a evaluarse, son nodos que realizan operaciones muy básicas que no justifican el realizar una prueba para cada una y se puede implementar una sola prueba para este proceso. El algoritmo utilizado en Javascript se muestra en la figura 277.

Figura 277

Algoritmo de comparativa del resultado obtenido con el esperado del orden de las instrucciones

```
describe('Completar mensaje SCARA', function () {
  it('Completar mensaje con retorno de pieza', function () {
    for (let i = 0; i < (result6.length * 2); i++) {
      if (i % 2 == 0) {
        result7[i] = result6[i / 2] + "x0x1x1x0";
      } else {
        result7[i] = "0x0x0x0x0x3";
      }
    }
    assert.deepEqual(result7, ["144.85x156.92x0x1x1x0",
      "0x0x0x0x0x3",
      "107.25x139.34x0x1x1x0",
      "0x0x0x0x0x3",
      "92.97x19.47x0x1x1x0",
      "0x0x0x0x0x3",
      "69.49x158.56x0x1x1x0",
      "0x0x0x0x0x3",
      "120.13x139.2x0x1x1x0",
      "0x0x0x0x0x3",
      "55.32x12.57x0x1x1x0",
      "0x0x0x0x0x3",
      "64.02x75.45x0x1x1x0",
      "0x0x0x0x0x3",
      "46.61x65.88x0x1x1x0",
      "0x0x0x0x0x3"]);
  });
});
```

Resultados de las pruebas unitarias

El código con todas las pruebas unitarias que se realizaron queda de la forma como se muestra en la Figura 278, posteriormente se procede a ejecutar el script usando el comando “npx mocha test.js” para obtener el resultado final, mostrado en la Figura 279.

Figura 278

Algoritmo para verificar el funcionamiento de las funciones de Node - RED

```
describe('Pruebas unitarias para el flujo', function () {
  describe('Mensaje recibido', function () {
    it('Separa el mensaje por saltos de línea', function () { ...
    });
    it('Separa el mensaje por tabulaciones', function () { ...
    });
    it('Elimina espacios en blanco', function () { ...
    });
  });
  describe('Formar mensaje posiciones', function () {
    it('Unir el mensaje normalizado', function () { ...
    });
  });
  describe('Cinemática inversa', function () {
    it('Cálculo de la cinemática inversa', function () { ...
    });
  });
  describe('Completar mensaje SCARA', function () {
    it('Completar mensaje con retorno de pieza', function () { ...
    });
  });
});
```

Figura 279

Resultado de las pruebas unitarias

```
PS C:\Users\alejo\test> npx mocha test.js

Pruebas unitarias para el flujo
  Mensaje recibido
    ✓ Separa el mensaje por saltos de línea
    ✓ Separa el mensaje por tabulaciones
    ✓ Elimina espacios en blanco
  Formar mensaje posiciones
    ✓ Unir el mensaje normalizado
  Cinemática inversa
    ✓ Cálculo de la cinemática inversa
  Completar mensaje SCARA
    ✓ Completar mensaje con retorno de pieza

6 passing (8ms)
```

Al ejecutar el comando se obtiene que todas las pruebas unitarias que se realizaron pasaron con éxito y en un tiempo de 8 milisegundos. Esta prueba refleja que cada una de las funciones que son utilizadas dentro de la programación de Node - RED se ejecuta de forma adecuada y su resultado es el esperado.

Capítulo 6: Resultados

Conclusiones

Las simulaciones realizadas del modelo mecánico y esquema eléctrico del robot SCARA determinan que el proceso de diseño y selección de materiales de los mecanismos y circuitos se realizó de forma adecuada. Las simulaciones mecánicas realizadas reflejan factores de seguridad y desplazamientos aceptables dentro de la normativa utilizada, y las simulaciones del circuito reflejan valores de consumo de potencia similares a los obtenidos por medición.

La construcción del robot SCARA en lo que corresponde al sistema mecánico y electrónico se realizó de manera satisfactoria, puesto que obtiene una respuesta confiable de los actuadores para los grados de libertad Q1 y Q2. Los grados de libertad presentaron una precisión de más del 97% para cada uno y adicionalmente muestra alta repetibilidad en la medición de ángulos, con valores que tienen baja dispersión para los ángulos de entrada.

