



UNIVERSIDAD INDOAMÉRICA
FACULTAD DE INGENIERÍA, INDUSTRIA Y
PRODUCCIÓN.
CARRERA DE INGENIERÍA INDUSTRIAL

TEMA:

**“SISTEMA DE ADQUISICIÓN DE TIEMPOS DE PRODUCCIÓN
APLICANDO IOT EN UNA EMPRESA DE MANUFACTURA”**

Trabajo de titulación previo a la obtención del título de Ingeniero Industrial, bajo la Modalidad de Propuesta Metodológica.

Autor(a)

José Luis Andagama Velásquez

Tutor(a)

Ing. Hernán Fabricio Espejo Viñán, MSc.

QUITO – ECUADOR

2023

**AUTORIZACIÓN POR PARTE DEL AUTOR PARA LA CONSULTA,
REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA
DEL TRABAJO DE TITULACIÓN**

Yo, José Luis Andagama Velásquez, declaro ser autor del Trabajo de Titulación con el nombre **“SISTEMA DE ADQUISICIÓN DE TIEMPOS DE PRODUCCIÓN APLICANDO IOT EN UNA EMPRESA DE MANUFACTURA”**, como requisito para optar al grado de Ingeniería Industrial y autorizo al Sistema de Bibliotecas de la Universidad Indoamérica, para que con fines netamente académicos divulgue esta obra a través del Repositorio Digital Institucional (RDI-UI).

Los usuarios del RDI-UI podrán consultar el contenido de este trabajo en las redes de información del país y del exterior, con las cuales la Universidad tenga convenios. La Universidad Indoamérica no se hace responsable por el plagio o copia del contenido parcial o total de este trabajo.

Del mismo modo, acepto que los Derechos de Autor, Morales y Patrimoniales, sobre esta obra, serán compartidos entre mi persona y la Universidad Indoamérica, y que no tramitaré la publicación de esta obra en ningún otro medio, sin autorización expresa de la misma. En caso de que exista el potencial de generación de beneficios económicos o patentes, producto de este trabajo, acepto que se deberán firmar convenios específicos adicionales, donde se acuerden los términos de adjudicación de dichos beneficios.

Para constancia de esta autorización, en la ciudad de Quito, a los veinte y tres días del mes de enero del 2023, firmo conforme:

Firma: _____

Autor: José Luis Andagama Velásquez

Número de Cédula: 1723708481

Dirección: Pichincha, Quito, Solanda, Solanda

Correo Electrónico: josephandagama2601@gmail.com

Teléfono: 022733384 - 0984063665

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación “SISTEMA DE ADQUISICIÓN DE TIEMPOS DE PRODUCCIÓN APLICANDO IOT EN UNA EMPRESA DE MANUFACTURA”, presentado por José Luis Andagama Velásquez, para optar por el Título de Ingeniero Industrial.

CERTIFICO

Que dicho trabajo de investigación ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del Tribunal Examinador que se designe.

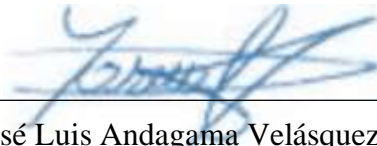
Quito, 25 de marzo del 2023

Ing. Hernán Fabricio Espejo Viñán. MSc.
CC:18019659358

DECLARACIÓN DE AUTENTICIDAD

Quien suscribe, declaro que los contenidos y los resultados obtenidos en el presente trabajo de investigación, como requerimiento previo para la obtención del Título de Ingeniero Industrial, son absolutamente originales, auténticos y personales y de exclusiva responsabilidad legal y académica del autor.

Quito, 25 de marzo del 2023



José Luis Andagama Velásquez

1723708481

APROBACIÓN TRIBUNAL

El trabajo de Titulación ha sido revisado, aprobado y autorizada su impresión y empastado, sobre el Tema: **“SISTEMA DE ADQUISICIÓN DE TIEMPOS DE PRODUCCIÓN APLICANDO IOT EN UNA EMPRESA DE MANUFACTURA”**, previo a la obtención del Título de Ingeniero Industrial, reúne los requisitos de fondo y forma para que el estudiante pueda presentarse a la sustentación del trabajo de titulación.

Quito, 25 de marzo del 2023

Ing. Topón Visarrea Blanca Liliana
MSc. LECTOR

Ing. Sarmiento Ortiz Fabian Alberto
MSc. LECTOR

DEDICATORIA

Dedico con todo mi corazón esta tesis a mi madre, pues sin ella no lo habría logrado, tu bendición a diario me protege. También quiero dedicárselo a mi padre, ya que tu presencia en mi vida me llena de alegría. Por todo su amor y apoyo incondicional doy este trabajo en ofrenda, los amo.

José Luis Andagama Velásquez.

AGRADECIMIENTO

Quiero extender un profundo agradecimiento a Dios y a mis padres; Cecilia, Jorge, sin su apoyo incondicional no habría podido llegar a esta cumbre. A mis amigos que su compañía fue más que solo risas, sobre todo Alisson, por su cálida compañía en todas las noches de desvelo; a Reidel por su ayuda incondicional. A mi tutor Hernán Espejo, por confiar en mí dándome la oportunidad de abordar este tema y a todos aquellos que formaron parte de este proceso hermoso.

Muchas Gracias.

José Luis Andagama Velásquez.

ÍNDICE GENERAL

TEMA:	i
APROBACIÓN DEL TUTOR.....	iii
DECLARACIÓN DE AUTENTICIDAD	iv
APROBACIÓN TRIBUNAL.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xv
ÍNDICE DE ANEXOS.....	xvii
RESUMEN EJECUTIVO	xviii
ABSTRACT	xix
CAPÍTULO I.....	1
INTRODUCCIÓN	1
Marco teórico.	6
Antecedentes.	22
Justificación.....	28
Objetivos.	30
Objetivo general.	30

Objetivos Específicos.....	30
CAPÍTULO II	31
INGENIERÍA DEL PROYECTO.....	31
Diagnóstico de la situación actual de la empresa.....	31
Área de estudio.....	35
Modelo operativo.....	35
CAPÍTULO III.....	38
PROPUESTA Y RESULTADOS ESPERADOS	38
Desarrollo de la propuesta.....	38
Recopilación de información.....	38
Análisis y evaluación de las necesidades.....	43
Detección de problemas.....	46
Selección de componentes.....	48
Diseño del prototipo.....	50
Programación del prototipo.....	55
Programación Emisor.....	58
Programación Receptor.....	64
Ensamble del prototipo.....	67
Pruebas de funcionamiento.....	68

Prueba de reinicio.....	74
Prueba inicio nuevo paquete.....	77
Visualización de datos finales de simulación.....	80
Prueba de conexión DNS para visualización remota	83
Resultados esperados.....	86
Análisis de costos.	94
CAPÍTULO IV	97
CONCLUSIONES Y RECOMENDACIONES.....	97
Conclusiones.	97
Recomendaciones.....	99
BIBLIOGRAFÍA.....	101
ANEXOS.....	103

ÍNDICE DE FIGURAS

Figura 1. Índice de confianza empresarial por sectores	1
Figura 2. Tamaño anual de la esfera de datos global	4
Figura 3. Diagrama de flujo para recolección de datos de producción en una manufacturera textil	5
Figura 4. Modelo OSI y suite de protocolos TCP/IP	7
Figura 5. Descripción general de la arquitectura basada en internet de las cosas..	9
Figura 6. Otros tipos de tecnología	16
Figura 7. Otros tipos de tecnología continuación.....	17
Figura 8.	20
Figura 9. Funcionamiento de primer sistema de adquisición de datos.....	22
Figura 10. Validación de datos en la programación del primer prototipo.....	23
Figura 11. Diagrama de flujo programación primer prototipo.....	24
Figura 12. Diagrama de flujo programación primer prototipo base de datos en MySQL.....	25
Figura 13. Diagrama de flujo, programación primer prototipo, disparador de datos (emisión de señal).....	26
Figura 14. Datos de producción que registra el primer prototipo	27
Figura 15. Diagrama de control estadístico tiempo empleado vs tiempo estándar de operación de la empresa	29
Figura 16. Tiempo de operación dispositivo vs Tiempo estándar de operación de la empresa.....	31
Figura 17. Línea ajustada y validación de datos a máximo 3 sigma.....	32

Figura 18. Puesto piloto en línea de producción	33
Figura 19. Requerimientos por las partes interesadas.....	42
Figura 20. Despliegue de la función de calidad	44
Figura 21. Flujo de datos del sistema anterior	50
Figura 22. Flujo de datos para el nuevo sistema propuesto	51
Figura 23. Flujo de trabajo anterior en un puesto de la línea de producción	52
Figura 24. Nuevo flujo de trabajo propuesto con el sistema en la línea de producción.....	53
Figura 25. Flujo de señales en el medio.....	54
Figura 26. Estructura de programación.....	55
Figura 27. Visualización remota a través de IoT.	65
Figura 28. Visualización final, programación orientada a objetos en la arquitectura por capas del sistema propuesto.....	66
Figura 29. Equipos de sistema (emisores).....	67
Figura 30. Ingreso datos equipo Raspberrypi 1(emisor 1).....	68
Figura 31. Envío de datos Raspberrypi 1(emisor 1).....	69
Figura 32. Visualización de datos Raspberrypi1 (emisor 1)	69
Figura 33. Ingreso datos equipo Raspberrypi2 (emisor 2).....	70
Figura 34. Envío de datos Raspberrypi2 (emisor 2).....	71
Figura 35. Visualización de datos Raspberrypi2 (emisor 2)	71
Figura 36. Ingreso datos equipo Raspberrypi3 (emisor 3).....	72
Figura 37. Envío de datos Raspberrypi3 (emisor 3).....	72
Figura 38. Visualización de datos Raspberrypi3 (emisor 3)	73

Figura 39. Visualización de datos integrados en Rasperryi4 (receptor/integrador)	73
Figura 40. Toma últimos datos registrados Rasperryi1 (emisor 1)	74
Figura 41. Toma últimos datos registrados Rasperryi2 (emisor 2)	74
Figura 42. Toma últimos datos registrados Rasperryi3 (emisor 3)	75
Figura 43. Finalización paquete Rasperryi1 (emisor1)	75
Figura 44. Finalización paquete Rasperryi2 (emisor2)	76
Figura 45.	76
Figura 46. Ingreso nuevo paquete Rasperryi1 (emisor1)	77
Figura 47. Ingreso nuevo paquete Rasperryi2 (emisor2)	78
Figura 48. Ingreso nuevo paquete Rasperryi3 (emisor3)	78
Figura 49.	79
Figura 50. Registro datos nuevo paquete Rasperryi2 (emisor2)	79
Figura 51. Registro datos nuevo paquete Rasperryi3 (emisor3)	80
Figura 52. Visualización de datos finales de simulación Rasperryi1 (emisor 1)	80
Figura 53. Visualización de datos finales de simulación Rasperryi2 (emisor 2)	81
Figura 54. Visualización de datos finales de simulación Rasperryi3 (emisor 3)	81
Figura 55. Visualización de datos Integrados (receptor/integrador) LAN.....	82
Figura 56. Prueba enlace dominio con IP dispositivo para acceso en la red.....	83
Figura 57. Port forwarding (redireccionamiento de puerto)	84

Figura 58. Visualización de datos integrados, Raspberrypi4 (receptor/integrador) WAN.....	85
Figura 59. Mejora en la gestión del flujo de información.....	86
Figura 60. Facilidad en la operatividad del sistema.....	88
Figura 61. Inclusión de IoT.....	89
Figura 62. Datos integrados de varios equipos en un solo punto.....	91
Figura 63. Cambio periférico de entrada.....	92

ÍNDICE DE TABLAS

Tabla 1. Ejemplos de tecnología presente en IoT según UIT.....	10
Tabla 2. Tipos de manufactura por su posicionamiento en la cadena de producción económica.	18
Tabla 3. Modelo operativo seleccionado.....	36
Tabla 4. Modelo operativo-primera fase	37
Tabla 5. Modelo operativo-segunda fase.....	37
Tabla 6. Desarrollo de la propuesta (primera fase)	38
Tabla 7. Ficha técnica descriptiva del dispositivo.....	38
Tabla 8. Especificaciones demandadas por las partes interesadas	41
Tabla 9. Calificación de especificaciones por partes interesadas.....	43
Tabla 10. Resultados del análisis QFD	45
Tabla 11. Orden de problemas de acuerdo a la priorización calculada	46
Tabla 12. Seudocódigo del sistema inicial	47
Tabla 13. Oportunidades de mejora	47
Tabla 14. Desarrollo de la propuesta (segunda fase).....	48
Tabla 15. Componentes del sistema	49
Tabla 16. Arquitectura de programación.....	55
Tabla 17. Tipo de programación (orientada a objetos).	56
Tabla 18. Seudocódigo del equipo emisor – programación orientada a objetos, adquisición e integración.....	58
Tabla 19. Seudocódigo emisor – programación orientada a objetos, visualización	

web.....	62
Tabla 20. Programación autoarranque del programa en linux	63
Tabla 21. Seudocódigo receptor – programación orientada a objetos,	
visualización web.....	64
Tabla 22. Programación en Linux	65
Tabla 23. Cronograma sugerido para implementación.....	93
Tabla 24. Costos referenciales de elementos utilizados.	94
Tabla 25. Costos implementación en la empresa (fijos)	95
Tabla 26. Costo total de implementación del sistema	96

ÍNDICE DE ANEXOS

Anexo 1. Datos integrados por IoT	103
Anexo 2. Código del sistema de adquisición de tiempos de producción en Python	109
Anexo 3. Código visualización web (emisor)	122
Anexo 4. Formulario obtenido de entrevista para valoración de requerimientos	126

UNIVERSIDAD INDOAMÉRICA

FACULTAD DE INGENIERÍA, INDUSTRIA Y PRODUCCIÓN

CARRERA DE INGENIERÍA INDUSTRIAL

**TEMA: “SISTEMA DE ADQUISICIÓN DE TIEMPOS DE PRODUCCIÓN
APLICANDO IOT EN UNA EMPRESA DE MANUFACTURA”**

AUTOR: José Luis Andagama Velásquez

TUTOR: MSc. Hernán Fabricio Espejo Viñán

RESUMEN EJECUTIVO

El presente trabajo de titulación tiene como objetivo el desarrollar soluciones a las oportunidades de mejora presentes en el sistema de comunicación para la adquisición de datos, ubicado en una empresa de confección de la ciudad de Quito. Mediante la aplicación del internet de las cosas IoT se propone; una mejora a la lógica de programación, pasando de la lectura de 5 parámetros a 8 lo que identifica al operador, puesto de trabajo, máquina utilizada y la actividad del lote de producción. Logrando así la adquisición e integración de datos de tiempos de producción en un solo punto, dando un salto a la industria 4.0 en la empresa, que para esta investigación la mayor parte de operaciones de manufactura son desarrollados en entornos hombre-máquina. De esta manera el sistema propuesto aporta a la probabilidad de establecer una comunicación con la red global, que permitirá a las partes interesadas tener acceso a datos de producción de manera remota desde cualquier parte del mundo. El sistema de adquisición de tiempos de producción deja la posibilidad a futuro la incorporación de nuevos dispositivos de envío de señal, la incorporación de complementos o sistemas de tratamiento de datos, generación de reportes automáticos, proyecciones, control de producción, balanceo de líneas, entre tantas otras aplicaciones que se podrán ajustar a los indicadores de gestión de las empresas que así lo demanden, todo esto vía internet.

DESCRIPTORES: adquisición, integración, IoT, tiempos de producción

UNIVERSIDAD INDOAMÉRICA

FACULTAD DE INGENIERÍA, INDUSTRIA Y PRODUCCIÓN

CARRERA INGENIERIA INDUSTRIAL

**THEME: “PRODUCTION TIME ACQUISITION SYSTEM APPLYING IOT
IN A MANUFACTURING COMPANY”**

AUTHOR: José Luis Andagama Velásquez

TUTOR: MSc. Hernán Fabricio Espejo Viñán

ABSTRACT

The objective of this degree work is to develop solutions to the improvement opportunities present in the communication system for data acquisition, located in a clothing company in the city of Quito.

Through the application of the internet of things IoT is proposed; an improvement to the programming logic, going from reading 5 parameters to 8, which identifies the operator, job position, machine used and the activity of the production batch. Thus, achieving the acquisition and integration of production time data at a single point, giving a leap to industry 4.0 in the company, which for this research most manufacturing operations are developed in man-machine environments. In this way, the proposed system contributes to the probability of establishing communication with the global network, which will allow interested parties to have access to production data remotely from anywhere in the world.

The production time acquisition system allows for the future incorporation of new signal sending devices, the incorporation of complements or data processing systems, automatic report generation, projections, production control, line balancing, among others. many other applications that can be adjusted to the management indicators of the companies that request it, all via the Internet.

KEYWORDS: acquisition, integration, IoT, production times

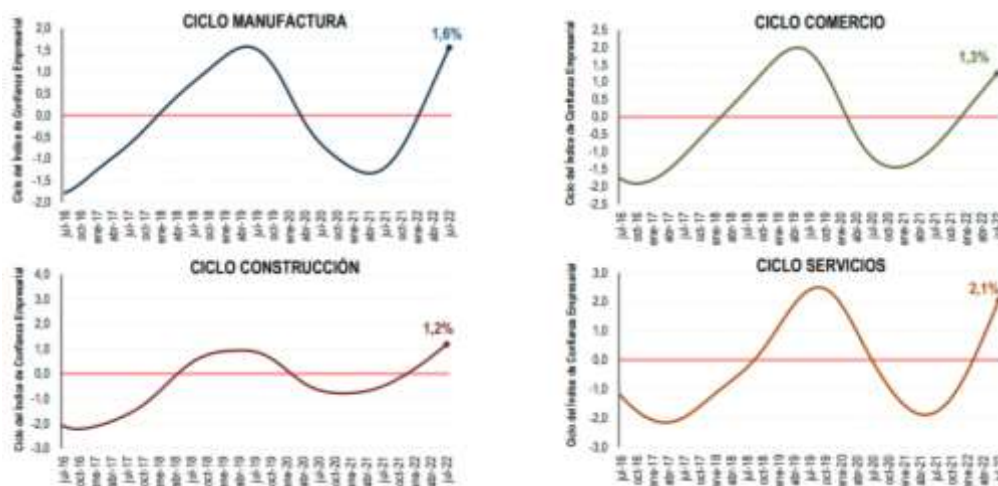
CAPÍTULO I

INTRODUCCIÓN

La tecnología con el pasar del tiempo ha ido cambiando, mejorando, con forme las necesidades del ser humano lo demanda. Desde la primera revolución industrial dada en 1760 hasta la industria 4.0, la tecnología ha evolucionado ha pasos exorbitantes; La industria manufactura no es la excepción. Según el banco central del Ecuador la industria de la manufactura es la segunda con mayor índice de confianza superado solo por los servicios lo cual podría representar un gran impacto en la sostenibilidad económica del país si esta decae de alguna manera.

