



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL
CLOUD COMPUTING PARA EL ROBOT KUKA KR3 R540.**

AUTOR: ROMERO CASTILLO, RONALD FLAVIO

DIRECTOR: ING. OROZCO BRITO, LUIS ALBERTO. MSc.

SANGOLQUÍ

2020



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES.

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL CLOUD COMPUTING PARA EL ROBOT KUKA KR3 R540.” fue realizado por el señor **ROMERO CASTILLO, RONALD FLAVIO** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 17 de enero del 2020

ING. LUIS ALBERTO OROZCO BRITO, MSc.

DIRECTOR

C. C.: 1710443803



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES.

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, ROMERO CASTILLO, RONALD FLAVIO, con cédula de identidad N°. 1723018121, declaro que el contenido, ideas y criterios del trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL CLOUD COMPUTING PARA EL ROBOT KUKA KR3 R540” es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 17 de enero del 2020

RONALD FLAVIO ROMERO CASTILLO

C. C.: 1723018121



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES.

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, **ROMERO CASTILLO, RONALD FLAVIO** autorizo a la Universidad de las Fuerzas Armadas ESPE, publicar el trabajo de titulación: “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL CLOUD COMPUTING PARA EL ROBOT KUKA KR3 R540**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 17 de enero del 2020

RONALD FLAVIO ROMERO CASTILLO

C. C.: 1723018121

DEDICATORIA

Mi trabajo de titulación está dedicado a mis padres, Flavio Romero y Mireya Castillo, Quienes en el transcurso de los años de vida que tengo, me han sabido guiar y corregir. Apoyándome y acompañándome en cada decisión, y es gracias a ellos, que hoy en día me convierto en ingeniero, les agradezco infinitamente por su amor, paciencia, trabajo y dedicación.

A mi hermano Henry, su esposa Katy, quien me han acompañado desde el inicio de mi carrera, apoyando y alentándome a seguir adelante, a perseverar y jamás rendirme, a fin de conseguir mis sueños.

A mi hermanita Silvia, quien, a la distancia, siempre estuvo brindándome su amor, estando pendiente de mí y del progreso de mi carrera, enseñándome que, con determinación, paciencia y valor todos los sueños se pueden alcanzar.

A mi tía, María Castillo, quien me ha apoyado incondicionalmente durante toda mi vida, aconsejándome y tratando de que sea mejor persona día a día.

AGRADECIMIENTO

Agradezco infinitamente a Dios y a la Virgen María, por todos los maravillosos años de vida y por hoy, permitirme convertirme en ingeniero.

A la Universidad de las Fuerzas Armadas – ESPE, a todos y cada uno de los ingenieros y maestros, que compartieron su conocimiento y experiencias conmigo, permitiendo que hoy en día tenga las facultades académicas y morales, para convertirme en Ingeniero Electrónico en Automatización y Control del Ecuador.

Mi más sincero y profundo agradecimiento, al Ingeniero Luis Orozco, quien, me guio, aconsejo y ayudo a adaptarme, en mis primeros años en la Universidad, en su calidad de director de la carrera de Ingeniería Electrónica en Automatización y Control. Al finalizar mi carrera, se convirtió en mi Director de Tesis, direccionando y ayudándome en todos los aspectos referentes a mi proyecto de titulación, demostrando gran paciencia, dedicación y, sobre todo amor por la enseñanza.

ÍNDICE DE CONTENIDO

DEDICATORIA.....	i
AGRADECIMIENTO	ii
ÍNDICE DE CONTENIDO	iii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS	xiii
RESUMEN	xviii
ABSTRACT	xix
CAPÍTULO I.....	1
1. INTRODUCCIÓN.....	1
1.1. Antecedentes	1
1.2. Justificación e importancia	4
1.3. Alcance	5
1.4. Objetivos.....	8
1.4.1. Objetivo general	8
1.4.2. Objetivos específicos.....	8
CAPÍTULO II	10
2. MARCO TEÓRICO	10
2.1. Cloud Computing.	10
2.1.1. Infraestructura como servicio (IaaS)	11
2.1.2. Plataforma como servicio (PaaS)	12

2.1.3.	Software como servicio (SaaS)	13
2.1.4.	Red como servicio (NaaS).....	13
2.2.	Servidores web.....	14
2.2.1.	Dirección IP	14
2.2.2.	Dominio	14
2.3.	Lenguajes web.....	15
2.3.1.	HTML- HyperText Markup Language.....	15
2.3.1.1.	Interfaz gráfica.....	15
2.3.2.	CSS- Cascading Style Sheets.....	17
2.3.3.	JAVA Script.....	17
2.3.4.	PHP- Hypertext Pre-Processor.....	17
2.4.	API's - Application Programming Interface.....	18
2.4.1.	Ejemplos de uso.	18
2.5.	Plataformas abiertas Cloud Computing.....	18
2.5.1.	Google Cloud Platform.....	19
2.5.1.1.	Visión API.....	19
2.5.1.2.	Tensor Flow.....	20
2.5.1.3.	Speech API	20
2.5.1.4.	Google Cloud SDK - Software Development Kit	21
2.5.1.5.	Credenciales Google Cloud.....	21

2.5.2.	Heroku: Cloud Application Platform	22
2.5.3.	Git.....	23
2.5.3.1.	GitHub.....	23
2.5.4.	Django	24
2.5.5.	Amazon Web Services. (AWS) - Cloud Computing Services.....	25
2.5.5.1.	Seguridad.....	25
2.5.5.2.	Servidor Virtual Privado. (VPS).....	25
2.6.	Visión artificial.....	26
2.6.1.	Características:	26
2.6.2.	Aplicaciones:.....	26
2.6.3.	Factores que intervienen.	27
2.6.3.1.	Cámaras.....	27
2.6.3.2.	Iluminación.....	28
2.6.4.	Open CV	29
2.6.4.1.	Características:.....	29
2.7.	Estructuras y protocolos de integración.....	30
2.7.1.	Profinet	30
2.7.2.	TCP/IP	31
2.7.2.1.	Protocolo de control de transmisión	31
2.7.2.2.	Protocolo de internet	31

2.7.2.3.	También incluye más de 100, protocolos entre los que están:	32
2.7.3.	HTTP - Hypertext Transfer Protocol	33
2.7.4.	Modbus	34
2.7.5.	Comunicación serial	35
2.7.6.	FTP - File Transfer Protocol	36
2.7.7.	OPC Server	37
2.7.8.	Sockets.....	37
2.8.	PLC Simatic S7-1200.....	38
2.8.1.	Web Server.....	39
2.8.2.	Comandos AWP.....	39
2.9.	KUKA KR3 R540	40
2.9.1.	Controlador KR C4	41
2.9.1.1.	Funciones del controlador KR C4	41
2.10.	Lenguajes de programación.	41
2.10.1.	KRL- Kuka Robot Language	41
2.10.2.	Lader	42
2.10.2.1.	Sus principales características son:.....	42
2.10.3.	Python	43
CAPÍTULO III		45
3. DISEÑO DEL SISTEMA Y APLICACIÓN.		45

3.1.	Arquitectura de Interconexión.	45
3.2.	Capa física.....	46
3.2.1.	Robot y el controlador.	46
3.2.2.	PLC.....	46
3.2.3.	Herramienta Térmica.....	47
3.2.3.1.	Características:.....	48
3.2.4.	Cámara Web.....	49
3.2.4.1.	Características de la cámara web:.....	49
3.3.	Capa lógica.....	50
3.3.1.	Servidor web del PLC.....	50
3.3.2.	Computador local	51
3.3.3.	Servidor Virtual.....	51
3.3.4.	Servidor FTP.....	51
3.3.5.	Control de Calidad.....	51
3.3.6.	Sitio de Alojamiento web.	52
3.4.	Capa de Aplicación.	52
3.4.1.	Interfaz web del PLC.....	52
3.4.2.	Interfaz web Cloud	52
3.5.	Arquitectura de componentes	53
3.6.	Sistema de corte.	53

3.6.1.	Módulo de seguridad.....	53
3.6.2.	Interfaz web del PLC.....	55
3.6.3.	Bridge-Lader.....	57
3.6.4.	Control de movimiento.....	58
3.6.5.	Corte láser.....	59
3.7.	Sistema de control de calidad.....	59
3.7.1.	Captura de imagen.....	59
3.7.2.	Envio de Imagen.....	60
3.7.3.	Servidor FTP.....	61
3.7.4.	Identificación de imagen en el servidor FTP.....	61
3.7.5.	Módulo de detección de Errores.....	63
3.7.6.	Interfaz web Cloud.....	64
3.8.	Arquitectura Cloud Computing.....	65
3.8.1.	Red como servicio.....	65
3.8.2.	Software como servicio.....	66
3.8.3.	Plataforma como servicio.....	67
3.8.4.	Infraestructura como servicio.....	67
3.9.	Diseño de prácticas de laboratorio.....	68
3.9.1.	Práctica 1: Introducción a los servidores web.....	68
3.9.1.1.	Objetivo General.....	68

3.9.1.2.	Objetivo Específicos.	68
3.9.1.3.	Descripción general de actividades.....	68
3.9.2.	Práctica 2: Servicios Cloud Computing.....	71
3.9.2.1.	Objetivo General.....	71
3.9.2.2.	Objetivo Específico.....	71
3.9.2.3.	Descripción general de actividades.....	71
CAPÍTULO IV.....		74
4. IMPLEMENTACIÓN DE ARQUITECTURA CLOUD COMPUTING.....		74
4.1.	Activación de Web Server	74
4.2.	Configuración del módulo de seguridad.....	76
4.3.	Interfaz web del PLC.....	76
4.3.1.	Tamaño de la aplicación web.....	77
4.3.2.	Actualización de aplicación web.....	77
4.3.3.	Comunicación con variables del PLC.	78
4.4.	Bridge-Lader.	79
4.4.1.	Aspectos importantes.....	82
4.5.	Control de movimiento.....	84
4.5.1.	Configuración Profinet del robot KUKA KR3 R540.....	84
4.6.	Corte láser	87
4.7.	Captura de imagen.....	90

4.8.	Envío de Imagen.....	90
4.9.	Servidor Virtual.....	92
4.10.	Servidor FTP.....	93
4.11.	Procesamiento de imagen.	96
4.11.1.	Sincronización del servidor FTP y el procesamiento de imagen.....	96
4.11.2.	Procesamiento de Imagen 2.....	97
4.12.	Modulo de detección de errores	97
4.12.1.	Sitio de Alojamiento web.....	99
CAPÍTULO V		101
5.	PRUEBAS Y RESULTADOS.....	101
5.1.	Pruebas y resultados de funcionamiento.....	101
5.1.1.	Control de movimiento.....	101
5.1.2.	Interfaz web del PLC.....	102
5.1.3.	Corte térmico.....	103
5.1.4.	Sistema de control de calidad.....	105
5.1.5.	Interfaz web Cloud.	106
5.1.6.	Tiempo de proceso Cloud Computing.....	108
5.1.6.1.	Residencial.	108
5.1.6.2.	Institucional.	108
5.1.6.3.	Servidor virtual.	109

5.1.1.	Velocidad de red de proceso Cloud Computing.	109
5.1.1.1.	Residencial.	109
5.1.1.2.	Institucional.	110
5.1.1.3.	Servidor virtual.	110
5.2.	Resultados de aprendizaje de estudiantes	111
5.3.	Análisis de resultados.	112
CAPÍTULO VI.....		115
6.	CONCLUSIONES Y RECOMENDACIONES	115
6.1.	Conclusiones	115
6.2.	Recomendaciones.....	117
7.	BIBLIOGRAFÍA.....	118
8.	ANEXOS.....	125

ÍNDICE DE TABLAS

Tabla 1. Etiquetas HTML.	16
Tabla 2. Comandos AWP	39
Tabla 3. Características KUKA KR3 R540.	40
Tabla 4. Símbolos básicos de Lader.	43
Tabla 5. Resultados del sistema de control de calidad.	105
Tabla 6. Tiempo de proceso Cloud Computing residencial.	108
Tabla 7. Tiempo de proceso Cloud Computing institucional.	108
Tabla 8. Tiempo de proceso Cloud Computing en el servidor virtual.	109
Tabla 9. Velocidad de proceso Cloud Computing residencial.	109
Tabla 10. Velocidad de proceso Cloud Computing institucional.	110
Tabla 11. Velocidad de proceso Cloud Computing en el servidor virtual.	110

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de microservicios.	6
Figura 2. Sistema Cloud Computing del robot KUKA KR3 R540.	7
Figura 3. Arquitectura Cloud Computing.	11
Figura 4. Ejemplos de IaaS.	12
Figura 5. Ejemplos de PaaS.	12
Figura 6. Ejemplos de SaaS.	13
Figura 7. Ejemplo URL.	14
Figura 8. Dirección IP.	14
Figura 9. Estructura básica HTML.	16
Figura 10. Google Cloud Platform.	19
Figura 11. Visión API.	20
Figura 12. TensorFlow.	20
Figura 13. Speech API.	20
Figura 14. Heroku: Cloud Application Platform.	22
Figura 15. Django.	24
Figura 16. Cámara lineal.	27
Figura 17. Cámara matricial.	27
Figura 18. Cámara digital.	27
Figura 19. Cámara web.	28
Figura 20. Cámara IP.	28
Figura 21. Red Profinet.	30

Figura 22. Protocolo TCP/IP.....	31
Figura 23. Modelo TCP/IP.....	32
Figura 24. Protocolo HTTP.....	33
Figura 25. Relación maestro-esclavo.....	34
Figura 26. Tipos de conexiones seriales.	36
Figura 27. PLC S7-1200.	38
Figura 28. KUKA KR3 R540.....	40
Figura 29. Arquitectura de Interconexión.	45
Figura 30. Resistencia eléctrica de corte.	47
Figura 31. Diodo láser.	48
Figura 32. Cabezal láser.....	49
Figura 33. Cámara web.....	50
Figura 34. Arquitectura de componentes.....	53
Figura 35. Algoritmo del módulo de seguridad.	54
Figura 36. Estructura de la interfaz web del PLC.	55
Figura 37. Algoritmo de la interfaz web del PLC.	56
Figura 38. Algoritmo para diseñar el Bridge-Lader.	57
Figura 39. Algoritmo de control de movimiento del robot.....	58
Figura 40. Activación del Pacer	59
Figura 41. Algoritmo de Captura de imagen	60
Figura 42. Algoritmo de envío de Imagen.	60
Figura 43. Algoritmo de recepción FTP.....	61
Figura 44. Algoritmo para el procesamiento de la imagen 1.	62
Figura 45. Algoritmo para el procesamiento de la imagen 2.	62

Figura 46. Algoritmo de detección de errores.....	63
Figura 47. Estructura de la interfaz web Cloud.....	64
Figura 48. Implementación red como servicio.....	66
Figura 49. Uso de software como servicio	66
Figura 50. Uso de plataforma como servicio	67
Figura 51. Uso de infraestructura como servicio.	67
Figura 52. Activación de Web Server.	69
Figura 53. Código HTML, relacionado con AWP	69
Figura 54. Lenguaje Lader, relacionado con el código AWG y HTML.	70
Figura 55. Servidor virtual remoto.	72
Figura 56. Configuración de propiedad del PLC S7-120.	74
Figura 57. Activación de Web Server.	74
Figura 58. Páginas de usuario en el PLC S7-1200.....	75
Figura 59. Bloque del servidor web.	75
Figura 60. Instrucción WWW.	75
Figura 61. Usuarios con acceso a la aplicación web.	76
Figura 62. Autorización de acceso a la aplicación web.....	76
Figura 63. Código CSS.....	77
Figura 64. Instrucción de acondicionamiento de tamaño.	77
Figura 65. Función para evitar recarga de la aplicación web.....	78
Figura 66. Sintaxis del lenguaje AWP.....	78
Figura 67. Código para modificar el estado de las variables de PLC.....	79
Figura 68. IP del robot KUKA.	79
Figura 69. a) IP del PLC, b) IP del computador.....	80

Figura 70. Opciones de TIA Portal.....	80
Figura 71. Librería GSDML de KUKA.....	81
Figura 72. Controlador KR C4 de KUKA.....	81
Figura 73. Ubicación de controlador KR C4 en TIA Portal.....	81
Figura 74. Configuración de entradas y salidas módulo Profinet KR C4.....	82
Figura 75. Entradas y salidas Profinet.....	83
Figura 76. Programación de entradas y salidas en el PLC.....	83
Figura 77. Comunicación Profinet.....	84
Figura 78. Configuración inicial WorkVisual.....	85
Figura 79. Unidad de control Profinet activa.....	85
Figura 80. Configuración de módulo Profinet.....	86
Figura 81. Configuración de entradas y salidas Profinet KUKA.....	86
Figura 82. Inicialización de variables en C++.....	87
Figura 83. Bucle infinito While.....	87
Figura 84. Control del láser en C++.....	88
Figura 85. Diseño del circuito de control en Proteus.....	88
Figura 86. PCB del circuito de control.....	89
Figura 87. Placa del circuito de control.....	89
Figura 88. Imagen de la pieza cortada.....	90
Figura 89. Cliente FTP en CMD.....	91
Figura 90. Cliente FTP en Python.....	91
Figura 91. Cliente FTP por software.....	92
Figura 92. Servidor virtual privado (VPS).....	93
Figura 93. Server Manager.....	94

Figura 94. Web Server (IIS).....	94
Figura 95. Configuración de usuario y clave FTP	94
Figura 96. Internet Information Services (IIS) Manager	95
Figura 97. Adición sitio FTP.....	95
Figura 98. Permisos de salida.....	95
Figura 99. Carpeta FTP del servidor.	96
Figura 100. Sincronización de FTP y procesamiento de imagen.	96
Figura 101. Respuesta de Google Cloud Visión.	97
Figura 102. Bordes de la imagen cortada.	98
Figura 103. Defectos de calidad en la pieza cortada.	98
Figura 104. Aprobación o rechazo de la pieza cortada.....	99
Figura 105. Aplicación Cloud Computing.....	100
Figura 106. Control de movimiento del robot KUKA KR3 R540.	101
Figura 107. Ventanas de la aplicación web.	102
Figura 108. Ajuste de velocidad y potencia 1.	103
Figura 109. Ajuste de velocidad y potencia 2.....	103
Figura 110. Ajuste de velocidad y potencia 3.....	103
Figura 111. Ajuste de velocidad y potencia 4.....	104
Figura 112. Ajuste de velocidad y potencia 5.....	104
Figura 113. Ajuste de velocidad y potencia 6.....	104
Figura 114. Piezas cortadas con láser.....	105
Figura 115. Resultados de la aplicación Cloud Computing.....	107
Figura 116. Softwares en el servidor virtual.....	107
Figura 117. Interfaz gráfica de la práctica 1.	111

Figura 118. Resultados obtenidos en la práctica 2. 111

RESUMEN

El presente trabajo describe y analiza la potencialidad de las aplicaciones y servicios Cloud Computing, aplicado al proceso de corte térmico y control de calidad mediante visión artificial. Para ello se implementa un sistema multiplataforma, en el que intervienen los principales proveedores de servicios en la nube, tales como: Amazon Web Server y Google Cloud Platform, entre otros, integrados a una red local híbrida en la que existen protocolos tanto de comunicación industrial y de comunicación ethernet. La red local está conformada por un robot KUKA KR3 R540, con su respectivo controlador KR C4, conectado mediante Profinet a un PLC s7-1200, el cual, por su parte está conectado a un cabezal láser, que efectúa el corte de piezas de fomix y aun computador local, que cumple la función de capturar una imagen de la pieza cortada y enviarla la mediante protocolos de comunicación FTP, una vez configuradas las credenciales de identificación, siendo de esta forma el cliente de un servidor virtual, ubicado en la nube de Amazon Web Services, donde se ejecuta todo el proceso de control de calidad y las aplicaciones Cloud Computing. Por su parte, para la ejecución de todo el proceso existe una aplicación web, alojada en el Web Server propio del PLC.

PALABRAS CLAVE:

- **CLOUD COMPUTING**
- **CORTE LÁSER**
- **VISIÓN ARTIFICIAL**
- **ROBOT KUKA KR3 R540**
- **PLC S7-1200**

ABSTRACT

The present project describes and analyzes the potential of Cloud Computing applications and services, applied to the thermal cutting process and quality control through artificial vision. For this, a multiplatform system is implemented, in which the main cloud service providers are involved, as: Amazon Web Server and Google Cloud Platform, among others, integrated into a hybrid local network in which there are both industrial communication protocols and ethernet communication. The local network is made up of a KUKA KR3 R540 robot, with its respective KR C4 controller, connected by Profinet to PLC s7-1200, which, in turn, is connected to a laser head, which cuts phoenix parts and a local computer, that fulfills the function of capturing an image of the cut piece and sending it through FTP communication protocols, once the identification credentials have been configured, thus being the client of a virtual server, located in the Amazon Web Services cloud, where the entire quality control process and Cloud Computing applications are executed. For its part, for the execution of the entire process there is a web application, hosted on the PLC's own web server.

KEY WORD:

- **CLOUD COMPUTING**
- **LASER CUT**
- **ARTIFICIAL VISION**
- **ROBOT KUKA KR3 R540**
- **PLC S7-1200**

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Antecedentes

La Universidad de las Fuerzas Armadas ESPE, posee en su laboratorio de Robótica Industrial, siete células robotizadas basadas en tecnología KUKA y SIEMENS conformadas cada una por: un robot KUKA KR3 R540, un controlador KUKA KR C4 Compac, un PLC s7-1200, un SWITCH CSM1277 y un panel de visualización KTP700 PN Basic, que están limitadas a un control local y una arquitectura cerrada por lo que es importante mejorar su funcionalidad y ampliar la cantidad de aplicaciones existentes; como se ha realizado en los siguientes ejemplos:

El desarrollo de aplicaciones industriales para células robotizadas multipropósito, ha tomado una gran relevancia en las investigaciones realizadas por parte de los estudiantes de la Universidad de las Fuerza Armadas- ESPE, en las cuales se implementaron sistemas de paletizado de piezas de acrílico por parte del Sr. Julio Mariño y selección de botellas mediante visión artificial por parte de la Srta. Aracely Pillajo, en el laboratorio de Robótica Industrial y PLC de la ESPE, durante el año 2019.

En (Castillo, 2013) se realizó el “Diseño de un laboratorio Cloud Computing de robots”, aplicado a un sistema de manufactura flexible; el cual contiene un robot CNC (controlado numéricamente por computadora) y un robot industrial tipo antropomórfico, este entorno es denominado flexible porque procesa varias cantidades y productos, permitiendo ajustar su comportamiento de acuerdo a la demanda, que la manufactura tradicional no puede lograr. Para el control se utiliza TeamViewer, un software que permite la conexión remota a otro dispositivo.

Entre sus características está compartir y controlar escritorios de otros computadores, que tengan instalado el software y compartan previamente la clave de acceso. A continuación, se plantea la réplica de un Teach pendant por hardware (consola de programación) y se realiza la vinculación con el sistema de robot flexible mediante un servidor LAMP (sistema de infraestructura de internet). Obteniendo como resultados un servidor tipo LAMP para la administración del tiempo de uso de los equipos, así como un dispositivo para manipular el Teach pendant del robot por hardware.

En (Rosado, 2008) se realizó el control Cloud Computing de un robot ABB IRB140 y se adicionó elementos de seguridad como: barreras fotoeléctricas para poder controlar con seguridad el robot a distancia. Para el control por hardware se utilizó la tarjeta Combi con el protocolo de comunicación DeviceNet, que envía y recibe información del PLC conectado al robot y para gestionar el ingreso al robot se utiliza el Servidor Web Apache, obteniendo resultados positivos ya que se logró desarrollar un entorno en el que personas sin conocimientos previos de programación pudieran interactuar con un robot mediante teleoperación , sin la necesidad de estar físicamente en el mismo lugar donde se encuentra el robot .

La enseñanza de programación de robots industriales tiene alta relevancia como se muestra en (Ariza & Amaya, 2011), el artículo describe el desarrollo y funcionamiento de un laboratorio Cloud Computing para la enseñanza de automatización, el control Cloud Computing se aplica al robot manipulador Mitsubishi RV-E2, el robot se encuentra ubicado en un laboratorio de CIM (Manufactura integrada por computadora) de la Universidad Militar de Nueva Granada, para este fin se realizó una aplicación web que se ejecuta en los servidores propios de la Universidad y se instaló una cámara que transmite el video de la estación de trabajo, como servidor remoto se utilizó Java System Application Server Platform Edition, y la aplicación web fue desarrollada en el

software NetBeans, el servidor de base de datos fue MySQL Server para almacenar toda la información dinámica de la aplicación web y como software de acceso remoto se utilizó winconnect XP. Luego de todo el proceso se optimizó la utilización de los equipos para realizar prácticas de apoyo académico. Ya que estos equipos tienen un costo elevado y es difícil que una institución académica pueda contar con varios de ellos, representó una gran mejora para el aprendizaje de los estudiantes asignando horarios establecidos para el ingreso de cada uno y permitiendo acceder al sistema desde cualquier lugar con acceso a internet.

Por su parte en (Alian, Fernández, Martínez, & Ortiz, 2007), se implementó todo un laboratorio de control basado en internet, para lo cual se empleó un sistema de experimentación remoto que puede ser en lazo abierto o cerrado, con controladores que se ejecutan en el servidor o en el cliente de acuerdo a lo requerido por el usuario. El servidor utilizado es un computador personal Pentium-III que se encuentra conectado por un lado a la red y por el otro al laboratorio mediante el uso de tarjetas de adquisición de datos PC_LAB_816, la aplicación web realizada se encarga de gestionar el flujo de usuarios y seleccionar el modo de operación, ya sea en lazo abierto o cerrado.

Al concluir el proyecto los resultados fueron satisfactorios, *“debido a que la experiencia adquirida, demuestra que un entorno de experimentación remota es factible y que los alumnos se adaptan rápidamente a un escenario educativo que implique la utilización de laboratorios remotos”*. (Alian, Fernández, Martínez, & Ortiz, 2007)

1.2. Justificación e importancia

Analizando la investigación de los antecedentes, se puede denotar que las limitaciones de la célula robotizada multipropósito de la ESPE, referentes al control local y una arquitectura cerrada, se pueden solucionar mediante los servicios que proporciona el Cloud Computing, por lo que es importante desarrollar un sistema que permita realizar aplicaciones reales como: el corte térmico de fomix y el control de calidad mediante visión artificial, ejecutando órdenes a distancias aproximadas de 15 metros y con una arquitectura abierta.

Adicionalmente la propuesta planteada en el presente proyecto, tiene un gran impacto en desarrollo académico y del aprendizaje, ya que permite a los estudiantes acceder a servidores virtuales que alojan programas de desarrollo como: Python, Arduino, KUKA Sim Pro, Proteus, entre otros, desde cualquier lugar conectado a internet.

En Ecuador existen varias empresas dedicadas al ensamblaje de autos y la fabricación de tuberías que poseen un vasto número de robots conectados a internet en sus plantas de producción, y esto se debe a la revolución de la industria 4.0 que está fundamentada en la conexión de los dispositivos y máquinas industriales a la internet. Es el caso de la ensambladora de autos General Motor Ecuador (GM OBB) que ha comenzado a aplicar los desarrollos de la industria 4.0 en su planta en Quito; Trabajando con realidad virtual, simulaciones, impresiones 3D y capacitando a sus operarios con gafas virtuales. (Ramos X. , 2018).

En base a los nuevos avances de la industria ecuatoriana el presente proyecto encaja en las propuestas de desarrollo empresarial de nuestro país, ya que el objetivo de los robots no es reemplazar a los humanos sino brindarles mayor confort y seguridad en el trabajo.

El corte térmico, tiene una gran aceptación en el mercado ecuatoriano como se menciona en (Tikay, 2018), ya sea para realizar elementos decorativos para el hogar o comerciales. La empresa Tikay se especializa en el corte y grabado de materiales delicados como papeles, cartulinas y materiales limpios como acrílicos y plásticos.

Los equipos utilizados para realizar los cortes poseen una gran potencia, generando altas temperaturas y desprendimiento de gases, que puede resultar en un potencial peligro para la seguridad de los operarios, por tal motivo, el presente proyecto pretende aplicar Cloud Computing para realizar la ejecución de este proceso a distancias de 15 metros, permitiendo salvaguardar la integridad de los operarios, solventado así los requerimientos planteados en el Art,58 del Reglamentos de Seguridad y Salud de la República Del Ecuador. (Empleo, 2012)

1.3. Alcance

El desarrollo del presente proyecto se centrará en ampliar la funcionalidad del robot KUKA KR3 R540 y los componentes de la célula robotizada multipropósito, disponible en el laboratorio de Robótica Industrial y PLC de la Universidad de la Fuerzas Armadas ESPE; a fin de potenciar su utilización en la industria 4.0. mediante Cloud Computing (empleando servidores online) y una interfaz gráfica accesible desde cualquier dispositivo con conexión a internet.