La arquitectura de comunicación mediante la utilización de los servicios web del proveedor Amazon y Node - RED se implementó de manera satisfactoria, puesto que las pruebas de carga del sistema reflejan que está diseñado para soportar una carga de 500 usuarios de manera simultánea.

La utilización de Amazon Web Services permitió el alojamiento del servidor en la nube para el acceso remoto a la interfaz HMI desde varias localidades por medio de una página de inicio de sesión. Esto se reflejó al realizar pruebas de acceso desde diferentes redes en distintas ubicaciones.

La interfaz humano-máquina en Node - RED se diseñó e implementó de manera satisfactoria al cumplir las pautas de la normativa utilizada; esto se refleja en la prueba de usabilidad realizada a usuarios, obteniendo un puntaje cercano a 80 sobre 100 en la escala SUS, lo que refleja que el producto desarrollado es adecuado.

El algoritmo de visión artificial implementado cumple de manera satisfactoria la detección de los objetos de prueba que se propone en la aplicación por defecto, ya que presenta una precisión de más del 95% en cada una de las clases implementadas, lo que brinda confiabilidad al usar el algoritmo.

Recomendaciones

El funcionamiento del robot SCARA refleja una alta precisión para las instrucciones enviadas por el usuario de forma manual y automática por visión artificial, con un tiempo promedio de recolección y posicionamiento de objetos de 25 segundos por pieza. Se recomienda implementar elementos mecánicos y electrónicos que permitan mejorar el tiempo de recolección sin afectar a la precisión.

La arquitectura de comunicación implementada soporta alta carga de usuarios de forma simultánea, sin embargo, durante la utilización del sistema pueden existir conflictos en la visualización de los elementos de control en la HMI con varios usuarios manipulando a la vez, por lo que es recomendable utilizar de forma individual la plataforma.

El comportamiento de recolección de objetos por visión artificial, al ser modificable por el usuario al reprogramar el bloque de función correspondiente, puede presentar fallos si se realiza una configuración inadecuada, por lo que se recomienda revisar de forma meticulosa el código de la programación para evitar comportamientos erróneos al ejecutar instrucciones.

El algoritmo de visión artificial funciona dentro de los parámetros esperados y con buenos niveles de precisión, sin embargo, se recomienda mantener un espacio con buena iluminación debido a que el tipo de sensor que se utilizó es bastante sensible a dichos cambios y genera fallas en la detección de los objetos de prueba en el algoritmo actual.

El micro ordenador que se colocó en el robot responde de forma satisfactoria ante los requerimientos actuales manteniendo niveles manejables de temperatura y capacidad de procesamiento, pero de requerir realizar una tarea más compleja en el reconocimiento de

objetos con visión artificial, este podría saturarse y dejar de funcionar de forma apropiada, por lo que se recomienda implementar un micro ordenador con mayor capacidad de procesamiento.

Trabajos futuros

Implementación de un sistema de base de datos para la visualización de gráficos históricos de mayor tiempo, en lo que concierne a valores tomados por los grados de libertad del robot, de tal manera que se refleje de forma más detallada las acciones realizadas con la celda de manufactura.

Implementación de un sistema de inicio de sesión de mayor seguridad, utilizando métodos de validación con las cuentas de correo institucionales de estudiantes y personal de la Universidad de las Fuerzas Armadas para el acceso a la HMI que controla al robot SCARA.

Diseñar celdas de manufactura inteligente colaborativa que funcionen en conjunto con el robot manipulador diseñado e implementado, de tal forma que sea posible realizar simulaciones de procesos industriales de manera remota, para el aprendizaje de estudiantes de ingeniería.