Figura 1.

Índice de confianza empresarial por sectores



Nota. Porcentajes de desviación respecto a la tendencia por industrias julio 2016 – julio 2022. Fuente (Banco central del Ecuador, 2022). Elaborado por (Banco central del Ecuador, 2022).

Se puede evidenciar que la industria manufacturera es la segunda mayor que mantiene un incremento económico después de la industria de los servicios.

Según el (Banco central del Ecuador, 2022). Cualquier declive en el desarrollo de la productividad en esta industria no solo significa un impacto económico a nivel país, sino que también significa un

impacto a la sostenibilidad de las industrias ecuatorianas con respecto a la competencia internacional que elaboren productos similares o del mismo tipo.

Como aplicación al igual que muchos posibles escenarios a esta industria puede comenzar en lo más básico, medidas, tiempos, datos de producción. Para poder analizar, controlar, evaluar un entorno de producción, es primordial para toda empresa ya que gracias a la adquisición de dichos datos permite la elaboración de indicadores de gestión y a su vez la integración para la comparación con otros datos de la empresa.

Si bien la forma de obtención de datos ya es un reto para las empresas, encontrar la manera de comunicar, entrelazar, integrar esa información puede parecer más complicado. Afortunadamente en 1983 bajo el protocolo TCP/IP el departamento de defensa de los estados unidos crea la red denominada arpa net que con el tiempo quedaría únicamente con la definición de “Internet”. “Es así como nace un nuevo sistema que permite interconectar equipos de computación e intercambiar información, aunque pasaría varios años hasta que el mismo se pudiera transferir de forma inalámbrica” (Bahillo, 2022).

“El 12 de marzo de 1989 Tim Berners Lee describió por primera vez el protocolo de transferencias de hipertextos que daría lugar a la primera web utilizando tres nuevos recursos: HTML, HTTP y un programa llamado Web Browser. Un año después Internet nacía de forma cerrada dentro del CERN, y en agosto de 1991, por fin, los usuarios externos al CERN comenzaron a poder acceder a esa información” (Bahillo, 2022).

Es aquí donde las industrias pueden aplicar no solo a modo de mejora si no de reingeniería, con el IoT (Internet of Things), o por su traducción al español (Internet de las cosas)

“El IoT tiene como objetivo unificar todo en nuestro mundo bajo una infraestructura común, brindándonos no solo el control de las cosas que nos rodean, sino también manteniéndonos informados

del estado de las cosas” (Madakam et al., 2015).

Entre otras denominaciones al internet de las cosas también se lo conoce como: “Web of things, Internet of objects, Embedded intelligence, Connected devices and technology, Cyber Physical Systems, etc.”(Madakam et al., 2015).

Tomar este principio de comunicación del internet, más transponer a los objetos en procesos diferentes para cumplir objetivos en común, el IoT puede ser vista desde un aspecto macro como una salida a más de un problema en las industrias.

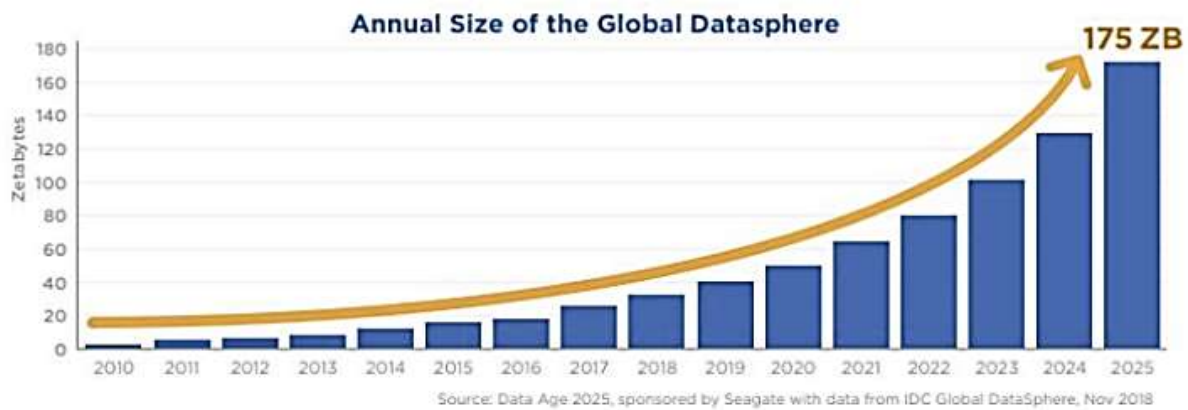
“La importancia que tienen los datos en las empresas llega a tal punto de considerarlos un activo más de la misma, pero los datos son valores que por sí solos no dicen nada, a menos que se le asigne a una variable en un caso de estudio” (*La Importancia de La Calidad de Los Datos En Las Empresas*, 2020).

Los datos procesados son útiles para la toma de decisiones, esto puede ser; mezclar la información con indicadores, compara su cambio en el tiempo, comparar las magnitudes obtenidas y conectadas a un objeto de estudio para obtener conclusiones, que ayudaran entre muchos otros análisis a la toma de decisiones empresariales.

En la **Figura 2** se puede visualizar el incremento de las empresas en el tiempo, que han optado por la adquisición de datos a través de la industria 4.0 e internet de las cosas.

Figura 2.

Tamaño anual de la esfera de datos global



Nota. Evolución en tamaño de datos tomados por las empresas 2010- 2022. Fuente. Digitization of the world from Edge to Core. Elaborado por. Digitization of the world from Edge to Core.

El crecimiento de los datos es exponencial, la mayor parte del volumen de información actualmente es guardado en la nube por servidores externos, siendo los dispositivos IOT conectados a la red (más de 150.000 millones) los que registrarán mayores crecimientos.

“Más del 50% de las empresas carecen del control total de sus datos, siendo la mejora de su seguridad. Se estima que en el año 2025 el volumen de los datos en el mundo será 175 veces más que en el año 2011 y que cada persona interactuará con dispositivos unas 4.800 veces al día” (*La Importancia de La Calidad de Los Datos En Las Empresas, 2020*).

Por otro lado la obtención de datos en la producción, son un factor determinante para la regulación y optimización de recursos, puesto que en un proceso productivo que no recibe medida y control alguno dificulta la toma de decisiones y análisis del entorno.

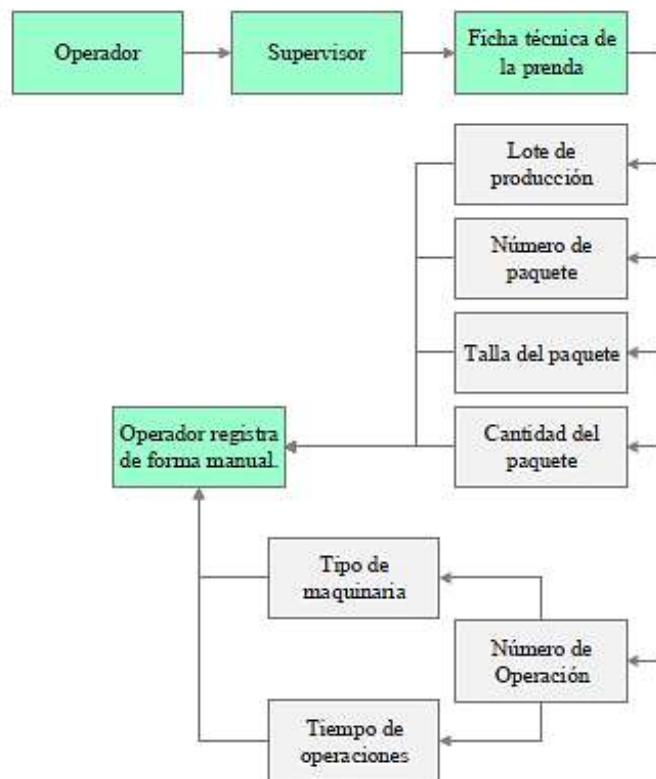
Conforme la empresa crece, estos análisis son cada vez más complejos, la trazabilidad de todos los procesos desemboca en la producción, pero esta a su vez contiene procedimientos. “De todas las

empresas de la industria de la manufactura la industria que ejecuta sus procesos manuales son los que más complicación tienen al momento de obtener datos de producción”(Cuyachamín, 2022) por ejemplo en una empresa textil el proceso de ensamblaje es uno de los más cruciales puesto que es aquí donde se ensambla y da forma al producto final.

En la **Figura 3** se puede visualizar el diagrama de flujo de la operación para recolección de datos de una empresa de manufactura textil, el cual muestra alguna de las variables necesarias a tomar en cuenta para seguir la trazabilidad del producto.

Figura 3.

Diagrama de flujo para recolección de datos de producción en una manufacturera textil



Nota. El operador es encargado de llenar manualmente los datos. Fuente (Cuyachamín, 2022).

Elaborado por (Cuyachamín, 2022).

Marco teórico.

Internet de las cosas (IoT).

“El Internet de las cosas se inspiró inicialmente en miembros de la comunidad RFID, quienes se refirieron a la posibilidad de descubrir información sobre un objeto etiquetado al navegar por una dirección de Internet o una entrada de base de datos que corresponda a un RFID o Near Field Comunicación en particular” (Madakam et al., 2015).

“El Internet de las cosas (IoT) permitió a los usuarios traer objetos físicos a la esfera del mundo cibernético, esto fue posible gracias a diferentes tecnologías de etiquetado como NFC, RFID y código de barras 2D que permitía identificar y referir objetos físicos a través de Internet” (Madakam et al., 2015).

También se recalca que el IoT incluye todo tipo de tecnologías sofisticadas como: “redes informáticas y de comunicación externas, recopilación de tecnología de la información, tecnología de comunicación remota, tecnología de transmisión de información remota, análisis de inteligencia de información de medidas marítimas y tecnología de control, etc.” (Madakam et al., 2015).

El IoT puede ser interpretada estructuralmente de varias formas, entre las principales se encuentran: “Arquitectura de la UIT, Arquitectura del foro IoT, Qian Xiaocong, Arquitectura de Zhang Jidong, Arquitectura de Kun Han, Shurong Liu, Dacheng Zhang y Ying Han” (Madakam et al., 2015).

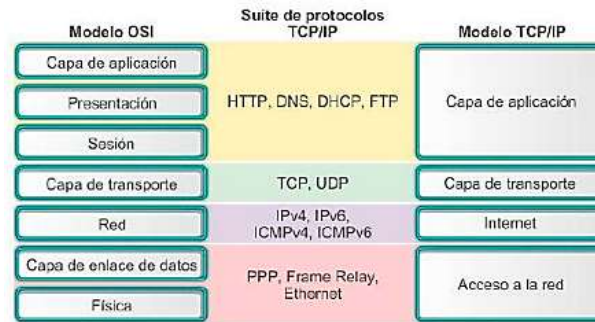
Por los convenios y tratados internacionales ratificados por el Ecuador, o en su defecto, a lo establecido en el Reglamento General a la presente Ley orgánica de Telecomunicaciones basa mucha de sus estructuras en base a la Unión Internacional de Telecomunicaciones (UIT). Por lo tanto, en el IoT aplicado a este trabajo se toma la arquitectura de la misma que lleva sus iniciales “Arquitectura de la UIT”. Por punto de normativa hace la más óptima para llevar a cabo.

De acuerdo con las recomendaciones de la (UIT), la red del Internet de las Cosas debe constar de:

capa de detección, capa de acceso, capa de red, capa de software intermedio, capas de aplicación.
 Véase de mejor manera en la **Figura 4**.

Figura 4.

Modelo OSI y suite de protocolos TCP/IP



Nota. Fuente (Madakam et al., 2015). Elaborado por (Madakam et al., 2015)

Cabe recalcar que el modelo OSI de la ISO, es un modelo de referencia para los protocolos de la red el cual se basa en la arquitectura de red de la UIT, mas no es una arquitectura de red.

Capas del Internet de las cosas (IoT)

Capa física.

La capa física también se conoce como capa de percepción o capa inferior de la arquitectura. Esta capa contiene dispositivos heterogéneos (por ejemplo, sensores y actuadores) que recopilan información y la envían a la capa superior de la arquitectura denominada capa de red para su posterior procesamiento.

Capa de red.

La capa de red también se conoce como capa de comunicación, que es la capa central de la arquitectura basada en IoT. Esta capa depende de redes básicas como redes de sensores inalámbricos (WSN) e Internet y redes de comunicación. La principal responsabilidad de la capa de red es transmitir los datos recopilados por la capa física y conectar varios dispositivos de la red, como servidores y

elementos inteligentes.

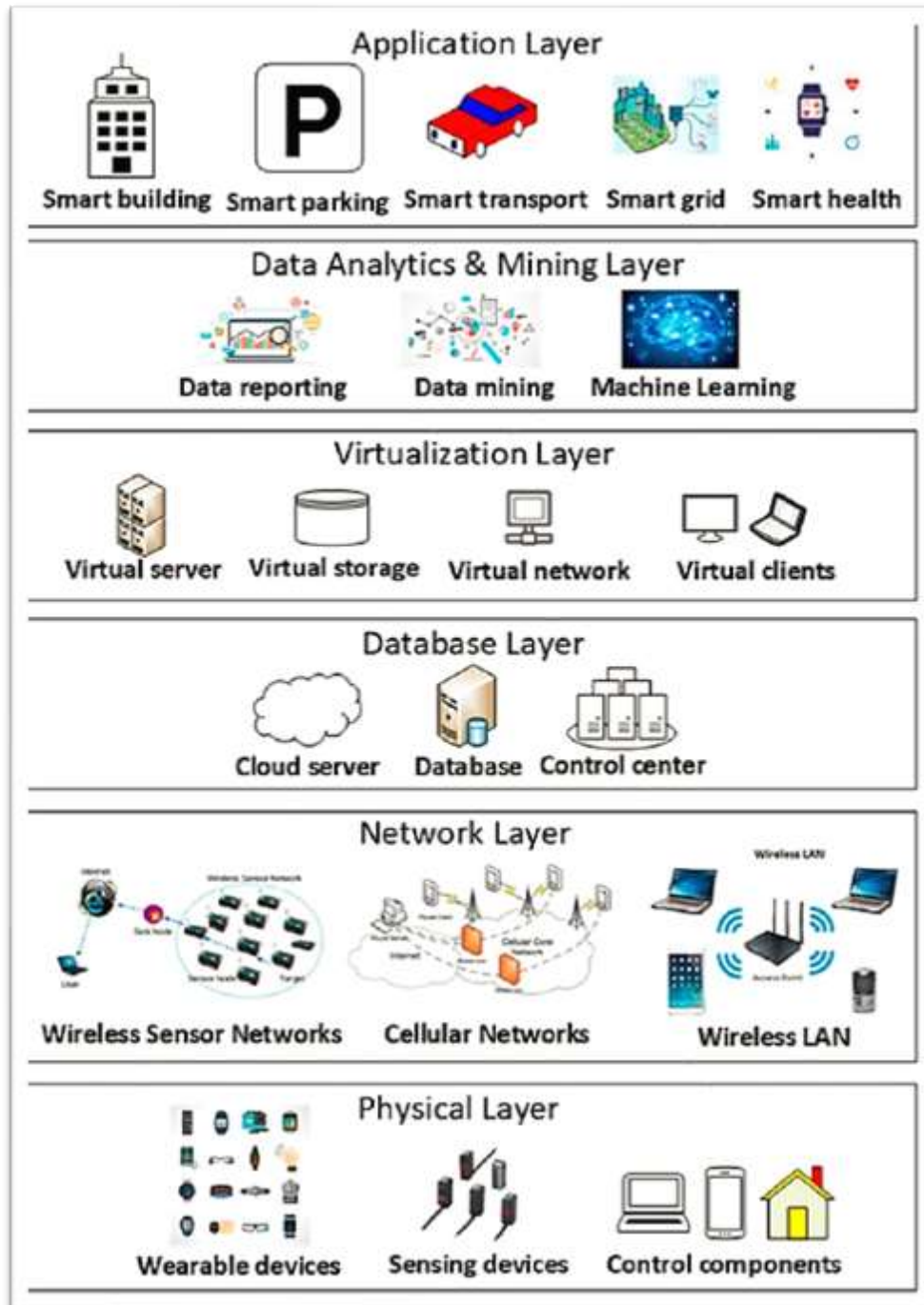
Base de datos.

Otra capa dentro de la arquitectura de la IoT es la base de datos, según (Madakam et al., 2015) la capa de base de datos también se conoce como capa de soporte y opera en estrecha colaboración con las capas superiores de la arquitectura. Se compone de servidores de bases de datos y sistemas informáticos inteligentes. La responsabilidad principal de esta capa es brindar soporte para los requisitos de las aplicaciones a través de enfoques de computación inteligente, como la computación en la nube o perimetral.

En la **Figura 5** se puede visualizar un esquema jerárquico de las capas mencionadas en el Internet de las cosas, cabe recalcar que es una integración de varias tecnologías.

Figura 5.

Descripción general de la arquitectura basada en internet de las cosas



Nota. La capa física es la primera capa, seguida por la capa de red que esta a su vez alberga a la capa de base de datos. Tomado de (Al-Turjman et al., 2022), Elaborado por: (Al-Turjman et al., 2022).