Para demostrar la ampliación y mejora de las funcionalidades del robot, se propone realizar 2 aplicativos que emplean servicios/servidores alojados en la nube: el corte térmico de planchas de fomix y el control de calidad de las piezas cortadas, mediante visión artificial.

En la primera etapa del proyecto, se implementará una puerta de enlace para conectar los dispositivos de hardware (controlador KR C4, PLC s7-1200, PC) a la nube. Después, se analizarán e implementarán los servicios/servidores de Cloud Computing más adecuados y

necesarios para ejecutar los aplicativos de corte y control de calidad, considerando una estructura basada en microservicios vinculados mediante API's (Interfaz de programación de aplicaciones), para conseguir un sistema robusto; ya que cada microservicios es independiente, y al existir algún desperfecto que interrumpa alguno, los demás siguen funcionando, como se muestra en la Figura 1.

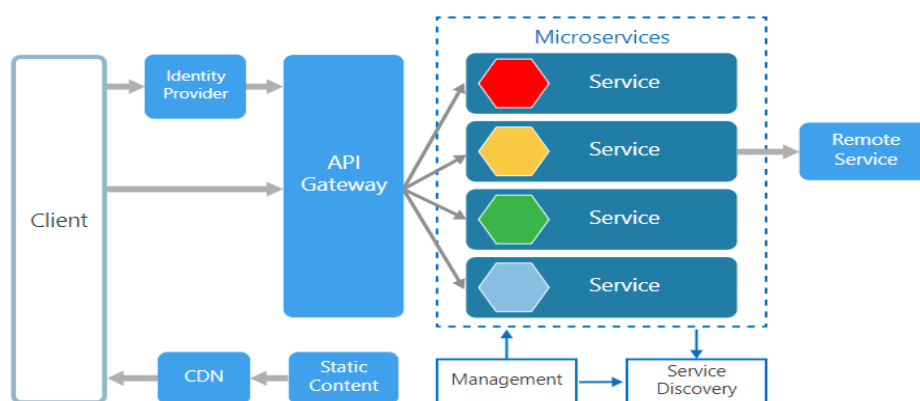


Figura 1. Arquitectura de microservicios.

Fuente: (Hydra, 2019)

Al culminar el corte térmico en la plancha de fomix, se efectuará un control de calidad de las piezas cortadas, mediante la utilización de microservicios de visión Cloud Computing como: Google Cloud Vision u otros.

Para superar el control de calidad, mismo que será diseñado con criterio de aceptación humana, la pieza cortada deberá poseer bordes regulares y sin quemaduras que distorsionen su forma. El suministro de materia prima y almacenamiento de piezas que superen el control de calidad se realizará de forma manual.

Como parte del sistema se desarrollará una interfaz gráfica, que podrá ser visualizada en dispositivos como: tablets, celulares, computadores entre otros, la cual permitirá al usuario ejecutar tres programas de corte previamente definidos, que se podrán modificar o adicionar de forma local en el controlador KR C4, adicionalmente, se contará con un microservicio de base de datos, en la que estarán almacenados los usuarios que pueden acceder.

El posible diseño esquemático que se llevará a cabo para lograr un sistema Cloud Computing para la estación robotizada multipropósito de la ESPE, se observa en la Figura 2.

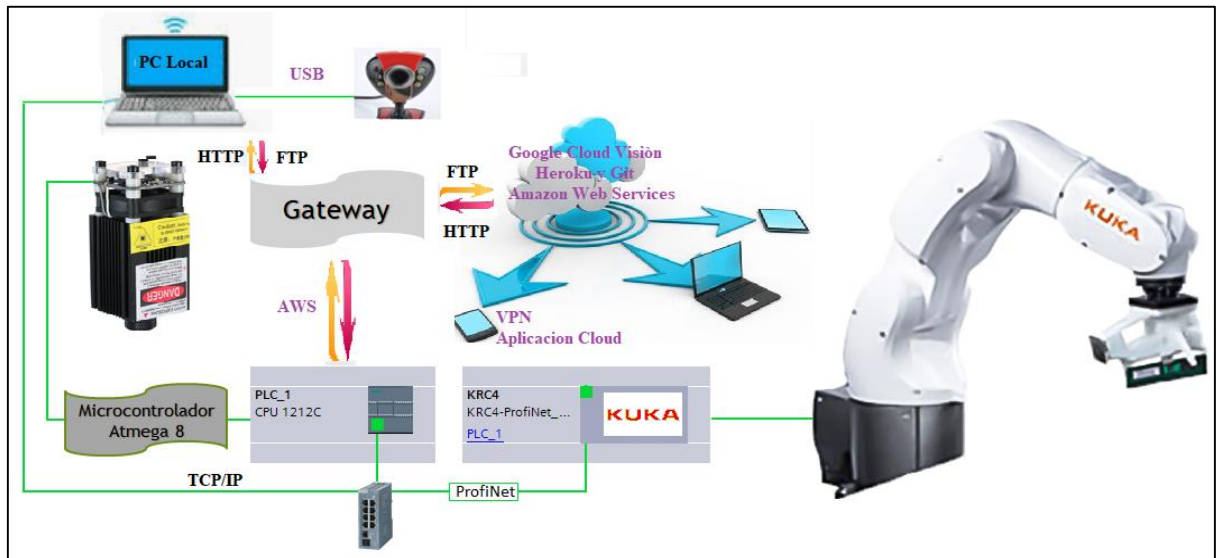


Figura 2. Sistema Cloud Computing del robot KUKA KR3 R540.

Finalmente, se desarrollarán dos prácticas de laboratorio para facilitar el aprendizaje de los estudiantes de Robótica Industrial en servidores web y servicios Cloud Computing, que permitan comparar el desempeño de control local y el del sistema Cloud Computing del robot KUKA KR3 R540.

1.4. Objetivos

1.4.1. Objetivo general

- Diseñar e implementar un sistema de control Cloud Computing para ampliar la funcionalidad del robot KUKA KR 3 R540, aplicado al corte térmico de planchas de fomix y control de calidad mediante servicios de visión artificial, a fin de potenciar su participación en la industria 4.0, mediante la utilización de servidores online y dispositivos con conectividad ethernet.

1.4.2. Objetivos específicos

- Implementar una puerta de enlace para conectar los dispositivos de hardware (controlador KUKA KR C4, PLC s7-1200, PC) a la nube.
- Definir las diferentes estrategias de integración de los servidores online (APIs, servidor web, servidor de base de datos, etc.), con los dispositivos de hardware utilizados.
- Implementar un sistema de corte térmico para planchas de fomix, mediante Cloud Computing y la célula robotizada multipropósito de la ESPE.
- Implementar un sistema de visión Cloud Computing, que permita realizar el control de calidad en las piezas de fomix cortadas.
- Realizar una interfaz gráfica que permita el acceso a dispositivos con conexión a internet. (celulares, computadores, tablets) para ejecutar los diferentes programas de corte.

- Desarrollar dos prácticas del laboratorio de Robótica Industrial para fortalecer el aprendizaje de los estudiantes, basadas en servidores web y procesos en la nube, que permitan comparar el desempeño del control local y el del sistema Cloud Computing del robot KUKA KR3 R540.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Cloud Computing.

Se refiere a la forma en que los usuarios administran procesos, que se encuentran alojados en la nube, tales como: datos, aplicaciones, servidores, entre otros.

Su historia comienza en el año 1961 cuando Jhon McCathy, sugiere el concepto de computación en la nube, para los años 90`s las empresas empiezan a ofertar servicios de redes virtuales privadas, en las que se podía adquirir aplicaciones empresariales a través de un servidor web, esto en el año 1999 por parte de la empresa “*salesforce*”. Entrando al nuevo milenio Amazon, lanza Amazon Web Services (AWS) que, disponía de servicios de alojamiento en la nube, análisis, integración de aplicaciones, administración y control, entre otros muchos más. (Service, 2019)

En el año 2006, Google presenta su aplicación Google Docs, que trajo un gran avance a la computación en la nube y la colocó en la vanguardia de la tecnología informática. A su vez Amazon introdujo Elastic Compute Cloud (EC2), Logrando que las empresas alquilen equipos virtuales, en los que puedan ejecutar sus propias aplicaciones informativas.

Posteriormente, se realizaron colaboraciones entre, International Business Machines Corporation (IBM), Google y universidades de los Estados Unidos, para realizar avances del Cloud Computing, logrando en el año 2009, que Google lanzará su plataforma Google Apps, que ofertaba aplicaciones útiles, baratas y confiables.

En 2010, se organiza el concepto de arquitectura por capas: Infraestructura (IaaS), Plataforma (PaaS), programas (SaaS) y Red (NaaS). (Concepción, 2019)

Hoy en día, se espera que el Cloud Computing, siga evolucionando ya que las herramientas que proporciona permiten supervisar, todo tipo de actividades en ese entorno, tales son: la implementación de recursos adicionales a un proceso, el seguimiento de su funcionamiento, la integración de los datos e incluso la recuperación de información ante desastres. Adicionalmente ofrece un control administrativo de la infraestructura, las aplicaciones, la plataforma y los datos, que con su unión conforman la nube. (Concepción, 2019)

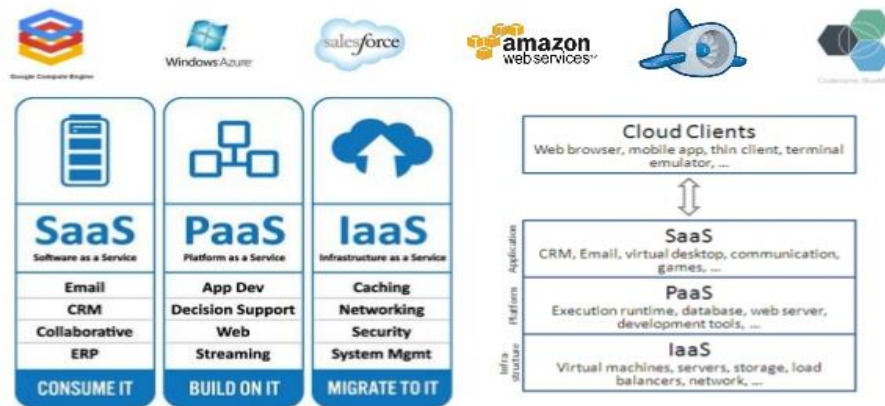


Figura 3. Arquitectura Cloud Computing.

Fuente: (Lopez, 2014).

2.1.1. Infraestructura como servicio (IaaS)

Provee de la infraestructura necesaria para ejecutar aplicaciones, ofrece: servidores, capacidad de proceso, espacio de almacenamiento y equipamiento físico. Además de permitir una escalabilidad semiautomática contratando más o menos recursos según sea la necesidad, y su ventaja más importante es su contratación, solo durante el tiempo de uso, evitando el costo masivo de la compra de servidores físicos, algunos ejemplos de IaaS se muestran en la Figura 4. (Concepción, 2019)

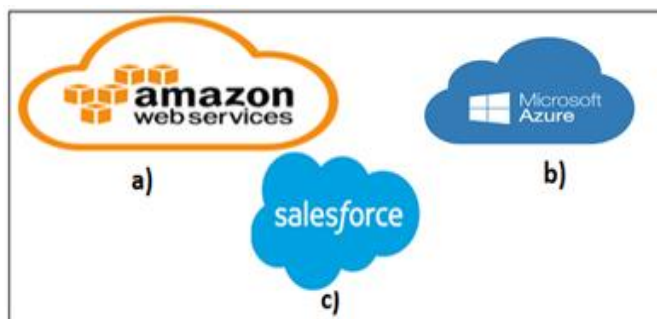


Figura 4. Ejemplos de IaaS.

Fuente : a) (Service, 2019) ,b) (Azure, 2019),c) (Salesforce, 2019).

2.1.2. Plataforma como servicio (PaaS)

Es un entorno de desarrollo e implementación completo en la nube, el cual se contrata a un proveedor alojado en la nube, permitiendo acceder a él a través de internet. Este entorno posee infraestructura como: servidores, almacenamiento, redes, herramientas de desarrollo, servicios de inteligencia empresarial (IB). Permitiendo así que los usuarios implementen sus propias aplicaciones y gestionarlas realizando procesos de: compilación, pruebas, administración y actualización. Evitando los gastos que supone la compra de licencias de software y servidores físicos, algunos ejemplos de PaaS se muestran en la Figura 5. (Concepción, 2019)

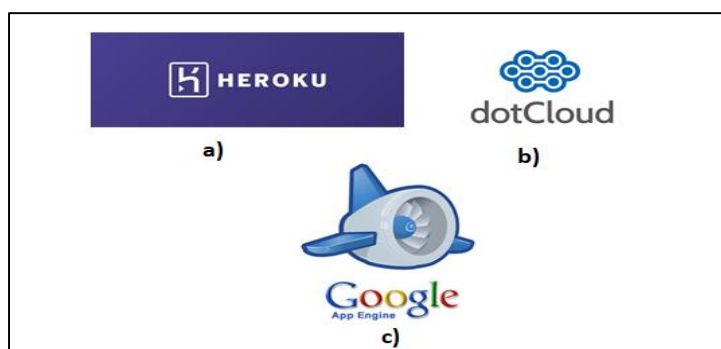


Figura 5. Ejemplos de PaaS.

Fuente : a) (Heroku, 2019) ,b) (DotCloud, 2019),c). (Google, 2019)

2.1.3. Software como servicio (SaaS)

Permite a los usuarios utilizar aplicaciones basadas en la nube, a través de un buscador con conexión a internet, sin la necesidad de instalaciones de software adicionales. Los ejemplos más comunes son: herramientas ofimáticas, calendarios o correos electrónicos.

SaaS es una solución de software que se adquiere de un proveedor alojado en la nube, el cual se encarga de administrar el hardware o software necesario para ofertar dicho servicio, que se basa en un modelo de pago por su uso, garantizando la disponibilidad, seguridad de los datos y aplicaciones contratadas. Además de permitir a organizaciones poner en marcha aplicaciones con un costo inicial reducido, algunos ejemplos de SaaS se muestran en la Figura 6. (Azure, 2019)



Figura 6. Ejemplos de SaaS.

Fuente: (Sevilla, 2010)

2.1.4. Red como servicio (NaaS).

Es la optimización de recursos, considerando los procesos informáticos y de red como un todo, en alguno de los casos, con una red virtual, que puede estar alojada en la Capa 2 (Ethernet, VLAN) o en Capa 3 (IP, VPN) del modelo OSI. (Azure, 2019)

2.2. Servidores web.

Es un programa que tiene como cualidad fundamental aceptar peticiones HTTP de un navegador web, alojar páginas y atender peticiones de los clientes. Para direccionar exactamente al sitio web al que se desea acceder, se debe ingresar una dirección única conocida como URL (Uniform Resource Identifier), en ella se encuentra información como: el protocolo, el dominio y la ruta en el servidor, como se detalla en la Figura 7.. (Ramos A. , 2011)



Figura 7. Ejemplo URL.

2.2.1. Dirección IP

Son direcciones utilizadas por el protocolo de internet para identificar a un dispositivo en la red de forma única. El protocolo permite la comunicación entre dispositivos de una misma red. Una IP es un código con cuatro valores numéricos separados por un punto, que van desde el 0 al 255. (Ramos A. , 2011)

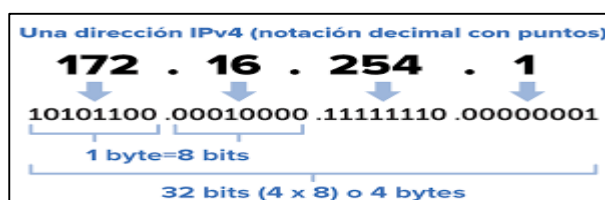


Figura 8. Dirección IP.

2.2.2. Dominio

Es una forma de reemplazar las direcciones IP, por palabras ya que, es más práctico recordar un nombre que varios números. (Ramos A. , 2011)

2.3. Lenguajes web.

Permiten proporcionar de estructura y estética a las aplicaciones web, intentando facilitar el trabajo de los desarrolladores y pueden ser aptos para el desarrollo de páginas dinámicas, estáticas, de clientes o servidores. (Murcia, 2017)

2.3.1. HTML- HyperText Markup Language

Es un lenguaje de marcado de hipertexto, permitiendo ordenar el contenido de una página web, se despliega en base a atributos, contenidos y etiquetas, que puede ser interpretado por un navegador, mostrando así las páginas web. Su primera versión fue publicada en 1991, por Timothy John Berners. (Reyes, 2018)

2.3.1.1. Interfaz gráfica.

El Web Server integrado en el PLC S7- 1200, acepta únicamente lenguaje HTML y sus complementos CSS y Java Scripts, para el desarrollo de aplicaciones web, pero no lenguajes como: PHP o Python. Por lo que al ser el lenguaje más utilizado en desarrollo web y la única opción, se desarrollará la interfaz gráfica en HTML.