Bibliografía

- Abdeetal, M., & Kermani, M. (2019). *An open-source integration platform for multiple peripheral modules with Kuka robots*. Obtenido de <https://doi.org/10.1108/IR-11-2019-0237>
- Amazon. (2022). *Amazon EC2*. Obtenido de <https://aws.amazon.com/ec2/>
- Bedaka, A., Vidal, J., & Lin, C. (2019). *Automatic robot path integration using three-dimensional vision and offline programming*. Obtenido de <https://doi.org/10.1007/s00170-018-03282-w>
- Brooke, J. (1995). *SUS: A quick and dirty usability scale*. . Obtenido de https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale
- Candell, R., Kashef, M., Yongkang, L., & Sebti, F. (2019). *A SysML representation of the wireless factory work cell*. Obtenido de <https://doi.org/10.1007/s00170-019-03629-x>
- Cardona, N. (2015). *Celdas flexibles de manufactura*. Obtenido de <https://prezi.com/skz4btking8r/celdas-flexibles-de-manufactura/>
- Cheng, Y. (2012). *Challenges and opportunities of the internet of things*. Obtenido de <https://ieeexplore.ieee.org/abstract/document/6164978/authors#authors>
- Co, H. T. (2015). *Global Connectivity Index*. Obtenido de <http://www.huawei.com/minisite/gci/en/index.html>
- Cortés, J. (2008). *Cinemática del brazo articulado PUMA*. Obtenido de https://personal.us.es/jcortes/Material/Material_archivos/Articulos%20PDF/RobotPUMA.pdf
- Cortés, J. (2008). *Representación de Denavit – Hartenberg*. Obtenido de https://personal.us.es/jcortes/Material/Material_archivos/Articulos%20PDF/RepresentDH.pdf

Costa-Castelló, R., Vallés, M., Jiménez, L., Díaz-Guerra, L., Valera, A., & Puerto, R. (2020).

Integración de dispositivos físicos en un laboratorio remoto de control mediante diferentes plataformas: Labview, Matlab y C/C++. *Revista iberoamericana de automática e Informática Industrial*, 7(1), 23-24.

doi:<https://polipapers.upv.es/index.php/RIAI/article/view/8503/8624>

Dirección de Educación Técnica de Buenos Aires. (2020). *Morfología básica de un robot*.

Obtenido de <https://cdn.continuemosestudiando.abc.gob.ar/uploads/e5644eb9-713f-45a0-9be2-552c4945bf33.pdf>

Fraccaroli, E., Lora, M., & Fummi, F. (2020). *Automatic generation of analog/Mixed signal virtual platforms for smart systems*. . Obtenido de <https://doi.org/10.1109/TC.2020.2970699>

FS Global. (2021). *¿Cuál es la diferencia entre modelo OSI y modelo TCP/IP?* Obtenido de <https://community.fs.com/es/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html>

Fujii, N., & Koike, N. (2017). *IoT remote group experiments in the cyber laboratory: A FPGA-based remote laboratory in the hybrid cloud*. Obtenido de <https://ieeexplore.ieee.org/abstract/document/8120315>

Gallegos, M. (2005). *Introducción a los servicios web*. Obtenido de

<http://repositorio.utn.edu.ec/bitstream/123456789/624/3/CAPITULO%20IV.pdf>

García, E. (2012). *Visión artificial*. Obtenido de

<https://ftp.isdi.co.cu/Biblioteca/BIBLIOTECA%20UNIVERSITARIA%20DEL%20ISDI/COLECCION%20DE%20LIBROS%20ELECTRONICOS/LE-1069/LE-1069.pdf>

Gausemeier, J., & Moehring, S. (2022). *VDI 2206 - A new guideline for desing of mechatronic systems*. Obtenido de

<https://www.sciencedirect.com/science/article/pii/S1474667017340351/pdf?md5=063a08023132d8ba61be0771710cc0f0&pid=1-s2.0-S1474667017340351-main.pdf>