Tecnologías presentes en las HetNets del IoT

“El IoT se puede considerar como HetNets, donde diferentes redes de distintos protocolos se integran y conectan entre sí para brindar mejores servicios a las personas, como redes de teléfonos inteligentes, redes sociales, Internet y redes industriales”(Al-Turjman et al., 2022).

La arquitectura de red, tecnológicamente puede estar integrada por: identificación por radiofrecuencia (RFID), protocolo de internet (IP), código de producto electrónico, código de barras, fidelidad inalámbrica (Wi-Fi), Bluetooth, ZigBee ZigBee, comunicación de archivo cercano (NFC), actuadores, redes de sensores inalámbricos (WSN), inteligencia artificial (IA)

En la **Tabla 1** se puede visualizar las principales características de cada una de las tecnologías presentes en la arquitectura del IoT según la UIT.

Tabla 1.

Ejemplos de tecnología presente en IoT según UIT

Tecnología	Características principales	Clasificación	Componentes	Usos comunes
Identificación por radiofrecuencia (RFID)	Sistema que transmite la identidad de un objeto o persona de forma inalámbrica, mediante ondas de radio en forma de un número de serie. La tecnología RFID juega un	RFID activo, RFID pasivo y RFID semipasivo.	Etiqueta, el lector, la antena, el controlador de acceso, el software y el servidor	Distribución, rastreo, monitoreo de pacientes, aplicaciones militares, etc.

	papel importante en IoT para resolver problemas de identificación.			
Protocolo de Internet (IP)	Principal protocolo de red utilizado en Internet. Principal protocolo de comunicaciones en el conjunto de protocolos de Internet para transmitir diagramas a través de los límites de la red.	IPv4 e IPv6. (Cada versión define una dirección IP de manera diferente).	Clase A, Clase B, Clase C, Clase D y Clase E.	Identificación de un ordenador en el internet.
Código de producto electrónico (EPC)	Código de 64 o 98 bits grabado electrónicamente en una etiqueta RFID y destinado a diseñar una mejora en el sistema de código de barras EPC. (número de serie único del producto).	Servicio de nombres de objetos (ONS), Servicio de descubrimiento de EPC (EPCDS), Servicios de información de EPC (EPCIS) y Servicios de seguridad de		Estandarización de la tecnología de código electrónico de productos.

EPC (EPCSS).				
Código de barras	Forma diferente de codificar números y letras mediante el uso de una combinación de barras y espacios de ancho variable.	Alfanumérico, numérico y bidimensional.	Códigos de respuesta rápida (QR). Bar Code Book.	Métodos alternativos de técnicas de entrada de datos. Etiquetas ópticas legibles por máquina adheridas a artículos que registran información relacionada con el artículo.
Fidelidad inalámbrica (Wi-Fi)	Wireless Fidelity (Wi-Fi) es una tecnología de red que permite que las computadoras y otros dispositivos se comuniquen a través de una señal inalámbrica.	802.11, 802.11a, 802.11b, 802.11g, 802.11n.		Conectividad de red de área local inalámbrica (WLAN) de alta velocidad
Bluetooth	Tecnología de radio económica y de corto alcance que elimina la necesidad de cableado privado			Creación de redes de área personal (PAN). Creación de conjunto de

	entre dispositivos (periféricos) con un alcance efectivo de 10 a 100 metros Usa especificación del estándar IEEE 802.15.1.		dispositivos Bluetooth que comparten un canal común para la comunicación e intercambio de información.
ZigBee ZigBee	Protocolos desarrollados para mejorar las características de las redes de sensores inalámbricos, para red inalámbrica de baja potencia basado en el estándar IEEE 802.15.4 Alcance de alrededor de 100 metros y un ancho de banda de 250 kbps.	Topologías: estrella, árbol de clúster y malla	Domótica, agricultura digital, controles industriales, monitoreo médico y sistemas de energía. Necesidad de transmisión de datos de baja velocidad, rango de transmisión corto, escalabilidad, y protocolo flexible.
Comunicación de archivo	Por sus siglas en ingles Near Field		Simplificar las transacciones,

cercano (NFC)	<p>Communication (NFC) es un conjunto de tecnología inalámbrica de corto alcance a 13,56 MHz, requiere una distancia de 4 cm. Funciona en ambientes sucios, no requiere línea de visión, método de conexión fácil y simple.</p>	<p>intercambio de contenido digital y la conexión de dispositivos electrónicos. Inicialización intuitiva de redes inalámbricas</p>	
Actuadores	<p>Convierte energía en movimiento. Se necesita fluido hidráulico, corriente eléctrica o alguna otra fuente de energía. Los sistemas hidráulicos y neumáticos permiten aumentar la fuerza y el par de un motor más pequeño.</p>	<p>Eléctrico: motores de CA y CC, motores paso a paso, solenoides. Hidráulico: usa fluido hidráulico para activar el movimiento. Neumático: usa aire comprimido para activar el movimiento.</p>	<p>Impulsar o dar inicio al movimiento en sistemas mecánicos. Convierten energía en trabajo.</p>
Redes de	<p>Red inalámbrica</p>	<p>Militar,</p>	

sensores inalámbricos (WSN)	de dispositivos autónomos espacialmente distribuidos que utilizan sensores para monitorear de manera cooperativa condiciones físicas o ambientales, como la temperatura, el sonido, la vibración, la presión, etc.	seguridad nacional, atención médica, monitoreo de agricultura de precisión, fabricación, monitoreo de hábitats, detección de incendios forestales e inundaciones, etc.
Inteligencia Artificial (IA)	Refiere a entornos electrónicos que son sensibles y responden a la presencia de personas. En un mundo de inteligencia artificial, los dispositivos funcionan en concierto para ayudar a las personas a llevar a cabo todas sus	Integrado, Consciente del contexto, Personalizado, Adaptativas, Anticipadoras.

tareas.

Nota. Fuente (Madakam et al., 2015). Elaborado por Autor.

Como se puede observar las HetNets por si solas ya conforman una integración propia de otras tecnologías, lo que hace del IoT una herramienta fuerte y sólida para cualquier aplicación.

Aunque las tecnologías antes mencionadas son las más usuales en la aplicación de IoT, existen muchas más que entre otras siguen y seguirán en desarrollo, se puede visualizar de mejor manera en la

Figura 6 y Figura 7.

Figura 6.

Otros tipos de tecnología

Abreviado	Nombre
6LoWPAN	IPv6 sobre redes de área personal inalámbricas de baja potencia
Ataque de disponibilidad	Ataque de disponibilidad
AI	Inteligencia artificial
COMO	Sistema autónomo
AV	vehículo autónomo
BLE	Bluetooth de baja energía
BN	Red negra
Estación base	Estación base
Análisis de contenido	Análisis de contenido
Circuito cerrado de televisión	Circuito cerrado de televisión
CoAP	Protocolo de aplicación restringida
CS	detección llena
DDoS	Denegación de servicio distribuida
DoS	Negación de servicio
EVC	Carga electrónica de vehículos
FI	Información falsa
GPS	Sistema de Posicionamiento Global
HAI	Información del área de inicio
ÉL	Cifrado homomórfico
HetNet	Red heterogénea
HTTP	Protocolo de Transferencia de Hipertexto
CIE	Red centrada en la información Tecnología de la
TIC	información y la comunicación
DNI	Sistema de detección de intrusos
IHS	Sistema de salud inteligente

Nota. Otros tipos de tecnología que pueden integrar la HetNet. Fuente (Al-Turjman et al., 2022),

Elaborado por (Al-Turjman et al., 2022)

Figura 7.

Otros tipos de tecnología continuación

Internet de las cosas	Internet de las Cosas
ES	Suplantación de identidad
SU	Sistema de transporte inteligente
KMS	Sistema de gestión de llaves
LTE	Evolución a largo plazo
MCS	Detección de multitudes móvil
ML	Aprendizaje automático
M2M	máquina a máquina
mensaje	Modificación de mensajes
MQTT	Transporte de telemetría de Message Queue Server
P2P	De igual a igual
PPBS	Esquema biométrico que preserva la privacidad
PPDM	Minería de datos que preserva la privacidad
QoS	Calidad de servicio
rfid	Identificación por radiofrecuencia
SDN	Redes definidas por software
SG	Red inteligente
SSL	Capa de sockets seguros
ejército de reserva	Análisis de tráfico
TTP	Tercero de confianza
UAV	Protocolo de datagrama de usuario de
UDP	vehículos aéreos no tripulados
Naciones Unidas	Naciones Unidas
UR	registro unificado
VoIP	Voz sobre IP
WSN	Red de sensores inalámbricos

Nota. Otros tipos de tecnología que conforma las HetNets. Fuente (Al-Turjman et al., 2022),

Elaborado por (Al-Turjman et al., 2022)

Seguridad en el Internet de las cosas

Debido al entorno complejo en el cual se desarrolla el IoT, se requiere tecnologías efectivas para superar las problemáticas de seguridad y privacidad en los entornos inteligentes. “Es beneficioso modelar los patrones de dispersión de datos en las WSN.118” (Al-Turjman et al., 2022).

“En los sistemas inteligentes, se requiere autenticación para garantizar que los servicios en sistemas heterogéneos solo puedan ser accedidos por usuarios autorizados.119Los dispositivos IoT en las ciudades inteligentes son capaces de autenticar la propia red y otros nodos en la red y los mensajes de

las estaciones de gestión. Además, dado que la cantidad de autenticación de datos está aumentando drásticamente, es crucial desarrollar tecnologías efectivas y avanzadas para garantizar una autenticación exacta y en tiempo real en las ciudades inteligentes”(Al-Turjman et al., 2022).

“La confidencialidad es otro requisito para asegurar las ciudades inteligentes. Evita que la información sea objeto de una fuente equivocada o de ataques pasivos. En las redes IoT, los atacantes pueden acceder a los dispositivos y escuchar en secreto la comunicación. Por lo tanto, es importante realizar investigaciones sobre técnicas basadas en el cifrado para proteger la confidencialidad de la transmisión de datos entre nodos. Esto, a su vez, ayuda a tener un sistema de comunicación confiable”(Al-Turjman et al., 2022).

La Industria Manufacturera y el IoT

Se entiende por manufactura a todo proceso que tiene como fin convertir materia prima en bienes elaborados o semielaborados, que posteriormente son destinados al mercado.

Por otro lado, la industria manufacturera es toda industria que transforma materias primas o bienes semiprocesados directamente en bienes finales de consumo.

Tabla 2.

Tipos de manufactura por su posicionamiento en la cadena de producción económica.

Tipos de Manufactura	Principales Características	Entornos de Producción usuales	Categorías de Procesos usuales
Tradicional	Convierte la materia prima en bienes para su comercialización directa o indirecta.	MTO, ATO, MTS, ETO	Proceso de trabajo Por lotes Repetitivo Continuo
Intermedia	Convierte la materia prima en bienes semielaborados para alimentar otras industrias manufactureras.	MTO, ATO, MTS	Por lotes Repetitivo
Mecánica	Convierte la materia prima en	MTO, ATO, MTS	Por lotes Repetitivo

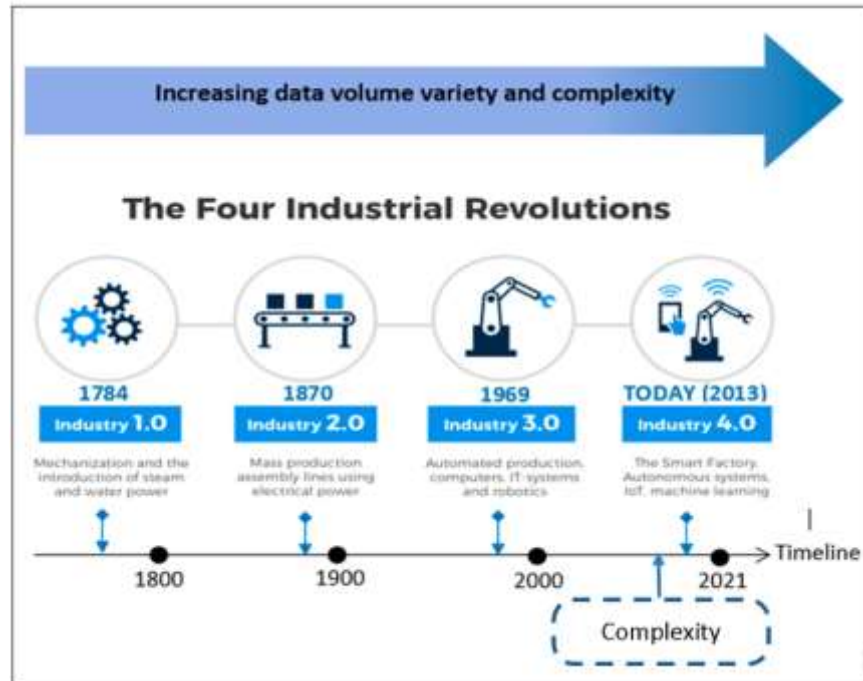
	herramientas y maquinarias necesarias para los procesos industriales de otros tipos, sean o no del sector secundario.		
Residual	Emplean como materia prima los sobrantes y residuos de otros procesos industriales.	ATO, MTS	Repetitivo Continuo

Nota. Cadena de producción de bienes de la economía. Fuente (Chapman, s. f.), (Industria Manufacturera - Concepto, Tipos, Ejemplos y Características, 2022). Elaborado por Autor.

Según (Abd Rahman et al., 2021). Las empresas de manufactura han adoptado ampliamente el enfoque de fabricación ajustada también conocida por sus siglas en inglés como lean manufacturing (LM). Desde la primera revolución industrial hasta la actualidad, la manera en la que se realizan los procesos productivos en las empresas ha ido cambiando, dando así denominaciones a cada etapa en la que sucedió de algún modo una revolución industrial como se puede visualizar en la **Figura 8**.

Figura 8.

Línea de tiempo hasta industria 4.0



Nota. Fusión de los sistemas físicos y digitales dentro de la evolución de la Industria 4.0, los sistemas de producción pueden actualizarse aplicando ambas nociones y elevando la excelencia operativa a un nuevo nivel. Fuente (Abd Rahman et al., 2021). Elaborado por (Abd Rahman et al., 2021).

La Industria 4.0 liderada por los datos asegura grandes avances en la producción.

El gran volumen de datos pertinentes de los sistemas de producción emplea servidores, sensores y computación en la nube, lamentablemente no existen sistemas estándar que gestionen los mismos en el contexto de la Industria 4.0. “Estudios previos realizados por investigadores en el mismo campo, demostró que no existía una plataforma estándar para gestionar las nuevas tecnologías en LM, análisis de datos e Internet de las cosas (IoT)”(Abd Rahman et al., 2021).

Datos de producción

Toda empresa independiente de su índole necesita medir su desempeño y esto se realiza en base a datos recolectados, las empresas de manufactura no son la excepción. En el punto de producción es primordial medir el tiempo que se demora en realizar una operación. Para esto el uso de tecnología de análisis, software de recopilación de datos, sensores y otras herramientas, permite a las industrias manufactureras obtener datos sobre una multitud de aspectos relativos a sus plantas y su empresa en general.

“La evaluación y contextualización de los datos contribuye a revelar información que ayuda a mejorar el proceso de control de la producción, aumentar la eficiencia, reducir costes, incrementar la calidad de los productos y mucho más” (*Recopilar Datos En Fábricas Mejora La Eficiencia | COPA-DATA*, s. f.).

Es esencial optimizar los procesos de producción, aumentar la productividad y garantizar la calidad del producto.

Para el logro de estos objetivos se requiere una base de datos precisa mediante; “El registro de datos de producción proporciona la base para la optimización de la producción, incluida la seguridad de la producción y el control de calidad. Algunos de los datos registrados incluyen datos de máquinas y procesos (como contadores, estado y resultados de medición), así como datos de energía. También se toman registros de control de calidad, como análisis de resultados de laboratorio o valores de contador fuera de línea. Los datos registrados incluyen información sobre los procedimientos de mantenimiento, como las horas de funcionamiento y los ciclos. Además, el registro de datos incluye entradas manuales, como valores de laboratorio e información adicional”(*Sistemas de Adquisición de Datos de Producción (PDA) Con Plant IT y Brewmaxx*, s. f.).

“Por la expresión, datos no son fiables datos no confiables, existe un gran debate puesto que, si bien

es cierto con malos datos malos resultados, pero también se puede alegar que, con datos, hay de dónde arrancar para mejorar. Si el operario no anota todo lo realizado, es malo y no sirve, pero si lo realiza esta bien cuando completa la planilla, esto no funciona en los procesos sostenibles en el tiempo, el operario cumple con el pedido durante un período corto y a partir de las siguientes comienza a fallar en el relleno de los datos, al cabo de un período más corto se puede tener un resultado igual de hipotético que si no se obtuviera datos”(Qué Importancia Tienen Los Datos En Su Producción, s. f.).

Antecedentes.

Descripción de la realidad

Royaltex S.A. es una empresa de manufactura ecuatoriana que produce uniformes a nivel nacional y producción de pantalones jeans como producto principal.

La empresa posee un prototipo de adquisición de datos desarrollado por (Cuyachamín, 2022) un egresado de la Universidad Tecnológica Indoamérica (UTI), el cual fue desarrollado en la placa popular Raspberrypi 3b+, con lenguaje Python, MySQL y php. Basa su lógica de programación en la validación de datos y envió de tiempos a la base de datos local hecha en MySQL. Para luego pasar a su visualización en una página web de servidor local el cual permite descargar los datos recolectados en formato .xls. El flujo se puede visualizar de mejor forma en **Figura 9**.

Figura 9.