Para mejorar y facilitar la fluidez de la escritura del código HTML, se utiliza un editor como Dreamweaver, Visual Studio Code entre otros. Estos programas integran plantillas de desarrollo, respecto al lenguaje HTML, aumentando la rapidez con la que se desarrolla una aplicación web, la estructura básica de una aplicación web escrita en HTML, empieza con un título y el cuerpo de la interfaz.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Documento sin título</title>
6 </head>
7 <body>
8     Cuerpo de la aplicación WEB.
9 </body>
10 </html>

```

Figura 9. Estructura básica HTML.

Por lo general el primer archivo de una aplicación web, lleva el nombre de index y en el cuerpo de código contiene una serie de etiquetas descritas en la Tabla 3, que ayudan a dar forma a la aplicación.

Tabla 1.
Etiquetas HTML.

Etiqueta	Descripción
!DOCTYPE	Tipo de documento (versión de html empleada)
html	Engloba todo el documento
<!-- ... -->	Comentario (sólo visible en el código fuente)
head	Delimita el encabezado del documento
title	Título del documento (se muestra en la pestaña del navegador)
base	URI base para direcciones relativas
link	Enlace a otros archivos (hoja de estilo, etc.)
meta	Metainformación sobre el documento
style	Hoja de estilo incluida en el documento
body	Delimita el cuerpo del documento
article	Artículo
section	Sección
nav	Navegación
aside	Lateral
h1 a h6	Encabezado (de nivel 1 a 6)
header	Cabecera
footer	Pie
address	Dirección (información sobre el autor)
hr	Separador
div	División
blockquote	Cita larga (que incluye varios párrafos)

2.3.2. CSS- Cascading Style Sheets

Es una hoja de estilo en cascada y se utiliza para proporcionar características como: colores, formas y márgenes a elementos escritos en lenguajes de marcado, separando así el contenido, de la parte visual. Su desarrollo estuvo a cargo de W3C (World Wide Web Consortium) y se presentó en 1996, desde ese entonces existe una gran relación entre HTML y CSS. (Bravo, 2019)

2.3.3. JAVA Script

Es un lenguaje sencillo de programación, que permite dotar de características dinámicas a elementos escritos en lenguajes de marcado, se pueden ejecutar en el navegador (Cliente). Los Scripts son funciones que se ejecutan ante algún evento sucedido en el navegador, de ese modo se pueden añadir efectos como: que una imagen cambie al dar clic sobre ella. Su desarrollo fue realizado por Netscape, basado en lenguaje Java. (Navarrete, 2000)

2.3.4. PHP- Hypertext Pre-Processor

Es un lenguaje de programación que permite realizar procesos dinámicos en elementos escritos en lenguajes de marcado, pero a diferencia de Java Script, que se ejecuta en el cliente, PHP lo hace en el servidor, haciendo invisible el código, para los usuarios. Fue desarrollado en 1994 por Rasmus Lerdorf, con la intención de facilitar el diseño dinámico de páginas web. (Definición, 2019)

2.4. API's - Application Programming Interface

Es un conjunto de normativas y códigos que las aplicaciones siguen para poder gestionar la comunicación entre ellas, sirviendo de igual forma de interfaz gráfica, para facilitar la interacción entre el usuario y el software.

Las API's, se utilizan para comunicarse con: sistemas operativos, bases de datos, o protocolos de comunicación, además de posibilitar un sinfín de servicios de procesamiento de información, monitoreo y control de sistemas.

Su mayor fortaleza, es la posibilidad de hacer uso de funciones existentes en otro entorno, ya sea este una infraestructura o plataforma, para no desarrollar algo ya existente, sino simplemente hacer uso de ello, abaratando costos y tiempos de desarrollo. (Merino, 2014)

2.4.1. Ejemplos de uso.

- Las empresas de distribución de comida a domicilio como: Uber Eats y Rappi tienen sus propias aplicaciones, sin embargo, dentro de dichas aplicaciones se encuentra la API's de Google maps.
- Aplicaciones de conversión de imágenes a texto, cuentan con la API de Cloud Visión.

2.5. Plataformas abiertas Cloud Computing.

Son entornos PaaS, tanto de desarrollo como de oferta de servicios, carece de control de la infraestructura, pero permiten ejecutar aplicaciones propias de los usuarios y contratar aplicaciones desarrolladas por la plataforma, además de posibilitar los sistemas multiplataforma, en las que el usuario puede contratar diversos servicios de distintos proveedores, a fin de implementar el mejor sistema para su aplicación.

Adicionalmente, el servidor o conjunto de servidores interconectados entre sí, que ofertan servicios de control y monitoreo, forman parte del internet de las cosas (IoT). (Campoverde, 2015)

2.5.1. Google Cloud Platform

Es una suite, donde se reúnen todas las aplicaciones de desarrollo web, que oferta Google, abarcando temas sobre: Machine Learning, Cloud Computing, Data Storage, Data Analytics, Networking, etc. Además de proveer herramientas de diseño, que permiten a los usuarios realizar pruebas y publicar sus propias aplicaciones en la plataforma, garantizando una gran estabilidad, seguridad y sobre todo facilidad en la escalabilidad. (Garcia, 2018)

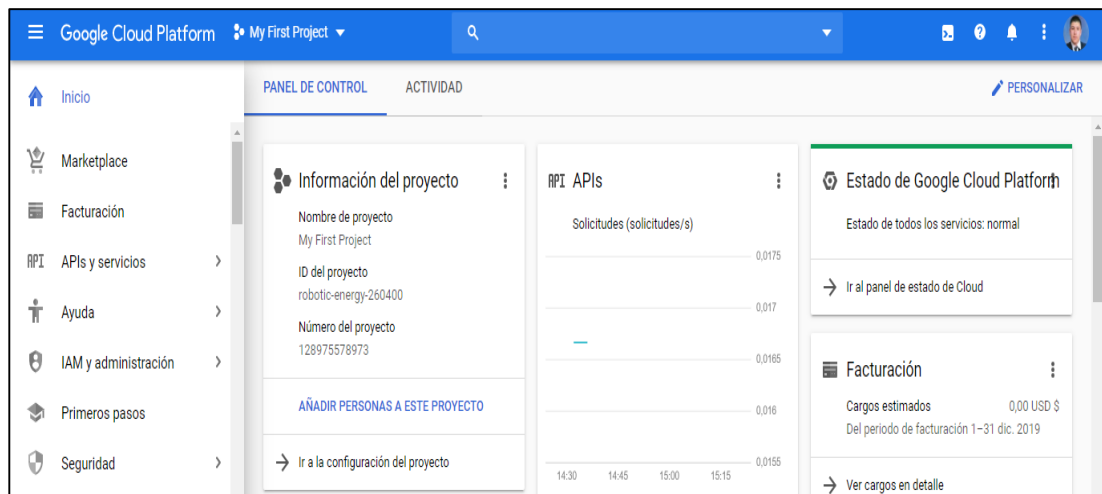


Figura 10. Google Cloud Platform.

Fuente: (Google, 2019)

2.5.1.1. Visión API

Es un modelo entrenado en el reconocimiento de imágenes, es capaz de detectar rostros, modelos de automóviles, animales, extraer texto de imágenes e identificar objetos. (Garcia, 2018)



Figura 11. Visión API.

Fuente: (Google, 2019)

2.5.1.2. Tensor Flow

Es una librería de código abierto, que permite entrenar un sistema mediante redes neuronal, Además de tener una gran compatibilidad con lenguajes de programación establecidos como: Python, C++, Java y GO. (Garcia, 2018)

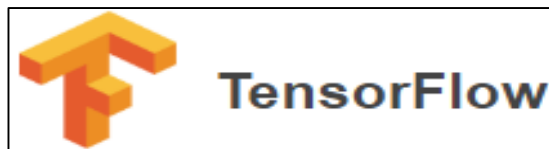


Figura 12. TensorFlow.

Fuente: (TensorFlow, 2019)

2.5.1.3. Speech API

Es una aplicación utilizada, para transformar información de audio a texto fácilmente, con alta precisión en condiciones desfavorables de ruido, aplicando modelos de entrenamiento basados en redes neuronales, siendo capaz de detectar más de 80 idiomas, lo que lo hace ideal para trabajar a nivel multicultural o internacional. (Garcia, 2018)



Figura 13. Speech API.

Fuente : (Google, 2019)

2.5.1.4. Google Cloud SDK - Software Development Kit

Es un entorno de desarrollo de software, que posee un conjunto de herramientas para la plataforma Google Cloud, permitiendo ejecutar líneas de comandos bq, gsuit, gcloud. Y su principal función es permitir acceder mediante líneas de comandos, a las aplicaciones de Google Cloud Platform, previa obtención de credenciales (Permisos de Acceso). Adicionalmente pueden ser ejecutadas de forma automatizada o de forma interactiva en secuencias de comandos.

Las bibliotecas de cliente para los diferentes lenguajes de programación Java, .NET, GO, PHP, Python, Ruby se instalan a través de gestores de paquetes. (Google C. , 2019)

Características:

- Herramienta kubectl
- Comando cmdlets de PowerShell (Windows)
- Herramienta de línea de comandos gcloud
- Herramienta gsutil

2.5.1.5. Credenciales Google Cloud.

Son un conjunto de permisos y autenticaciones, que solicita Google Cloud Platform, para acceder a sus aplicaciones a través de línea de comandos, el archivo que contiene esta credencial tiene extensión .json y se obtiene siguiendo los pasos descritos en (Cloud, 2019).

2.5.2. Heroku: Cloud Application Platform

“Es una plataforma como servicio (PaaS), que permite a los desarrolladores construir, ejecutar y operar aplicaciones completamente en la nube.” (CLI, 2019)

Al ser una plataforma como servicio, los usuarios se concentran totalmente en el desarrollo de aplicaciones, y su principal ventaja es permitir trabajar con una gran variedad de lenguajes de programación como: Python, PHP, Ruby. Aunque no permite la interacción con lenguajes de marcado como HTML, por lo que se necesita vincular la ejecución de un HTML dentro de PHP. (Romero, 2019)

Características:

- Permite la ejecución de aplicaciones ligeras de forma gratuita.
- Su sistema está apoyado en GitHub.
- Instala los programas necesarios para la ejecución de aplicaciones de forma semiautomática, si es una aplicación PHP, se encarga de la totalidad de la ejecución, pero si es una aplicación en Python, se debe especificar versión y complementos necesarios para la ejecución de la aplicación.

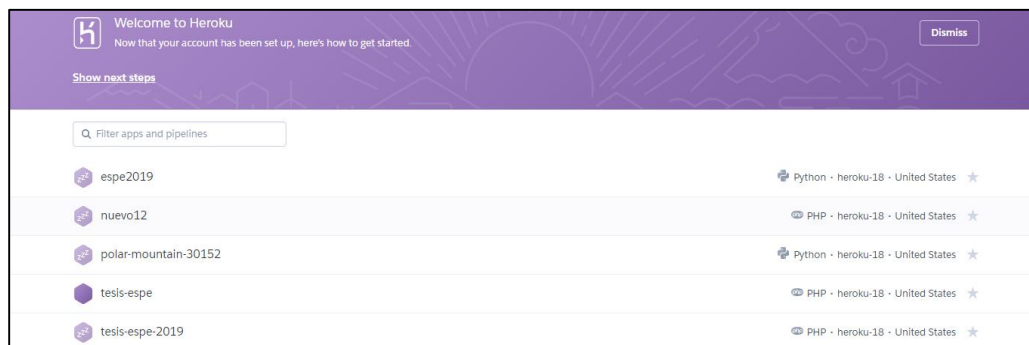


Figura 14. Heroku: Cloud Application Platform.

Fuente: (CLI, 2019)

2.5.3. Git

Es un programa para gestionar proyectos y aplicaciones, que permite identificar y controlar la versión de un código. Fue desarrollado por Linus Torvalds, que a su vez también es el creador de Linux.

Su principal característica, es permitir alertar a los programadores de todos los cambios que ha sufrido un código o una aplicación en general, como adicional permite respaldar la información de forma eficiente, ya que si cierta información no ha sufrido cambio el sistema la deja intacta, y no la vuelve a cargar en el repositorio, mejorando los tiempos de ejecución. (Robles, 2018)

Características

- Genera un repositorio de versiones actuales y anteriores.
- Proporciona un listado de los cambios que se han realizado.
- Permite regresar a versiones anteriores.
- Proporcionan la posibilidad de trabajar en ramas, teniendo varias posibilidades de código a utilizar.

Nota: Una rama es una copia del repositorio original, donde se han modificado códigos, versiones o características de una aplicación, para probar su funcionamiento.

2.5.3.1. GitHub

Es una red social diseñada, para la interacción entre desarrolladores, que en la actualidad es uno de los repositorios más grandes. (Bravo1, 2019)

2.5.4. Django

Es un framework web, que posee un conjunto de herramientas referenciales para desarrollar aplicaciones web de forma rápida y sencilla. Django es gratuito y de código abierto, desarrollado en Python, por lo que es una de las opciones más utilizadas por los programadores para realizar aplicaciones Cloud Computing, con código base de Python. (docs, 2019)

Fue desarrollado en los años 2003 y 2005, por programadores de sitios web referentes al periodismo.

Los puntos fuertes de Django, son el mapeo de URL, vistas, modelos y plantillas previamente configuradas y probadas. Además de incluir diferentes herramientas destinadas al desarrollo web, como:

- Serialización de datos
- Formularios.
- Cacheo.
- Autenticación y permisos de usuarios.
- Sitio de Administración.



Figura 15. Django.

Fuente: (Pythones, 2019)

2.5.5. Amazon Web Services. (AWS) - Cloud Computing Services

Es la plataforma Cloud Computing más completa y utilizada a nivel mundial, ofreciendo más de 175 servicios integrales de centros de datos, millones de clientes, incluyendo empresas emergentes e instituciones gubernamentales, están utilizando AWS para reducir costos, aumentar su eficiencia e innovar rápidamente.

Ofrece tecnología de infraestructura de cómputo (VPS), almacenamiento, bases de datos, y tecnologías de aprendizaje automático e inteligencia artificial. (Services, 2019)

2.5.5.1. Seguridad

AWS, está diseñado para ser el entorno más robusto y seguro de Cloud Computing, su infraestructura se creó para cumplir con los estándares de seguridad de ejércitos, bancos internacionales, e instituciones con altos estándares de confidencialidad. (Service, 2019)

2.5.5.2. Servidor Virtual Privado. (VPS)

Son computadores privados capaces de funcionar como servidores, ubicados en la nube, que se contratan mediante un proveedor.

Los VPS funcionan bajo su propio sistema operativo, con direcciones IP, privadas y públicas, las cuales permiten acceder a su entorno desde cualquier lugar del mundo, en ellos pueden ejecutarse aplicaciones de desarrollo, bases de datos, juegos entre otros. Su costo dependerá del proveedor y de sus características como: capacidad de almacenamiento, memoria RAM y velocidad de internet.

2.6. Visión artificial

Es un campo perteneciente a la inteligencia artificial que, mediante la aplicación de herramientas adecuadas, permite el procesamiento y análisis de información capturada en imágenes digitales. (Etitudela, 2019)

Los procesos que componen los sistemas de visión artificial son:

- Captación de imagen.
- Instrucciones.
- Procesamiento.
- Ejecución.

2.6.1. Características:

- Realizar Controles de Calidad.
- Automatizar tareas repetitivas.
- Evaluar el estado de objetos, sin contacto físico.

2.6.2. Aplicaciones:

- Relaciones espaciales entre varios objetos.
- Identificación e inspección de objetos.
- Mediciones Tridimensionales.
- Control de Calidad.

2.6.3. Factores que intervienen.

2.6.3.1. Cámaras

Es el elemento que permite capturar la imagen, para su posterior procesamiento, en el mercado existen varios tipos de cámaras como son:

- **Cámaras lineales.** -Realizan un barrido del objeto capturado, junto a un desplazamiento longitudinal. (Etitudela, 2019)

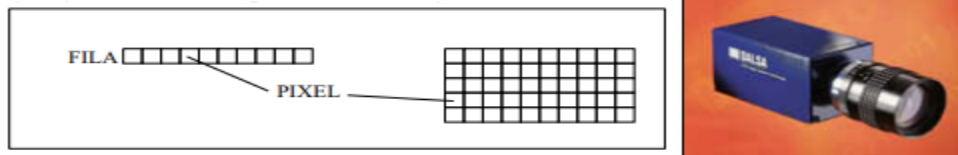


Figura 16. Cámara lineal.

- **Cámara matricial.** – Posee sensores fotosensibles, que cubren el área que está formada por una matriz de pixeles posibilitando la captura de imágenes. (Etitudela, 2019)



Figura 17. Cámara matricial.

- **Cámara digital.** - Utiliza sensores electrónicos para digitalizar imágenes.



Figura 18. Cámara digital

- **Cámara web.** - Son cámaras digitales que se conectan a un computador generalmente por puerto USB, para capturar y transmitir información gráfica, ya sea imágenes o video. (Pérez1, 2016)



Figura 19. Cámara web

- **Cámara IP.** – Son videocámaras digitales especializadas en la transmisión de información a través de internet.



Figura 20. Cámara IP

2.6.3.2. Iluminación

Es el proceso más crítico en visión artificial, y su propósito es controlar la forma en que las cámaras capturan imágenes, ya que las cámaras capturan la luz reflejada de los objetos, por tal motivo una iluminación de calidad y controlada es primordial. (Eitudela, 2019)

2.6.4. Open CV

Es una biblioteca de aprendizaje y visión por computadora de código abierto, que permite tener una infraestructura común en aplicaciones de visión artificial. Adicionalmente al ser un producto con licencia de software libre permisiva, da la facilidad a las empresas que requieran esta biblioteca, de modificar su código a conveniencia.

OpenCV, posee más de 2500 programas y códigos optimizados, para realizar tareas de aprendizaje y visión. Estos programas se utilizan habitualmente para la detección y reconocimiento de rostros, adquisidor de modelos 3D de objetos, detectar y clasificar objetos, entre otros.

La biblioteca es ampliamente utilizada en grupos de desarrollo e investigación, organismos estatales y en una gran variedad de empresas como: Google, Microsoft, Sony, Honda y muchas más.

Ofrece interfaces en lenguajes de programación (C++, Matlab, Python, Java) y es compatible con los siguientes sistemas operativos: Linux, Android, Windows, Mac OS. (OpenCV, 2019)

2.6.4.1. Características:

- Vinculación con Google Cloud.
- Multiplataforma.
- Librerías para Python.
- Escrito Nativamente en C++.

2.7. Estructuras y protocolos de integración.

2.7.1. Profinet

Es el principal estándar industrial para intercambiar información entre controladores (PLC's, DCSs, PAC's) y dispositivos (sistemas de visión, lectores RIFD, Bloque I/O, Instrumentos de Procesos, otros controladores). (Profinet, 2019)

Se basa en Ethernet, para facilitar la integración de los procesos empresariales de fabricación y se puede integrar a redes de buses de campo existentes, protegiendo inversiones previas y mirando al futuro.

Profinet es modular, flexible e integral, por lo que permite realizar actualizaciones en cualquier momento un equipo a la vez, sin la necesidad de actualizar todo. Además de tener una relación plena con comunicaciones TCP/IP, con el fin de ofrecer soluciones escalables en tiempo real, un ejemplo de red Profinet se observa en la Figura 21. (Profibus, 2017)

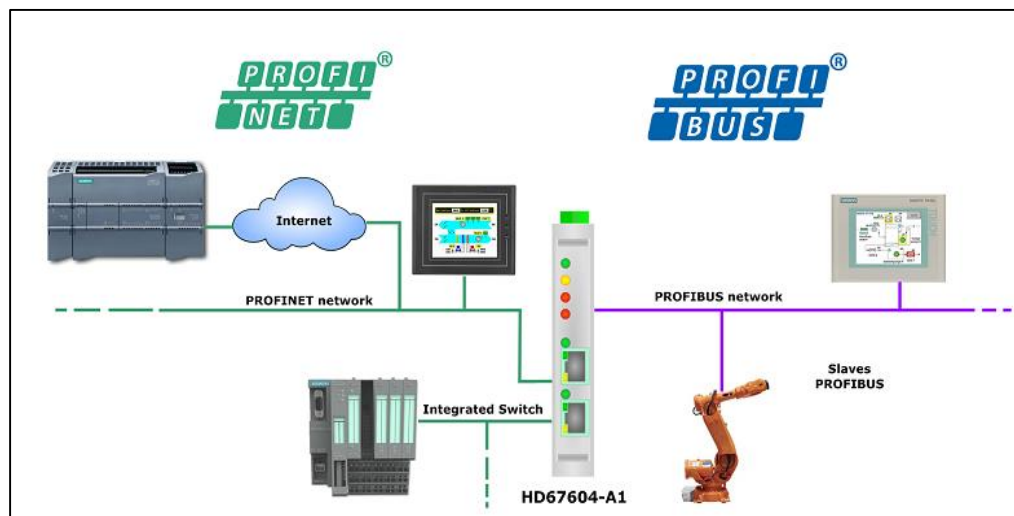


Figura 21. Red Profinet.

Fuente: (ADF, 2019)

2.7.2. TCP/IP

Es un conjunto de protocolos, que posibilitan el intercambio de información entre redes y dispositivos de control a través de internet. Fue desarrollado por el departamento de Defensa de los Estados Unidos en 1972 y su implementación fue a través de ARPANET (Red de Área Extensa). (Pérez & Merino, 2008)

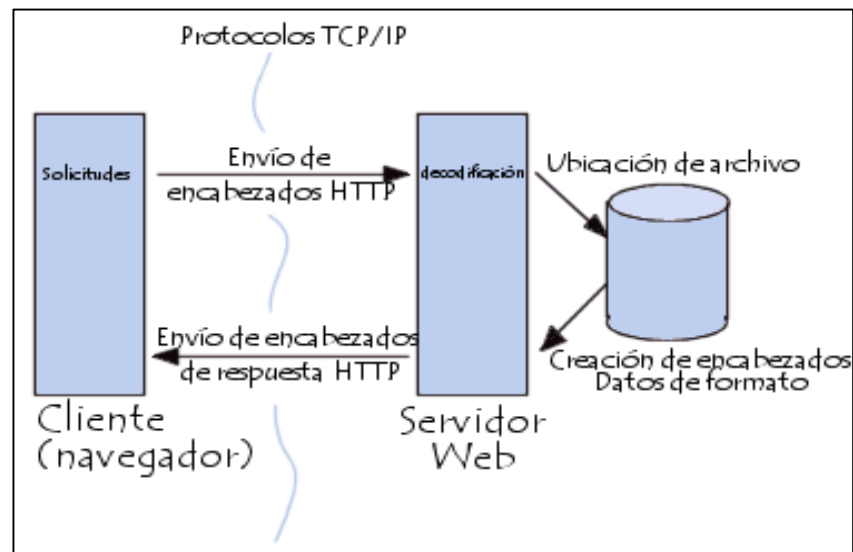


Figura 22. Protocolo TCP/IP.

Fuente: (Pérez & Merino, 2008)

2.7.2.1. Protocolo de control de transmisión

Proporciona un método de transporte de datos muy fiable dentro del nivel de transporte del modelo OSI.

2.7.2.2. Protocolo de internet

Ofrece la posibilidad de direccionar otras máquinas, haciéndolo en el nivel referencial de red.

2.7.2.3. También incluye más de 100, protocolos entre los que están:

- HTTP, para acceder a sitios web.
- FTP (File Transfer Protocol), Traslación de archivos
- FTP (File Transfer Protocol), Recepción de Archivos
- ARP (Address Resolution Protocol), direccionamiento.
- POP (Post Office Protocol).
- RMT, realiza copias de archivos en una unidad.
- DDAT'ES , muestra fechas y horas guardadas.
- SUPDUP Permite conectarse a otro servidor de la red.

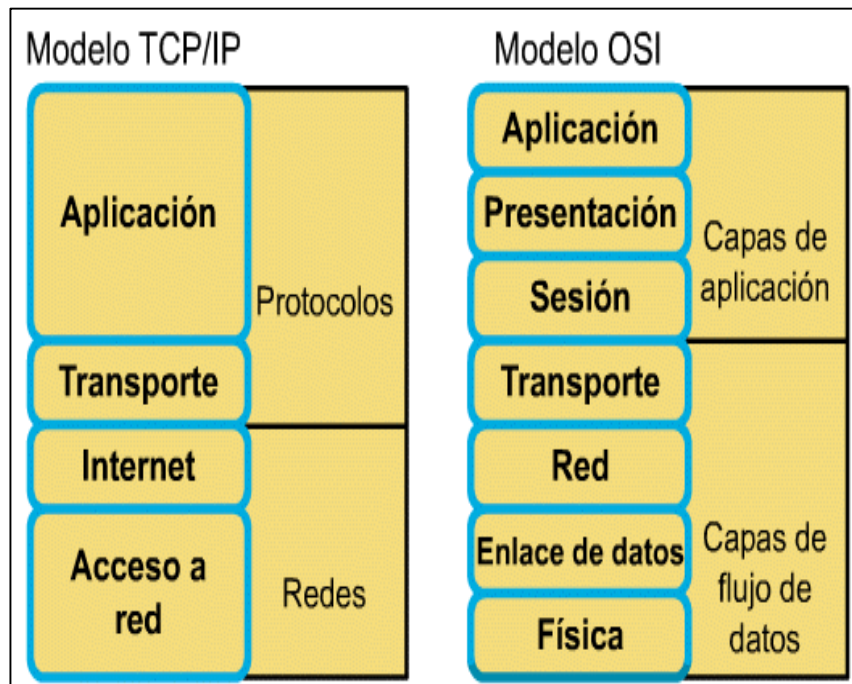


Figura 23. Modelo TCP/IP.

Fuente: (Roa, 2003)

2.7.3. HTTP - Hypertext Transfer Protocol

Es el protocolo de transferencia de datos de la web, es decir, el código que permite a un elemento solicitante y al elemento que contiene la información solicitada interactuar.

HTTP, delimita la forma y la semántica de la información para entablar la comunicación entre los diferentes componentes de la web (clientes, servidores, proxis), Fue desarrollado por la World Wide Web Consortium en 1999, teniendo la colaboración de Internet Engineering Task Force.

Su función principal es mediar entre las peticiones de un cliente y las respuestas que proporciona un servidor, para entablar una comunicación fluida y en el mismo lenguaje. Su funcionamiento es muy simple se basa en petición-respuesta entre un servidor y un usuario o cliente solicitante, el cliente genera un solicitud y el servidor entrega una respuesta de forma puntual y establece las pautas para iniciar, desarrollar y cerrar la transmisión de información, este proceso se realiza cada vez que el cliente efectúa una solicitud. (Raffino, 2019)

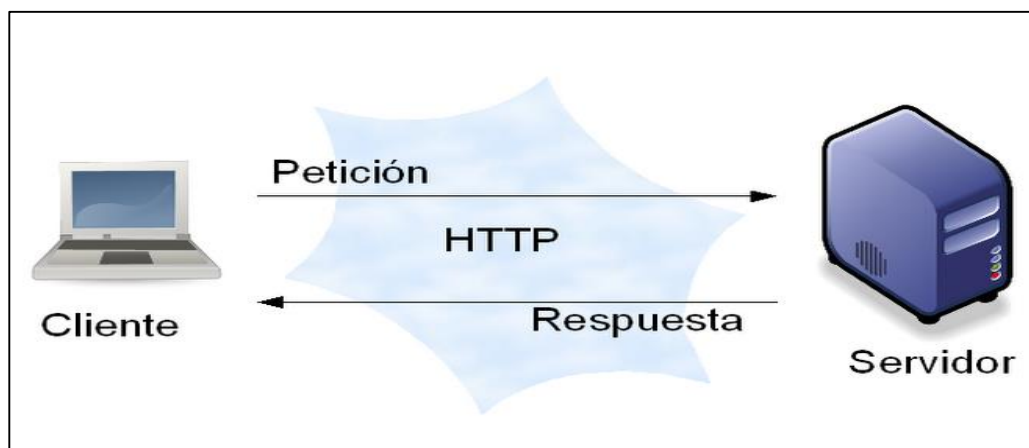


Figura 24. Protocolo HTTP.

Fuente: (Raffino, 2019)

2.7.4. Modbus

Es un protocolo de comunicación para dispositivos de automatización, desarrollado en 1979, en el nivel referencial de aplicación, transfiriendo información por una capa serial, y ha evolucionado al pasar de los años para incluir al protocolo serial, TCP/IP y UDP (User Datagram Protocol).

Modbus, basa su protocolo en solicitudes y respuestas o en relaciones maestro-esclavo, la comunicación se efectúa en pares, un elemento maestro inicia la solicitud y espera la respuesta de elemento esclavo, la información de las solicitudes y respuestas, y la capa de red por donde se trasmite dicha información, son definidas en las diferentes capas que posee el protocolo. (Instruments, 2019)

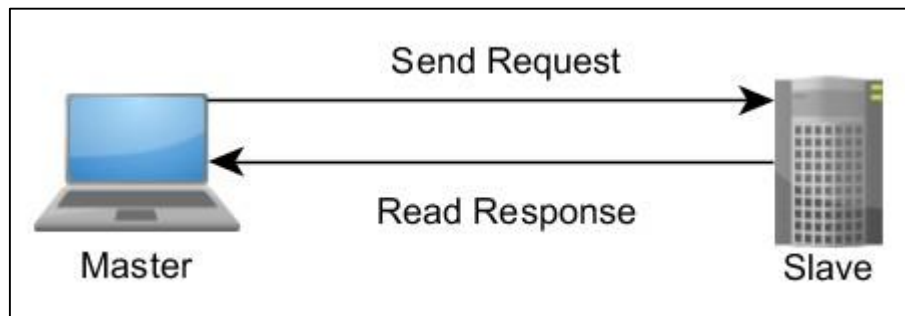


Figura 25. Relación maestro-esclavo.

Fuente: (Instruments, 2019)

Capas del Protocolo.

- Define la unidad de datos de protocolo (PDU)
- Define la unidad de datos de aplicación (ADU).

2.7.5. Comunicación serial

El puerto serial se encarga de la comunicación digital entre dos dispositivos por medio de conexiones RX y TX, permitiendo realizar tres modos de comunicación: Full-duplex, Half-duplex y Simplex. Es extremadamente sencillo de manejar, pero su principal desventaja es que solo se pueden comunicar dos dispositivos a la vez.

- **Full-duplex.** - Puede enviar y recibir información simultáneamente.
- **Half-duplex.** – Puede recibir y enviar información, una a la vez.
- **Simplex.** – Solo se puede enviar o solo recibir información.

La función de un puerto serial, es la de encapsular y des-encapsular paquetes binarios seriales, lo que significa transformar un byte de estructura paralela a un conjunto de pulsos seriales que se emiten en una línea de transmisión.

El protocolo serial define tres condiciones de uso: inicio de transmisión (IT), paridad (P) y fin de transmisión (FT), para el correcto funcionamiento de estas condiciones es necesario un oscilador que sincronice las transmisiones. El oscilador controla la velocidad del puerto serial, que se denomina BAUD 's. Entonces se tienen que agregar tres tipos de bits al byte de información para que exista comunicación serial: un bit de inicio, un bit de paridad y uno o dos bits de parada, lo que significa que a una velocidad de 9600 BAUD, se transmiten, 8 bits información, 1 bit de inicio y 1 bit de parada, lo que quiere decir que se envían 10 bits, por cada 8 de datos, que da una transmisión de datos reales de 960 Bytes por segundo . (Estrada, 2017)

El puerto serial tiene varios nombres: UART, USART o EUSART.

- UART – Universal Asynchronous Receiver
- USART – Universal Synchronous and Asynchronous Receiver and Transmitter,
- EUSART – Enhanced Universal Asynchronous Receiver and Transmitter

Los ordenadores en la actualidad, tanto de escritorio como portátiles, ya no tienen integrado el puerto serial, por lo que para lograr comunicar un sistema serial con un ordenador, se requiere un dispositivo adicional o convertidor que se denomina USB-TTL, lo que permite al protocolo de comunicación serial convertirse al protocolo de comunicación USB, con el que cuenta todos los ordenadores en la actualidad, las conexiones seriales más utilizadas se detallan en la Figura 26. (Estrada, 2017)

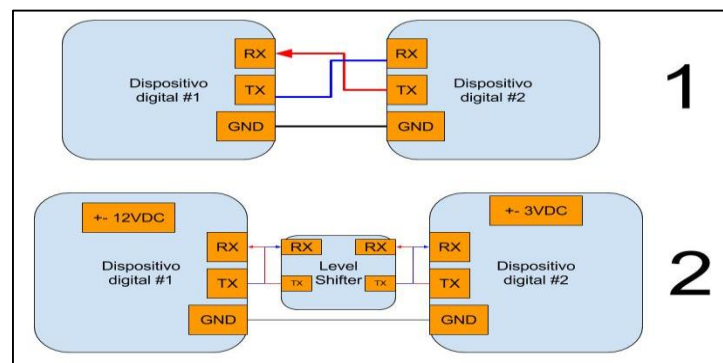


Figura 26. Tipos de conexiones seriales.

Fuente: (Estrada, 2017)

2.7.6. FTP - File Transfer Protocol

Es uno de los protocolos de transmisión de archivos más utilizados en la actualidad, a pesar de que su desarrollo fue en 1971. Está basado en un software que actúa de servidor, permitiendo el envío y recepción de archivos mediante TCP y un cliente el cual realiza solicitudes al servidor.

2.7.7. OPC Server

Es una tecnología de comunicación con una arquitectura cliente/servidor, que cumple con una o más especificaciones definidas, el servidor OPC hace de interfaz comunicando una fuente de datos (PLC's, DCS's, PAC's) y por el lado el receptor (HMI's, SCADA's), en relación el servidor OPC, es el esclavo del sistema y el cliente OPC es el maestro, la comunicación es bidireccional, permitiendo leer y escribir a través del OPC. (Matrikon, 2019)

- Servidor OPC DA. - Transmisión de datos en tiempo real.
- Servidor OPC HDA. - Acceso a Datos Históricos.
- Servidor OPC A&E.- Transfiere Alarmas y Eventos.
- Servidor OPC UA. - Permite a los Servidores OPC trabajar con cualquier tipo de datos.

2.7.8. Sockets

Es un método que ayuda a que dos aplicaciones se comuniquen, los sockets son la base de los protocolos de comunicación en internet, y son una característica universal en cualquier lenguaje de programación JAVA, Python, C#, entre otros; además de entablar comunicaciones locales o remotas. (Ramos A. , 2015)

Para que exista comunicación debe haber:

- Servidor: Aplicación que está a la espera de que se conecte el cliente.
- Cliente: Aplicación que se conecta al servidor.
- Cliente/Servidor: Aplicación que es cliente y servidor al mismo tiempo.

2.8. PLC Simatic S7-1200.

El PLC Siemens S7-1200, es un controlador modular para tareas de automatización sencillas, proporcionando flexibilidad y potencia para cubrir las necesidades de automatización en cada caso, está disponible en el mercado desde el año 2019. Su software de programación predilecto es TIA Portal en sus diferentes versiones y Step7.

Incorpora un microprocesador potente, circuitos de entrada y salida, Profiten y una fuente de alimentación, además de poseer un Web Server integrado y ofrecer protección por contraseña para acceder a la configuración de sus funciones. Permite seleccionar el tamaño de la memoria de programa y la de datos hasta 50 KB.

La velocidad de ejecución en instrucciones matemáticas es de 2.3 instrucciones/microsegundo y en el caso de instrucciones booleanas se ejecutan a 12 mega instrucciones/s. (Mercado, 2019)

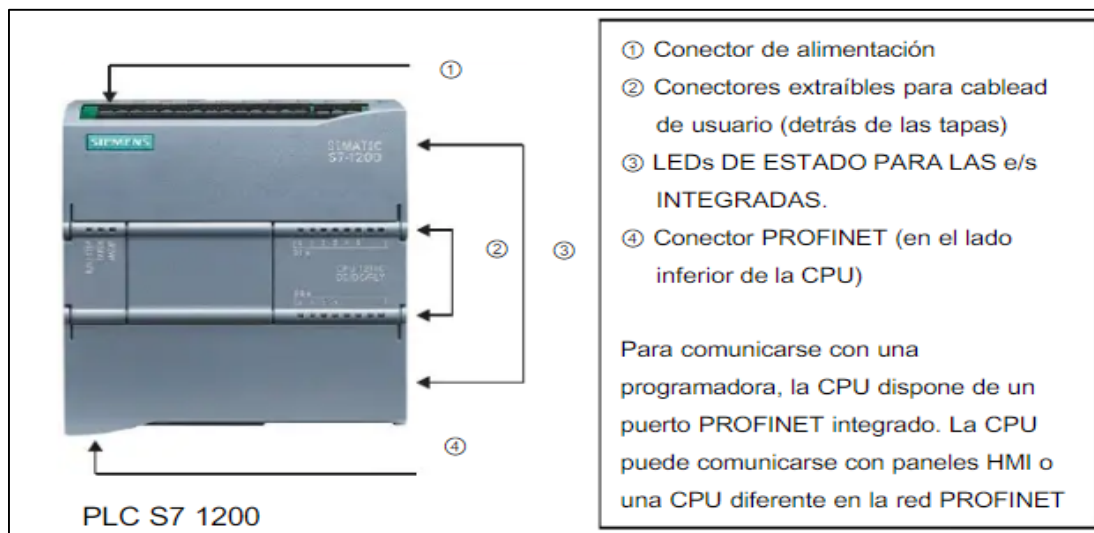


Figura 27. PLC S7-1200.

Fuente: (Mercado, 2019)

2.8.1. Web Server

Es un sistema integrado en el PLC SIMATIC s7-1200, que permite la lectura y escritura de información en la CPU vía web, también ofrece medios de desarrollo en HTML, para que los usuarios creen sus propias aplicaciones. Los códigos creados deben contener comandos AWP (Automation Web Programming), que son los códigos proporcionados por Siemens para acceder a la información contenida en el PLC. También se debe enfatizar que el servidor web no fue diseñado para reemplazar a sistemas SCADA, por lo que es posible encontrar dificultades para realizar algunas acciones. (Martínez, 2016)

El servidor web permite conectarse a él mediante HTTP y HTTPS. Contando con un sistema de delimitación de usuarios basado en la petición de usuario y contraseña.

2.8.2. Comandos AWP.

Los comandos Automation Web Programming (AWP) constituyen una sintaxis especial de comandos para el intercambio de datos entre la CPU y la interfaz web. Los comandos AWP se introducen en forma de comentarios de HTML y ofrecen las siguientes posibilidades descritas en la Tabla 2.

Tabla 2.
Comandos AWP

Función	Representación
Lectura de variables del PLC	<code>:= <Varname>:</code>
Escritura de variables del PLC	<code>< !—AWP_In_Variable Name =' <Varname>:'--></code>
Lectura de variables especiales	<code>< !—AWP_Out_Variable Name =' <Typ>::<name>:'--></code>
Escritura de variables especiales	<code>< !—AWP_In_Variable Name ='<Typ>::<name>:'--></code>

2.9. KUKA KR3 R540

Es un pequeño robot industrial, de con 6 grados de libertad, en la actualidad es el más rápido de su categoría y está diseñado género antropomórfico para efectuar una carga útil de 3kg y posee protección IP 40. (Eurobots, 2019)

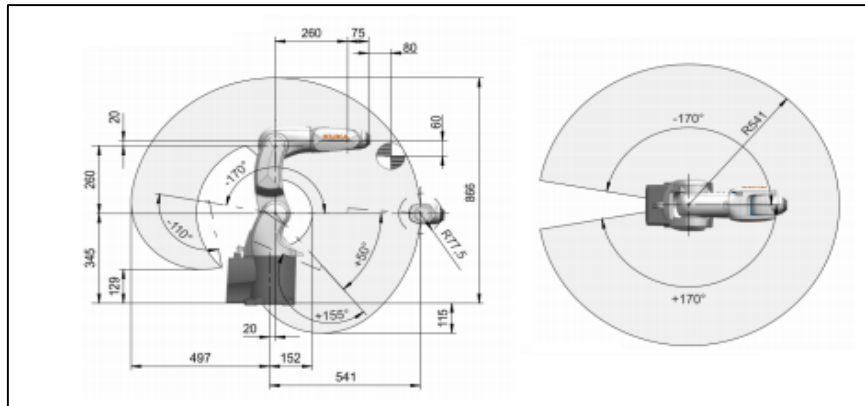


Figura 28. KUKA KR3 R540.

Fuente: (AG.Kuka, 2019)

Se caracteriza por tener una gran movilidad en espacios reducidos, siendo compacto y flexible, es ideal para aplicaciones con objetos pequeños, versatilidad y preciso, requiere mantenimientos simples y por su gran velocidad los tiempos de ejecución de los procesos se reducen. Su montaje puede ser realizado en suelos, paredes o techos. (Eurobots, 2019)

Tabla 3.

Características KUKA KR3 R540.

<i>Datos técnicos</i>	
<i>Alcance máximo</i>	<i>541 mm</i>
<i>Carga máxima</i>	<i>3 kg</i>
<i>respetabilidad</i>	<i>± 0.02 mm</i>
<i>Numero de ejes</i>	<i>6</i>
<i>Posición de montaje</i>	<i>Suelo, techo, pared</i>

2.9.1. Controlador KR C4

Es un controlador potente, seguro y flexible, posee una arquitectura estructurada que se centra en estándares de datos abierto y potentes y posee diversidad de controladores SafetyControl, RobotControl, MotionControl, LogicControl y ProcessControl, Adicionalmente tienen una base de datos conjunta, Profinet y una infraestructura que utilizan y comparten inteligentemente. (Eurobots, 2019)

2.9.1.1. Funciones del controlador KR C4

- Soporte de procesador multi-núcleo para un rendimiento escalable.
- Tarjetas de memoria integrada
- Rápida comunicación vía Gigabit Ethernet

2.10. Lenguajes de programación.

2.10.1. KRL- Kuka Robot Language

Es un lenguaje de programación similar al pascal, desarrollado para controlar robots, y que genera dos tipos de archivo con el mismo nombre, pero diferentes extensiones, un .dat que contiene archivos de datos permanentes y otro .src que contiene archivos de comandos de movimientos. Está basado en la normativa IRL y soporta multitareas. (Pérez & Fernández, 2002)