- González, A., Díaz, I., & Domínguez, M. (2019). *Desarrollo de laboratorio remoto de investigación mediante la utilización de tecnologías industriales multi conectadas*. Obtenido de <https://doi.org/10.17979/spudc.9788497497169.277>
- HiveMQ. (2022). *MQTT: The Messaging and Data Exchange Protocol of the IoT*. Obtenido de <https://www.hivemq.com/mqtt/mqtt-protocol/>
- IBM. (2021). *Maximo Application Suite: Manufactura*. Obtenido de <https://www.ibm.com/cloud/es/products/maximo/manufacturing>
- IBM. (2022). *Servicios Web*. Obtenido de <https://www.ibm.com/docs/es/was/9.0.5?topic=services-web>
- Instituto Internacional para la Educación Superior en América Latina y el Caribe. (2020). *COVID-19 y educación superior: De los efectos inmediatos al día después*. Obtenido de <http://www.iesalc.unesco.org/wp-content/uploads/2020/05/COVID-19-ES-130520.pdf>
- IONOS Digital Guide. (2020). *Web services: servicios de máquina a máquina*. Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/web-services/>
- Jarro, J., & Pesántes, F. (2021). *Desarrollo de un laboratorio remoto de un banco PLC en la Universidad Politécnica Salesiana con IoT*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/21399>
- Jaziri, I., Charaabi, L., & Jelassi, K. (2018). *Remote web-based control laboratories using embedded Linux and field-programmable gate array*. Obtenido de <https://doi.org/10.1177/0959651818776542>
- Jiménez, R., Avilés, O., & Mauledoux, M. (2018). *Remote Lab for Robotics Applications*. Obtenido de <https://doi.org/10.3991/ijoe.v14i01.7674>
- Juárez, C. (2020). *Visión artificial para mejorar la calidad*. Obtenido de <https://thefoodtech.com/seguridad-alimentaria/vision-artificial-para-mejorar-la-calidad/>

Kamath, R., Balachandra, M., & Prabhu, S. (2019). *Raspberry Pi as Visual Sensor Nodes in Precision Agriculture: A Study*. Obtenido de

<https://doi.org/10.1109/ACCESS.2019.2908846>

Kio Networks. (2021). *¿Qué son y para qué sirven los protocolos de comunicación de redes?*

Obtenido de [https://www.kionetworks.com/blog/data-center/protocolos-de-comunicación-de-](https://www.kionetworks.com/blog/data-center/protocolos-de-comunicación-de-redes#:~:text=Un%20protocolo%20es%20un%20conjunto,a%20través%20de%20una%20red)

[redes#:~:text=Un%20protocolo%20es%20un%20conjunto,a%20través%20de%20una%20red](https://www.kionetworks.com/blog/data-center/protocolos-de-comunicación-de-redes#:~:text=Un%20protocolo%20es%20un%20conjunto,a%20través%20de%20una%20red)

Kouzes, R., Myers, J., & Wulf, W. (1996). *Collaboratories: Doing science on the internet*.

Obtenido de https://www.researchgate.net/publication/220475379_Collaboratories_Doing_Science_on_the_Internet

Kumar, A., Srikanth, P., Nayyar, A., Sharma, G., Krishnamurthi, R., & Alazab, M. (2020). A

Novel Simulated-Annealing Based Electric Bus System Design, Simulation, and Analysis for Dehradun Smart City. Obtenido de <https://doi.org/10.1109/ACCESS.2020.2990190>

Liu, Q., Liu, Z., Xu, W., Tang, Q., Zhou, Z., & D, T. (2019). *Human-robot collaboration in*

disassembly for sustainable manufacturing. Obtenido de

<https://doi.org/10.1080/00207543.2019.1578906>

Llamas, L. (2019). *¿Qué es MQTT? Su importancia como protocolo IoT*. Obtenido de

<https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>

Londoño, J. (2017). *Diseño e implementación de un laboratorio remoto IOT para una planta*

térmica didáctica. Obtenido de <https://repository.udistrital.edu.co/handle/11349/5416>

Longo, E., Redondi, A., Cesana, M., Arcia-Moret, A., & Manzoni, P. (2019). *MQTT: a spanning*