Funcionamiento de primer sistema de adquisición de datos

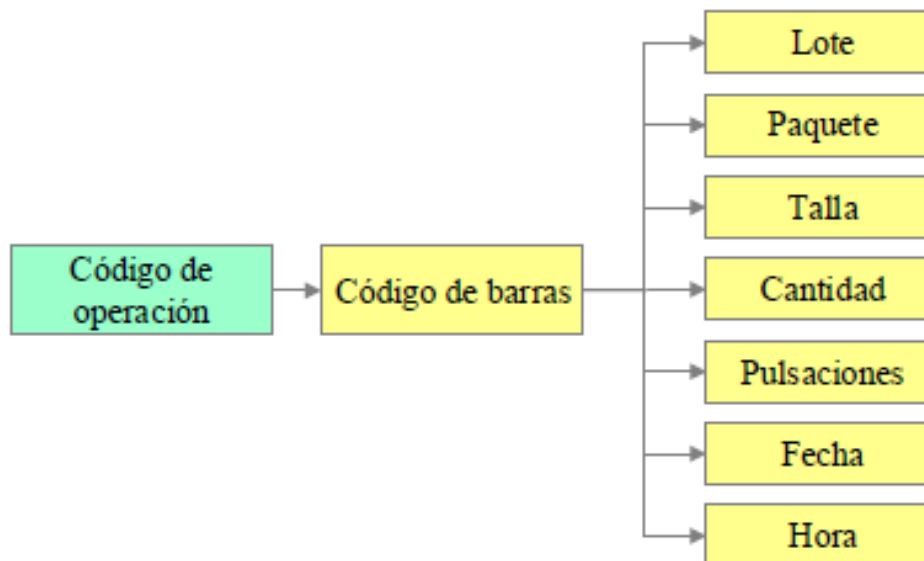


Nota. Fuente (Cuyachamín, 2022) . Elaborado por (Cuyachamín, 2022).

El flujo comienza con un emisor de señal el cual es presionado por el operario, luego esta señal es interpretada por la placa Raspberrypi 3b+ y da ejecución al programa desarrollado en la misma. Se puede identificar un problema con la lógica de programación puesto que el sistema de adquisición de datos debe ser autónomo y flexible en la validación de datos.

Figura 10.

Validación de datos en la programación del primer prototipo

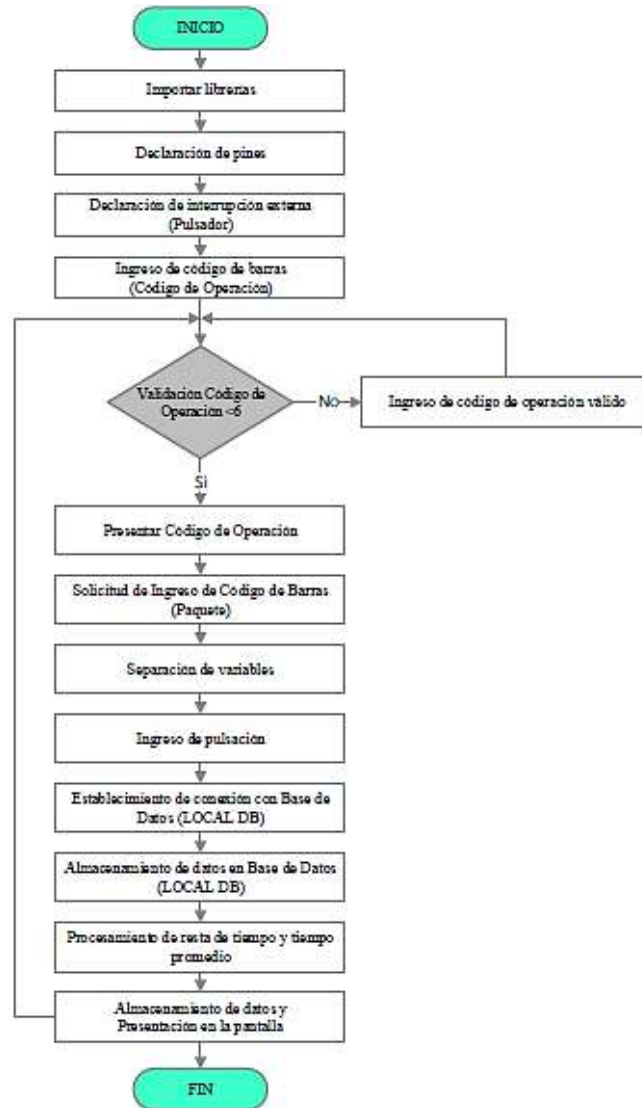


Nota. Validación de datos en la programación del prototipo inicial, para empezar un nuevo proceso es necesario reiniciar el ciclo. Tomado de (Cuyachamín, 2022) Elaborado por (Cuyachamín, 2022).

En la **Figura 11** se visualiza la lógica de programación del sistema desarrollado por (Cuyachamín, 2022) el cual es representado por un diagrama de flujo.

Figura 11.

Diagrama de flujo programación primer prototipo

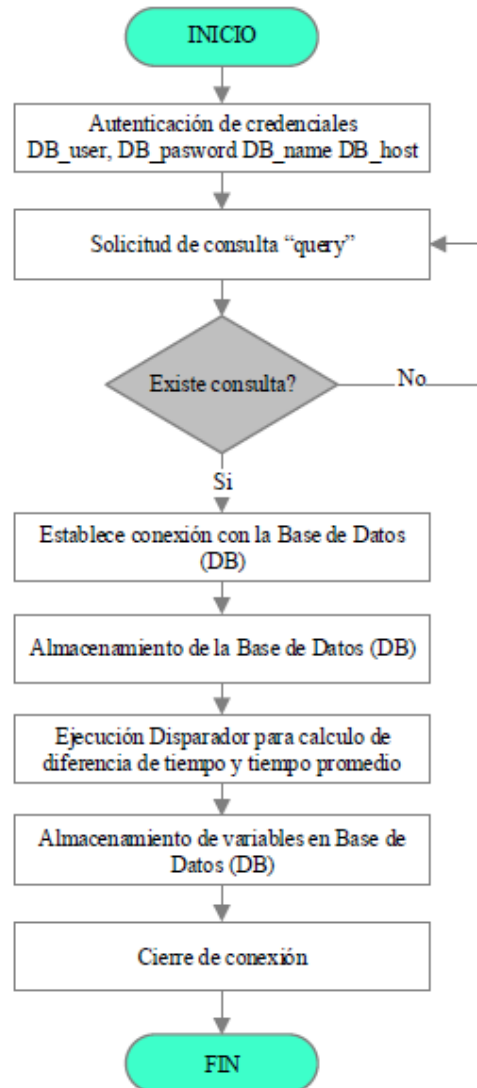


Nota. Estructura de lógica de programación desarrollada para el sistema de adquisición de datos que actualmente posee la empresa. Fuente (Cuyachamín, 2022). Elaborado por (Cuyachamín, 2022).

De la misma forma en la **Figura 12** y **Figura 13** se puede evidenciar un desaprovechamiento del uso de módulos (característica principal del lenguaje de programación Python) para que la estructura de la lógica sea flexible y no lineal.

Figura 12.

Diagrama de flujo programación primer prototipo base de datos en MySQL

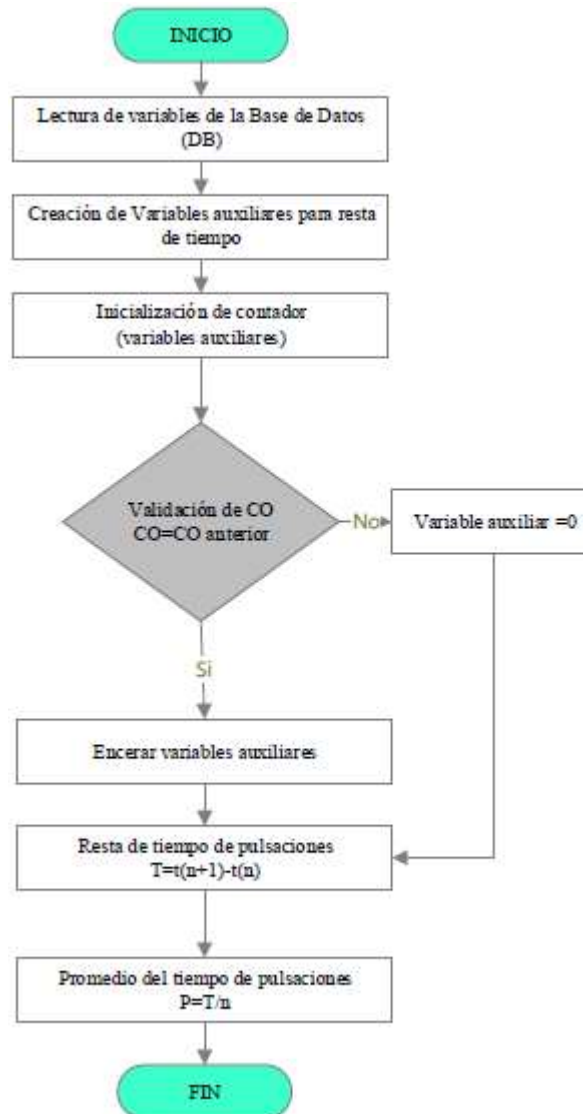


Nota. Fuente (Cuyachamín, 2022) Elaborado por (Cuyachamín, 2022).

Para la base de datos el programador optó por MySQL un lenguaje que permite almacenar y acceder a los datos a través de múltiples motores de programación.

Figura 13.

Diagrama de flujo, programación primer prototipo, disparador de datos (emisión de señal)



Nota. Fuente (Cuyachamín, 2022). Elaborado por (Cuyachamín, 2022).

El proceso de resta de tiempos es realizada en lenguaje MySQL con variables auxiliares, dicho proceso se lo podría realizar previamente ya en lenguaje Python con otra lógica de programación.

Finalmente, en la **Figura 14** se puede visualizar los datos recolectados por todo el primer sistema programado para posteriormente visualizar en una página web los resultados obtenidos.

Actualmente las variables que se necesitan en la empresa son más que las primeras en las que se basó este diseño, también existen errores en la denominación correcta de las variables, por ejemplo; código de barras representa a código de paquete y pulsaciones representa a número de pieza terminada.

Figura 14.

Datos de producción que registra el primer prototipo

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

[Exportar a Excel](#)

Código Operación	Código Barras	Lote	Paquete	Talla	Cantidad	Fecha	Pulsaciones	Hora	Tiempo	Promedio
359	28516123035	28516	12	30	35	2021-07-25	1	15:05:07	00:00:00	00:00:00
359	28516123035	28516	12	30	35	2021-07-25	2	15:05:29	00:00:22	00:00:11
359	28516123035	28516	12	30	35	2021-07-25	3	15:05:46	00:00:17	00:00:13
359	28516123035	28516	12	30	35	2021-07-25	4	15:06:01	00:00:15	00:00:13
280	28516122025	28516	12	20	28	2021-07-25	5	15:06:08	00:00:05	00:00:11

Nota. Las variables con los datos obtenidos del primer prototipo son código de operación, código barras, lote, paquete, talla, cantidad, fecha, pulsaciones, hora, tiempo, promedio. Fuente (Cuyachamín, 2022). Elaborado por (Cuyachamín, 2022).

A pesar de que la empresa de manufactura cuenta con el sistema de adquisición de datos (primer prototipo), este actualmente no está en uso, la recolección de datos de producción se sigue desarrollando de manera manual, el principal problema se puede deber a que el sistema inicial cumplía especificaciones piloto en su lógica de programación.

Justificación.

El presente trabajo es importante por los beneficios que traerá su desarrollo en las empresas de manufactura para la adquisición de datos, en especial en aquellas industrias manufactureras en las cuales predominan las labores de manera manual y mantener un control de producción es verdaderamente un reto con metodologías clásicas, sin la implementación de nuevas tecnologías disponibles.

El impacto que tendrá el presente trabajo es una mejora a la adquisición de datos en entornos de producción que predomine operación hombre-máquina, en especial aquellas industrias manufactureras que su distribución de planta es por células, aplicando sistemas tecnológicos de la información y comunicación para la adquisición de datos de producción. Permitiendo a los analistas desarrollar estrategias, toma de decisiones para los posibles eventos de su programa maestro de producción y minimizar costos de los mismos.

La utilidad que tendrá el sistema es la operatividad en la toma de tiempos de producción, ya que dado su entorno de la cual parte el diseño es fácil de usar y no interfiere en la suma de suplementos innecesarios al tiempo estándar de producción.

Los principales beneficiarios de este proyecto serán los analistas de datos, jefes de producción, departamentos que gestionen la trazabilidad de la producción y por último la empresa en general. Cabe recalcar que todo proyecto/análisis parte de datos, y el obtener datos de una manera más eficiente se traduce en mayor aprovechamiento de recursos en este caso el tiempo.

Según los estudios realizados por (Cuyachamín, 2022) demuestran que es factible el desarrollo de un sistema de integración de datos, **Figura 15** en un diagrama de control compara el tiempo promedio del operación obtenido por el sistema con el tiempo estándar de la operación proporcionado por la empresa, el cual muestra que no cruza el límite del umbral calculado, también en (Maldonado, 2022)

en el análisis de fiabilidad estadística de los datos desarrollado en 2022 muestra que solo existe una dispersión del 10% en lo cual lo hace un prototipo viable en la ejecución y toma de datos en una línea de producción con alta flexibilidad como lo es una estructura de célula de manufactura (LM). Esto hace viable la continuación y mejora del prototipo planteado inicialmente.

Figura 15.

Diagrama de control estadístico tiempo empleado vs tiempo estándar de operación de la empresa



Nota. Diagrama de control elaborado en la herramienta informática Minitab. Fuente (Cuyachamín, 2022). Elaborado por (Cuyachamín, 2022).

Tamaño de la muestra 128 datos, media de 70 segundos, desviación estándar de 1 segundo y promedio de 70 segundos (Cuyachamín, 2022).

Objetivos.

Objetivo general.

Diseñar un sistema de adquisición de tiempos de producción que integre los datos de varios puntos de una línea de manufactura, a una BD remota partiendo de un prototipo inicial, mediante la aplicación de IoT (internet de las cosas) para dar integración y disponibilidad en red de los datos adquiridos.

Objetivos Específicos.

1. Recopilar información del prototipo inicial y los requerimientos de las principales partes interesadas de la empresa, mediante una inspección cualitativa con herramientas de consulta, para determinar los atributos técnicos que posee el prototipo como las principales especificaciones demandadas.
2. Analizar los atributos técnicos obtenidos del prototipo inicial como las especificaciones demandadas por la empresa, mediante una comparación de priorización a través de una matriz QFD, para identificar las oportunidades de mejora a tratar en el diseño del nuevo prototipo.
3. Diseñar un sistema de adquisición de tiempos de producción, el cual implemente solución a las oportunidades de mejora detectadas; mediante reprogramación como cambio de arquitectura con ayuda de IoT, para optimizar código de programación, satisfacer las demandas de ejecución del sistema y permitir la visualización de datos en línea como los tiempos de producción.

CAPÍTULO II

INGENIERÍA DEL PROYECTO

Diagnóstico de la situación actual de la empresa.

La empresa actualmente realiza el registro de su producción manualmente, en una primera instancia y con la finalidad de validar la calidad de los datos, en (Cuyachamín, 2022) se obtiene una breve comparación entre los tiempos estándar de operación otorgado por la empresa con los tiempos tomados por el sistema **Figura 16**.

Pese a ello el obtener una validez estadística sobre la fiabilidad de los datos era necesaria, afortunadamente en (Maldonado, 2022) se destaca un análisis preciso sobre la validez de los datos obteniendo una fiabilidad de un máximo de 3 sigma y un mínimo de 1 sigma mediante una gráfica de línea ajustada **Figura 17**. Cabe recalcar que existe un único puesto de trabajo en la línea de producción en donde yace un prototipo del sistema, este pertenece al programa piloto para la implementación del mismo **Figura 18**.

Figura 16.

Tiempo de operación dispositivo vs Tiempo estándar de operación de la empresa



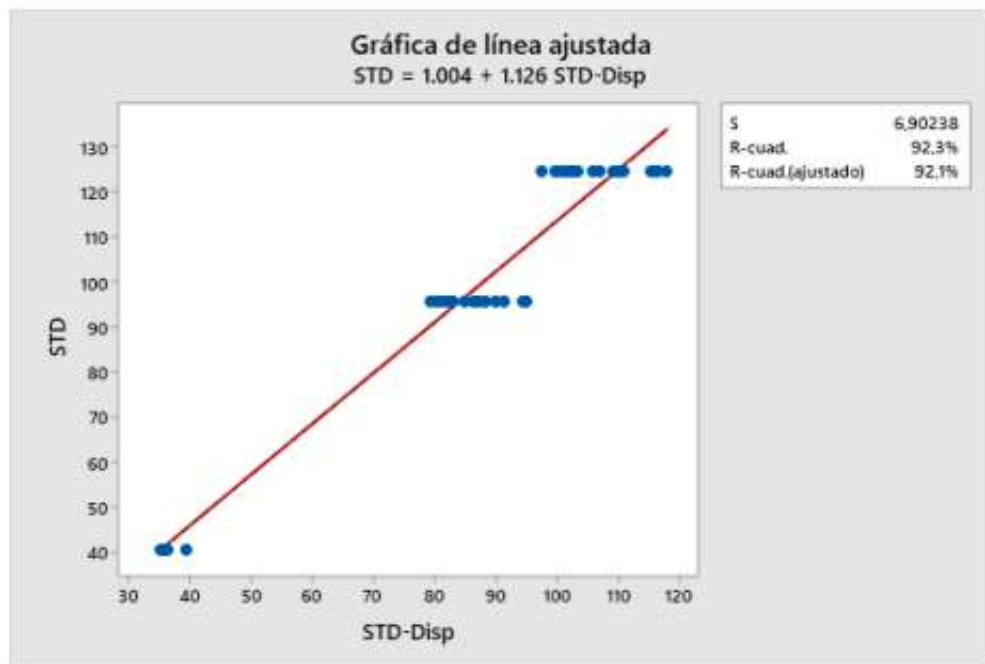
Nota. Fuente (Cuyachamín, 2022). Elaborado por (Cuyachamín, 2022).

Según la muestra realizada de 126 piezas en comparativa con el tiempo estándar analizado por la empresa, se puede observar que existen pocos datos que exceden este tiempo incluso los tiempos son

menores a los encontrados, lo cual se puede hipotetizar con una posible optimización al tiempo estándar de producción tomando en cuenta los suplementos respectivos necesarios en la operación y sin afectar los mismos. Para eliminar datos erróneos se realizó en (Maldonado, 2022) una validez de datos mediante línea ajustada.

Figura 17.