2.10.2. Ladder

Es el lenguaje más común de programación de PLC, también llamado diagrama de contactos o de escalera, es incorporado por todos los fabricantes de PLC. Y son esquemas de representación de la lógica de control de un proceso industrial.

Este lenguaje es igual en todos los PLC, pero cambia su forma de interpretación de acuerdo a cada fabricante, transformando el código de escalera en ensamblador y posteriormente cargándolo en el microprocesador del PLC. (Cano, 2013)

2.10.2.1. Sus principales características son:

- Instrucciones de entrada a la izquierda.
- Instrucciones de Salida a la derecha.
- Posibilidad de programar compuertas lógicas.
- Salidas y entradas en paralelo o serie.
- El procesador ejecuta el código de arriba a abajo y de izquierda a derecha.




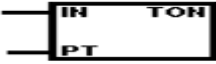
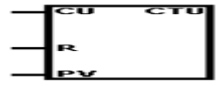
Las entradas son contactos normalmente abiertos o cerrados, que permiten o deniegan la activación de una salida y las salidas son equivalentes a la energización de un circuito eléctrico. En consecuencia, las entradas de un PLC están conectadas a pulsadores o sensores y las salidas están conectadas a los actuadores, que son los encargados de realizar el proceso industrial. (Cano, 2013)

En el lenguaje de escalera existen auxiliares adicionales a las entradas y salida como:

- **Contactos:** Pueden tomar solo dos estados o activado o desactivado.

- **Marcas:** Son variables lógicas virtuales que se utilizan para memorizar estados o acumular resultados.
- **Temporizadores:** Permiten medir el tiempo y efectuar una acción, pueden ser con retardo a la conexión o con retardo a la desconexión.
- **Contadores:** Posiciones de memoria que almacenan un valor numérico.
- **Tag's:** Nombres de identificación de los diferentes elementos que conforman el programa ladder.

Tabla 4.
Símbolos básicos de Lader.

<i>Nombre</i>	<i>Símbolo</i>
<i>Contacto normalmente abierto</i>	
<i>Contacto normalmente cerrado</i>	
<i>Bobina</i>	
<i>Timers</i>	
<i>Contadores</i>	

2.10.3. Python

Es un lenguaje de programación de alto nivel, fácil de aprender que cuenta con un sin número de librerías y algoritmos optimizados para desarrollar aplicaciones comunes y de alta demanda. Con un enfoque simple pero eficiente orientado a la programación de objetos,

posee una sintaxis elegante y tipado dinámico, permitiendo obtener programas más cortos que en JAVA, C o C++. (Brihuega, 2017)

- Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola línea.
- No es necesario la utilización de llaves para agrupar instrucciones, se las realiza mediante sangrías, obteniendo una estructura elegante y de fácil identificación de procesos.
- No se necesita declarar argumentos, ni variables.
- Es un lenguaje interpretado que no necesitamos compilar, ahorrándonos tiempo en el desarrollo y prueba de una aplicación.
- El código puede ser ejecutado en cualquier software que disponga de un intérprete.
- Multiparadigma: Programación asíncrona, Imperativo, Lógico, Funcional, Declarativo, Estructurado, Dirigido por eventos, Modular, Orientado a aspectos, Orientado a objetos. (Pythones, 2019)

CAPÍTULO III

3. DISEÑO DEL SISTEMA Y APLICACIÓN.

El presente capítulo se centra, en describir y detallar todos los subprocesos necesarios para el desarrollo de un sistema de control Cloud Computing.

3.1. Arquitectura de Interconexión.

La metodología empleada para el diseño del proyecto, se basa en una arquitectura de interconexión física de elementos, la cual se despliega en la Figura 29, permitiendo visualizar el entorno macro, de lo que se desea conseguir al finalizar el proyecto, a fin de realizar las aplicaciones de corte térmico y control de calidad, propuestas en el apartado 1.3.

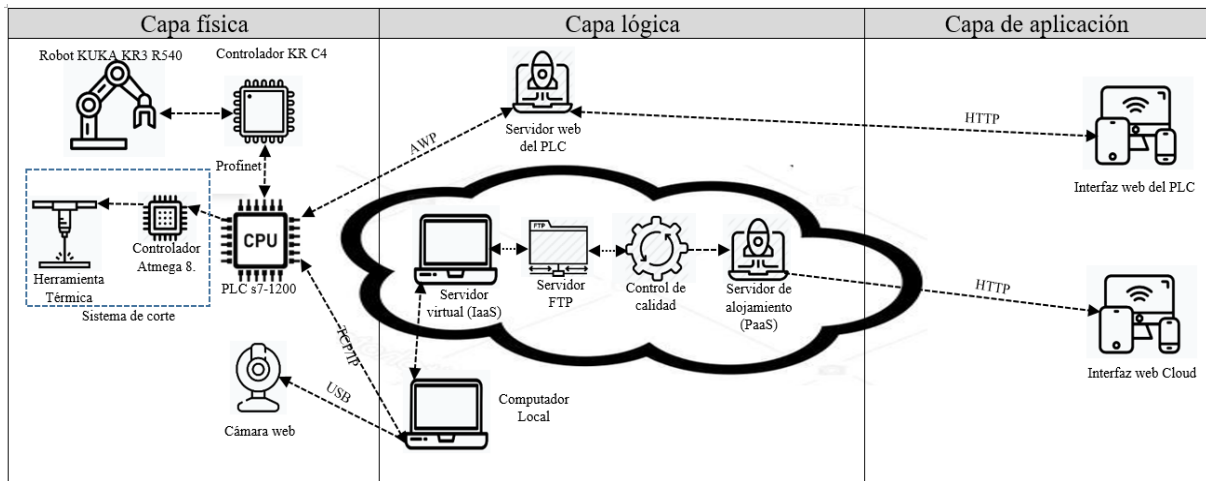


Figura 29. Arquitectura de Interconexión.

Como se muestra en la Figura 29, el sistema posee una capa física en la que se encuentra: el robot KUKA KR3 R540 con su respectivo controlador, la herramienta térmica, la cámara web y el PLC, los cuales se conectan a la capa de lógica, mediante un servidor web integrado en el hardware propio del PLC y también mediante un computador local, que

adquiere la información de la capa física a través de protocolos de comunicación TCP/IP y USB, enviando la información a la nube a través de un servidor FTP, ajado a su vez en un servidor virtual, generando de esta forma dos interfaces de usuario, una perteneciente al servidor web del PLC y otra perteneciente al sistema ubicado en la nube.

3.2. Capa física.

Corresponde a los elementos de hardware que conforman el sistema, tales como:

3.2.1. Robot y el controlador.

El robot KUKA, posee una estructura antropomórfica con 6 grados de libertad, lo que hace que sea ideal para efectuar el proceso de corte, ya que puede generar movimientos complejos y compuestos con sus diferentes articulaciones.

Su velocidad es regulable, teniendo altas velocidad de hasta 1m/s, que con la interacción de la herramienta térmica realizan grabados rápidamente, además con el manejo de bajas velocidades de hasta 0.001m/s, posibilita realizar cortes en materiales blandos como fomix, papel entre otros.

La comunicación predilecta del robot KUKA y el controlador C4, para interactuar con el PLC es Profinet, teniendo un módulo interno de 32 entradas y 32 salidas virtuales, las cuales permiten ejecutar programas KRL que se encargan del movimiento del robot.

3.2.2. PLC.

El PLC funciona como un puente que permite relacionar los elementos de la capa física y con la ayuda del computador local se relaciona con los elementos de la capa lógica. Así

pues, el PLC está conectado mediante Profinet al robot KUKA y a su controlador como se menciona en el apartado anterior.

También, está conectado a la herramienta térmica, las cuales en su mayoría poseen tecnología de control basada en TTL, por lo que al momento de la implementación se deberá realizar un circuito de adaptación que permita controlar el encendido/apagado de la herramienta, así como la regulación de su potencia.

Adictamente, posee un servidor web incorporado en su hardware, el que permite desplegar una interfaz web.

3.2.3. Herramienta Térmica.

Para el desarrollo del sistema de corte térmico, se debe adquirir, primeramente, la herramienta que efectuará los cortes, como opciones se tiene: una resistencia eléctrica o un cabezal láser.

La resistencia eléctrica, es un alambre de nicromo, que realiza el corte por contacto térmico, el cual dependiendo de su calibración logra obtener una gran precisión y buenos acabados, como su principal desventaja se presenta el desgaste y la variación de temperatura a medida que se la utiliza, por lo que se debe realizar mantenimientos y calibraciones constantes, e incluso el cambio de la totalidad de la herramienta, como principal ventaja, tienen un costo reducido y fácil de adquirir.



Figura 30. Resistencia eléctrica de corte.

Los sistemas de corte láser, se caracterizan por realizar procesos precisos de alta calidad con bajo mantenimiento, ya que están conformados por un diodo de luz y varios espejos, que reflejan la luz aumentando su potencia, intensidad y enfoque. Su principal ventaja es la exactitud y repetibilidad, que presentan al efectuar un proceso. Como principal desventaja se presenta su coste elevado.



Figura 31. Diodo láser.

La herramienta seleccionada para desarrollar el corte térmico, fue un cabezal láser, al garantizar su constancia y durabilidad en el tiempo, siendo ideal para grabar, cortar, garabatear, taladrar y ofrece elevada precisión. (SainSmart, 2019)

3.2.3.1. Características:

- Cuenta con gafas de protección visual color verdes.
- Material del disipador de calor: aluminio.
- Potencia de salida: 5,5 W.
- Longitud de onda: 445 nm láser azul.
- Voltaje: 12 V y Corriente: 2 A.
- Forma del haz de luz: punto (enfoque ajustable).
- Tiempo de elevación: >10.000 horas.

- Temperatura de funcionamiento: -40 ~ 167.0 °F.



Figura 32. Cabezal láser.

Fuente: (SainSmart, 2019)

3.2.4. Cámara Web.

En el presente proyecto, se desea evidenciar la potencialidad de las aplicaciones Cloud Computing, para desarrollar procesos industriales, en los que antes era necesarios adquirir dispositivos especializados y de costo elevado.

Por tal motivo, se utilizarán dispositivos con prestaciones mínimas y de bajo costo. Conociendo que, el dispositivo de captura de imagen, es primordial en los sistemas de control de calidad por visión artificial, se utilizará una cámara web, relacionada con Cloud Computing, con el fin de analizar, su desempeño e identificar si es una opción viable a ser implementada en un futuro.

3.2.4.1. Características de la cámara web:

1. Sensor CMOS.

2. Resolución 640 x 480
3. Interfaz USB.
1. 30 cuadros por segundo
2. Tamaño de sensor: 43.86 x 3.64 mm



Figura 33. Cámara web.

3.3. Capa lógica.

Es aquella capa que permite controlar y gestionar los elementos de la capa física emitiendo ordenes de ejecución de programas, recibiendo y enviando información. Los elementos que componen la capa lógica se describen a continuación:

3.3.1. Servidor web del PLC.

Es un servidor web incorporado en el hardware del PLC, en el que estará alojada la interfaz web del PLC, desarrollada en lenguaje HTML y que permitirá modificar el estado de sus variables mediante comandos AWP. La seguridad del servidor está basada en un protocolo de petición de usuario y contraseña.

Cuenta con varios niveles de acceso: permitiendo leer variables, modificarlas, cambiar contraseñas, adicionar variables, borrar variables, dependiendo de la configuración interna que se realice.

3.3.2. Computador local

Está conectado con el PLC a través del protocolo TPC/IP y es el encargado de enviar información desde la capa física a la capa lógica, en él se encuentran los programas de: captura de imagen y de envío de imagen los cuales se ejecutan luego de que el proceso de corte a finalizado.

3.3.3. Servidor Virtual

El servidor virtual recibe la imagen de la pieza cortada que es enviada por el computador local, mediante un servidor FTP interno, una vez se identifica la llegada de la imagen, se procede a realizar el control de calidad, al finalizar este proceso se publican los resultados en la aplicación web Cloud y se envía la respuesta de si la pieza fue rechazada o aceptada a la capa física.

3.3.4. Servidor FTP.

Es el encargado de recibir la imagen enviada por el computador local y guardarla para su posterior utilización en el control de calidad.

3.3.5. Control de Calidad.

Se encarga de procesar la imagen de la pieza cortada, almacenada en el servidor FTP, realizando procesos de reconocimiento de objetos, sus características y la detección de sus bordes, con el fin de identificar posibles errores e irregularidades de la pieza, generados en

el proceso de corte. Una vez realizado todo el proceso de control de calidad, se cargan los resultados en el sitio de alojamiento web y se devuelve como respuesta a la capa física, si la pieza fue aceptada o rechazada.

3.3.6. Sitio de Alojamiento web.

Permite almacenar toda la información referente al control de calidad, y publicarla en un dominio propio, a fin de que la información pueda ser visualizada a nivel mundial.

3.4. Capa de Aplicación.

Son los elementos que posibilita la interacción de los usuarios con los demás elementos del sistema:

3.4.1. Interfaz web del PLC.

La interfaz se encuentra alojada en el servidor web del PLC, cuenta con botones de interacción que permite activar/desactivar variables del programa Lader, el cual a su vez mediante la comunicación Profinet, ejecuta programas en KRL que gestionan el movimiento del robot. Para que se realice el corte, el programa en Lader, también incluye instrucciones de encendido/apagado del láser e intrusiones para regular su potencia.

3.4.2. Interfaz web Cloud

Toda la información necesaria para desplegar la interfaz se encuentra almacenada en el sitio de alojamiento web, en ella se puede visualizar los resultados del control de calidad, así como también información referente a los elementos que conforma el laboratorio de Robótica y PLC de la Universidad de las Fuerzas Armadas ESPE.

3.5. Arquitectura de componentes

La arquitectura de componentes, permite identificar todos los procesos que se deben llevar a cabo para realizar los aplicativos de corte y control de calidad, propuestos en el apartado 1.3.

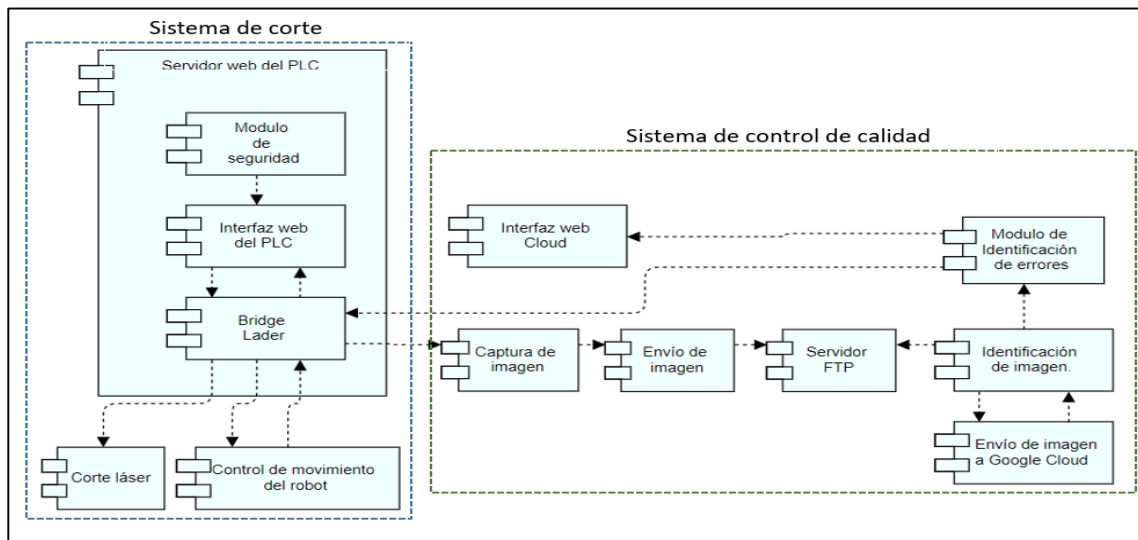


Figura 34. Arquitectura de componentes.

A continuación, se detalla cada uno de los procesos con su respectivo algoritmo de diseño:

3.6. Sistema de corte.

3.6.1. Módulo de seguridad.

La seguridad, es el parámetro más importante y crítico en la implementación de una aplicación web, ya que, si no se implementa de una forma adecuada, vuelve vulnerable de ataque internos o externos, a un proceso de automatización. Es por este motivo que la seguridad, estará sostenida en los procesos propios de Siemens. Que es una empresa; líder en la fabricación de equipos industriales para la automatización, con procesos y estándares de seguridad muy elevados.

El módulo de seguridad del PLC, se basa en un protocolo de petición de usuario y contraseña, en el que se puede configurar el nivel de acceso. El diseño del módulo de seguridad, se establece en el algoritmo de la Figura 35.

Para ingresar al módulo de seguridad, se lo debe hacer mediante el software de programación del PLC, en el cual se configura el nombre de usuario y la contraseña, así como también el nivel de acceso que tendrá dicho usuario, una vez realizadas las configuraciones se procede a cargarlas en el PLC.

Para identificar su correcto funcionamiento se ingresa a la dirección IP del PLC y se llena los apartados de usuario y la contraseña, previamente definidos, si los datos proporcionados son correctos permite desplegar la interfaz web.

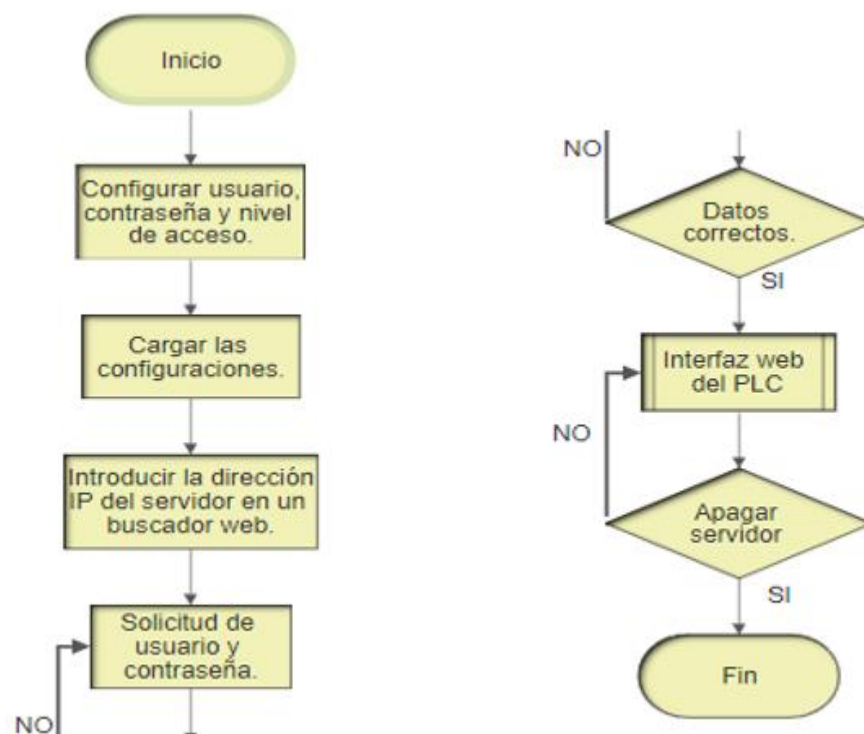


Figura 35. Algoritmo del módulo de seguridad.

3.6.2. Interfaz web del PLC.

La estructura de la interfaz web, está basada en un sistema de navegación amigable con el usuario, siendo capaces de desplegar la información deseada, mediante la selección en la barra de menú. La interfaz web además cuenta con información referente al laboratorio de robótica industrial y PLC de la Universidad de las Fuerzas Armadas - ESPE, como categorías de Robots, Temas de investigación y Aplicaciones, como se muestra en la Figura 36.

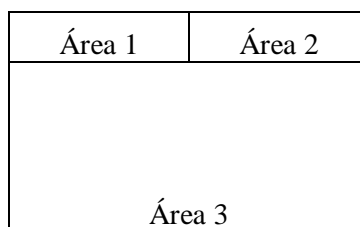


Figura 36. Estructura de la interfaz web del PLC.

En el Área 1, existe un logo de la Universidad de las Fuerzas Armadas ESPE, en el Área 2, existe un menú en el cual está conformado por: robots, temas, aplicación, desarrollo.

En el Área 3, existen diferentes ventanas con información referente al laboratorio de robótica y PLC como: Categorías de Robots, Temas de Investigación, Aplicaciones de Robótica, Descripción del proyecto, Botones de ejecución, Información del desarrollador.

El título de la interfaz, tiene un tamaño de 24 pixeles, en negrita y de color blanco, con un fondo tipo imagen, que corresponde a una postal de la Universidad de las Fuerzas Armadas-ESPE.

Los subtítulos tienen un tamaño de 16 pixeles, con letra negrita y color negro.

El texto en general, tiene un tamaño de 12 pixeles, con letra time new roman de color negro y fondo blanco. Por su parte las imágenes tienen un tamaño de 150x100 pixeles, centradas.

Pero la función principal de la interfaz web, es ejecutar los procesos de corte y control de calidad del sistema Cloud Computing. Los botones cumplen la función tanto de activación como desactivación de las variables del programa en lenguaje Lader. En la interfaz tienen un tamaño de 50x50 pixeles, de forma cuadrada y su relación con el lenguaje lader y AWP, se describe en el algoritmo de la Figura 37.

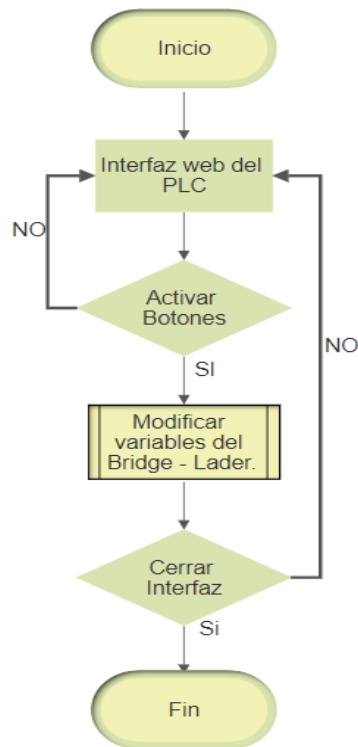


Figura 37. Algoritmo de la interfaz web del PLC.

La sintaxis de los comandos AWP es: “*AWP_IN_Variable NAME 'entrada 17'*” y se varia su valor o estado con “*name='entrada 17' value='0 o 1'*”

3.6.3. Bridge-Lader.

El programa en Lader cumple la función de puente entre los comandos ejecutados en la interfaz web del PLC y el programa de control de movimiento del robot en KRL. Así como también la activación/desactivación del láser. El algoritmo de diseño del Bridge- Lader se muestra en la Figura 38.

Una vez es enviada la instrucción de corte deseada, mediante la interfaz web del PLC, a la cual se accede a través de dispositivos como: tablets, celulares o computadores.

Se identifica la variable que cambio de estado, y se activa/desactiva dicha variable en el programa de Lader, el cual comunica a su vez al robot la secuencia de movimientos, que debe realizar para obtener el corte de la pieza deseada.

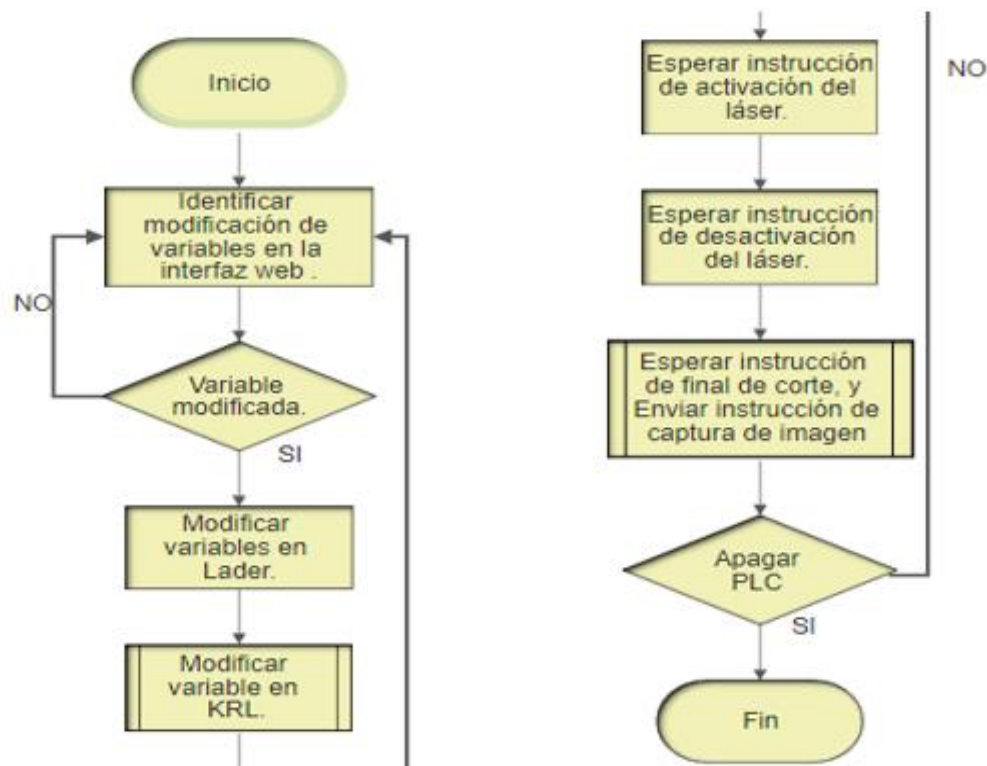


Figura 38. Algoritmo para diseñar el Bridge-Lader.

3.6.4. Control de movimiento.

El control de movimiento de robot se basa en un programa en KRL, el cual está a la espera de instrucciones emitidas por el PLC. De acuerdo a la instrucción del PLC el robot efectuar uno u otro movimiento. El algoritmo que se emplea para este fin se detalla en la Figura 39. Una vez que el robot recibe la instrucción del PLC, de empezar con el proceso de corte, se coloca en posición encima del fomix y envía de vuelta al PLC la instrucción de activar el láser.

Al activarse el láser, el robot comienza a realizar los movimientos necesarios para cortar la figura y al finalizar envía la instrucción de desactivar el láser.

Una vez a terminado el proceso de movimiento del robot y este vuelve a su posición inicial, envía la instrucción de captura de imagen al PLC, el cual la réplica al computador local.

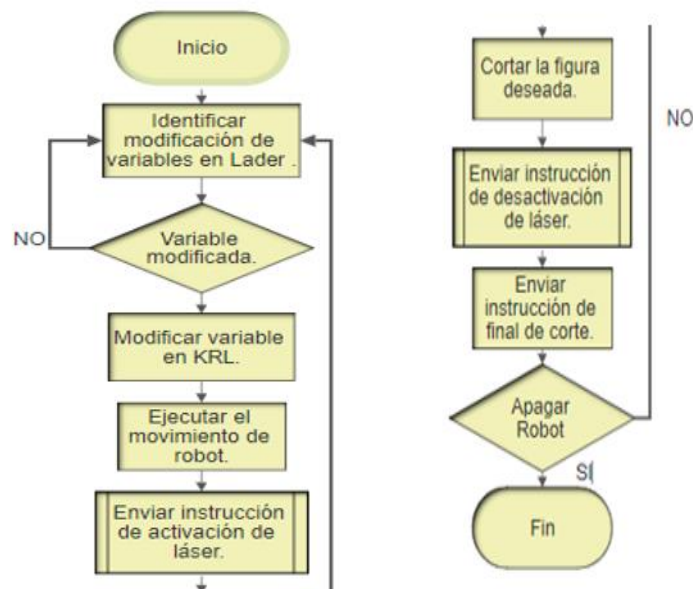


Figura 39. Algoritmo de control de movimiento del robot.

3.6.5. Corte láser.

El sistema de corte laser está a la espera de que el PLC, emita la instrucción de activación o desactivación, para poder ejecutar el corte. Pero cabe recalcar que, aunque es el PLC el que da la orden directa, es el programa en KRL el cual emite la instrucción en el momento en que se necesita que el cabezal láser este activado o no. El algoritmo que se ejecuta para la activación/desactivación del láser se muestra en la Figura 40.



Figura 40. Activación del Pacer

3.7. Sistema de control de calidad.

3.7.1. Captura de imagen.

Al terminar el proceso de corte, se envía una instrucción de captura de imagen al computador local, como se describe en el apartado 3.6.4. Una vez se recibe esta instrucción se activar la cámara web y se solicita la captura de imagen, al finalizar la captura se procede a almacenarla en la memoria interna de computador, dando paso al componente de envío de imagen. El algoritmo necesario para la captura de imagen se describe en la Figura 41.

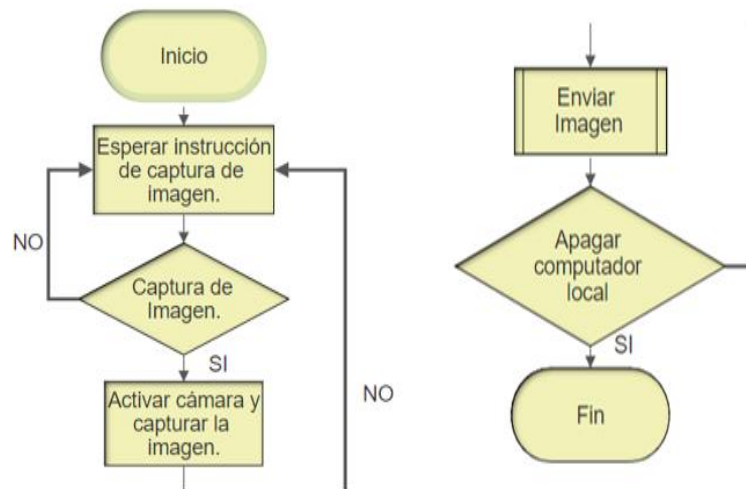


Figura 41. Algoritmo de Captura de imagen

3.7.2. Envío de Imagen.

Para el envío de imagen, se debe configurar las credenciales del servidor FTP que estará, alojado en el servidor virtual en la nube, luego se procede a leer la imagen y enviarla al servidor FTP a través de internet, este proceso se describe en la Figura 42.

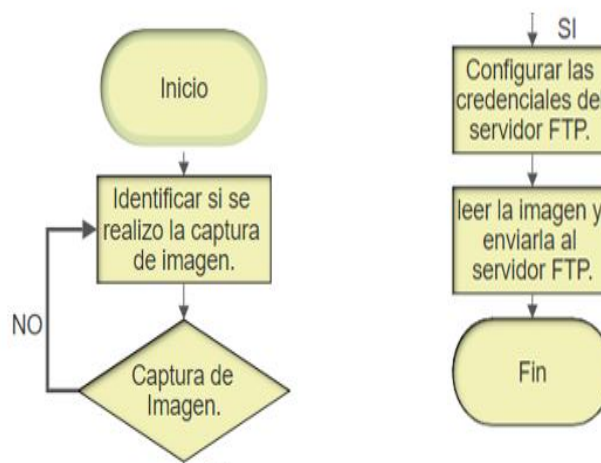


Figura 42. Algoritmo de envío de Imagen.

3.7.3. Servidor FTP.

El servidor FTP, se encarga de recibir y almacenar las imágenes de las piezas cortadas para luego ser procesadas, tal y como se detalla en el algoritmo de la Figura 43, dicho algoritmo se encuentra en un bucle infinito, ya que el servidor siempre está a la espera de que lleguen nuevas imágenes, solo finaliza cuando el servidor virtual se encuentra suspendido o apagado.

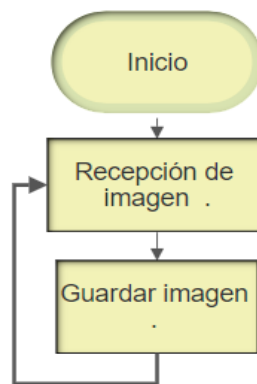


Figura 43. Algoritmo de recepción FTP.

3.7.4. Identificación de imagen en el servidor FTP.

El procesamiento de imágenes se realiza en dos etapas, la primera etapa detecta si en el servidor FTP existe alguna imagen a ser procesada, de ser el caso afirmativo procede a enviarla al procesamiento de imagen 2 y de ser el caso negativo sigue en espera de que llegue una imagen. El algoritmo para efectuar el procesamiento de imagen se despliega en la Figura 44.

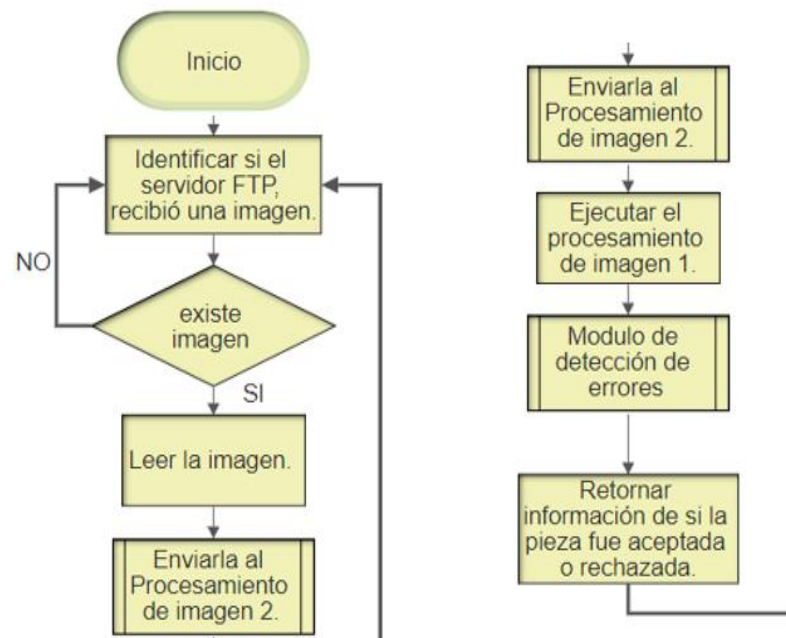


Figura 44. Algoritmo para el procesamiento de la imagen 1.

En él envió de imagen a Google Cloud, se extraen las características y atributos de la imagen mediante un servidor de visión artificial alojado en la nube y las retorna al procesamiento de imagen 1, como se muestra en el algoritmo de la Figura 45.

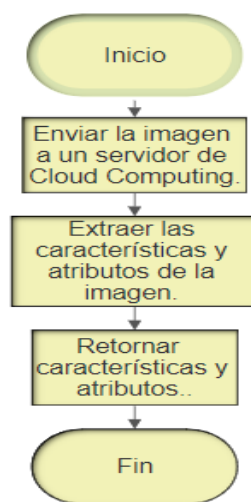


Figura 45. Algoritmo para el procesamiento de la imagen 2.

Una vez que se retorna las características y atributos de la figura cortada, en el procesamiento de imagen 1, se procede a extraer los bordes de la imagen original mediante un software de Vision Computing, al extraer los bordes de la figura cortada se los precede a comparar con los bordes de una figura perfecta, a fin de extraer los errores y desperfectos de la pieza. Luego de realizado todo el procesamiento de imagen y control de calidad se procede a retornar información de si la pieza ha sido aceptada o rechazada, al computador local y cual a su vez la remite al PLC.

3.7.5. Módulo de detección de Errores.

En el módulo de detección se procede a remarcar y contar los errores. Al finalizar todo el proceso, se carga la información del sistema de control de calidad en el sitio de alojamiento web como se muestra en el algoritmo de la Figura 46.

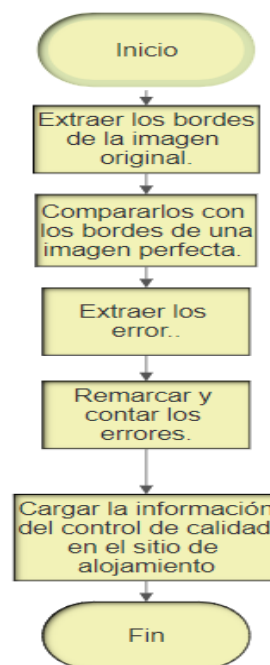


Figura 46. Algoritmo de detección de errores.

3.7.6. Interfaz web Cloud

La estructura de la interfaz web Cloud es similar en su estructura y contenido a la interfaz web del PLC, en su estructura y contenido, diferenciándose en algunos aspectos particulares, la aplicación web Cloud cuenta con toda la información referente al control de calidad, y no posee botones de ejecución. Pero al igual que la interfaz web del PLC, está basada en un sistema de navegación amigable con el usuario, siendo capaces de desplegar la información deseada, mediante la selección en la barra de menú. La interfaz web además cuenta con información referente al laboratorio de robótica industrial y PLC de la Universidad de las Fuerzas Armadas - ESPE, como categorías de Robot, Temas de investigación, Aplicaciones y resultados del control de calidad, como se muestra en la Figura 47.

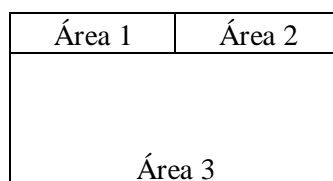


Figura 47. Estructura de la interfaz web Cloud.

En el Área 1, existe un logo de la Universidad de las Fuerzas Armadas ESPE, en el Área 2, existe un menú en el cual está conformado por: robots, temas, aplicación, desarrollo.

En el Área 3, existen diferentes ventanas con información referente al laboratorio de robótica y PLC como: Categorías de Robots, Temas de Investigación, Aplicaciones de Robótica, Descripción del proyecto, Botones de ejecución, Información del desarrollador

El título de la interfaz, tiene un tamaño de 24 pixeles, en negrita y de color blanco, con un fondo tipo imagen, que corresponde a una postal de la Universidad de las Fuerzas Armadas-ESPE.

Los subtítulos como: Categorías de Robots, Temas de Investigación, Aplicaciones de Robótica, Descripción del proyecto, Botones de ejecución, Información del desarrollador, tienen un tamaño de 16 pixeles, con letra negrita y color negro.

El texto en general, tiene un tamaño de 12 pixeles, con letra time mew roman de color negro y fondo blanco. Por su parte las imágenes tienen un tamaño de 150x100 pixeles, centradas.

3.8. Arquitectura Cloud Computing.

Se procede a describir cada una de las capas de la arquitectura Cloud Computing, a fin de interrelacionarla con la arquitectura de interconexión y de componentes:

3.8.1. Red como servicio.

La red implementada establece conexión entre el robot KUKA KR3 R540, PLC s7-1200, Cabezal láser y dispositivos de visualización/ejecución como: computadores, tablet y celulares. Realizando así las conexiones mostradas en la Figura 48.

- El PLC y el robot, se comunican mediante Profinet. (2.7.1 y 4.5.1).
- La comunicación entre el PLC y los dispositivos de visualización/ejecución se realiza, con la ayuda de un Gateway con Protocolo HTTP (2.7.3) y comandos AWP. (2.8.1 y 4.3.3).

- El PLC y el cabezal láser se conectan mediante un microcontrolador Atmega 8, para efectuar el corte deseado.

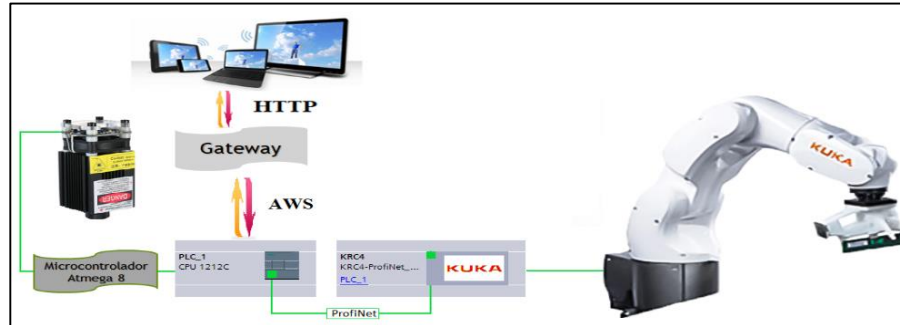


Figura 48. Implementación red como servicio.

3.8.2. Software como servicio.

Se desarrolla un programa en Python, el cual captura una imagen, cuando el PLC termina de cortar la pieza seleccionada, mediante una cámara web y la envía a la plataforma de Google Cloud, para ser procesada por la API de Visión, retornando como resultado sus características, e interaccionando con el Software OpenCV, se efectúa el control de calidad por Visión Artificial. Las conexiones necesarias para este proceso se muestran en la Figura 49.

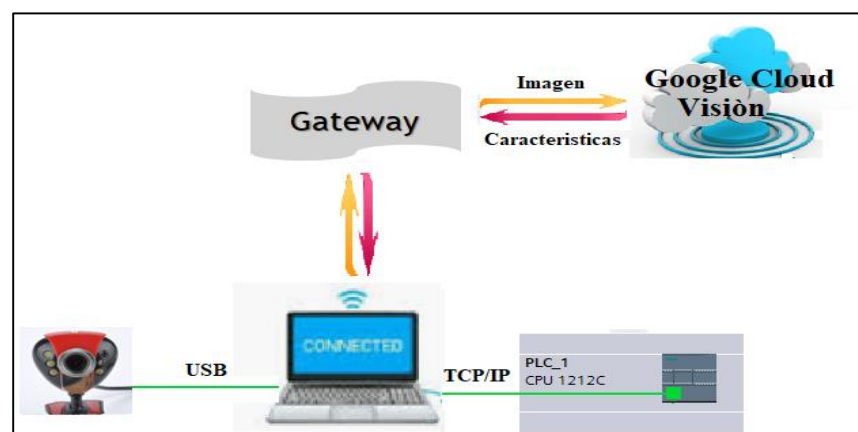


Figura 49. Uso de software como servicio

3.8.3. Plataforma como servicio.

A fin de poder mostrar los resultados del sistema de control de calidad, se crea una cuenta de usuario en la Plataforma Cloud Computing de Heroku. La cual permite desarrollar aplicaciones propias y cargar todos los recursos necesarios para su ejecución, a través de repositorios de Git.

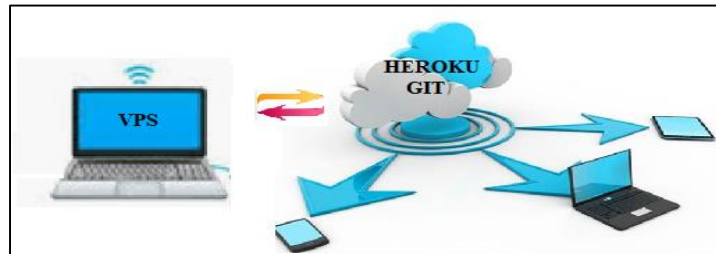


Figura 50. Uso de plataforma como servicio

3.8.4. Infraestructura como servicio.

Las conexiones necesarias para entablar comunicación entre un VPS y enviar archivos mediante FTP, se realizan mediante conexiones virtuales.

Un computador local, envía la imagen de la pieza cortada mediante protocolo FTP a un servidor virtual de Amazon Web Services, donde se guarda y se ejecuta todo el procesamiento de la imagen para realizar el control de calidad y se despliega la Aplicación Cloud Computing.

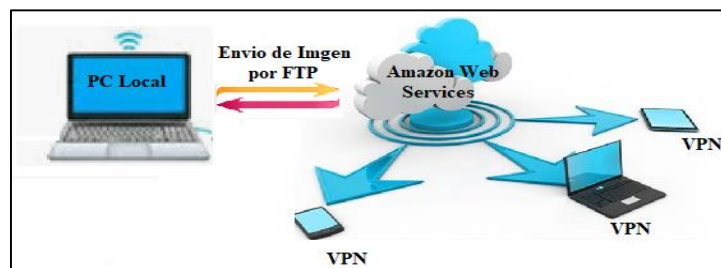


Figura 51. Uso de infraestructura como servicio.

3.9. Diseño de prácticas de laboratorio.

3.9.1. Práctica 1: Introducción a los servidores web.

3.9.1.1. Objetivo General.

Implementar una interfaz gráfica que permita, ejecutar movimientos en el robot KUKA KR3 R540, con ayuda del PLC s7-1200.

3.9.1.2. Objetivo Específicos.

- Establecer comunicación Profinet, entre el PLC s7-1200 y el Robot KUKA KR3 R540.
- Configurar el Web Server, propio del PLC, permitiendo que exista comunicación bidireccional entre las variables del PLC y las de la interfaz gráficas.
- Desarrollar e implementar una interfaz web en lenguaje HTML, que incluya códigos AWP, con la finalidad de modificar el estado de las variables del PLC.
- Efectuar códigos en lenguaje Lader Y KRL, que permitan interactuar con la interfaz gráfica.

3.9.1.3. Descripción general de actividades.

La primera práctica está enfocada a la ejecución de algoritmos desarrollados en lenguaje Lader y KRL, mediante la utilización de una aplicación web. Para este fin se debe efectuar una serie de pasos:

1. Prerrequisitos:

- Instalación de software TIA Portal.
- Conocer el protocolo de comunicación Profinet.
- Instalación de un editor de lenguajes de web.
- Revisar la documentación sobre el Web server de los PLC's Siemens.

2. Activar el Web Server.

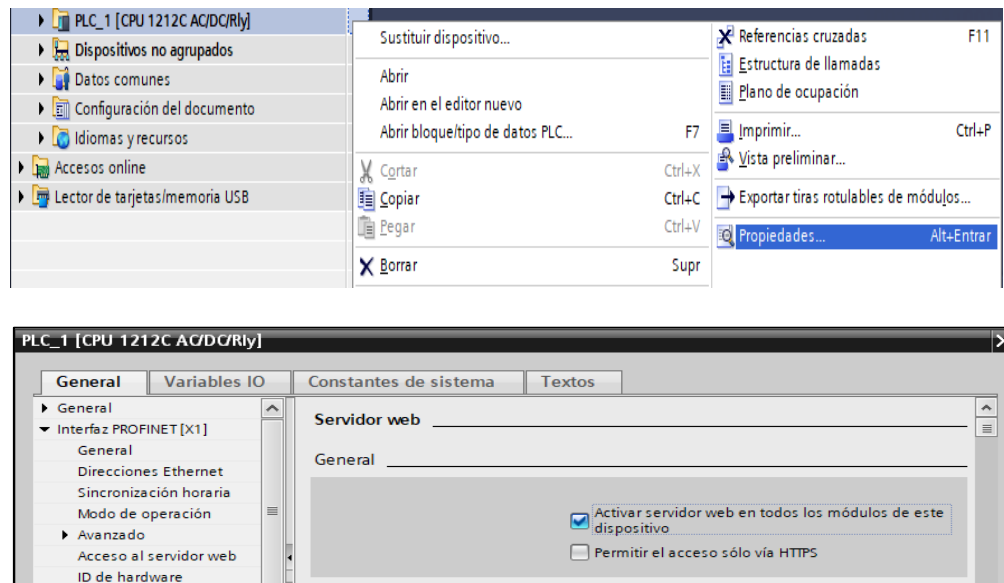


Figura 52. Activación de Web Server.

3. Crear un Botón de Ejecución en Lenguaje HTML, y relacionarlo con código AWP.