tree protocol for distributed MQTT brokers. Obtenido de

<https://arxiv.org/pdf/1911.07622.pdf>

- Macleod, C., Dobie, G., Pierce, S., Summan, R., & Morozov, M. (2018). *Machining-Based Coverage Path Planning for Automated Structural Inspection*. Obtenido de <https://doi.org/10.1109/TASE.2016.2601880>
- Magyari, A., & Chen, Y. (2021). FPGA Remote Laboratory Using IoT Approaches. *Electronics*, 10(18), 0-19. doi:<https://doi.org/10.3390/electronics10182229>
- Manou, E., Vosniakos, G., & Matsas, E. (2019). *Off-line programming of an industrial robot in a virtual reality environment*. Obtenido de <https://doi.org/10.1007/s12008-018-0516-2>
- Maoudj, A., Bouzouia, B., Hentout, A., Kouider, A., & Toumi, R. (2019). *Distributed multi-agent scheduling and control system for robotic flexible assembly cells*. Obtenido de <https://doi.org/10.1007/s10845-017-1345-z>
- Mar-Cornelio, O., Santana-Ching, I., & Gonzalez-Gulin, J. (2019). *Remote Laboratory System for practices of Control Engineering*. Obtenido de <https://doi.org/10.14483/23448350.14893>
- Márquez, M., Sánchez, M., Mejías, A., & Andújar, J. (2019). *Las ventajas del IoT en el diseño de laboratorios remotos: myGateway*. Obtenido de <https://doi.org/10.17979/spudc.9788497497169.294>
- Martínez, J., Sámano, V., Pérez, L., Uscanga, J., & Soto, H. (2021). Laboratorio remoto para sistemas de control discretos mediante internet de las cosas. *Pistas Educativas. Revista multidisciplinaria.*, 43(140), 161-180. doi:<http://itcelaya.edu.mx/ojs/index.php>
- Matsas, E., & Vosniakos, G. (2017). *Design of a virtual reality training system for human-robot collaboration in manufacturing tasks*. Obtenido de <https://doi.org/10.1007/s12008-015-0259-2>
- Miranda, R. (2016). *Cinemática y dinámica de robots manipuladores*. doi:https://books.google.es/books?hl=es&lr=&id=giF1EAAAQBAJ&oi=fnd&pg=PR3&dq=robot+manipuladores&ots=QWjzHFgxNR&sig=0fILemel99hCmXqd07b2Ni6KG_8#v=onepage&q=robot%20manipuladores&f=true

- Navarro, C., & Ramírez, M. (2018). *Mapeo sistemático de la literatura sobre evaluación docente (2013-2017)*. Obtenido de <https://doi.org/10.1590/S1678-4634201844185677>
- Okuda, M., Mizuya, T., & Nagao, T. (2017). *Development of IoT testbed using OPC UA and database on cloud*. Obtenido de <https://ieeexplore.ieee.org/document/8105726>
- Oracle. (2022). *What is IoT?*. Obtenido de <https://www.oracle.com/internet-of-things/what-is-iot/>
- Pertuz, C., & Salguero, J. (2021). *Diseño de un banco didáctico para el control de procesos analógicos mediante el uso de infraestructura IoT en un laboratorio de automatización*. Obtenido de <http://repository.uamerica.edu.co/handle/20.500.11839/8290>
- Power Prima. (2022). *Sistemas flexibles de manufactura*. Obtenido de <https://www.primapower.com/es/tecnologias/sistemas/sistemas-de-manufactura-flexible/psbb>
- Ramya, M., Purushothama, G., & Prakash, K. (2020). Design and Implementation of IoT Based Remote Laboratory for Sensor Experiments. *International Association of Online Engineering*, 14(9), 227-238. doi:<https://www.learntechlib.org/p/217837/>
- Rodríguez, E. (2015). *Análisis del sistema de medición mediante estudio de repetibilidad y reproducibilidad en Schott envases farmacéuticos SAS*. Obtenido de <https://repository.libertadores.edu.co/bitstream/handle/11371/475/RodriguezBarraganEusebio.pdf?sequence=2&isA>
- Rose, K., Eldridge, S., & Chapin, L. (2015). *Internet de las cosas - Una breve reseña*.
- Runji, J., & Lin, C. (2020). *Switchable Glass Enabled Contextualization for a Cyber-Physical Safe and Interactive Spatial Augmented Reality PCBA Manufacturing Inspection System*. Obtenido de <https://doi.org/10.3390/s20154286>
- Safeea, M., Neto, P., & Bearee, R. (2019). *On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case*. Obtenido de <https://doi.org/10.1016/j.robot.2019.07.013>

Scharfenberg, G., & Ludek, G. (2019). *New design methodology - using VHDL-AMS models to consider aging effects in automotive mechatronic circuits for safety relevant functions.*

Obtenido de <https://doi.org/10.23919/ae.2019.8867034>

Secretaria Nacional de Planificación. (2021). *Plan de Creación de Oportunidades.* Obtenido de www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-Creación-de-Oportunidades-2021-2025-Aprobado.pdf