Línea ajustada y validación de datos a máximo 3 sigma



Nota. Fuente (Maldonado, 2022). Elaborado por (Maldonado, 2022)

Como se puede visualizar en una muestra de 40,57 de una población de 99 grupos y tras realizar un análisis de validación de datos en (Maldonado, 2022), se obtiene un coeficiente de correlación ajustado del 92.11% y un coeficiente de correlación cuadrado del 92.3% obteniendo una correlación del 99.96% entre tiempos estándar establecidos por la empresa y los analizados estadísticamente validando así la calidad y fiabilidad de los datos obtenidos por el dispositivo.

Figura 18.

Puesto piloto en línea de producción



Nota. La empresa destina el estudio de datos y aplicación del sistema como programa piloto a un puesto de trabajo de la línea de producción, en la cual se puede observar el prototipo del sistema instalado.

El problema del presente sistema no es su diseño más es que, está incompleto como se detalla en el capítulo uno. La problemática principal es la falta de más puntos de adquisición del sistema, en la línea de producción, puesto que un solo punto de recolección no permite la posibilidad de obtener una visualización general del estado de la línea de producción.

Al habilitar el sistema en varios puntos se puede obtener tener varias comparativas de la línea y de esta manera a futuro se pueda dar un tratamiento a esos datos de manera fácil, como un balance de línea, control, estado y en un futuro hasta predecir cuellos de botella en la producción.

Según (SEDANO ALBERTO, 2021). Las ventajas que presenta un balance de línea son innumerables, desde aumentar la productividad hasta distribuir la carga laboral e identificar problemas existentes en la línea.

A su vez también señala que el conocer el tiempo de ciclo permitirá realizar una reducción de los puestos de trabajo o ampliación, dependiendo de la capacidad instalada y demandada calculada.

El conocer los tiempos empleados en una línea de producción es vital y más si es ajustada como es típico en las distribuciones por células de manufactura. Pero entre las distribuciones fijas orientadas al producto o proceso y las orientadas a células son, que estas últimas son susceptibles a tener un mayor costo al control de sus tiempos y balance de producción, por la flexibilidad con la cual están diseñados.

Por lo cual el presente sistema al tener una habilitación estratégica en diferentes puntos de la línea, permitirá realizar la toma de estos tiempos y la oportunidad de realizar un sin número de aplicaciones.

Área de estudio.

Dominio: Tecnología y Sociedad

Línea de investigación: Análisis, diseño y desarrollo de sistemas inteligentes aplicables a la industria.

Campo: Ingeniería Industrial.

Área: Producción.

Aspectos: Sistema de adquisición de tiempos de producción aplicando IoT

Objeto de estudio: Sistema de adquisición de tiempos de producción para la integración y visualización de los datos a través de IoT.

Periodo de análisis: agosto 2022 a marzo 2022

Modelo operativo.

Como modelo operativo se toma como referencia el utilizado en (Cuyachamín, 2022) el mismo es actualizado y adecuado a los objetivos/necesidades del nuevo sistema a diseñar, obteniendo como resultado un nuevo modelo operativo. **Tabla 3.**

Tabla 3.

Modelo operativo seleccionado

MODELO OPERATIVO		
Fase 1	Fase 2	
Identificar	Solucionar	
Evaluación	Diseño	Programación
Recopilación de información.		
Análisis y evaluación de las necesidades		
Detección de problemas con el antiguo diseño	Selección de componentes	
	Diseño del prototipo	Programación del prototipo
		Ensamble del prototipo
		Pruebas de funcionamiento

Nota. Fuente adecuado de (Cuyachamín, 2022). Elaborado por el Autor.

Tabla 4.

Modelo operativo-primera fase

Identificar			
Modelo	Resultados a obtener		
Evaluación	Recopilación de información.	Análisis y evaluación de las necesidades	Detección de problemas con el antiguo diseño

Nota. Fuente Autor. Elaborado por Autor

La primera fase tiene como objetivo identificar las principales problemáticas de diseño, en el prototipo que posee la empresa, mediante una inspección cualitativa en la ejecución del dispositivo y su lógica de programación con su objetivo de uso, para en la siguiente fase proponer solución a las problemáticas encontradas.

Tabla 5.

Modelo operativo-segunda fase

Solucionar			
Modelo	Resultados a obtener		
Diseño	Selección de componentes	Diseño del prototipo	
Programación	Programación del prototipo	Ensamble del prototipo	Pruebas de funcionamiento

Nota. Fuente Autor. Elaborado por Autor

La segunda fase tiene como objetivo el dar soluciones a las oportunidades de mejora detectadas en la primera fase, diseñando un sistema de adquisición de tiempos de producción mediante la aplicación del Internet de las Cosas para disponer de datos en línea de los procesos de manufactura.

CAPÍTULO III

PROPUESTA Y RESULTADOS ESPERADOS

Desarrollo de la propuesta.

Tabla 6.

Desarrollo de la propuesta (primera fase)

Identificar			
Objetivo: Identificar las principales problemáticas de diseño en el prototipo que posee la empresa, mediante una inspección cualitativa con ponderación en la ejecución del dispositivo, lógica de programación con las especificaciones requeridas por el entorno, para en la siguiente fase proponer solución a las problemáticas encontradas.			
Modelo	Resultados a obtener		
Evaluación	Recopilación de información.	Análisis y evaluación de las necesidades	Detección de problemas

Nota. Las herramientas utilizadas para esta fase son: ficha descriptiva del dispositivo y análisis QFD.

Recopilación de información.

Tabla 7.

Ficha técnica descriptiva del dispositivo

Dispositivo	Sistema de comunicación para la adquisición de datos
Descripción física	El sistema de comunicación consta de un microcontrolador “raspberry pi 3b+” como punto central, donde se realiza el procesamiento/ejecución del sistema, consta de 2 periféricos de entrada y uno de salida, siendo un pulsador NC, y un lector de código de barras como entradas y una pantalla LCD de 3.5” como salida.

El lector de barra toma los datos iniciales, da inicio a un contador el cual mide el tiempo de operación para después enviar los datos tomados a una base de datos, el proceso es visible en el periférico de salida, pero además su diseño permite visualizar los resultados en una página web en la red LAN.

Modelo	Primer prototipo		
Marca		
Fabricante	Cuyachamín Chanaluisa Segundo Alex	Origen	Ecuador

CARACTERÍSTICAS GENERALES

Peso	45g	Altura	3.4cm	Ancho	7.59 cm	Largo	12.19cm
-------------	-----	---------------	-------	--------------	---------	--------------	---------

Imagen del dispositivo

Características técnicas

- Placa base Raspberry Pi 3 tipo B+
- Procesador Broadcom BCM2837B0
Cortex-A53 64-bit SoC @ 1,4 GHz
- RAM: 1 GB LPDDR2 SDRAM
- Conexiones inalámbricas: Bluetooth 4.2 BLE, Wi-Fi Dual Band b/g/n/ac
- Conexión por cable: Gigabit Ethernet
- Decodificación y codificación H.264 1080p30
 - Admite PoE (PoE HAT no suministrado)
 - Lector de tarjetas micro-SD
 - Header GPIO 40 pines
 - Admite boot PXE
- Puerto de cámara CSI para conectar la cámara Raspberry Pi
- Puerto de visualización DSI para conectar la pantalla táctil Raspberry Pi
 - 4 USB 2.0



- Compatible con los accesorios de Pi 3

Modelo B

- Alimentación 5V / 2,5 A
- Pantalla táctil de 3,5" electro resistivo.
- Lector código de barras Code 128, Code 39.



Función:

Sistema informático con lenguaje de alto nivel para toma de datos necesarios a saber en una línea de producción, donde los procesos se realizan en su mayor parte Hombre-Maquina.

Atributos:

Lógica de programación	Spaguetti, serie	estructura de visualización de datos	Pantalla del equipo y pagina web
lenguaje de programación	Python	procesamiento	Raspberrypi 3b+, MySQL.
periférico de entrada	Lector código de barras, pulsador NC	tipo de energía del sistema	Energía eléctrica
periférico de salida	Pantalla Lcd "3.5	almacenamiento	Local microSD
conectividad	Inalámbrica Wifi	tamaño de datos	255 bytes

Nota. Fuente (Cuyachamín, 2022). Elaborado por el Autor.

El prototipo existe solo en punto de la línea de producción, el mismo fue diseñado en base a las necesidades del momento, las cuales se detalla más a profundidad en (Cuyachamín, 2022).

Tabla 8.

Especificaciones demandadas por las partes interesadas

Partes interesadas	Principales especificaciones demandadas
Operador del prototipo	Eficacia en la adquisición de datos.
Jefe de mecánica	Eficiencia y ergonomía en la pulsación para envío de datos.
Jefe de ingeniería	Fiabilidad en los datos recolectados.
Investigador	Integración en más de un punto de la línea de producción. Visualización remota de los datos en cualquier parte del mundo. Interface gráfica amigable para el operario. Eficiencia operativa en la visualización de datos en la pantalla. Autonomía. Inicio en los últimos datos registrados. Plug and play (rápido y fácil de usar).

Nota. Fuente Royaltex. Elaborado por el Autor.

Dichas especificaciones fueron obtenidas mediante una entrevista al operador del prototipo, jefe de mecánica, jefe de ingeniería e investigador, a continuación, se procedió a dar una valoración de importancia a dichas especificaciones **Figura 19**.

Figura 19.

Requerimientos por las partes interesadas.

Requerimientos del sistema

El presente formulario esta diseñado para realizar una priorización a los requerimientos mencionados por las partes interesadas.

1. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
Que grado de importancia es para usted la Eficacia en la adquisición de datos en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

2. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
Que grado de importancia es para usted la Eficiencia y ergonomía en la pulsación para envío de datos, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

3. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.

Nota. Formulario obtenido a partir de las especificaciones requeridas por las partes interesadas. Fuente empresa de manufactura Royaltex. Elaborado por el Autor.

Análisis y evaluación de las necesidades.

Con el fin de cubrir las necesidades de la empresa y encontrar posibles mejoras en el prototipo inicial, se establece una Matriz de relación entre requerimientos del cliente, en este caso las principales demandas por las partes interesadas **Tabla 8** con los principales atributos del producto el sistema inicial **Tabla 7** para realizar una ponderación comparativa a través de una matriz QFD permitiendo de esta manera encontrar que atributo técnico del sistema inicial incide más en lograr el cumplimiento de dichas demandas.

Par lo cual se toma la calificación dada a cada especificación obtenidas de la encuesta de valoración hacia las partes interesadas.

Tabla 9.

Calificación de especificaciones por partes interesadas

Partes interesadas	Operador del prototipo	Jefe de mecánica	Jefe de ingeniería	Investigador	Total
Pregunta 1	5	5	5	5	5
Pregunta 2	5	4	3	4	4
Pregunta 3	5	5	5	5	5
Pregunta 4	5	5	5	5	5
Pregunta 5	3	2	3	4	3
Pregunta 6	1	2	3	2	2
Pregunta 7	3	2	3	4	3
Pregunta 8	5	4	3	4	4
Pregunta 9	5	5	5	5	5
Pregunta 10	5	5	5	5	5

Nota. Fuente Royaltex. Elaborado por el Autor

La ponderación obtenida refleja el grado de importancia con respecto a las especificaciones técnicas.

Figura 20.

Despliegue de la función de calidad

IMPORTANCIA PONDERADA DE LOS REQUISITOS TECNICOS		1,92	1,85	1,31	1,00	1,23	1,23	2,15	1,38	1,46	1,23	
Cómo?		Lógica de programación	Lenguaje de programación	Periférico de entrada	Periférico de salida	Conectividad	Estructura de Visualización de datos	Procesamiento	Tipo Energía del sistema	Almacenamiento de datos	Tamaño de dato	
Qué?		Importancia para el cliente [1-5]										
1. REQUERIMIENTOS DE LOS CLIENTES		1	2	3	4	5	6	7	8	9	10	
1	Eficacia en la adquisición de datos	5	5	3	5	1	1	1	3	2	4	4
2	Eficiencia y ergonomía en la pulsación para envío de datos	4	4	3	5	1	5	1	3	3	1	1
3	Fiabilidad en los datos recolectados	5	5	5	5	2	5	2	4	2	5	3
4	Integración en mas de un punto de la linea de producción	5	5	5	3	3	5	5	5	2	5	5
5	Visualización remota de los datos en cualquier parte del mundo	3	5	3	1	1	5	5	4	3	5	5
6	Interface gráfica amigable para el operario	2	4	5	1	5	3	3	3	3	3	3
7	Eficiencia operativa en la visualización de datos en la pantalla	3	4	3	3	5	2	2	4	2	2	2
8	Autonomía	4	5	4	5	4	5	5	5	4	5	4
9	Inicio en los ultimos datos registrados	5	5	3	1	4	2	2	4	2	5	5
10	plug and play (rapido y facil de usar)	5	5	4	4	4	5	4	4	3	5	4
EVALUACION DE IMPORTANCIA	ABSOLUTA	196	156	144	118	157	121	162	104	171	152	
	RELATIVA	2	2	1	1	2	1	2	1	2	1	
EVALUACIÓN DE INGENIERÍA		377	288	188	118	193	149	349	144	250	187	
		3	2	2	1	2	1	3	1	2	2	

Nota. Fuente Royaltex. Elaborado por el Autor.

La evaluación ponderada de los requisitos del cliente respecto a los del producto se da gracias a la suma de los productos de las ponderaciones de las matrices involucradas, importancia de especificaciones para el cliente, importancia entre atributos técnicos y relación de las dos matrices anteriormente mencionadas. Dichos resultados se pueden visualizar de una mejor forma en la Tabla 10.

Tabla 10.

Resultados del análisis QFD

Atributos del dispositivo	Lógica de programación	Lenguaje de programación	Periférico de entrada	Periférico de salida	Conectividad	Estructura de Visualización de datos	Procesamiento	Tipo Energía del sistema	Almacenamiento de datos	Tamaño de dato
Evaluación de importancia Absoluta	377	288	188	118	193	149	349	144	250	187
Evaluación de importancia Relativa	3	2	2	1	2	1	3	1	2	2

Nota. Fuente empresa de manufactura Royaltex. Elaborado por el Autor

De todos los atributos presentes en el sistema de comunicación para la adquisición de datos, los que presentan un mayor índice directo para el cumplimiento de las especificaciones son la lógica de programación como primer lugar y en segundo el procesamiento del sistema. El resto de los atributos se puede tomar como secundarios como orden prioritario.

Detección de problemas.

Tabla 11.

Orden de problemas de acuerdo a la priorización calculada

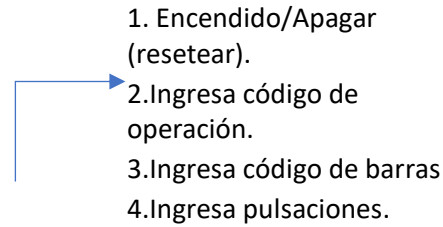
Lógica de programación	3
Procesamiento	3
Lenguaje de programación	2
Almacenamiento de datos	2
Conectividad	2
Periférico de entrada	2
Tamaño de dato	2
Estructura de Visualización de datos	1
Tipo Energía del sistema	1
Periférico de salida	1

Nota. Fuente empresa de manufactura Royaltex. Elaborado por el Autor.

Para encontrar las causas que provocan que la lógica de programación no cumpla con las especificaciones demandadas, es necesario evaluar a través de un pseudocódigo el flujo que sigue la lógica programada y determinar si es posible reciclar código al momento de proponer la nueva lógica de programación. El procesamiento independiente de los componentes del nuevo sistema también será reestructurado de acuerdo a la capacidad del hardware y el uso que se le dé al mismo.

Tabla 12.

Seudocódigo del sistema inicial



Nota. La flecha indica un bucle presente. Fuente (Cuyachamín, 2022). Elaborado por el Autor.

Se puede ver que la lógica de programación que sigue el sistema es limitada y no da la posibilidad de ingresar más variables, como el inicio en los últimos datos registrados o la autonomía que demanda las especificaciones. Por lo tanto, se puede concluir en que es necesario proponer una nueva lógica de programación en la cual solviente las debilidades del primer prototipo, respetando en lo posible las herramientas utilizadas iniciales.

Tabla 13.

Oportunidades de mejora

Atributos influyentes	Oportunidades de mejora
Lógica de programación	Gestión del flujo de información Entorno de programación. Inclusión de IoT
Procesamiento	Operatividad de los dispositivos. Integrar datos en un solo punto. Datos a la espera de ser procesados/analizados. Mejor periférico de entrada

Nota. Los atributos influyentes generales presentan oportunidades de mejora. Fuente Autor. Elaborado por Autor.

Tabla 14.

Desarrollo de la propuesta (segunda fase)

Solucionar			
Objetivo: Proponer una solución a las oportunidades de mejora detectadas en la primera fase, mediante la aplicación de herramientas de elección para cumplir con las especificaciones para la cual es requerido el sistema.			
Modelo	Resultados a obtener		
Diseño	Selección de componentes	Diseño del prototipo	
Programación	Programación del prototipo	Ensamble del prototipo	Pruebas de funcionamiento

Nota. Las herramientas utilizadas para esta fase son: clasificación por Hardware y software, recomendaciones, diagrama de flujo, pseudocódigo, programación orientada a objetos.

Selección de componentes.

Con el fin de mantener la trazabilidad del proyecto, la selección de los componentes para el nuevo sistema se mantiene con los predecesores, puesto que el objetivo de optimizar hardware no es una prioridad en el momento, los mismos fueron obtenidos con ponderación en (Cuyachamín, 2022).

Se aplica la recomendación de cambio al periférico de entrada de pulsaciones por un sensor infrarrojo dado por (Maldonado, 2022).

Se obtiene una tabla con los componentes a usar para el sistema. **Tabla 15.**

Tabla 15.