```

<!-- AWP_In_Variable Name='entrada 17' -->
<form method="post" action="" id="arranca" name="arranca">
  <input type="submit" value="Start 17" id="boton">
  <input type="hidden" name="entrada 17" value="1">
</form>
a)

```

Figura 53. Código HTML, relacionado con AWP

4. Establecer comunicación Profinet, entre el robot y el PLC.
5. Relacionar el lenguaje Lader, con los nombres de las variables del botón de ejecución.

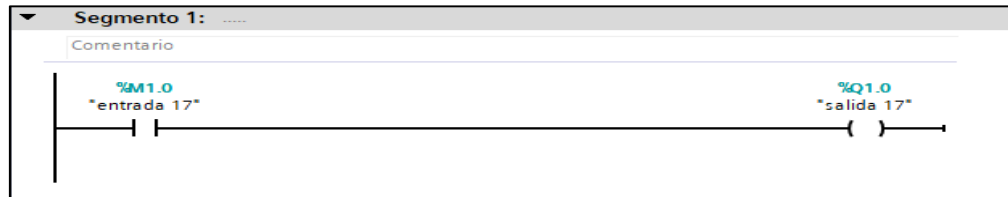


Figura 54. Lenguaje Lader, relacionado con el código AWG y HTML.

6. Identificar la variación del estado de las entradas del robot, para ejecutar el proceso de control de movimiento.

```
EXT BAS(BAS_COMMAND :IN,REAL :IN)
```

```
DECL AXIS HOME
```

```
INT I
```

```
IF $IN[17]== TRUE THEN
```

```
  LIN {X 0,Y 0, Z 50, A 180, B -1.35, C -180}
```

```
  LIN {X 50,Y 0, Z 50, A 180, B -1.35, C -180}
```

```
  LIN {X 50,Y 50, Z 50, A 180, B -1.35, C -180}
```

```
PTP HOME
```

7. Cargar y ejecutar todos los Programas.

3.9.2. Práctica 2: Servicios Cloud Computing.

3.9.2.1. Objetivo General.

Realizar el corte láser de figuras de fomix, e identificar objetos mediante servicios de Cloud Computing.

3.9.2.2. Objetivo Específico.

- Establecer comunicación Profinet, entre el PLC s7-1200 y el Robot KUKA KR3 R540.
- Utilizar sistemas de reconocimiento por visión artificial, provistos por la plataforma de Google Cloud.
- Cortar mediante un cabezal láser, figuras de fomix, que puedan ser identificadas, por visión artificial.
- Ejecutar el sistema de corte, mediante una aplicación web.

3.9.2.3. Descripción general de actividades

La práctica número 2, está enfocada en el corte de piezas de fomix y el reconocimiento de objetos a través de servicios de visión artificial en la nube. Para lograr la implementación de esta práctica se debe seguir una serie de pasos descritos a continuación.

1. Prerrequisitos.

- Haber realizado la práctica 1.
- Un computador con acceso al escritorio remoto.

2. Realizar y cargar todos los programas de la práctica 1.
3. Realizar el código en KRL, con la activación y desactivación del láser, que se efectúa con el encendido o apagado de la salida 17, de robot KUKA KR3 R540.

```
EXT BAS(BAS_COMMAND :IN,REAL :IN)
```

```
DECL AXIS HOME
```

```
INT I
```

```
IF $IN[17]== TRUE THEN
```

```
  LIN {X 0,Y 0, Z 50, A 180, B -1.35, C -180}
```

```
  $SYN OUT[17], State = TRUE, Delay = 0
```

```
  LIN {X 50,Y 0, Z 50, A 180, B -1.35, C -180}
```

```
  LIN {X 50,Y 50, Z 50, A 180, B -1.35, C -180}
```

```
  $SYN OUT[17], State =FALSE, Delay = 0
```

```
PTP HOME
```

4. Abrir el escritorio remoto, y conectarse al servidor virtual con el usuario y contraseña provista por el tutor.



Figura 55. Servidor virtual remoto.

5. Una vez dentro de servidor virtual, abrir Python.
6. Llamar las bibliotecas y credenciales de Google Cloud Vision.

```
# Importar las librerías de gogle cloud.
from google.cloud import vision
from google.cloud.vision import types

#Enviar credenciales a google cloud.

os.environ['GOOGLE_APPLICATION_CREDENTIALS']=
"C:/Users/Administrator/Desktop/FInal/pythonl/Scripts/Tesis/Credenciales
Tesis.json"
```

7. Seleccionar la imagen, en la cual se quiere identificar el objeto. Y enviarla a Google Cloud Platform.

```
imagen1='C:/FTP/Corte/pieza cortada.png'
client = vision.ImageAnnotatorClient()
file_name = os.path.join(os.path.dirname(__file__),imagen1)
# Loads the image into memory
with io.open(file_name, 'rb') as image_file:
    content = image_file.read()
image = vision.types.Image(content=content)
# Performs label detection on the image file
response = client.label_detection(image=image)
atributos = response.label_annotations
print("Descripcion de la imagen:")
#print(atributos)
for atributo in atributos:
    print ("El objeto es un: {0}, Con una Certeza del: {1}
%".format(atributo.description,atributo.score))
```

8. Ejecutar el Programa de Python, para obtener las características y atributos de la imagen.

CAPÍTULO IV

4. IMPLEMENTACIÓN DE ARQUITECTURA CLOUD COMPUTING

4.1. Activación de Web Server

Para que la aplicación Web, se ejecute dentro del PLC, se debe activar y configurar el Web Server. Para ello en el software TIA Porta, se realiza los siguientes pasos:

1. Clic derecho en “PLC_1” y seleccionar “Propiedades”.

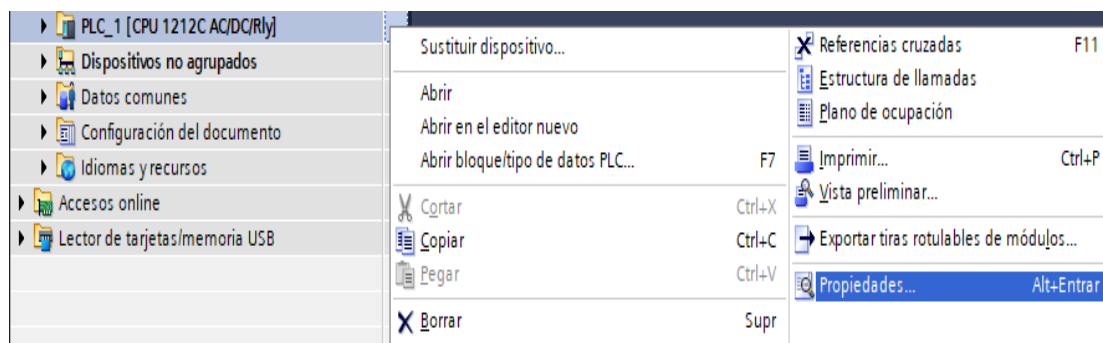


Figura 56. Configuración de propiedad del PLC S7-120.

2. Clic en la casilla de “Activar servidor web”.

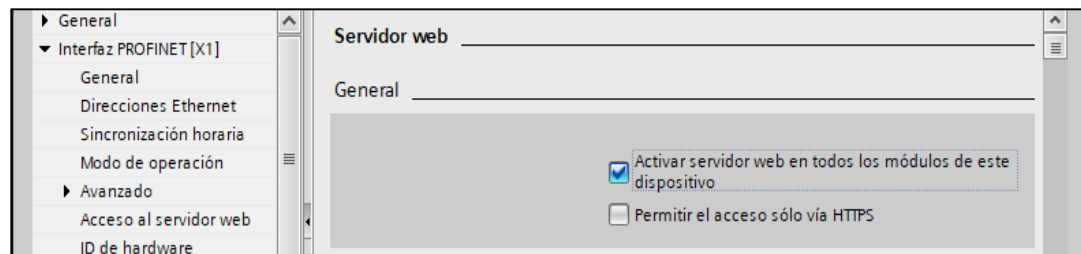


Figura 57. Activación de Web Server.

3. Se configura la aplicación de usuario, buscando la ubicación del archivo HTML y seleccionando el Archivo index.html, luego se da clic en generar bloques.

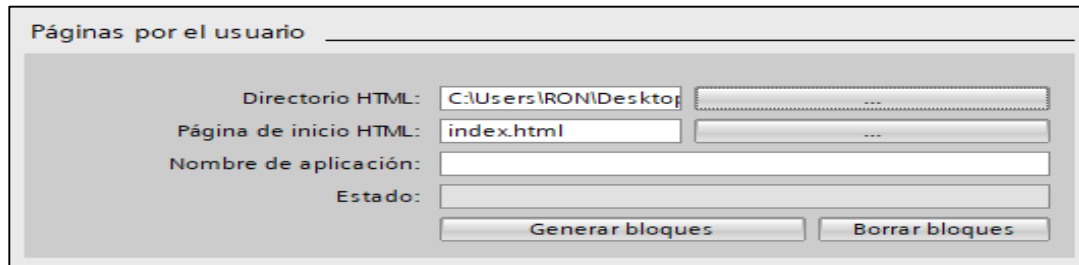


Figura 58. Páginas de usuario en el PLC S7-1200.

4. Clic en “Aceptar” y se observa, que se generan los bloques DB333 y DB334.

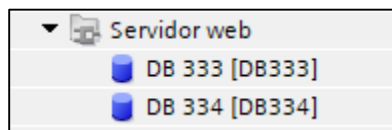


Figura 59. Bloque del servidor web.

5. Para finalizar, se configura la instrucción WWW, como se muestra en la Figura 60.

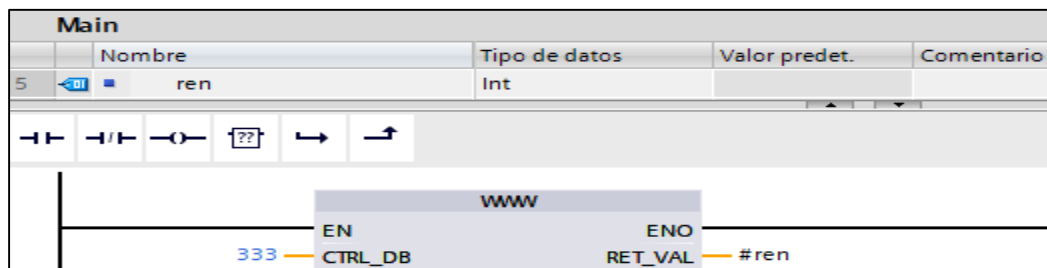


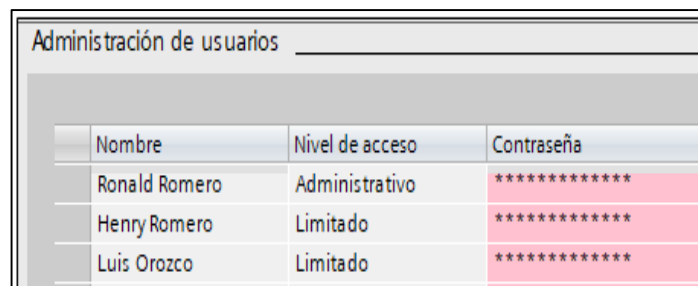
Figura 60. Instrucción WWW.

6. Luego de cargar el proyecto en el PLC, la aplicación web esta lista para ser usada, accediendo desde un buscador a la dirección IP 172.31.1.2.

4.2. Configuración del módulo de seguridad.

El módulo de seguridad de PLC, está basado en un protocolo de petición de usuario y contraseña, con los cuales los usuarios tendrán diferentes niveles de acceso de acuerdo a la figura 62.

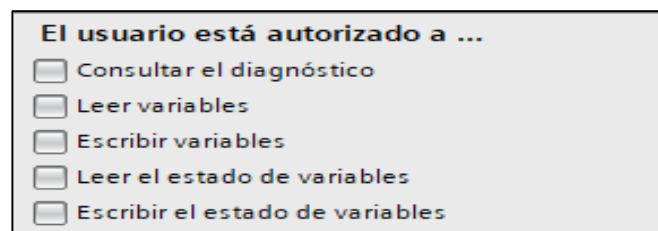
La Administración de usuarios de Siemens, posee una base de datos, en la cual se delimita a los usuarios que pueden ingresar a la aplicación web, como se muestra en la Figura 61 .



Nombre	Nivel de acceso	Contraseña
Ronald Romero	Administrativo	*****
Henry Romero	Limitado	*****
Luis Orozco	Limitado	*****

Figura 61. Usuarios con acceso a la aplicación web.

Por su parte, el nivel de accesos de cada uno de ellos. Tiene varias opciones de autorización, como se detalla en la Figura 62.



El usuario está autorizado a ...

- Consultar el diagnóstico
- Leer variables
- Escribir variables
- Leer el estado de variables
- Escribir el estado de variables

Figura 62. Autorización de acceso a la aplicación web.

4.3. Interfaz web del PLC

Para realizar la interfaz se utilizará el Web Server, integrado en el PLC s7-1200 y lenguaje web HTML.

Aunque HTML, otorga formato y estructura a las aplicaciones web, descuida en su totalidad la estética, por lo que se incorpora estilos en lenguaje CSS, que ayudan a mejorar la presentación visual de una aplicación web, haciéndola más amigable y fluida para el usuario. El código CSS, otorga atributos a un texto, botón o imagen como: color, tamaño y posición.