Sklivanitis, G., Gannon, A., Tountas, K., Pados, D., Batalama, S., Reichhart, S., . . . Diaz, D. (2018). *Airborne cognitive networking: Design, development, and deployment.* Obtenido de <https://doi.org/10.1109/ACCESS.2018.2857843>

Steed, C. (2018). *A simulation-based approach to develop a holonic robotic cell.* Obtenido de <https://doi.org/10.1108/IR-07-2018-0149>

TIBCO. (2021). *What is Industrial Internet of Things?* Obtenido de <https://www.tibco.com/fr/reference-center/what-is-industrial-internet-of-things-iiot>

Torres, F. (2019). *Robot manipulador SCARA.* . Obtenido de https://www.researchgate.net/figure/Figura-1-Robot-manipulador-SCARA_fig1_335149942

Universidad Anáhuac Cancún. (2019). *Fisioterapia, un paso de la igualdad.* Obtenido de <https://www.anahuac.mx/cancun/fisioterapia-un-paso-de-la-igualdad>

Universidad de Vigo. (2004). *Servicios web.* . Obtenido de <https://ccia.esei.uvigo.es/docencia/SCS/0910/transparencias/Tema4.pdf>

Universidad Santiago de Chile. (2016). *Robots de Configuración Mezclada o SCARA (RRP).* Obtenido de <http://www.udesantiagovirtual.cl/moodle2/mod/book/view.php?id=24911&chapterid=228>

Valenzuela, F. (2015). *Aplicaciones de las metodologías QFD y diseño de experimentos a un proceso de manufactura en un contexto Seis Sigma.* Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lat/valenzuela_c_f/

- Valner, R., Kruusamäe, K., & Pryor, M. (2018). *TeMoto: Intuitive multi-range telerobotic system with natural gestural and verbal instruction interface*. Obtenido de <https://doi.org/10.3390/robotics7010009>
- Vargas, J., Cuero, J., & Torres, D. (2020). Laboratorios Remotos e IOT una oportunidad para la formación en ciencias e ingeniería en tiempos del COVID19: Caso de Estudio en Ingeniería de Control. *Revista Espacios*, 41(42), 188-198.
doi:<http://www.sweetpoison.revistaespacios.com/a20v41n42/a20v41n42p16.pdf>
- Vázquez, R., & Burgos, A. (2018). Laboratorio Remoto VISIR como herramienta pedagógica. *Extensionismo, Innovación Y Transferencia Tecnológica*, 4(0), 303-309. doi:
<https://revistas.unne.edu.ar/index.php/eitt/article/view/2896/2565>
- Vidal, S. (2019). *Aplicación de herramientas de calidad para evaluar características nutricionales de alimento balanceado para aves*.
doi:<https://www.dspace.espol.edu.ec/retrieve/3c165a53-99a6-4e12-97f4-b5a15b75d58a/D-CD102984.pdf>
- Wang, B. (2020). *Smart Manufacturing and Intelligent Manufacturing: A Comparative Review*.
- Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L., Kircaliali, A., & Harik, R. (2020). *A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence*. Obtenido de <https://doi.org/10.1016/j.jmsy.2020.06.012>
- XLSTAT. (2022). *Análisis detallado de sensibilidad y especificidad*. . Obtenido de <https://www.xlstat.com/es/soluciones/funciones/analisis-detallado-de-sensibilidad-y-especificidad>
- Yang, Y., Shen, S., Lui, K., Lee, K., Chen, J., Ding, H., . . . Shen, Y. (2017). *Ultrasonic robotic system for noncontact small object manipulation based on Kinect gesture control*. Obtenido de <https://doi.org/10.1177/1729881417738739>

- Yu, S., & Han, J. (2021). *Virtual reality platform-based conceptual design and simulation of a hot cell facility*. Obtenido de <https://doi.org/10.1007/s00170-021-07139-7>
- Zhang, J., & Letaief, K. (2019). *Mobile edge intelligence and computing for the internet of vehicles*. Obtenido de <https://doi.org/10.1109/JPROC.2019.2947490>
- Zhu, C., Pires, N., & Azar, A. (2019). *A novel multi-brand robotic software interface for industrial additive manufacturing cells*. Obtenido de <https://doi.org/10.1108/IR-11-2019-0237>

Apéndices