Componentes del sistema

Hardware			
Componente		Clase	Función
Microcontrolador	Raspberry pi 3B	CPU	Procesamiento del sistema
Lector código de barras		Periférico de entrada	Lectura e ingreso de datos de producto
Sensor infrarrojo	EB0043059	Periférico de entrada	Registro de tiempos y datos de producción
Pantalla LCD de 3.5"	Raspberry pi	Periférico de salida	Visualización para operario
Fuente de energía	Raspberry pi	Alimentación	Transformación de voltaje y carga apta para el sistema
Carcasa de plástico	Raspberry pi	Protección externa	Protección de CPU
Software			
Sistema operativo	Linux	Gestionador de recursos	Gestionador de los recursos de hardware y proveedor de servicios a los programas de aplicación de software.
Entorno de programación	- PYTHON - MySQL - PHP - HTML	Lenguaje de programación	Englobar todas las tareas necesarias para el funcionamiento del programa, codificar, generar, depurar, actualizar, integrar, testear, validar y ejecutar.

Nota. Fuente (Cuyachamín, 2022). Elaborado por. Autor

Los lenguajes que conforman el entorno de programación se comunican en las capas del IoT, siendo Python el lenguaje integrador y MySQL el gestor en la capa de base de datos. Por otro lado,

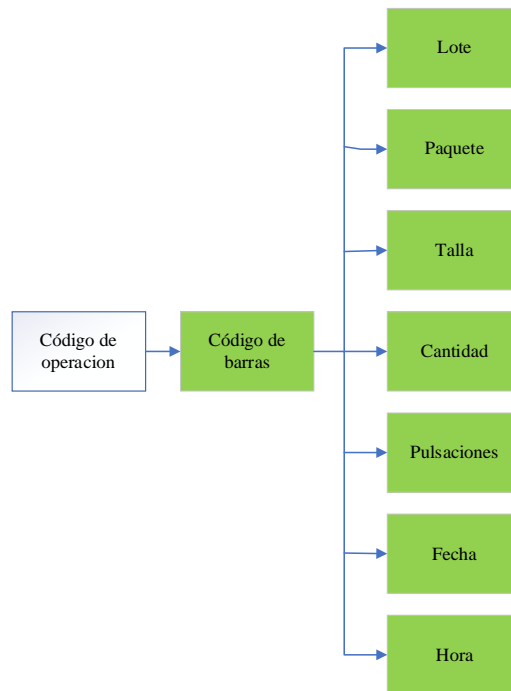
PHP y HTML actúan en la capa de aplicación.

Diseño del prototipo.

Como punto de partida se toma el flujo de datos del primer diseño, esto con el fin de determinar la manera en que el sistema maneja los datos que recolecta **Figura 21.**

Figura 21.

Flujo de datos del sistema anterior



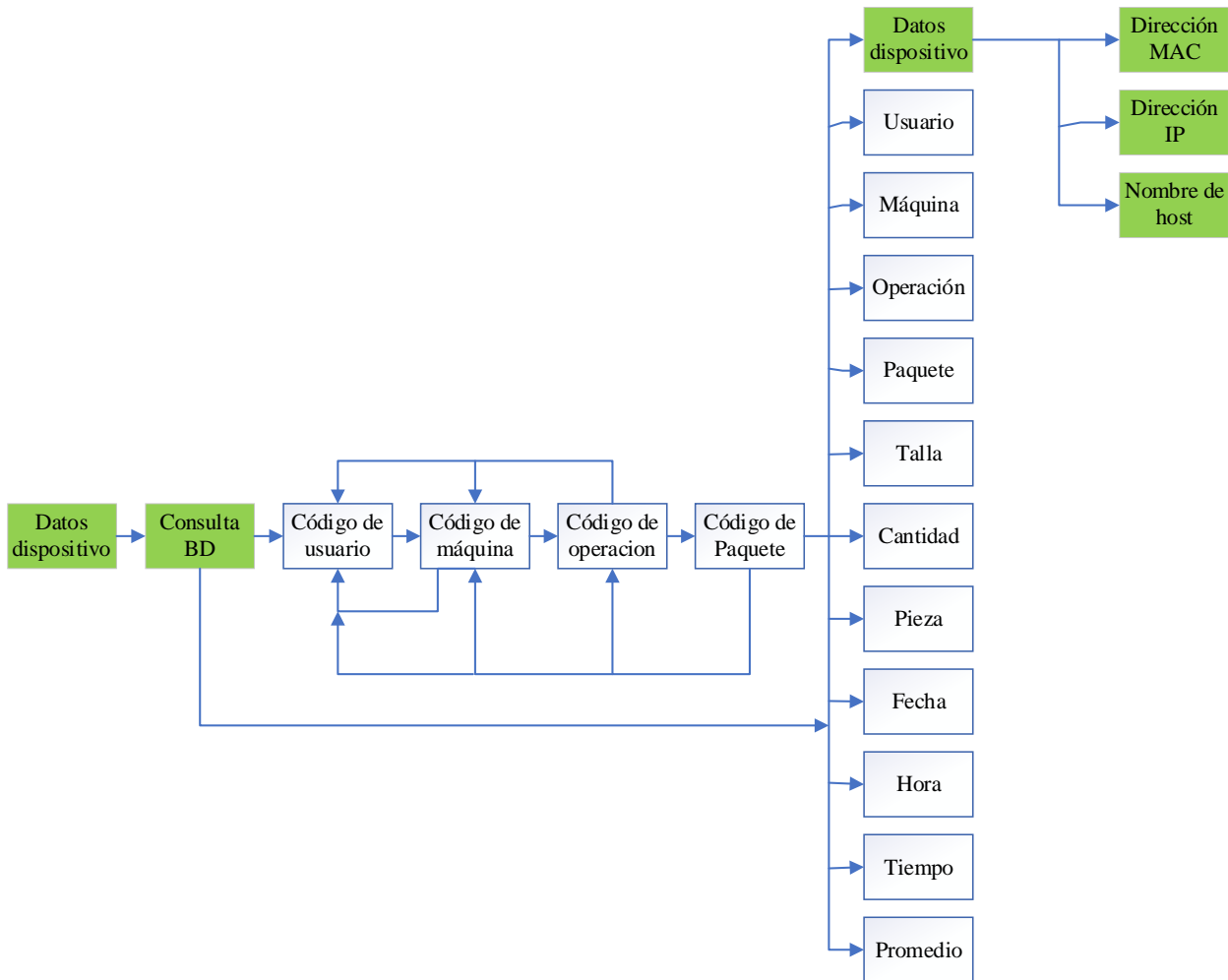
Nota. Fuente (Cuyachamín, 2022). Elaborado por el Autor

Al igual que la lógica de programación, el flujo que mantienen los datos es lineal por lo cual presenta la desventaja de la necesidad repetir el proceso por completo para cambiar los datos, lo cual operativamente ocasiona problemas tomando en cuenta que son actividades repetitivas. También se puede observar la ausencia de variables de identificación de dispositivo máquina en la que se realiza la operación y operario, variables necesarias para seguimiento de calidad.

Por lo cual se propone un flujo dinámico de información. **Figura 22.**

Figura 22.

Flujo de datos para el nuevo sistema propuesto

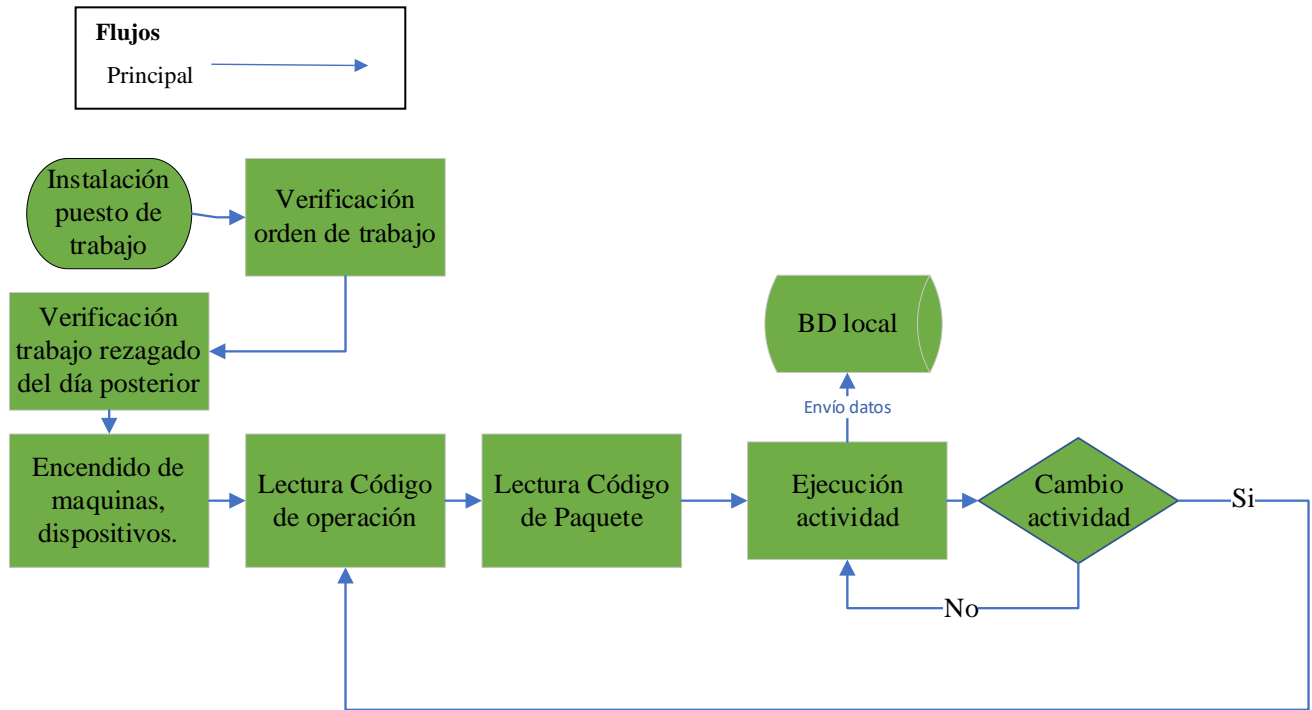


Nota. Fuente empresa de manufactura Royaltex. Elaborado por Autor.

En comparación con el antiguo flujo de datos el presente propuesto pretende solucionar los problemas operativos anteriormente mencionados, al hacer el sistema adaptable a las necesidades de cambio del operario como una conexión directa a la base de datos da la continuidad del proceso ante una repentina pausa.

Figura 23.

Flujo de trabajo anterior en un puesto de la línea de producción



Nota. Fuente Royaltex. Elaborado por el Autor.

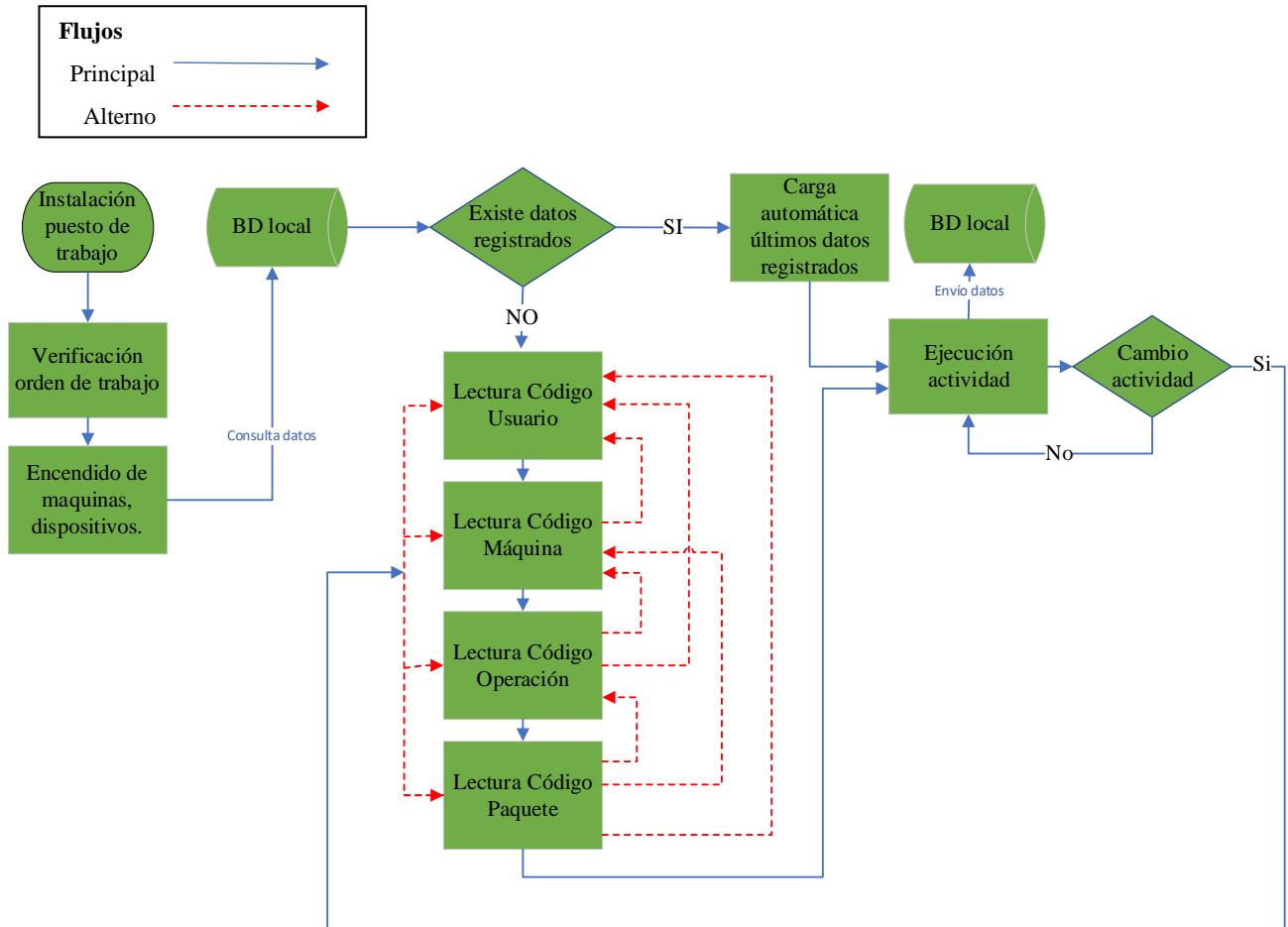
El operario inicia con la instalación en su puesto de trabajo para después encender la maquinaria necesaria y dispositivos a ocupar, una vez realizado ello procede a leer el código de operación seguido por el código de paquete. Registrados estos códigos el operario procede a realizar la actividad una vez terminada la pieza del paquete, luego envía el tiempo tomado en la ejecución de dicha operación para después verificar si existe más piezas por realizar, si el operario decide cambiar de actividad obligatoriamente necesita ingresar todos los datos anteriormente mencionados; código de operación y paquete para poder realizar su nueva actividad.

Esto significa un problema operativo, ya que el sistema no presenta un control adecuado para identificar en qué pieza se encuentra el trabajador ya que, al cambiar la actividad necesariamente debe ingresar todos los datos requeridos nuevamente.

Se propone una mejora al flujo de trabajo **Figura 24**.

Figura 24.

Nuevo flujo de trabajo propuesto con el sistema en la línea de producción



Nota. Fuente Autor elaborado por Autor.

A diferencia del anterior flujo de trabajo, el propuesto evita al operario la verificación del trabajo rezagado del día anterior, puesto que el sistema lo verifica automáticamente, dando al operario los datos de la última pieza registrada en el sistema y permitiéndole dar continuidad si el paquete ingresado quedo inconcluso.

El operario inicia con la instalación en su puesto de trabajo para después verificar la orden de trabajo entregada en el día, enciende la máquina y dispositivos a usar, el nuevo sistema realiza la

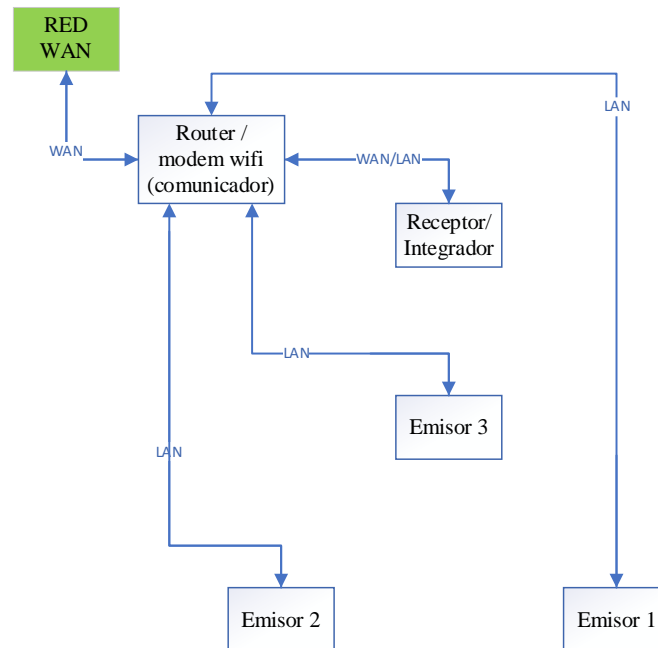
consulta a la base de datos y pregunta si existen datos registrados, si la respuesta es afirmativa carga los últimos datos informando al operario la última pieza en la que se quedó, si el operario decide cambiar la actividad únicamente cambia los datos que el necesite modificar. Cualquier flujo alterno es válido puesto que el operario puede modificar según así lo requiera.

De esta manera la inclusión del sistema de adquisición de tiempos evita interponerse en la operatividad del usuario.

Como medio, el sistema se comunica entre sus equipos a través de internet por medio de ondas electromagnéticas por tecnología wifi evitando cableado en la implementación de IoT siendo, WLAN o LAN como intranet de la empresa (local) y WAN como red de área amplia (pública). Los equipos denominados como emisores se encargan de todo el procedimiento antes mencionado, la ejecución, cálculo y registro de datos.

Figura 25.

Flujo de señales en el medio



Nota. Fuente (Al-Turjman et al., 2022). Elaborado por el Autor.

El equipo denominado como receptor/integrador cumple la función de compilador de datos para después, dar una posterior visualización a través del internet de manera remota en cualquier parte del mundo.

Programación del prototipo.

Figura 26.

Estructura de programación

Arquitectura de programación	Por Capas
Tipo de programación	Orientada a objetos

Nota. Fuente (Al-Turjman et al., 2022). Elaborado por el Autor.