```
.availability-calendar-toolbar {  
  padding-bottom: 15px;  
  font-size: 1.4em;  
  margin-bottom: 30px;  
  color: #fff;  
}
```

Figura 63. Código CSS.

4.3.1. Tamaño de la aplicación web.

El primer problema que se presenta al realizar una aplicación web, es la diversidad de tamaños de pantallas en los que será visualizada, por lo que se debe agregar instrucciones de acondicionamiento de tamaño, que permiten a una aplicación web, desplegarse tanto en computadores, celulares o Tablet, sin generar distorsiones en el contenido. Estas instrucciones son “meta” y su estructura se puede visualizar en la Figura 64.

```
<meta name="viewport" content="width=device-width,  
| initial-scale=1, shrink-to-fit=no">
```

Figura 64. Instrucción de acondicionamiento de tamaño.

4.3.2. Actualización de aplicación web

Se visualiza que la aplicación web, se recarga cada vez que existe un cambio de estado en alguna de sus variables, generando retardos y tiempos de espera innecesarios. Este

problema se resuelve haciendo una llamada a una función en Java Script. Que se puede visualizar en la Figura 65.

```

$(document).ready(function() {
    // Enviar la información de los botones sin recargar la página.
    $.ajaxStart(function() {
        $('#loading').show();
        $('#result').hide();
    }).ajaxStop(function() {
        $('#loading').hide();
        $('#result').fadeIn('fast');
    });
    $('#arranca, #para, #reinicia').submit(function() {
        $.ajax({
            type: 'POST',
            url: $(this).attr('action'),
            data: $(this).serialize(),
            success: function(data) {
                $('#result').html(data)
            }
        })
        return false;
    });
});

```

Figura 65. Función para evitar recarga de la aplicación web.

4.3.3. Comunicación con variables del PLC.

La comunicación del PLC, con la aplicación web se realiza a través de comandos AWP, los cuales se insertan en el código HTML, al inicio de todo, y se los declara con el mismo nombre que tienen las variables en el PLC, para poder ser leídas o modificadas desde la aplicación web. La sintaxis del lenguaje AWP se muestra en la Figura 66.

```

<!-- AWP_In_Variable Name='entrada 17' -->
<!-- AWP_In_Variable Name='entrada 18' -->
<!-- AWP_In_Variable Name='entrada 19' -->
<!-- AWP_In_Variable Name='entrada 20' -->
<!-- AWP_In_Variable Name='entrada 21' -->

```

Figura 66. Sintaxis del lenguaje AWP.

Para poder modificar o leer el estado de una variable, se debe identificar en la etiqueta “name” y en la etiqueta “id” se vincula con la función de Java Script de acuerdo a su propósito, en este caso en particular se modificarán las entradas de PLC, por lo que se

generarán botones con el método “post”, que ayuden a activar o desactivar las variables. La Figura 67 a), muestra el código necesario para activar una variable al presionar un botón, y la Figura 67 b), muestra el código para desactivarla.

```

<form method="post" action="" id="arranca" name="arranca">
  <input type="submit" value="Start 17" id="boton">
  <input type="hidden" name="entrada 17" value="1">
</form>
a)

```

```

<form method="post" action="" id="para" name="para">
  <input type="submit" value="Stop 17" id="boton">
  <input type="hidden" name="entrada 17" value="0">
</form>
b)

```

Figura 67. Código para modificar el estado de las variables de PLC.

4.4. Bridge-Lader.

Para la configuración Profinet en el PLC, se utiliza el Software TIA Porta, que es desarrollado por Siemens.

Luego de instalar el programa TIA Portal, se conecta cables Ethernet entre el computador, PLC y Robot KUKA, mediante un switch. Y se verifica que sus direcciones IP, se encuentren en la misma subred.

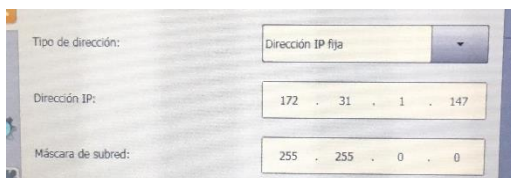


Figura 68. IP del robot KUKA.

En este caso en particular la asignación de la IP de robot KUKA es 172.31.1.147 como se muestra en la Figura 68. Por lo tanto, el computador y el PLC deberán estar en la misma subred.

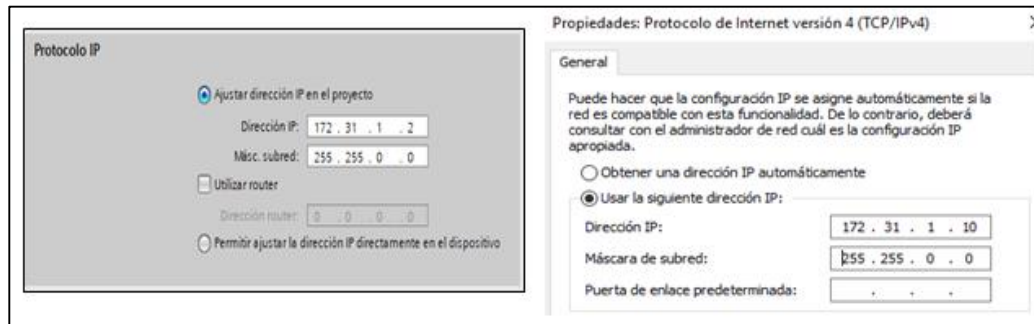


Figura 69. a) IP del PLC, b) IP del computador.

La Figura 69 a), muestra la IP asignada al PLC, correspondiente a 172.31.1.2 y la Figura 69 b), muestra la IP asignada al computador, correspondiente a 172.31.1.10, con esta asignación, se asegura, que todos los elementos estén bajo la misma subred.

Luego de verificar la asignación de todas las direcciones IP, se procede a configurar la comunicación Profinet, en el PLC.

Se Abre el programa TIA Portal, se lo configura con el PLC S7-1200 y se agrega las librerías del controlador KR C4 de KUKA. Para ello se siguen los siguientes pasos:

1. Clic en “*Opciones*”- “*Administrar archivos de descripción de dispositivos*”

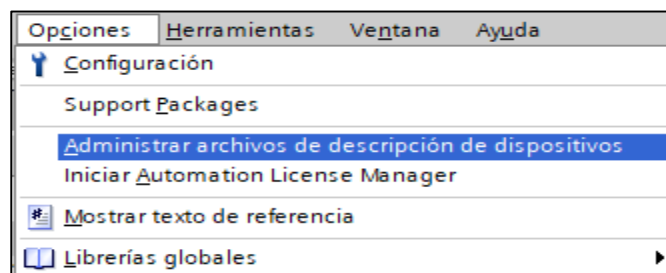


Figura 70. Opciones de TIA Portal.

2. Buscar la carpeta GSDML, que se encuentra en los archivos de instalación de WorkVisual.

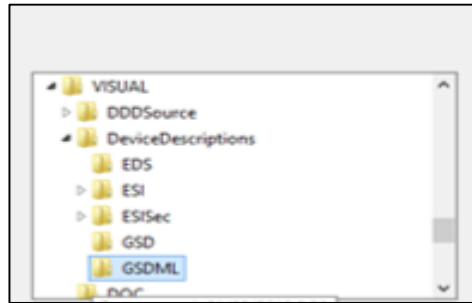


Figura 71. Librería GSDML de KUKA.

- Se instala y agrega el bloque del controlador KR C4 de KUKA. Que tiene la forma mostrada en la Figura 72.



Figura 72. Controlador KR C4 de KUKA.

- La ubicación del controlador KR C4, se muestra en la Figura 73.



Figura 73. Ubicación de controlador KR C4 en TIA Portal.

A continuación, se configura el módulo Profinet del controlador KR C4, como se muestra en la Figura 74.

- Nombre de Interfaz. – KRC4
- Número de entradas :32
- Número de salidas: 32
- Puerto: Port1

Módulo	Rack	Slot	Direcció..	Direcció..	Tipo	Referencia	Firmware
▼ KRC4	0	0			KRC4-ProfiNet_3.3	0x300	V3.3
▼ Interface 1	0	0 X1			KRC4		
Port1	0	0 X1 P1			Port1		
32 digital in- and outputs_1	0	2	1...4	1...4	32 digital in- and o...	KUKA Device IO32	

Figura 74. Configuración de entradas y salidas módulo Profinet KR C4.

4.4.1. Aspectos importantes.

En este punto ya se puede programar tanto el PLC, como el robot KUKA, para ello se debe conocer la asignación, que se efectuó de entradas y salidas de los dispositivos.

Tipo	Dirección	Tipo	Dirección
Bool	%I1.0	Bool	%Q1.0
Bool	%I1.1	Bool	%Q1.1
Bool	%I1.2	Bool	%Q1.2
Bool	%I1.3	Bool	%Q1.3
Bool	%I1.4	Bool	%Q1.4
Bool	%I1.5	Bool	%Q1.5
Bool	%I1.6	Bool	%Q1.6
Bool	%I1.7	Bool	%Q1.7

Figura 75. Entradas y salidas Profinet

Las entradas van desde I1.0 hasta I4.7 y las salidas van desde Q1.0 hasta Q4.7, de forma consecutiva como se puede comprobar en la Figura 75.

Para relacionar la programación del PLC, con la del robot KUKA, se debe tener muy en claro que las salidas Profinet del PLC son las entradas Profinet del Robot KUKA y viceversa.

Así pues, si se activa la salida 17 del PLC (Q1.0), al mismo tiempo se activa la entrada 17 del robot KUKA, para denotar este evento, se implementa la siguiente línea de código.

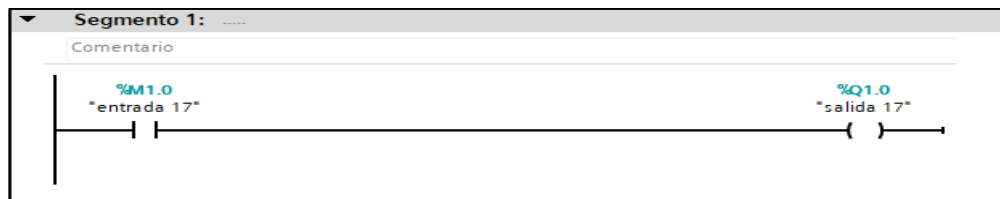


Figura 76. Programación de entradas y salidas en el PLC.

En consecuencia, la entrada 17 de Robot KUKA, se activa, pudiendo leerse con el siguiente código. `IF $IN[17]== TRUE THEN`, o de lo contrario `$OUT[17], TRUE,2` para activar la entrada 17 del PLC (I0.1).

4.5. Control de movimiento.

Para controlar el movimiento del robot KUKA KR3 R540, se establece comunicación entre el PLC y el Robot; mediante el estándar de comunicación Profinet, el cual está integrado de forma nativa en el PLC S7-1200 de Siemens y en el controlador KR C4 del robot KUKA. Cabe recalcar que, aunque el hardware necesario para la comunicación Profinet, está incluido en el controlador KR C4, su activación tiene un costo adicional.

4.5.1. Configuración Profinet del robot KUKA KR3 R540.

La configuración de estándar de comunicación Profinet, se realiza a través del software WorkVisual, que es propio de la marca KUKA.

Una vez instalado el programa WorkVisual, se verifica que el “*Protocolo de Internet (TCP/IPv4)*” este en “*Obtener una dirección IP automáticamente*” y se conecta un cable ethernet desde la “*salida RJ45*” del computador al “*puerto X69*” del controlador KR C4. Como denota la Figura 77.

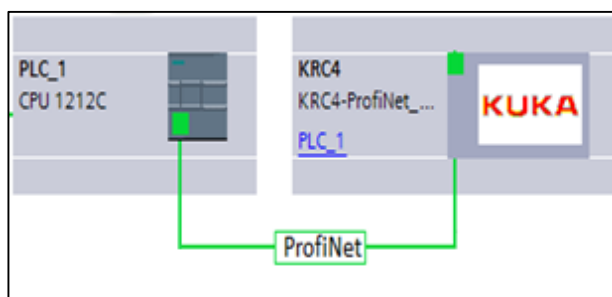


Figura 77. Comunicación Profinet.

Luego, se abre el programa WorkVisual y se realiza una serie de pasos para obtener la configuración inicial:

1. Presionar “*Buscar*”.
2. Clic en “*Refrescar*.”
3. Si la conexión del cable ethernet, está correctamente aparece “*Cell WINDOWS*”, si no es el caso verificar las conexiones.
4. Desplegar “*Cell WINDOWS*” y dar doble clic en el archivo seleccionado en la Figura 78, para Abrir.
5. Una vez abierto se procede a guardarlo, con otro nombre acorde al proyecto. “*Archivo*”- “*Guardar Como*”- “*Nuevo Nombre*”- “*Guardar*”.

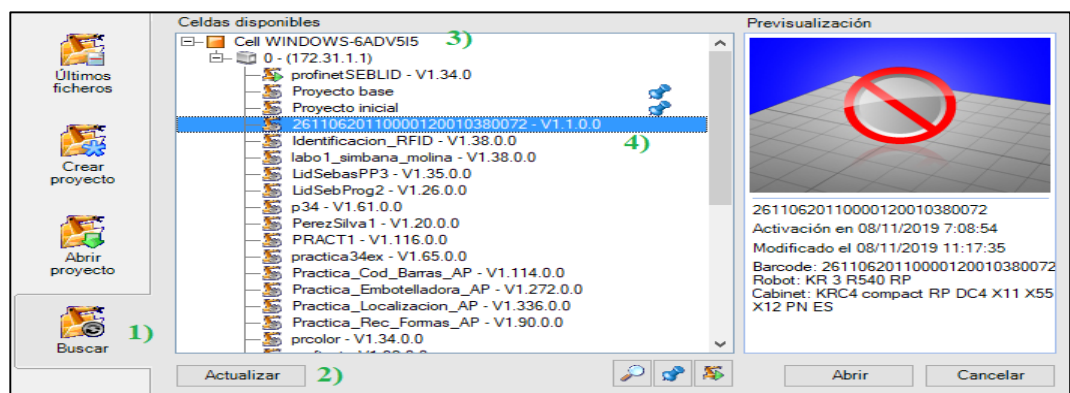


Figura 78. Configuración inicial WorkVisual.

6. Clic Derecho en “*Windows_AJBINBL*” y “*Poner como unidad de control activa*”

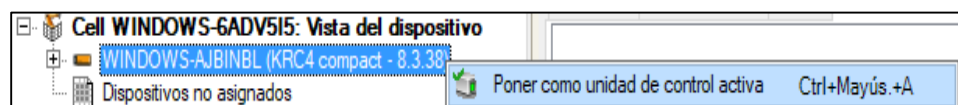


Figura 79. Unidad de control Profinet activa.

7. Doble clic en “*PROFINET*” y se realiza las configuraciones mostradas en la Figura 80.

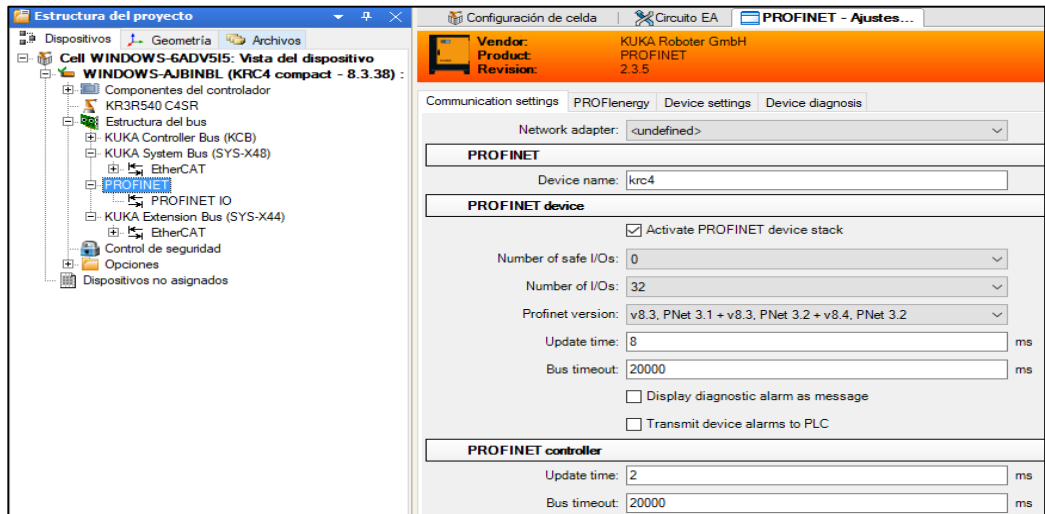


Figura 80. Configuración de módulo Profinet.

8. Se configura las entradas y salidas del módulo Profinet, a partir de número 17 hasta la 48, ya que desde la 0 hasta la 16 están reservadas para EtherCAT y se procede a carga el proyecto.

Name	Tipo	Descripción	E/S	E/S	Name	Tipo	Descripción
SIN[17]	BOOL		←	←	02:01:0001 Input	BOOL	
SIN[18]	BOOL		←	←	02:01:0002 Input	BOOL	
SIN[19]	BOOL	Entrada	←	←	02:01:0003 Input	BOOL	
SIN[20]	BOOL		←	←	02:01:0004 Input	BOOL	
SIN[21]	BOOL		←	←	02:01:0005 Input	BOOL	
SIN[22]	BOOL		←	←	02:01:0006 Input	BOOL	
SIN[23]	BOOL		←	←	02:01:0007 Input	BOOL	
SIN[24]	BOOL		←	←	02:01:0008 Input	BOOL	
SIN[25]	BOOL		←	←	02:01:0009 Input	BOOL	
SIN[19]	BOOL	Entrada	←	←	02:01:0001 Input	BOOL	
SIN[20]	BOOL		←	←	02:01:0001 Output	BOOL	
SIN[21]	BOOL		←	←	02:01:0002 Input	BOOL	
SIN[22]	BOOL		←	←	02:01:0002 Output	BOOL	
SIN[23]	BOOL		←	←	02:01:0003 Input	BOOL	
SIN[24]	BOOL		←	←	02:01:0003 Output	BOOL	
SIN[25]	BOOL		←	←	02:01:0004 Input	BOOL	

1 bit(s) seleccionados en 1 señal(es) 1 bit(s) seleccionados en 1 señal(es)

Figura 81. Configuración de entradas y salidas Profinet KUKA.

4.6. Corte láser

Los cabezales láser, están diseñados para interactuar con sistemas de control TTL, tecnología de la cual carece el PLC, por lo que se debe realizar algunos acoples.

La integración de láser con el PLC, se realizará a través de un microcontrolador Atmega 8, el cual posee tecnología TTL y permite controlar el encendido/apagado del láser, así como su potencia.

Para ello se desarrolla un pequeño código en lenguaje C++, que se ejecuta en el microcontrolador, en un bucle infinito mientras el sistema este fusionando.

Como primer paso, se inicializa las variables y se configura la salida PWM de microcontrolador. A fin de regular la potencia del láser.

```
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB=0b00000011;
    DDRD=0b11110000;
    TCCR1A=0b10000011;    //Configurar el PWM
        TCCR1B=0b00000001; //Sin preescalador
    TCNT1 =0b00000000;    //No se modifica
    OCR1A=1;              //Inicializar el TOP en cero
    int a=1;
    int aux=1020;
```

Figura 82. Inicialización de variables en C++.

1. Se procede a generar un bucle infinito con la instrucción While.

```
while (1)
{
}
```

Figura 83. Bucle infinito While.

2. Finalmente, se procede a realizar la activación del láser y la modulación de su potencia, mediante PWM.

```

if((PIND&(1<<PIND2)))
{
  if(a==1)
  {
    OCR1A=aux;
  }
  PORTD|=(1<<DDR6); //Activación de Laser.
  PORTD|=(1<<DDR7);

  if ((PIND&(1<<PIND0))&OCR1A=) { //Si el boton 1 esta activado
    OCR1A=OCR1A+5; //Incrementar la modulaciòn
    _delay_ms(50);
  }
  if ((PIND&(1<<PIND1))) { //Si el boton 2 esta activado
    OCR1A=OCR1A-5; //Decrementar la modulaciòn
    _delay_ms(50);
  }
  a=0;
  aux=OCR1A;
}
else
{
  OCR1A=0; // Apagar Bit CER0
  PORTD&=~(1<<DDR6);
  PORTD&=~(1<<DDR7);
  a=1;
}
}

```

Figura 84. Control del láser en C++.

3. Una vez realizado el código, se efectúa el diseño del circuito, que se realiza en el software “Proteus”, considerando las entradas/salidas programadas y la salida PWM.

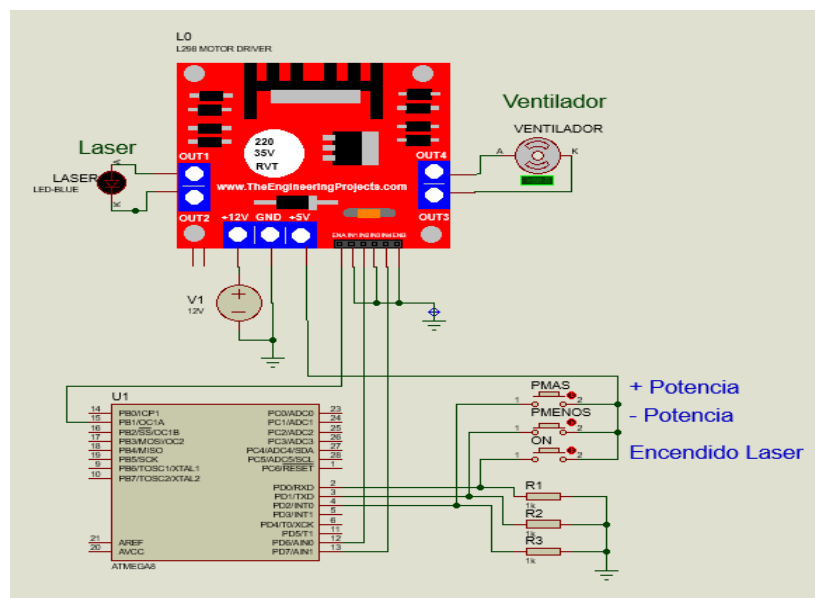


Figura 85. Diseño del circuito de control en Proteus.

4. Al obtener un diseño del circuito funcional y del agrado, se procede a generar las pistas PCB en la sección “ARES” del software “Proteus”.

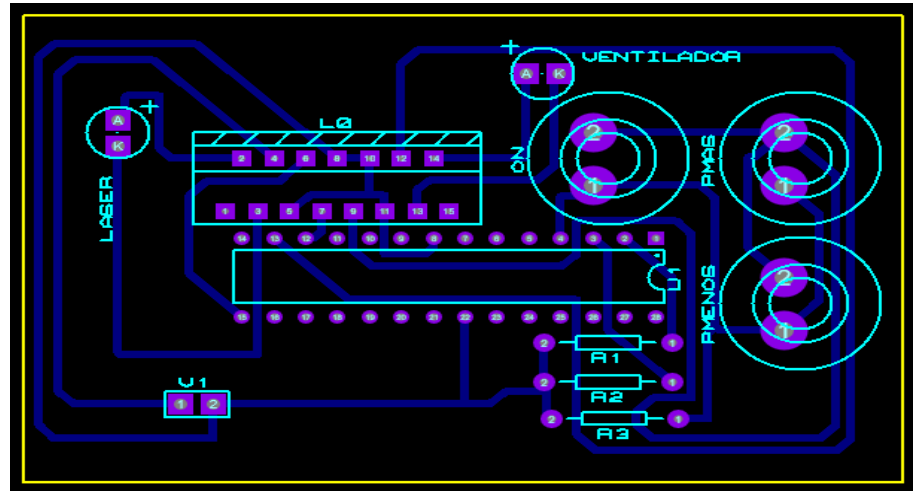


Figura 86. PCB del circuito de control.

5. Luego se realiza el montaje, como se ilustra en la Figura 87.

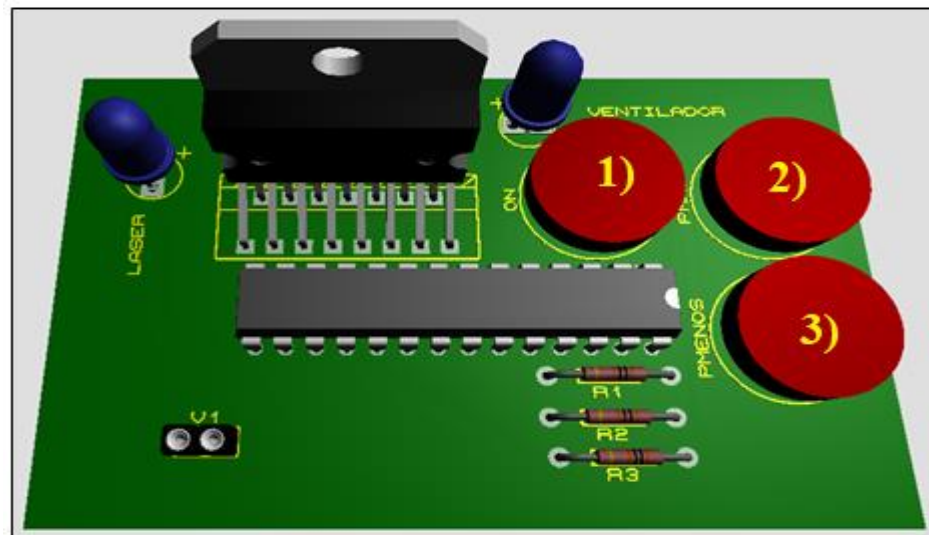


Figura 87. Placa del circuito de control.

6. Para finalizar, se conecta las entradas: ON (1), PMAS (2) y PMENOS (3) mostradas en la Figura 87, con las salidas el PLC.

4.7. Captura de imagen.

Para realizar la captura de imagen, se utiliza lenguaje de programación Python y su librería Open CV.

Se instala el programa Python Versión 3.7 y la librería OpenCV, con el comando “*pip install OpenCV*”. La captura de la imagen se realiza mediante el comando “*cap = cv2.VideoCapture(1)*” como se muestra en la Figura 88 y código de ejecución se muestra en al Anexo A1.



Figura 88. Imagen de la pieza cortada

4.8. Envío de Imagen.

El envío de imagen se realiza mediante un cliente FTP, que se encarga de almacenar información en el servidor FTP o de extraerla, dependiendo de las necesidades del usuario.

Para ello existen tres formas comunes:

1. Por el CMD interno de los sistemas operativos, con el comando ftp, introduciendo la dirección de servidor FTP, el usuario y contraseña. Los comandos del cmd se muestran en la Figura 89.

```

Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 3.16.88.255
Connected to 3.16.88.255.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (3.16.88.255:(none)): optimusprime
331 Password required
Password:
230 User logged in.
ftp> _

```

Figura 89. Cliente FTP en CMD

- Mediante Python, con ayuda de la librería `ftplib` y sus comandos FTP. A los que de igual forma que en el método anterior se le proporciona la información del servidor FTP. El código del cliente FTP se muestra en la Figura 90.

```

from ftplib import FTP
import cv2

servidor = "3.16.88.255"
user = "optimusprime"
passw = "XXXXXXXXXXXXXX"

try:
    conexion = FTP(servidor)
    conexion.login(user, passw)
    print("conexion")
except Exception:
    print("conexion error")

```

Figura 90. Cliente FTP en Python

- El tercer método, es el más sencillo, es un software que sincroniza una carpeta local con el servidor FTP. En este caso se utiliza el programa “*FTPbox*”, en el cual se debe llenar la información solicitada, tal como se muestra en la Figura 91. Y estará listo para usarse.

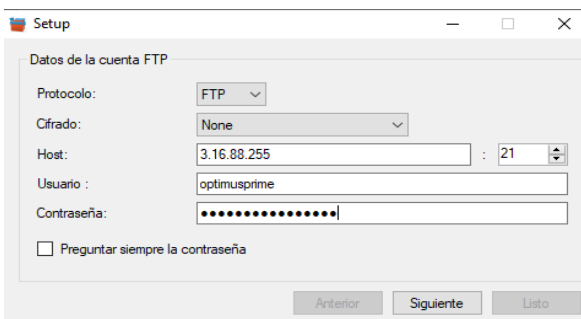


Figura 91. Cliente FTP por software.

Luego de establecida la comunicación con el servidor FTP, mediante alguno de los métodos anteriormente mencionados se procede al envío de imágenes de las piezas cortadas.

4.9. Servidor Virtual

Se realiza la contratación de un servidor virtual, que corresponde a la denominación IaaS (infraestructura como servicio) en la arquitectura Cloud Computing, mediante el proveedor Amazon Web Services, debido a su alto nivel de seguridad, ideal para procesos industriales, además de una amplia variedad de sistemas operativos y una velocidad de conexión a la red, superior a los 200mb/s,

En el servidor virtual, se almacena y ejecuta todos los procesos, necesarios para efectuar el control de calidad y desplegar la aplicación Cloud Computing.

Ya que en su interior cuenta con:

- Sistema Operativo Windows Server 2016
- Python 3.7.5, Git, Heroku, Google Cloud, OpenCV
- Web Server FTP

Evitando, así la necesidad de instalar todos los programas en un computador local, salvo con la excepción del proceso de captura de imagen, el cual necesita tener un intermediario para realizar la captura de imagen y enviarla por FTP al servidor virtual. Al cual se puede acceder mediante la función de “*Escritorio Remoto*”



Figura 92. Servidor virtual privado (VPS)

4.10. Servidor FTP.

El servidor FTP, se ejecuta internamente como un complemento adicional de un ordenador virtual o físico. Para su integración, se realiza diferentes pruebas en sistemas operativos, como Windows Server 2008, 2012, 2016, 2019, Rasbian y Ubuntu. Siendo Windows Server 2016, el sistema operativo que presentó, el mejor desempeño, al ejecutar la aplicación.

Para inicializar el Servidor FTP, se debe instalar el complemento Web Server de Windows. Siguiendo los pasos mencionados a continuación:

1. Se abre, “*Serve Manager*” y se selecciona “*Add Roles and Features*”.

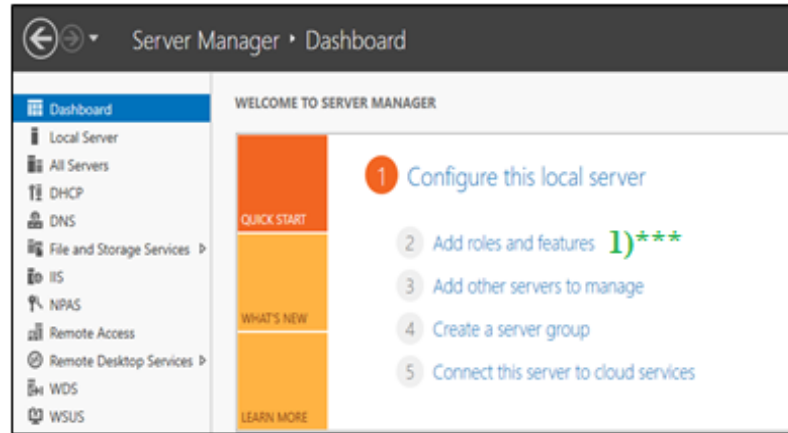


Figura 93. Server Manager.

2. Se marca el complemento “*Web Server (IIS)*” y se lo instala.



Figura 94. Web Server (IIS)

3. Luego de su instalación, se configura el nombre de usuario FTP y su contraseña, para ello, se ingresa “*Computer Management*” en el buscador de Windows, desplegar “*Local User and Groups*”, seleccionar “*New User*” y configurar el nombre y clave deseados.

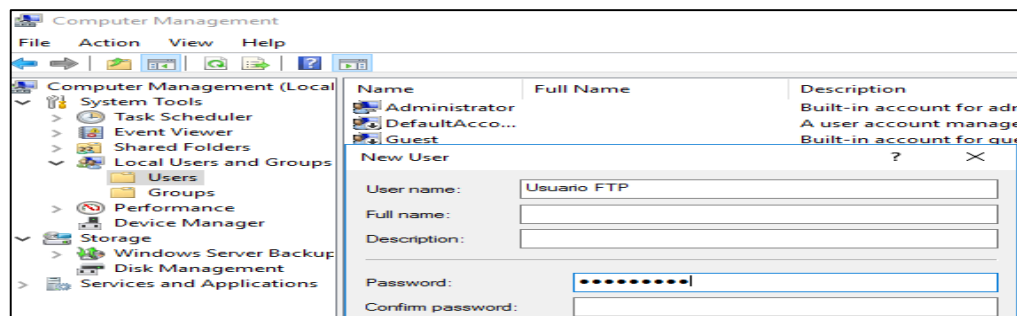


Figura 95. Configuración de usuario y clave FTP

4. Seleccionar “Tool”, opción “*Internet Information Services (IIS) Manager*”

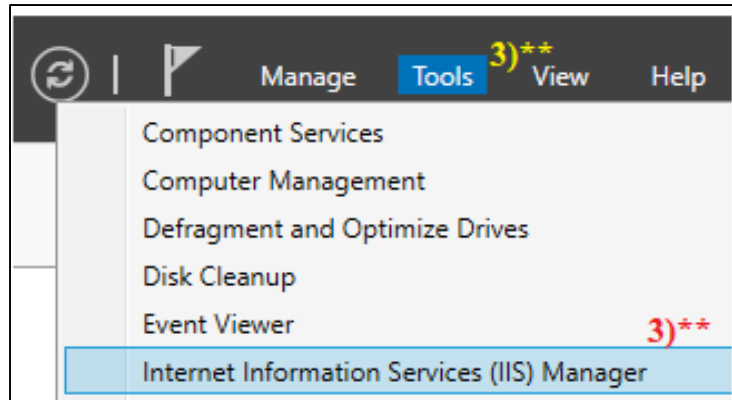


Figura 96. Internet Information Services (IIS) Manager

5. Clic derecho en “*EC2AMAZ*”, opción “*Add FTP Site*”

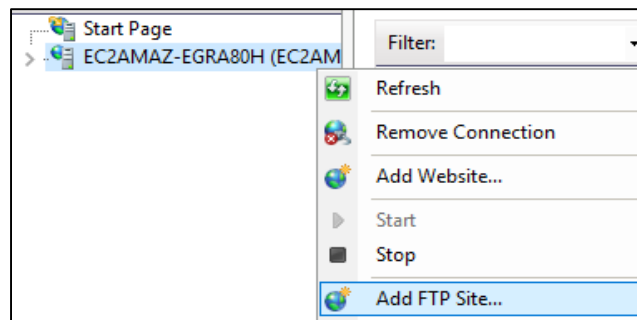


Figura 97. Adición sitio FTP.

6. Se otorgan permisos de salida en el firewall.

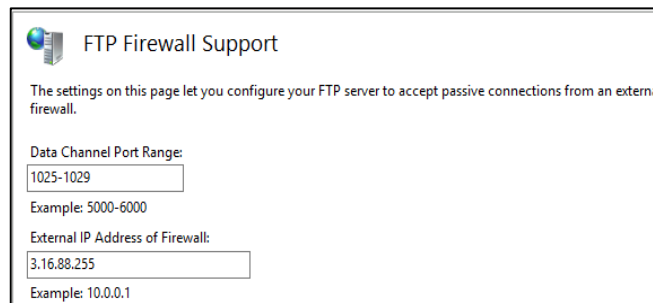


Figura 98. Permisos de salida.

7. Para Finalizar, se ingresa el nombre al servidor FTP y se lo vincula a una carpeta local, en la cual el cliente FTP, podrá cargar o extraer información.

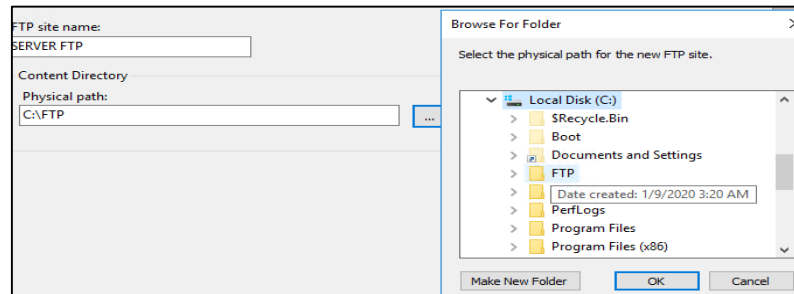


Figura 99. Carpeta FTP del servidor.

Al finalizar toda la serie de pasos mencionados anteriormente el servidor FTP, está listo para recibir las imágenes de la pieza cortada que envía el cliente.

4.11. Procesamiento de imagen.

4.11.1. Sincronización del servidor FTP y el procesamiento de imagen.

La sincronización, se realiza mediante un protocolo de consulta, en el que Python está constantemente preguntando al servidor FTP, si la imagen de la pieza cortada, se ha entregado: si la respuesta es negativa, Python sigue en espera y si la respuesta es afirmativa, se ejecuta todo el proceso de control de calidad, y vuelve a entrar en estado de espera hasta que se entregue una nueva imagen de la pieza cortada. El código necesario para efectuar el protocolo de consulta se muestra en la Figura 100.

```
import os.path
while (1):
    archivo = os.path.exists("CC:/FTP/Corte/pieza cortada.png")
    print(archivo)

    if (archivo== True):
        print("Ingresar aqui el proceso de control de calidad.")
    else:
        print("Segir en Espera")
```

Figura 100. Sincronización de FTP y procesamiento de imagen.

4.11.2. Procesamiento de Imagen 2.

Una vez obtenida la imagen, se procede a analizarla mediante los servicios de Cloud Computing de Google, específicamente con la API de Visión, para ello se agrega la librería de cliente de Google Cloud Visión a Python, con el comando “*pip install --upgrade google-cloud-vision*” y se genera las credenciales de acceso, siguiendo los pasos de (Cloud, 2019).

Luego se llama a las librerías de Cliente Cloud Visión y se envía las credenciales, así como la imagen a procesarse a Google Cloud Visión. Mediante los códigos descritos en el Anexo A2, obteniendo como resultado una serie de características y atributos que posee la imagen descritos en la Figura 101.

```

Descripcion de la imagen:
El objeto es un: Black, Con una Certeza del : 0.9571874141693115 %
El objeto es un: Triangle, Con una Certeza del : 0.894115686416626 %
El objeto es un: Triangle, Con una Certeza del : 0.7034509181976318 %
El objeto es un: Black-and-white, Con una Certeza del : 0.6834213137626648 %
El objeto es un: Cone, Con una Certeza del : 0.6325452327728271 %
El objeto es un: Space, Con una Certeza del : 0.5419734120368958 %
Es un Triangulo , Con una Certeza del: 0.894115686416626 %
Es un Triangulo , Con una Certeza del: 0.7034509181976318 %

```

Figura 101. Respuesta de Google Cloud Visión.

Una vez obtenida las características y atributos de la imagen, se retorna los resultados.

4.12. Modulo de detección de errores

Una vez obtienen los resultados del procesamiento de imágenes 2, se realiza el suavizado de la imagen mediante una función Gaussiana, que convierte una imagen normal a escala de grises, el comando utilizado es “*gauss = cv2.GaussianBlur(img, (5,5), 0)*”, luego se extrae los bordes

de la imagen con la función canny, para la cual el código es “*canny = cv2.Canny(gauss, 10, 300)*”, obteniendo como resultado la Figura 102.

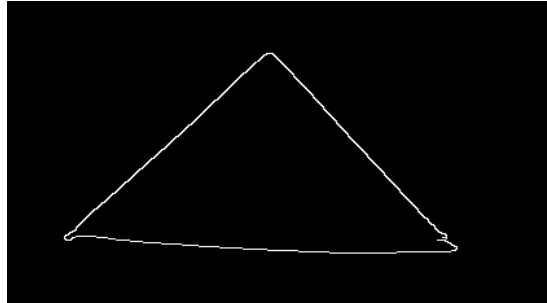


Figura 102. Bordes de la imagen cortada.

Al obtener los bordes originales de la imagen, se generan bordes perfectos del objeto cortado, con el comando “*approxs = cv2.approxPolyDP(cnt,epsilon,True)*” e identificar las diferencias entre la figura originales y la aproximación con “*diff_total = cv2.absdiff(contornos, aproxx)*” para así obtener los errores de corte que posee el objeto, como se observa en la Figura 103. El código de su obtención se despliega en el Anexo A3.

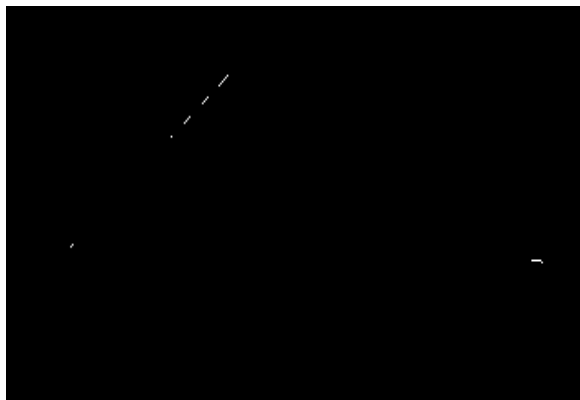


Figura 103. Defectos de calidad en la pieza cortada.

Para finalizar, se cuentan y remarcan los errores identificados, aprobando o rechazando, la pieza. Como se muestra en la Figura 104.

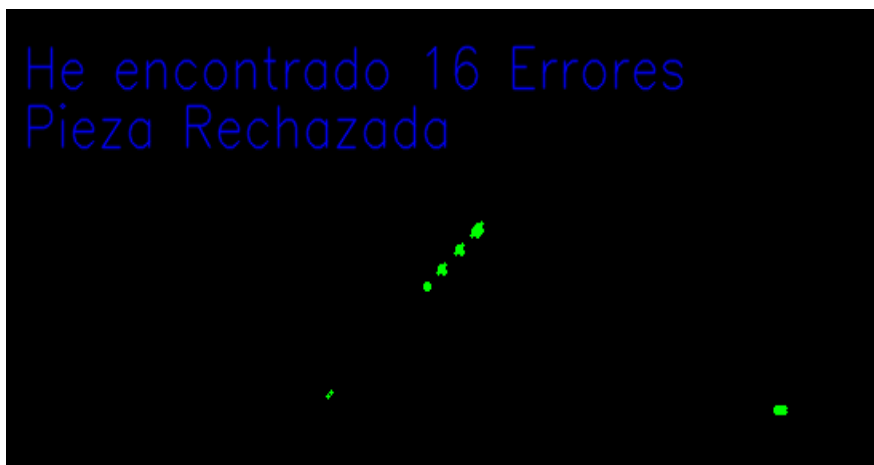


Figura 104. Aprobación o rechazo de la pieza cortada.

4.12.1. Sitio de Alojamiento web.

Se seleccionó a la plataforma Cloud Computing Heroku, que permite desarrollar aplicaciones propias y respaldadas bajo el repositorio Git. Permitiendo recupera información ante fallos y publicar aplicaciones pequeñas de forma gratuita, Además de tener compatibilidad con Python y lenguajes web PHP.

Para su utilización, como primer paso se debe crear cuentas de usuario, en Heroku y Git dirigiéndose a las siguientes ubicaciones, para Heroku en (CLI, 2019) y para Git en (Git, 2019). Una vez obtenida las cuentas, se descarga Heroku CLI de (Heroku, 2019), y se lo instala.

A fin de obtener, las librerías de Git, se ejecuta el código “*pip install gitpython*” en Python. Y se procede a subir la información referente al control de calidad a la plataforma Cloud Computing de Heroku. Los pasos a seguir se describen a continuación.

1. Inicio de sesión :

```
$ git config --global user.name Ron ██████████R
$ git config --global user.email scan█████████@hotmail.com
```

\$ heroku login

2. Se genera una ubicación web, con el comando “*git.remote*” y el nombre de la aplicación web que se desea.

\$ heroku git:remote -a tesis2020

3. Para generar un repositorio, se ejecuta “*git init*” y se adiciona todos los archivos necesarios para la implementación de la aplicación web, con el comando “*git add.*”.
4. Se carga el repositorio en la nube, con “*git push heroku master*”
5. Finalmente, se abre la aplicación ejecutando “*heroku open*”,

Una vez realizado todo el proceso anterior se obtiene, la aplicación Cloud Computing, accesible en cualquier parte de mundo, ubicada en la dirección <https://espe2020.herokuapp.com/>, como se muestra en la Figura 105. Todas las imágenes de conforman la interfaz Cloud se encurtan en el A8.

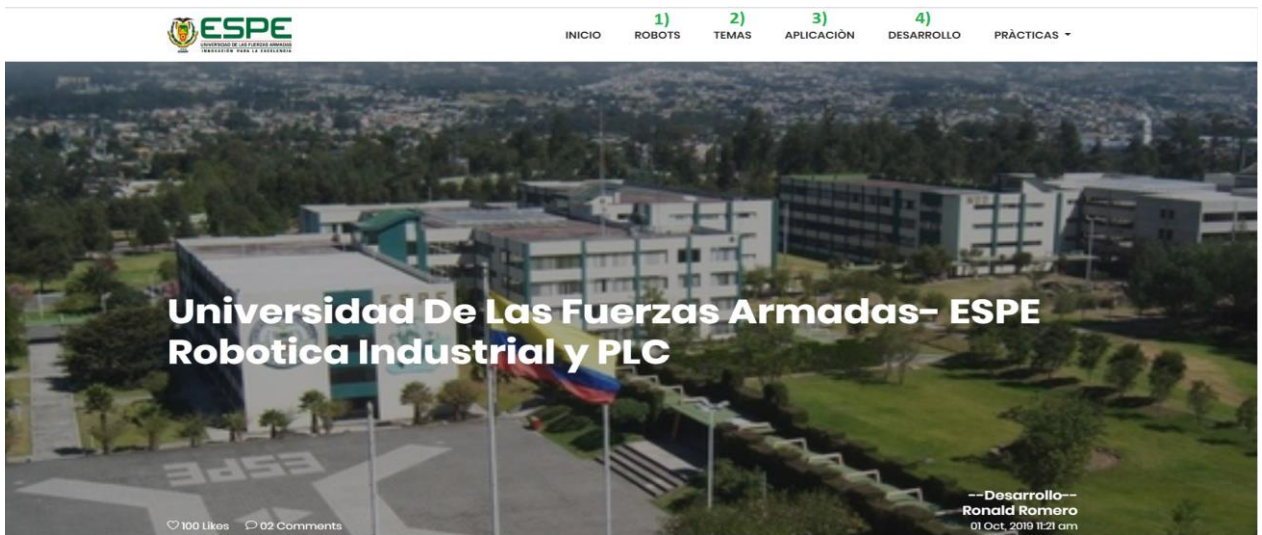


Figura 105. Aplicación Cloud Computing.

CAPÍTULO V

5. PRUEBAS Y RESULTADOS.

5.1. Pruebas y resultados de funcionamiento.

5.1.1. Control de movimiento.

Se controla el movimiento del robot KUKA KR3 R540, mediante comunicación Profinet con el PLC, siendo capaces de activar/desactivar entradas, salidas y ejecutar programas en KRL.

Para comprobar su correcto funcionamiento se ejecuta un pequeño programa el cual permite mover al robot mediante la aplicación web del PLC.