Por la estructura que maneja el IoT, se toma como arquitectura de programación del sistema por capas y una programación orientada a objetos para los equipos que conforman el mismo, esto ayudara a utilizar en lo posible el código antiguo que cumpla las especificaciones planteadas para el presente diseño, además de la posibilidad de realizar cambios a futuro en el código sin que afecte a todo el sistema.

Tabla 16.

Arquitectura de programación

Capa	Descripción	Lenguaje de programación	Aplicación
Capa de presentación	Se especializa en operaciones de interacción con el usuario mediante pantalla, teclado o	PHP, HTML	Pantalla

mouse			
Capa de reglas	Se especializa en operaciones de procesamiento de datos, por ejemplo, mediante fórmulas de cálculo	PYTHON	Memoria
Capa de datos	Se especializa en operaciones con los medios de almacenamiento como el disco duro	MySQL	Medio de almacenamiento

Nota. Fuente (Al-Turjman et al., 2022). Elaborado por el Autor.

La arquitectura por capas hace énfasis a la estratificación de funciones a realizar por especialización, algunos lenguajes son más efectivos para unas funciones que otros y según el IoT que también maneja una arquitectura por capas hace de esto una compatibilidad buena entre el sistema y el medio.

Tabla 17.

Tipo de programación (orientada a objetos).

Clase	Objeto	Atributos	Métodos
Sistema de adquisición de tiempos de producción	Emisor	Adquisición	- Código de usuario. -Código de máquina. -Código de operación. -Código de paquete.

		<ul style="list-style-type: none"> -Conexión BD local. -Extractor lote, paquete, talla, cantidad del código de Paquete. -Toma ultimo datos registrados en BD local. -Validación en encendido. -Ejecución cronometro. -Validación cambio de paquete.
	Integración	<ul style="list-style-type: none"> -Conexión BD local. -Conexión BD madre. -Migración de datos.
	Visualización	-Visualización de datos Pagina web.
Receptor	Visualización	-Visualización de datos Pagina web.

Nota. Fuente (Al-Turjman et al., 2022). Elaborado por Autor.

La programación orientada a objetos busca la facilidad de organización de código clasificándolo; por clase, objeto, atributos y métodos. Este tipo de programación es muy útil puesto a que ayuda a evitar reescribir código y ordenar de mejor manera lo que se da o no al programa para que ejecute.

En este caso el sistema de adquisición de tiempos de producción es la clase y los objetos se dividen por el equipo emisor y el equipo receptor.

Como siguiente paso a realizar, se establece la elaboración de los pseudocódigos de los métodos a programar, esto es clave puesto que es la elaboración de la lógica de programación antes de pasar a un lenguaje de programación de alto nivel, en el caso de la presente Python, pero es un paso a seguir previo a la elaboración de la programación propiamente dicha.

Programación Emisor

Tabla 18.

Seudocódigo del equipo emisor – programación orientada a objetos, adquisición e integración.

Main

1. Encendido.
 2. Importa librerías necesarias “datetime, socket, getmac”.
 3. Encera variables “MAC, IP, HostName”.
 4. Toma datos de dispositivo en las variables con el mismo nombre “MAC, IP, HostName”.
 5. Imprime en pantalla los datos.
 6. Almacena en atributos objeto “ObjRaspberry”.
 7. Ejecuta método “Toma ultimo datos registrados en BD local”.
-

Toma ultimo datos registrados en BD local

1. Importa librerías necesarias “pymysql, time, threading”.
 2. Recoge últimos datos registrados en la BD local.
 3. Ejecuta método Migración de datos.
 4. Ejecuta método, Validación en encendido.
-

Validación en encendido

1. Define variable global “contador”.
 2. Encera variable.
 3. Define variable “contador” es igual a “pieza” más 1.
 4. Imprime último paquete registrado.
 5. Imprime última pieza registrada.
 6. Imprime variable “contador”.
 7. Ejecuta método, Ejecución cronómetro.
-

Migración de datos

1. Encera variable “conexión”.
2. Trata de establecer conexión a BD madre: *si* valida último dato enviado con ultimo registrado y envía diferencia.
3. Ejecuta método “Conexión BD madre”.
4. *No* repite hasta obtener conexión a BD madre.

Ejecución cronometro

1. Importa librerías necesarias “RPi.GPIO, pymysql, time, threading”.
2. Declara variable global “CodigoNuevo, Sensor”.
3. Define variable “sensor” a pin 32 de GPIO
4. Configura GPIO con el modelo BOARD
5. Declara GPIO 32 como entrada.
6. Recoge fecha inicio y hora inicio.
7. Imprime fecha inicio y hora inicio.
8. Define función paro por programación multihilo.
9. Pide ingresar variable por teclado “CodigoNuevo”.
10. Inicia bucle, mientras “CodigoNuevo” no este definido ejecuta sentencia de envío de datos
11. Crea query para inserto de datos en BD local
12. Ejecuta método “Conexión BD local”
13. *Si* rompe el bucle ejecuta sentencia validación de código ingresado y ejecuta el método de validación acorde.
14. *No* ejecuta el método “Ejecución cronometro” nuevamente.

Métodos de validación

Código de usuario

1. Declara variable global “Usuario”.
 2. Pide ingreso del dato y lo almacena en la variable antes declarada.
 3. Extrae primer carácter de usuario y lo almacena en “firstValidacion”.
 4. Extrae la longitud del string y lo almacena en “secondValidacion”.
 5. Sentencia ejecuta solo *si* ‘firstValidacion= "E"’ and ‘secondValidacion == "6"’ Y ejecuta método “Código de máquina”.
-

6.No imprime en pantalla “Ingresa un Código de Usuario Valido”. Y ejecuta método “Código de usuario”.

Código de máquina

- 1.Declara variable global “Maquina”.
 - 2.Pide ingreso del dato y lo almacena en la variable antes declarada.
 3. Extrae primer carácter de “Maquina” y lo almacena en “firstValidacion”.
 4. Extrae la longitud del string y lo almacena en “secondValidacion”.
 5. Sentencia ejecuta solo *si* ‘firstValidacion= "E"’ and ‘secondValidacion == "6"’
 - 6.Establece variable global “Usuario”
 - 7.Usuario es igual a máquina. Y ejecuta método “Código de máquina”.
 8. Sentencia ejecuta solo *si* firstValidacion== "M" and secondValidacion == 6. Y ejecuta método “Código de operación”.
 - 6.No imprime en pantalla “Ingresa un Código de Maquina Valido”. Y ejecuta método “Código de máquina”.
-

Código de operación.

- 1.Declara variable global “Operación”.
 - 2.Pide ingreso del dato y lo almacena en la variable antes declarada.
 3. Extrae primer carácter de “Operación” y lo almacena en “firstValidacion”.
 4. Extrae la longitud del string y lo almacena en “secondValidacion”.
 5. Sentencia ejecuta solo *si* ‘firstValidacion= "E"’ and ‘secondValidacion == "6"’
 - 6.Establece variable global “Usuario”
 - 7.Usuario es igual a máquina. Y ejecuta método “Código de máquina”.
 8. Sentencia ejecuta solo *si* firstValidacion== "M" and secondValidacion == 6. Y ejecuta método “Código de operación”.
 - 9.Sentencia ejecuta método “Código de paquete” solo *si* secondValidacion == 5
 - 6.No imprime en pantalla “Ingresa un Código de operación Valido”. Y ejecuta método “Código de operación”.
-

Código de paquete

- 1.Declara variable global “Operación”.
 - 2.Pide ingreso del dato y lo almacena en la variable antes declarada.
 3. Extrae primer carácter de “Operación” y lo almacena en “firstValidacion”.
-

-
4. Extrae la longitud del string y lo almacena en "secondValidacion".
 5. Busca si existe "," en el string y lo almacena en la variable "thirdValidacion".
 6. Sentencia ejecuta solo si 'firstValidacion= "E"' and 'secondValidacion == "6"'
 7. Establece variable global "Usuario".
 8. Usuario es igual a máquina. Y ejecuta método "Código de máquina".
 9. Sentencia ejecuta solo si firstValidacion== "M" and secondValidacion == 6. Y ejecuta método "Código de operación".
 10. Sentencia ejecuta método "Extractor lote, paquete, talla, cantidad del código de Paquete" y "Validación cambio de paquete" solo si "thirdValidacion" es diferente a -1.
 11. Sentencia ejecuta método "Código de paquete" solo si secondValidacion == 5.
 12. No imprime en pantalla "Ingresa un Código de operación Valido". Y ejecuta método "Código de operación".

Extractor lote, paquete, talla, cantidad del código de paquete

1. Declara variable global "lote, paquete, talla, cant".
2. Realiza explode de datos y los almacena en ese orden.

Validación cambio de paquete

1. Declara variable global "contador".
2. Define variable "contador" como 1
3. imprime en pantalla '<<Pieza #', contador,">>'
4. Ejecuta método "Ejecución cronometro"

Conexión BD local

1. Importar librerías necesarias
2. Trata de conectar con credenciales dadas "DB_HOST, DB_USER, DB_PASS, DB_NAME".
3. Ejecuta query de entrada
4. verifica si la transacción es exitosa
5. Si. cierra conexión.
6. No, realiza rollback a como estaba al inicio.

Conexión BD madre.

1. Importar librerías necesarias
 2. Trata de conectar con credenciales dadas "DB_HOST, DB_USER, DB_PASS, DB_NAME".
 3. Ejecuta query
-

-
4. verifica si la transacción es exitosa
 5. Si. cierra conexión.
 6. No, realiza rollback a como estaba al inicio.
-

Nota. Fuente Autor. Elaborado por el Autor.

La relación que mantienen los métodos hace del objeto una estructura muy dinámica en su programación, como se puede observar en los pseudocódigos se estructura el atributo por métodos y se hace el llamado al método necesitado.

Tabla 19.

Seudocódigo emisor – programación orientada a objetos, visualización web.

Consulta BD

1. Define variables para conexión a BD local.
 2. Define consulta en código SQL y ordénalo por id descendente
 3. cierra conexión
-

Definición extensión .xls para descarga de datos

1. Establece encabezados para navegador
 2. Define encabezados para que navegador interprete que descargará un archivo de Excel.
-

Construcción de interface grafica web

1. Construye una tabla HTML formato A4.
 2. Inserta figura de cronometro
 3. Establece encabezados Mac, Ip, Hostname, Usuario, Maquina, Operación, lote, paquete, talla, cantidad, pulsaciones, fecha, hora, tiempo, promedio.
 4. establece filas para inserto.
-

Inserto de datos consultados en tabla para visualización

1. Establece relación datos consultados con encabezados creados.
 2. Inserta información
-

Nota. Fuente (Cuyachamín, 2022). Elaborado por el Autor

Tabla 20.

Programación autoarranque del programa en linux

Arranque del programa automático

1. Encendido dispositivo
 2. Ejecuta script de arranque en la carpeta autostart (este hace un llamado a la ejecución del programa al iniciar el sistema operativo)
-

Nota. Fuente Autor. Elaborado por el Autor.

Con el fin de respetar la arquitectura planteada los atributos de adquisición e integración del objeto emisor se desarrolla en un script en lenguaje Python, el cual se puede visualizar el orden en el pseudocódigo **Tabla 18**.

Se separa el atributo de visualización web en otro script **Tabla 19** dicho script se realiza en lenguaje php, ya que su protocolo de comunicación es http (el script se ejecuta automáticamente al realizar un llamado por la red Lan).

Finalmente se realiza la programación en el sistema operativo de linux con la ejecución automática del script con los atributos adquisición e integración **Tabla 20**.

Programación Receptor

Tabla 21.

Seudocódigo receptor – programación orientada a objetos, visualización web.

Consulta BD

1. Define variables para conexión a BD local.
 2. Define consulta en código SQL y ordénalo por id descendentemente
 3. cierra conexión
-

Definición extensión .xls para descarga de datos

1. Establece encabezados para navegador
 2. Define encabezados para que navegador interprete que descargará un archivo de Excel.
-

Construcción de interface grafica web

1. Contruye una tabla HTML formato A4.
 2. Inserta figura de cronometro
 3. Establece encabezados Mac, Ip, Hostname, Usuario, Maquina, Operación, lote, paquete, talla, cantidad, pulsaciones, fecha, hora, tiempo, promedio.
 4. Establece filas para inserto.
-

Inserto de datos consultados en tabla para visualización

1. Establece relación datos consultados con encabezados creados.
 2. Inserta información
-

Nota. Fuente Autor. Elaborado por el Autor.

Tabla 22.

Programación en Linux

Registro/Enlace Dominio automático DNS

1. Encendido dispositivo

2. Ejecuta crontab, enlace automático IP publica a dominio establecido datosp.freemyip.com (es necesario realiza un redireccionamiento al puerto 80 del router para que atienda la llamada de la red)

Nota. Fuente Autor. Elaborado por el Autor.

Figura 27.

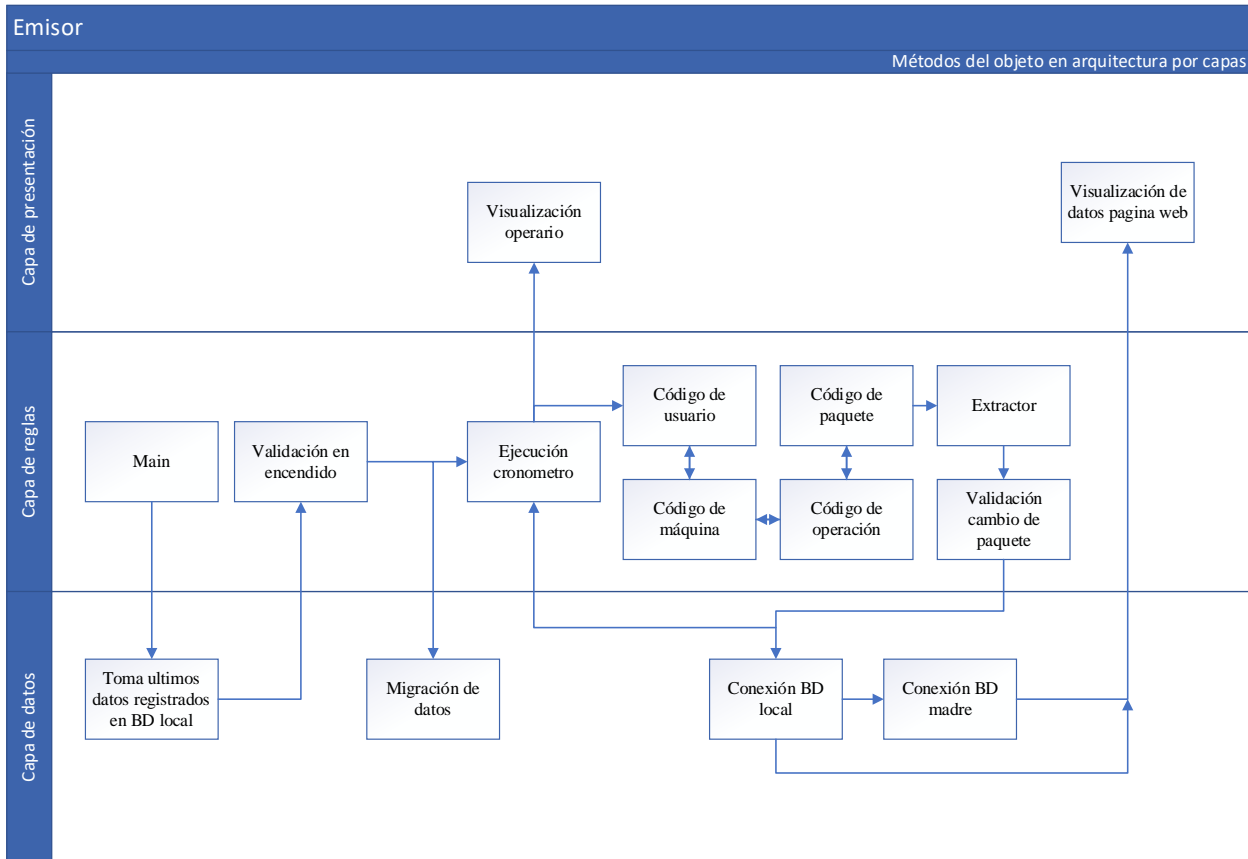
Visualización remota a través de IoT.



Nota. Fuente Autor. Elaborado por Autor

Figura 28.

Visualización final, programación orientada a objetos en la arquitectura por capas del sistema propuesto.



Nota. Fuente Autor. Elaborado por Autor

En este diagrama de recorrido, se puede apreciar de mejor manera como está estructurada la programación con la arquitectura inicial planteada con la programación orientada a objetos. La relación que mantienen los métodos es lineal, mencionando que el método migración y el método cambio se realizan por programación multihilo una característica propia de Python que permite ejecutar simultáneamente un método independientemente si se está ejecutando otro método.

Ensamble del prototipo.

Figura 29.

Equipos de sistema (emisores).



Nota. Fuente Autor. Elaborado por el Autor.

Para el ensamble del sistema se anexo un socket para la independencia del sensor infrarrojo, esto con fines de independizar los componentes que conforman el equipo; sensor, lector código de barra, fuente de poder, dando facilidad en el caso de que se requiera cambiar el tipo de periférico de entrada para él envío de datos a la BD.

Pruebas de funcionamiento.

Como punto de arranque, todas las bases de datos están vacías y se procede a introducir los datos de: usuario, maquina, operación y paquete respectivamente.

Para ello se realiza fallos al introducir los códigos como pruebas de fuego y verificación que las sentencias de validación Esten funcionando.

Figura 30.

Ingreso datos equipo Raspberrypi 1(emisor 1)



```
python3
Archivo Editar Pestañas Ayuda
-----
Datos de Dispositivo >>>
Ip: 192.168.1.9/ Hostname: raspberrypi1
Mac: b8:27:eb:b5:97:e2
-----
Ingrese codigo de Usuario: 32011,1,10,34
Ingresa unCodigo de Usuario Valido
Ingrese codigo de Usuario: E11111
Ingrese codigo de Maquina: M11111
Ingrese codigo de Operación: 02242
Ingrese codigo de Paquete: 32011,1,10,34
-----
                                0
                                <<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:03:10
```

Nota. Fuente Autor. Elaborado por Autor

Figura 31.