```
EXT BAS(BAS_COMMAND :IN,REAL :IN)
DECL AXIS HOME
IF $IN[17]== TRUE THEN
LIN {X 0,Y 0, Z 50, A 180, B -1.35, C -180}
LIN {X 50,Y 0, Z 50, A 180, B -1.35, C -180}
LIN {X 50,Y 50, Z 50, A 180, B -1.35, C -180}
PTP HOME
```

Como se muestra en la Figura 106, se pudo lograr efectuar el movimiento del robot a través de la interfaz web del PLC y un pequeño programa de prueba en KRL.



Figura 106. Control de movimiento del robot KUKA KR3 R540.

5.1.2. Interfaz web del PLC.

La estructura de la interfaz web mostrada en la Figura 107, está basada en un sistema de navegación amigable con el usuario, siendo capaces de desplegar la información deseada, mediante la selección en la barra de menú. La interfaz web además cuenta con información referente al laboratorio de robótica industrial y PLC de la Universidad de las Fuerzas Armadas - ESPE, como categorías de Robot, Temas de investigación y Aplicaciones.

Pero la función principal de la aplicación web, es ejecutar los procesos de corte y control de calidad del sistema Cloud Computing.

Así se obtiene una interfaz web, con 6 ventanas de visualización integradas para la ejecución del presente proyecto como se muestra en la Figura 107 y en el Anexo A8. La cual se puso a prueba en: computadores, tablet y celulares, funcionando correctamente y adaptándose al tamaño de la pantalla de cada dispositivo.

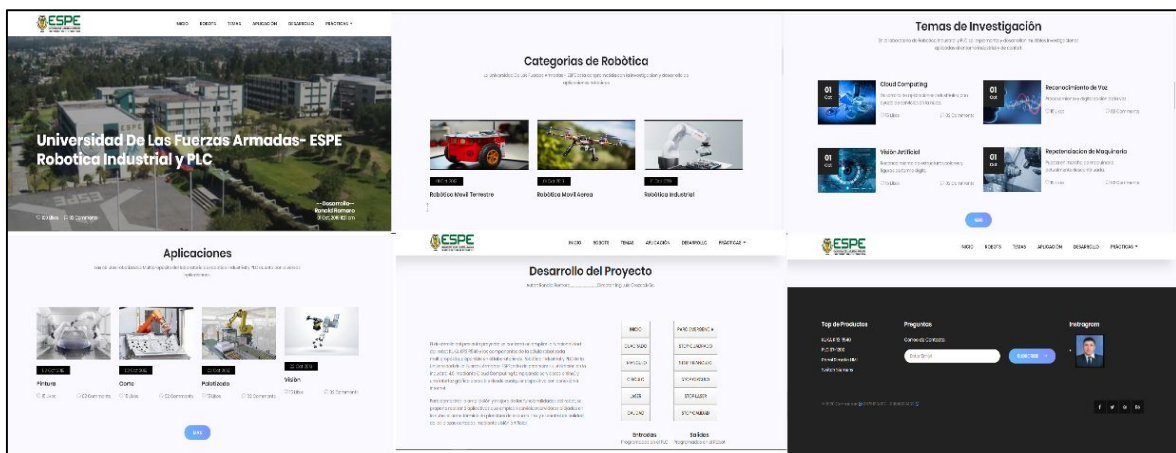


Figura 107. Ventanas de la aplicación web.

5.1.3. Corte térmico.

En el presente apartado, se realiza el ajuste de velocidad y potencia, a fin de encontrar la velocidad y potencia óptima para cortar fomix.

- Ajuste 1. Velocidad 0.07m/s - Potencia del 50%: Marca levemente la superficie, casi de forma imperceptible, excepto en la parte final, donde genera una quemadura visible, pero sin llegar a traspasar.



Figura 108. Ajuste de velocidad y potencia 1.

- Ajuste 2. Velocidad 0.05mm/s - Potencia del 75%: Marca notoriamente la superficie, pero sin llegar a traspasarla, excepto en la parte final, donde si se realiza un leve corte.

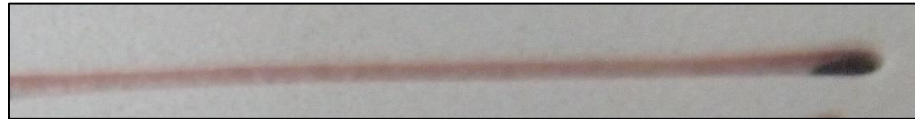


Figura 109. Ajuste de velocidad y potencia 2.

- Ajuste 3. Velocidad 0.03m/s - Potencia del 100%: Realiza una marcación profunda, dejando una pequeña membrana.



Figura 110. Ajuste de velocidad y potencia 3.

- Ajuste 4. Velocidad 0.01m/s - Potencia del 100%: Realiza el corte de mala calidad, dejando bordes irregulares y en algunos tramos una fina membrana.



Figura 111. Ajuste de velocidad y potencia 4.

- Ajuste 5. Velocidad 0.007m/s - Potencia del 100%: Realiza un corte de buena calidad, con bordes regulares y uniformes.



Figura 112. Ajuste de velocidad y potencia 5.

- Ajuste 6. Velocidad 0.002m/s - Potencia del 100%: Realiza un corte profundo, pero al moverse tan lentamente, genera una gran cantidad de calor que deforma la pieza cortada, dándole bordes irregulares y de mala calidad.



Figura 113. Ajuste de velocidad y potencia 6.

De acuerdo a los ajustes realizados, la velocidad óptima de operación es de 0.007 m/s y una potencia del 100%, ya que, existe la posibilidad de reducir la potencia, pero también se tendrá que reducir la velocidad, con el fin de que se pueda efectuar el corte, aumentando significativamente el tiempo de ejecución.



Figura 114. Piezas cortadas con láser.

5.1.4. Sistema de control de calidad.

Para comprobar el correcto funcionamiento del sistema de control de calidad, se realiza una toma de datos de 50 muestras, en las que se compara los resultados obtenidos mediante el sistema de control de calidad y criterios humanos de aceptación o rechazo.

Los criterios de aceptación humana son: bordes regulares y sin quemaduras excesivas que distorsionen la forma de la pieza.

Los criterios de aceptación del control de calidad por visión artificial son: bordes regulares comparados con una imagen de una pieza perfecta, según los criterios humanos anteriormente mencionados.

Tabla 5.

Resultados del sistema de control de calidad.

N ^o	Criterio de aceptación humana	Respuesta del sistema de control de calidad.	Acierto del sistema de control de calidad.	N ^o	Criterio de aceptación humana	Respuesta del sistema de control de calidad.	Acierto del sistema de control de calidad.
1	Aceptar	Aceptar	V	26	Rechazar	Rechazar	V
2	Rechazar	Rechazar	V	27	Aceptar	Aceptar	V
3	Aceptar	Aceptar	V	28	Rechazar	Rechazar	V
4	Rechazar	Rechazar	V	29	Aceptar	Aceptar	V
5	Aceptar	Aceptar	V	30	Rechazar	Rechazar	V
6	Rechazar	Rechazar	V	31	Aceptar	Aceptar	V
7	Aceptar	Rechazar	F	32	Rechazar	Rechazar	V

CONTINÚA

8	Rechazar	Rechazar	V	33	Aceptar	Aceptar	V
9	Aceptar	Aceptar	V	34	Rechazar	Rechazar	V
10	Rechazar	Rechazar	V	35	Aceptar	Aceptar	V
11	Aceptar	Aceptar	V	36	Rechazar	Rechazar	V
12	Rechazar	Rechazar	V	37	Aceptar	Aceptar	V
13	Aceptar	Aceptar	V	38	Rechazar	Rechazar	V
14	Rechazar	Rechazar	V	39	Aceptar	Aceptar	V
15	Aceptar	Aceptar	V	40	Rechazar	Rechazar	V
16	Rechazar	Rechazar	V	41	Aceptar	Aceptar	V
17	Aceptar	Aceptar	V	42	Rechazar	Rechazar	V
18	Rechazar	Rechazar	V	43	Aceptar	Aceptar	V
19	Aceptar	Aceptar	V	44	Rechazar	Rechazar	V
20	Rechazar	Rechazar	V	45	Aceptar	Aceptar	V
21	Aceptar	Aceptar	V	46	Rechazar	Rechazar	V
22	Rechazar	Rechazar	V	47	Aceptar	Rechazar	F
23	Aceptar	Aceptar	V	48	Rechazar	Rechazar	V
24	Rechazar	Rechazar	V	49	Aceptar	Aceptar	V
25	Aceptar	Aceptar	V	50	Rechazar	Rechazar	V

5.1.5. Interfaz web Cloud.

Se implementa un sistema Cloud Computing de Arquitectura total, el cual consta de Software como servicio (Google Cloud Visión), Plataforma Como Servicio (Heroku y Git), Infraestructura como Servicio (Amazon Web Services), además de incluir una aplicación propia que efectúa el control de calidad de piezas cortadas por láser, dando como resultado su aceptación o rechazo y registrando cada una, con hora y fecha. Al finalizar todo el proceso de control de calidad, se publican los resultados en la plataforma Heroku, a los cuales se puede acceder en la dirección: <http://espe2020.herokuapp.com/>

La interfaz web Cloud de la Figura 115, se probó accediendo a ella desde diferentes lugares y redes de internet, a fin de comprobar si se puede visualizar a nivel mundial se probó en un servidor virtual alojado en Estados Unidos y otro alojado en Ecuador.



Figura 115. Resultados de la aplicacion Cloud Computing

Adicionalmente en el servidor virtual de Amazon Web Services, no solo se ejecuta la aplicación propia, también existen softwares que puedes ser usados por el usuario como:

- Python – Arduino - Proteus
- Kuka SIM PRO 1.1 – WorkVisual
- Sistema de envío y almacenamiento de información a través de FTP.
- Heroku CLI - Git - WinRAR



Figura 116. Softwares en el servidor virtual.

5.1.6. Tiempo de proceso Cloud Computing.

Se realiza una toma de datos del tiempo que tarda en ejecutarse la aplicación Cloud Computing, la metodología usada para la toma de datos, se basa en la observación directa del proceso, en la que se envían 10 imágenes de 105 Kb y se toma el tiempo mediante un cronometro, desde que se ejecuta el proceso de envío de imagen hasta la carga de la información en el sitio de alojamiento y su apertura en el buscador Google Chrome. Este proceso se realiza en tres entornos diferentes:

5.1.6.1. Residencial.

Tabla 6.

Tiempo de proceso Cloud Computing residencial.

Tamaño de Imagen	Tiempo de Proceso (s)	Xi-X	(Xi-X)²
imagen 1	42,41	-0,105	0,011025
imagen 2	42,43	-0,085	0,007225
imagen 3	40,84	-1,675	2,805625
imagen 4	42,73	0,215	0,046225
imagen 5	45,78	3,265	10,660225
imagen 6	42,2	-0,315	0,099225
imagen 7	41,64	-0,875	0,765625
imagen 8	42,66	0,145	0,021025
imagen 9	42,25	-0,265	0,070225
imagen 10	42,21	-0,305	0,093025
Número de datos :	10	Suma:	14,57945
Valor Medio :	42,515	Desviación	1,2727682

5.1.6.2. Institucional.

Tabla 7.

Tiempo de proceso Cloud Computing institucional.

Tamaño de Imagen	Tiempo de Proceso	Xi-X	(Xi-X)²
imagen 1	34,65	-2,095	4,389025
imagen 2	37,76	1,015	1,030225
imagen 3	41,23	4,485	20,115225
imagen 4	36,63	-0,115	0,013225
imagen 5	32,59	-4,155	17,264025
imagen 6	33,65	-3,095	9,579025
imagen 7	37,19	0,445	0,198025
imagen 8	35,41	-1,335	1,782225

CONTINÚA

imagen 9	39,78	3,035	9,211225
imagen 10	38,56	1,815	3,294225
Número de datos :	10	Suma:	66,87645
Valor Medio :	36,745	Desviación	2,72593409

5.1.6.3.Servidor virtual.

Tabla 8.

Tiempo de proceso Cloud Computing en el servidor virtual.

Tamaño de Imagen	Tiempo de Proceso (s)	Xi-X	(Xi-X) ²
imagen 1	23,21	0,124	0,015376
imagen 2	24,21	1,124	1,263376
imagen 3	22,37	-0,716	0,512656
imagen 4	25,51	2,424	5,875776
imagen 5	22,59	-0,496	0,246016
imagen 6	23,06	-0,026	0,000676
imagen 7	21,93	-1,156	1,336336
imagen 8	23,43	0,344	0,118336
imagen 9	21,92	-1,166	1,359556
imagen 10	22,63	-0,456	0,207936
Número de datos :	10	Suma:	10,93604
Valor Medio :	23,086	Desviación	1,1023228

5.1.1. Velocidad de red de proceso Cloud Computing.

Al igual que la toma de datos realizada, para los tiempos de ejecución, se realiza una toma de datos de la velocidad de internet en cada uno de los entornos, la metodología utilizada para este proceso se basa en la observación directa, mediante la medición de la velocidad de internet con la ayuda del sitio online <https://www.test-velocidad.com/>

5.1.1.1. Residencial.

Tabla 9.

Velocidad de proceso Cloud Computing residencial.

Tamaño de Imagen	Velocidad de Proceso (Mb/s)	Xi-X	(Xi-X) ²
imagen 1	3,81	0,179	0,032041
imagen 2	3,69	0,059	0,003481
imagen 3	3,56	-0,071	0,005041

CONTINÚA

imagen 4	3,55	-0,081	0,006561
imagen 5	3,55	-0,081	0,006561
imagen 6	3,63	-0,001	1E-06
imagen 7	3,51	-0,121	0,014641
imagen 8	3,63	-0,001	1E-06
imagen 9	3,53	-0,101	0,010201
imagen 10	3,85	0,219	0,047961
Número de datos :	10	Suma:	0,12649
Valor Medio :	3,631	Desviación	0,11855144

5.1.1.2. Institucional.

Tabla 10.

Velocidad de proceso Cloud Computing institucional.

Tamaño de Imagen	Velocidad de Proceso (Mb/s)	Xi-X	(Xi-X)²
imagen 1	100,15	-4,615	21,298225
imagen 2	106,34	1,575	2,480625
imagen 3	103,65	-1,115	1,243225
imagen 4	98,78	-5,985	35,820225
imagen 5	110,56	5,795	33,582025
imagen 6	105,23	0,465	0,216225
imagen 7	108,69	3,925	15,405625
imagen 8	104,29	-0,475	0,225625
imagen 9	106,69	1,925	3,705625
imagen 10	103,27	-1,495	2,235025
Número de datos :	10	Suma:	116,21245
Valor Medio :	104,765	Desviación	3,59339595

5.1.1.3. Servidor virtual.

Tabla 11.

Velocidad de proceso Cloud Computing en el servidor virtual.

Tamaño de Imagen	Velocidad de Proceso(Mb/S)	Xi-X	(Xi-X)²
imagen 1	223	-35,46	1257,4116
imagen 2	261,5	3,04	9,2416
imagen 3	260,8	2,34	5,4756
imagen 4	263,9	5,44	29,5936
imagen 5	264,9	6,44	41,4736
imagen 6	271	12,54	157,2516
imagen 7	267,5	9,04	81,7216
imagen 8	257,5	-0,96	0,9216
imagen 9	260,3	1,84	3,3856
imagen 10	254,2	-4,26	18,1476
Número de datos :	10	Suma:	1604,624
Valor Medio :	258,46	Desviación	13,3525861

5.2. Resultados de aprendizaje de estudiantes

Los resultados obtenidos al desarrollar la práctica 1, fueron satisfactorios, ya que se logró implementar una interfaz gráfica simple, la cual consta de un botón de activación, permitiendo la comunicación con el PLC, que a su vez envía la orden de ejecución del programa en KRL, para efectuar el movimiento del robot.



Figura 117. Interfaz gráfica de la práctica 1.

Por otra parte, en la práctica 2, se utilizó todos los programas desarrollados en la práctica 1, con la debida adaptación de encendido del cabezal láser y la ejecución del programa de reconocimiento de objetos. Obteniendo al finalizar la pieza cortada en fomix, con sus respectivas características.

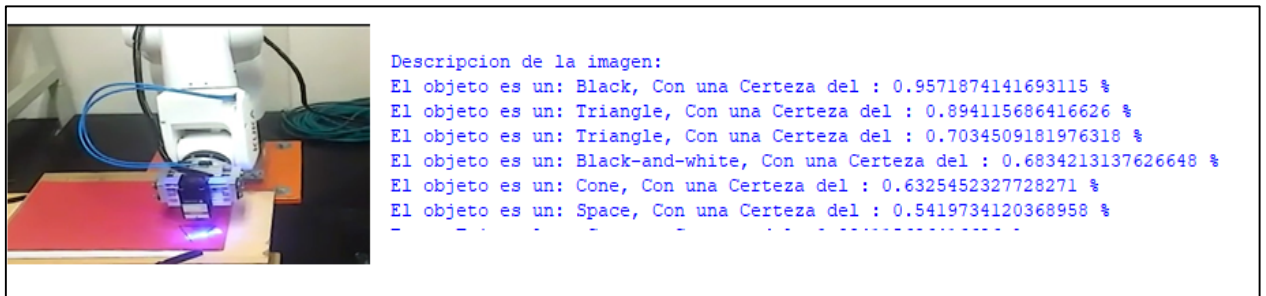


Figura 118. Resultados obtenidos en la práctica 2.

5.3. Análisis de resultados.

El funcionamiento, del control de movimiento, vinculado con la aplicación de ejecución y el corte láser, es satisfactorio, logrando cortar láminas de fomix con la figura deseada a la perfección, y con una gran calidad. Teniendo bordes regulares, precisos y sin quemaduras excesivas.

En cuanto, al sistema de control de calidad, logra cumplir con su función, siendo capaz de identificar errores en los bordes de las piezas cortadas, siempre y cuando superen el milímetro de profundidad de daño. Para conocer su porcentaje de acierto, se recurre a la información proporcionada por la Tabla 4.

$$\frac{N^{\circ} \text{ de aciertos} * 100\%}{N^{\circ} \text{ de muestras}} = \text{Porcentaje de acierto}$$

$$\frac{48 * 100\%}{50} = 96 \%$$

Ecuación 1. Porcentaje de acierto del sistema de control de calidad.

Se logra identificar que el proceso de control de calidad acierta en un 96%, con lo que se puede establecer que es un sistema eficiente, para la detección de fallos en piezas de fomix cortadas por láser, a pesar de no contar con un ambiente controlado de iluminación.

Por su parte, el tiempo de ejecución de la Aplicación Cloud Computing mejora notablemente al ser ejecutada, en el servidor virtual, en comparación con la ejecución local. Esto se debe a las grandes velocidades de internet proporcionadas por Amazon Web Services alcanzando los 258,46 Mb/s de acuerdo a la información de la Tabla 10. Y a la cercanía del servidor virtual, con los servidores de Google Cloud Platform, ya que los dos se encuentra en Estados Unidos.

Para identificar de forma matemática, la mejora obtenida al realizar el proceso mediante Cloud Computing, se comparará el tiempo de ejecución local con la del servidor virtual descritos en las Tablas 5, 6 y 7 respectivamente.

$$TP \text{ Residencial} - TP \text{ en el servidor} = \text{Diferencia de Tiempo}$$

$$42,515(s) - 23,086 (s) = 19.429(s)$$

$$\frac{\text{Diferencia de Tiempo} \times 100\%}{TP \text{ Residencial}} = \text{Porcentaje de Reducción\%}$$

$$\frac{19.429 \times 100\%}{42.515} = 45.69\%$$

Ecuación 2. Porcentaje de reducción de tiempo de proceso residencial.

$$TP \text{ Institucional} - TP \text{ en el servidor} = \text{Diferencia de Tiempo}$$

$$36,745 (s) - 23,086 (s) = 13.659 (s)$$

$$\frac{\text{Diferencia de Tiempo} \times 100\%}{TP \text{ Institucional}} = \text{Porcentaje de Reducción\%}$$

$$\frac{13.659 \times 100\%}{36,745} = 37.17\%$$

Ecuación 3. Porcentaje de reducción de tiempo de proceso institucional.

En el caso de la ejecución de la aplicación de forma local en una residencia con velocidad de internet de 3,631 Mb/s de acuerdo a la Tabla 8, se redujo el tiempo de ejecución de 42.515 a 23.086 segundos, representando un 45.69 % de mejora del sistema al realizarlo mediante servicios Cloud Computing.

Por su parte la ejecución de la aplicación de forma local en una institución, con velocidad de internet de 104,76 Mb/s de acuerdo a la Tabla 9, pasó de tardar 37,745 a 23.086 segundos, lo que representa un 37.17 % de mejora del sistema al ejecutarlo en servicios de Cloud Computing.

***Nota:** Se seleccionó como residencia una vivienda localizada al sur de la ciudad de Quito, con una velocidad de internet promedio, y como Institución al laboratorio de Robótica y PLC de la Universidad de las Fuerzas Armadas-ESPE.*

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Se desarrolló e implementó un sistema Cloud Computing de Arquitectura total multiplataforma, en el que se integran servicios de Google Cloud Platform, Heroku Cloud Application Platform y Amazon Web Services, denotando la flexibilidad y potencialidad de los proveedores Cloud Computing para trabajar en aplicaciones colaborativas. Mejorando la funcionalidad de la célula robotizada multipropósito al convertir su sistema de control local, a un sistema de control Cloud Computing.
- Se implementó el envío de información por medio de protocolos FTP, en la que existe un servidor virtual en la nube y un cliente local, al que se accede remotamente mediante credenciales, de dirección, nombre y contraseña, para la recepción y almacenamiento de la imagen a ser procesada. De esta forma se aísla al servidor de accesos externos, otorgando un mayor nivel de seguridad al sistema, evitando ataque ya que es invisible al exterior y a los usuarios.
- A fin de efectuar cortes precisos, con bordes regulares y de gran calidad en láminas de fomix. Los parámetros de calibración obtenidos fueron: una potencia del 100% (5.5 Watts) a una velocidad de 0.007 m/s.
- Para la obtención del sistema de control de calidad por visión artificial, se vincularon la API de Visión de Google Cloud y la librería de código abierto OpenCV, mediante Python, lo que permitió identificar: las características de la pieza cortada, sus atributos

y las irregularidades de corte, que existen en sus bordes, con un porcentaje de acierto del 96%. Todo este proceso se ejecuta en un servidor virtual, ubicado en la nube de Amazon Web Services.

- La interfaz gráfica para la ejecución del sistema Cloud Computing, se desarrolló en lenguaje HTML y se alojó en el Web Server propio del PLC s7-1200, lo que permitió controlar el estado de sus variables internas programadas, mediante comandos AWP de interacción propios de PLC. Al controlar el PLC, se pudo realizar el control de los demás elementos conectados.
- Se logró identificar una mejora sustancial en el tiempo de ejecución del proceso, al comparar los tiempos de ejecución en un servidor local y en un servidor virtual ubicado en la nube, ya que los servidores locales poseen velocidades de conexión reducidas, que van desde los 5 Mb/s hasta 100 Mb/s y por su parte los proveedores de VPN (servidores virtuales) proporcionan velocidades de conexión que superan los 200 Mb/s, lo que produce un envío de información y procesamiento en la nube de mayor velocidad, comprobando así que la adopción de PaaS (Plataforma como servicio), como los proporcionados por Amazon AWS, son la mejor opción para el desarrollo de aplicaciones industriales escalables, ya que reducen el tiempo de ejecución de procesos, una inversión inicial mínima y la facturación es en base a la demanda.
- Mediante entrevistas verbales realizadas a los estudiantes, se logró identificar grandes resultados, ya que los estudiantes, implementaron con éxito cada una de las prácticas, posibilitando así su continuación en temas de investigación o en trabajos de titulación.

- La implementación de un sistema Cloud Computing, a menudo tiene costos de prueba reducidos, ya que los proveedores otorgan saldos iniciales, como es el caso de Google Cloud, el cual proporciona 300 dólares de monto inicial para realizar pruebas de funcionamiento, pero una vez consumido el saldo inicial o pasado un año, cada petición realizada tiene un costo de 1.66 centavos. Por su parte Amazon Web Services proporciona un tiempo de uso de 720 horas mensuales durante un año, el cual si es sobrepasado tiene un costo 7,35 dólares al mes. Por esta razón, si se desea implementar una aplicación o sistema en la nube, a largo plazo se debe considerar los costos que conlleva contratar los servicios de Cloud Computing.

6.2. Recomendaciones

- Tener en cuentas las políticas de seguridad y privacidad de las instituciones donde se desarrolla un proyecto, ya que pueden delimitar las opciones o posibilidades de implementación de un sistema.
- Inspeccionar los dispositivos de hardware, e identificar si cuentan con todos los elementos de conexión y comunicación, requeridos para el desarrollo del proyecto.
- Investigar más de una posible solución a un problema, ya que algunas opciones pueden estar restringidas por las políticas de la institución o empresa, donde se desarrolla el proyecto.
- El sistema implementado, puede ser utilizado para diferentes materiales, ya sea: madera, cartulina, cartón, acrílico entre otros.

7. BIBLIOGRAFÍA

- ADF. (2019). *Profinet / profibus - converter*. Obtenido de http://adfweb.com/home/products/PROFINET_PROFIBUS.asp?language=ESP
- AG.Kuka. (2019). *El controlador del robot que viene de la nube*. Obtenido de <https://www.kuka.com/fr-fr/presse/nouvelles/2017/05/le-petit-robot-kr-3-agilus>
- Alian, N., Fernández, J., Martínez, A., & Ortiz, J. (2007). *Un Laboratorio de Ingeniería de Control Basado en Internet*. Recuperado el 2019, de <https://scielo.conicyt.cl/pdf/infotec/v18n6/art04.pdf>
- Ariza, C., & Amaya, D. (2011). *Laboratorio remoto para la enseñanza de la programación de un robot industrial*. Recuperado el 01 de 08 de 2019, de <http://web.usbmed.edu.co/usbmed/fing/v2n1/v2n1a7.pdf>
- Azure, M. (2019). *Software como servicio*. Obtenido de <https://azure.microsoft.com/es-es/overview/what-is-saas/>
- Bravo, G. (2019). *¿Qué es CSS?* Obtenido de <https://www.hostinger.es/>
- Bravo1, G. (2019). *¿Qué es GitHub y para qué se utiliza?* Obtenido de <https://www.hostinger.es/tutoriales/que-es-github/>
- Brihuega, D. A. (2017). *Criptografía sin secretos con Python*. Obtenido de <https://books.google.com.ec/books?id=eo6fDwAAQBAJ&pg=PA87&lpg=PA87&dq=Python+es+un+lenguaje+de+programaci%C3%B3n+poderoso+y+f%C3%A1cil+de+aprender.+Cuenta+con+estructuras+de+datos+eficientes&source=bl&ots=0Iox-E7EO3&sig=ACfU3U3mVJ1FsxZvNIR7p4kqYnAJgDDONg&>

Campoverde, A. (2015). *Cloud computing con herramientas open-source para Internet de las cosas*. Obtenido de <https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/view/712>

Cano, R. (2013). *Lenguaje Ladder*. Obtenido de <https://es.slideshare.net/rukita04/proy-1>

Castillo, R. (2013). *Diseño de un laboratorio remoto de robots y procesos de manufactura industriales*. Recuperado el 01 de 08 de 2019, de http://www.epistemus.uson.mx/revistas/articulos/15-08_DISENO%20DE%20LABORATORIO.pdf

CLI, H. (2019). *Welcome to Heroku*. Obtenido de <https://dashboard.heroku.com/apps>

Cloud, G. (2019). *Bibliotecas cliente de Vision*. Obtenido de <https://cloud.google.com/vision/docs/libraries?hl=es-419#client-libraries-usage-python>

Concepción, U. D. (2019). *Computación en Nube*. Obtenido de <http://www2.udec.cl/~aldconcha/cyp/index.html>

Definición, C. (2019). *PHP*. Obtenido de <https://conceptodefinicion.de/php/>

docs, M. W. (2019). *Introducción a Django*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducci%C3%B3n>

DotCloud. (2019). *Beneficios de usar Servicios Cloud*. Obtenido de <http://www.dotcloud.com.ar/>

Empleo, M. D. (2012). *Reglamento de Seguridad y Salud*. Obtenido de <http://www.trabajo.gob.ec/wp-content/uploads/downloads/2012/12/Reglamento-de-Seguridad-y-Salud-para-la-Construcción-y-Obras-Públicas.pdf>

Estrada, R. (2017). *Puerto Serial – protocolo y su teoría*. Obtenido de <https://hetprostore.com/TUTORIALES/puerto-serial/>

Etitudela. (2019). *Visión artificial*. Obtenido de <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

Eurobots. (2019). *Kuka kr 3 r540 agilus*. Obtenido de <https://www.eurobots.es/robot-usado-kr-3-r540-agilus-es.html>

Garcia, A. (2018). *¿Qué es google cloud platform?* Obtenido de <https://www.doctormetrics.com/google-cloud-platform/>

Git. (2019). *Login*. Obtenido de <https://github.com/login>

Google. (2019). *Google Cloud Platform*. Obtenido de <https://console.cloud.google.com/home/dashboard?project=robotic-energy-260400&hl=es>

Google, C. (2019). *Herramientas imprescindibles para Google Cloud Platform*. Obtenido de https://cloud.google.com/sdk/?&utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-skws-all-all-trial-e-dr-1007178-LUAC0009293&utm_content=text-ad-none-none-DEV_c-CRE_403088706946-ADGP_SKWS+%7C+Multi+~+Dev+%7C+SDK-KWID_43700042945981925-kw

Heroku. (2019). *The Heroku CLI*. Obtenido de <https://devcenter.heroku.com/articles/heroku-cli>

Hydra. (2019). *Arquitectura de Microservicios*. Obtenido de <https://www.hydra1303.com/language/es/pks-es/arquitectura-de-microservicios-microservices-para-kubernets/>

- Instruments, N. (2019). *Información Detallada sobre el Protocolo Modbus*. Obtenido de <https://www.ni.com/es-cr/innovations/white-papers/14/the-modbus-protocol-in-depth.html>
- Lopez, D. (2014). *Cloud Computing: una perspectiva tecnológica*. Obtenido de <https://es.slideshare.net/dipina/tocando-lanube-v2>
- Martínez, T. (2016). *Uso del Servidor Web Integrado en el PLC S7-1200 para Supervisión y Control Básico*. Obtenido de <https://electronicaymanufactura.blogspot.com/2016/09/uso-del-servidor-web-integrado-en-el.html>
- Matrikon. (2019). *Que es un Servidor OPC?* Obtenido de <https://www.matrikonopc.es/opc-servidor/index.aspx>
- Mercado, J. (2019). *Sistemas programables avanzados*. Obtenido de <https://books.google.com.ec/books?id=Kz-dDwAAQBAJ&pg=PA58&lpg=PA58&dq=La+CPU+incorpora+un+microprocesador,+una+fuelle+de+alimentaci%C3%B3n+integrada,+circuitos+de+entrada+y+salida,+PROFINET&source=bl&ots=zFtRHsim2N&sig=ACfU3U2QMwafNLGEhJoTaXN4Sm6tmLr2RQ&h>
- Merino, M. (2014). *¿Qué es una API y para qué sirve?* Obtenido de <https://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>
- Murcia, U. d. (2017). *Páginas web dinámicas*. Obtenido de <https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-13-paginas-web-dinamicas.html>

- Navarrete, T. (2000). *El lenguaje JavaScript*. Obtenido de https://s3.amazonaws.com/academia.edu.documents/54012715/javascript.pdf?response-content-disposition=inline%3B%20filename%3DEl_lenguaje_JavaScript.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20191230%2Fus-east-1%2Fs3%2Faws
- OpenCV. (2019). *Acerca de OpenCV*. Obtenido de <https://opencv.org/about/>
- Pérez, J. Y., & Fernández, N. L. (2002). *El lenguaje V+*. Obtenido de <http://www.etitudela.com/profesores/rpm/rpm/downloads/manualv.pdf>
- Pérez, J., & Merino, M. (2008). *TCP/IP*. Obtenido de <https://definicion.de/tcp-ip/>
- Pérez1, J. (2016). *Definición de cámara web*. Obtenido de <https://definicion.de/camara-web/>
- Profibus. (2017). *Profinet*. Obtenido de <http://profibus.es/profinet>
- Profinet, U. (2019). *¿Qué es Profinet?* Obtenido de <https://profinetuniversity.com/profinet-basico/que-es-profinet/>
- Pythones. (2019). *Qué es Python – Definición, características y ventajas*. Obtenido de <https://pythones.net/que-es-python-y-sus-caracteristicas/>
- Raffino, M. (2019). *HTTP*. Obtenido de <https://concepto.de/http/>
- Ramos, A. (2011). *Aplicaciones web*. Obtenido de <https://books.google.es/books?hl=es&lr=&id=LXs3YIMoeNgC&oi=fnd&pg=PA1&dq=q ue+es+un+servidor+web&ots=SIXPbnxxJI&sig=Lp9Ik2dFpeA0vzfBHPyRwGFZzew#v=onepage&q=Un%20servidor%20web%20es%20un%20programa%20dise%C3%B1ado&f=false>

- Ramos, A. (2015). *Que es un socket en programacion?* Obtenido de <https://evilnapsis.com/2015/11/12/que-es-un-socket-en-programacion/>
- Ramos, X. (2018). *La era del robot se instala en Ecuador.* Obtenido de <https://www.eluniverso.com/noticias/2018/08/17/nota/6909632/era-robot-se-instala-paso-lento-ecuador>
- Reyes, J. J. (2018). *¿Qué es HTML?* Obtenido de <https://devcode.la/>
- Roa, C. (2003). *Tendencias de la Computación.* Obtenido de <http://cesarroa.atspace.cc/Materias/Modelo%20OSI.pdf>
- Robles, V. (2018). *¿Que es Git y para que sirve?* Obtenido de <https://victorroblesweb.es/2018/04/28/que-es-git-y-para-que-sirve/>
- Romero, E. (2019). *Heroku – Una plataforma para la creación de aplicaciones.* Obtenido de <http://estebanromero.com/herramientas-emprender-desarrollar-proyectos/heroku-una-plataforma-para-la-creacion-de-aplicaciones/>
- Rosado, A. (2008). *Laboratorio remoto para la programación de robots.* Recuperado el 01 de 08 de 2019, de <http://e-spacio.uned.es/fez/eserv/taee:congreso-2008-1066/S3A02.pdf>
- SainSmart. (2019). *SainSmart 445nm/5.5W.* Obtenido de https://www.amazon.com/-/es/dp/B07RV8TG18?psc=1&ref=ppx_pop_dt_b_product_details
- Salesforce. (2019). *Salesforce Customer 360 Platform.* Obtenido de <https://www.salesforce.com/es/>
- Service, A. W. (2019). *AWS.* Obtenido de <https://aws.amazon.com/es/>

Services, A. W. (2019). *Informática en la nube con AWS*. Obtenido de <https://aws.amazon.com/es/what-is-aws/>

Sevilla, F. (2010). *Modelo SaaS: Software como Servicio – Computación en la nube*. Obtenido de <http://www.elementoscomunes.com/modelo-saas-software-como-servicio-computacion-en-la-nube/#.XidoGGhKhPY>

TensorFlow. (2019). *TensorFlow*. Obtenido de <https://www.tensorflow.org/>

Tikay. (2018). *Tikay* . Obtenido de <http://www.tikay.com.ec/corte-laser-en-ecuador-servicios.html>