Envío de datos Raspberrypi 1(emisor 1)

```
python3
Archivo Editar Pestañas Ayuda
HoraFin: 12:04:56
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 106.0
-----0-----
                <<Pieza # 2 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:04:56
/ 192.168.1.9 /
FechaFin: 2022/12/27
HoraFin: 12:06:36
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 100.0
-----0-----
                <<Pieza # 3 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:06:36
```

Nota. Fuente Autor. Elaborado por Autor.

Figura 32.

Visualización de datos Raspberrypi1 (emisor 1)

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

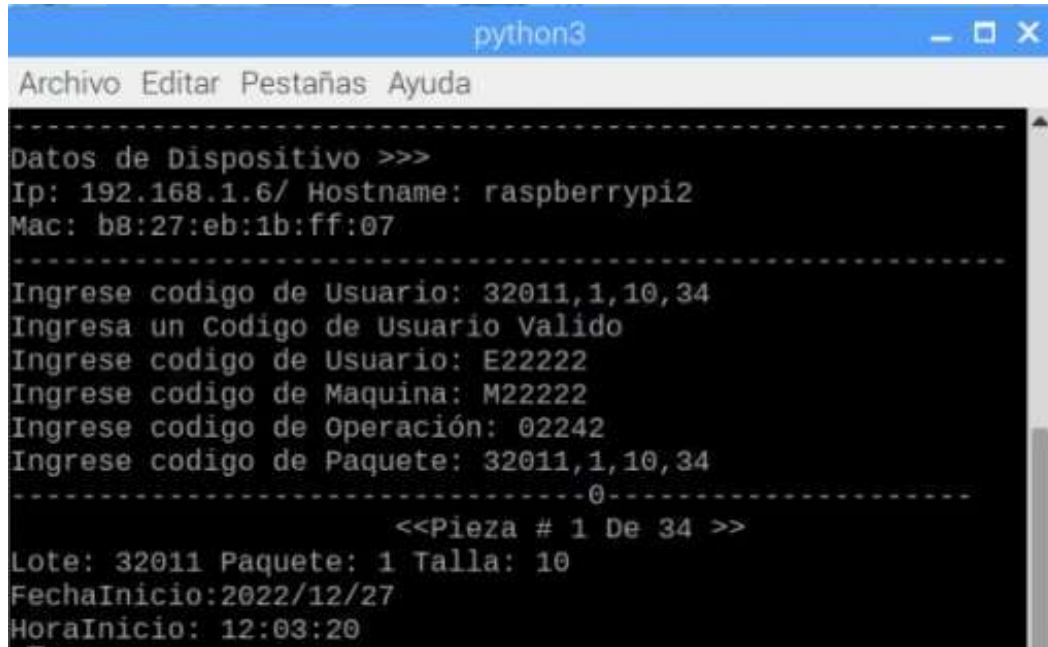
[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
98274cb8097a2	192.168.1.8	raspberrypi1	E111111	M111111	00242	32011	1	10	34	2	2022-12-27	12:04:56	100.0	100.00
98274cb8097a2	192.168.1.8	raspberrypi1	E111111	M111111	00242	32011	1	10	34	1	2022-12-27	12:03:10	106.0	106.00

Nota. Fuente Autor. Elaborado por Autor.

Figura 33.

Ingreso datos equipo Raspberrypi2 (emisor 2)



```
python3
Archivo  Editar  Pestañas  Ayuda
-----
Datos de Dispositivo >>>
Ip: 192.168.1.6/ Hostname: raspberrypi2
Mac: b8:27:eb:1b:ff:07
-----
Ingrese codigo de Usuario: 32011,1,10,34
Ingresa unCodigo de Usuario Valido
Ingrese codigo de Usuario: E22222
Ingrese codigo de Maquina: M22222
Ingrese codigo de Operación: 02242
Ingrese codigo de Paquete: 32011,1,10,34
-----0-----
                <<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:03:20
```

Nota. Fuente Autor. Elaborado por Autor

Figura 34.

Envío de datos Raspberrypi2 (emisor 2)

```
python3
Archivo Editar Pestañas Ayuda
HoraFin: 12:04:57
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 97.3
-----0-----
                <<Pieza # 2 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:04:57
/ 192.168.1.6 /
FechaFin: 2022/12/27
HoraFin: 12:06:38
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 101.0
-----0-----
                <<Pieza # 3 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:06:38
```

Nota. Fuente Autor. Elaborado por Autor

Figura 35.

Visualización de datos Raspberrypi2 (emisor 2)

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
08:27:eb:18:f0:07	192.168.1.6	raspberrypi2	02222	M22222	02242	32011	1	10	34	2	2022-12-27	12:04:57	101.0	99.15
08:27:eb:18:f0:07	192.168.1.6	raspberrypi2	02222	M22222	02242	32011	1	10	34	1	2022-12-27	12:06:38	97.3	97.30

Nota. Fuente Autor. Elaborado por Autor.

Figura 36.

Ingreso datos equipo Raspberrypi3 (emisor 3)



```
python3
Archivo Editar Pestañas Ayuda
-----
Ingrese codigo de Usuario: 02051
Ingresas unCodigo de Usuario Valido
Ingrese codigo de Usuario: 32011,2,12,34
Ingresas unCodigo de Usuario Valido
Ingrese codigo de Usuario: E11111
Ingrese codigo de Maquina: M11111
Ingrese codigo de Operación: E33333
Ingreses codigo de Maquina: M33333
Ingrese codigo de Operación: 02242
Ingrese codigo de Paquete: 32011,1,10,34
-----0-----
                <<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:04:06
```

Nota. Fuente Autor. Elaborado por Autor

Figura 37.

Envío de datos Raspberrypi3 (emisor 3)



```
python3
Archivo Editar Pestañas Ayuda
HoraFin: 12:05:08
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 61.7
-----0-----
                <<Pieza # 2 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:05:08
/ 192.168.1.5 /
FechaFin: 2022/12/27
HoraFin: 12:06:41
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 93.3
-----0-----
                <<Pieza # 3 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:06:41
```

Nota. Fuente Autor. Elaborado por Autor

Figura 38.

Visualización de datos Raspberrypi3 (emisor 3)

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
88:27:eb:31:e4:0b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	1	30	34	2	2022-12-27	12:00:00	93.3	77.50
88:27:eb:31:e4:0b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	1	30	34	1	2022-12-27	12:04:00	81.7	81.70

Nota. Fuente Autor. Elaborado por Autor

Figura 39.

Visualización de datos integrados en Rasperryi4 (receptor/integrador).

REGISTRO DE TIEMPOS DE PRODUCCIÓN INTEGRACION



Tabla de Códigos de Datos Integrados

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
88:27:eb:31:e4:0b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	1	30	34	2	2022-12-27	12:03:00	93.3	77.50
88:27:eb:1b:f0:07	192.168.1.6	raspberrypi2	E22222	M22222	02342	32011	1	30	34	2	2022-12-27	12:04:57	104.0	90.15
88:27:eb:b5:97:e2	192.168.1.8	raspberrypi1	E11111	M11111	02342	32011	1	30	34	2	2022-12-27	12:04:50	103.0	103.00
88:27:eb:31:e4:0b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	1	30	34	1	2022-12-27	12:04:00	81.7	81.70
88:27:eb:1b:f0:07	192.168.1.6	raspberrypi2	E22222	M22222	02342	32011	1	30	34	1	2022-12-27	12:03:30	87.3	87.50
88:27:eb:b5:97:e2	192.168.1.8	raspberrypi1	E11111	M11111	02342	32011	1	30	34	1	2022-12-27	12:03:10	106.0	106.00

Nota. Resultado de IoT, Visualización que se podrá acceder desde cualquier parte del mundo

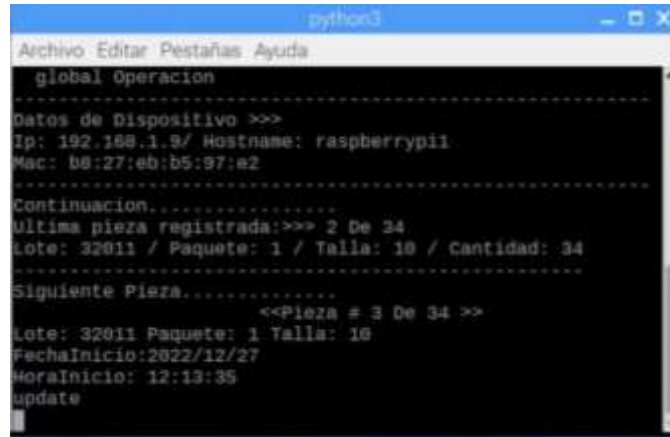
Fuente Autor. Elaborado por Autor

Prueba de reinicio

Con el fin de comprobar la toma de los últimos datos registrados se realiza un reboot al sistema.

Figura 40.

Toma últimos datos registrados Raspberrypi1 (emisor 1)



```
python3
Archivo Editar Pestañas Ayuda
global Operacion
-----
Datos de Dispositivo >>>
Ip: 192.168.1.9/ Hostname: raspberrypi1
Mac: b8:27:eb:b5:97:e2
-----
Continuacion.....
Ultima pieza registrada:>>> 2 De 34
Lote: 32011 / Paquete: 1 / Talla: 10 / Cantidad: 34
-----
Siguiente Pieza.....
                <<Pieza # 3 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:13:39
update
```

Nota. Interface grafica vista después de un reboot, inicia en la última pieza con los datos correspondientes a la misma registrada. Fuente Autor. Elaborado por Autor.

Figura 41.

Toma últimos datos registrados Raspberrypi2 (emisor 2)



```
python3
Archivo Editar Pestañas Ayuda
global Operacion
-----
Datos de Dispositivo >>>
Ip: 192.168.1.6/ Hostname: raspberrypi2
Mac: b8:27:eb:1b:ff:07
-----
Continuacion.....
Ultima pieza registrada:>>> 2 De 34
Lote: 32011 / Paquete: 1 / Talla: 10 / Cantidad: 34
-----
Siguiente Pieza.....
                <<Pieza # 3 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 11:51:52
update
```

Nota. Interface grafica vista después de un reboot, inicia en la última pieza con los datos correspondientes a la misma registrada. Fuente Autor. Elaborado por Autor.

Figura 42.

Toma últimos datos registrados Raspberrypi3 (emisor 3)



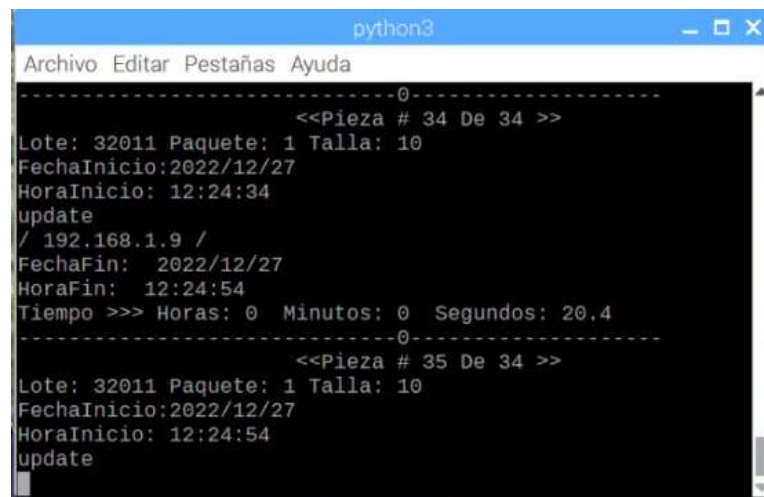
```
python3
Archivo Editar Pestañas Ayuda
-----0-----
global Operacion
-----0-----
Datos de Dispositivo >>>
Ip: 192.168.1.5/ Hostname: raspberrypi3
Mac: b8:27:eb:31:e4:0b
-----0-----
Continuacion.....
Ultima pieza registrada:>>> 2 De 34
Lote: 32011 / Paquete: 1 / Talla: 10 / Cantidad: 34
-----0-----
Siguiente Pieza.....
                <<Pieza # 3 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 11:51:18
update
```

Nota. Interface grafica vista después de un reboot, inicia en la última pieza con los datos correspondientes a la misma registrada. Fuente Autor. Elaborado por Autor.

Se continua con el proceso hasta finalizar el paquete...

Figura 43.

Finalización paquete Raspberrypi1 (emisor1)



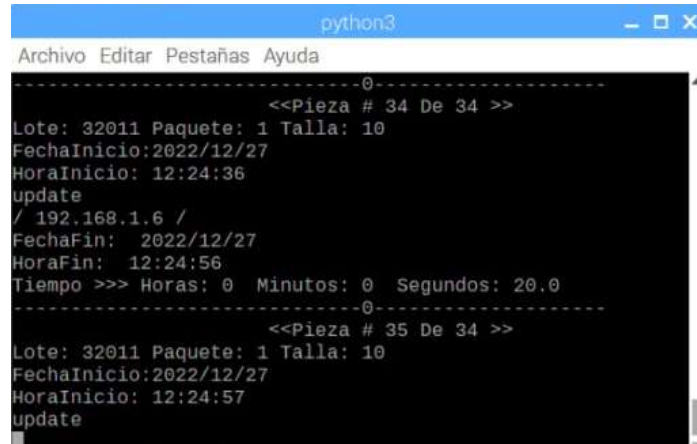
```
python3
Archivo Editar Pestañas Ayuda
-----0-----
                <<Pieza # 34 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:34
update
/ 192.168.1.9 /
FechaFin: 2022/12/27
HoraFin: 12:24:54
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 20.4
-----0-----
                <<Pieza # 35 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:54
update
```

Nota. Interface grafica en la cual se muestra que se ha completado 34 piezas de 34. Fuente Autor.

Elaborado por Autor.

Figura 44.

Finalización paquete Raspberrypi2 (emisor2)



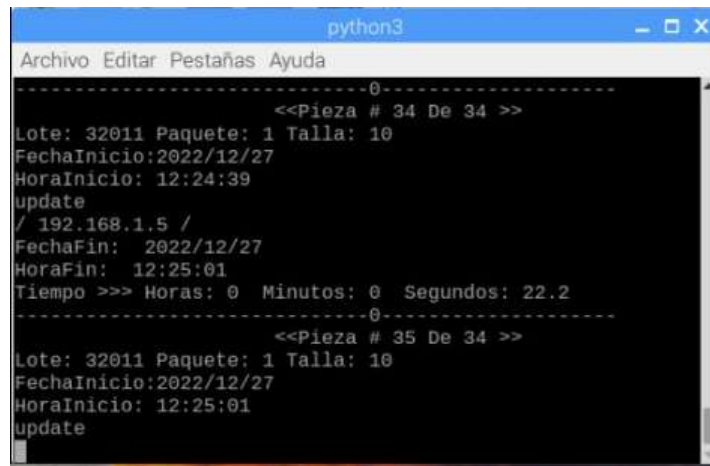
```
python3
Archivo Editar Pestañas Ayuda
-----0-----
<<Pieza # 34 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:36
update
/ 192.168.1.6 /
FechaFin: 2022/12/27
HoraFin: 12:24:56
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 20.0
-----0-----
<<Pieza # 35 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:57
update
```

Nota. Interface grafica en la cual se muestra que se ha completado 34 piezas de 34. Fuente Autor.

Elaborado por Autor.

Figura 45.

Finalización paquete Raspberrypi3 (emisor3)



```
python3
Archivo Editar Pestañas Ayuda
-----0-----
<<Pieza # 34 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:39
update
/ 192.168.1.5 /
FechaFin: 2022/12/27
HoraFin: 12:25:01
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 22.2
-----0-----
<<Pieza # 35 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:25:01
update
```

Nota. Interface grafica en la cual se muestra que se ha completado 34 piezas de 34. Fuente Autor.

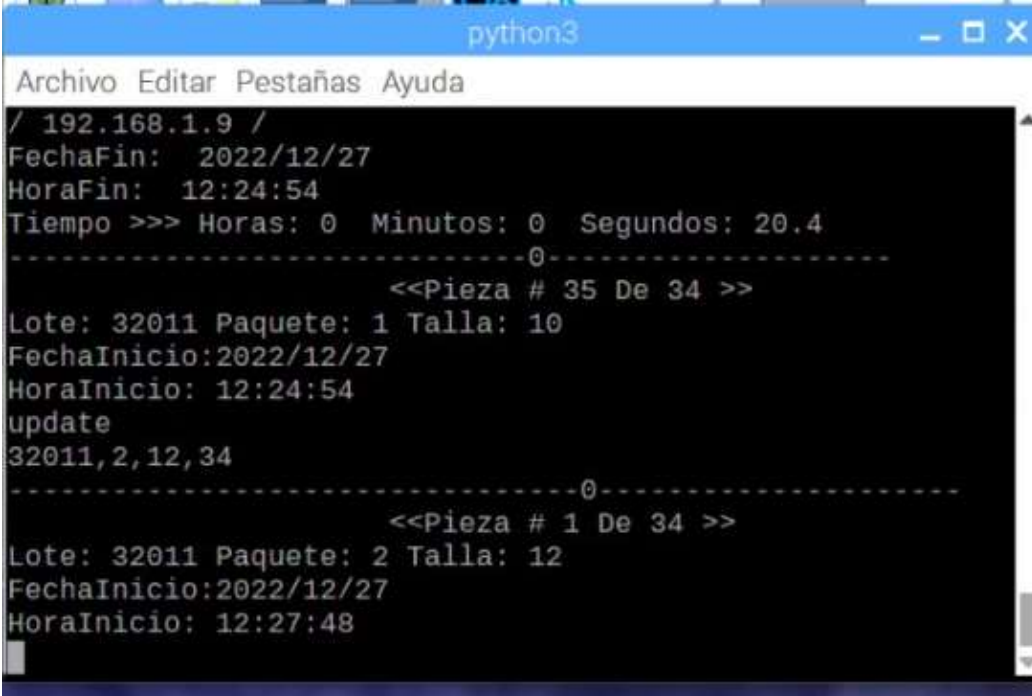
Elaborado por Autor.

Al no existir más piezas se cambia a otro paquete, la pieza 35 no se envía, si bien es cierto en este punto se puede incluir una sentencia para que frene el paquete y solicite el otro, pero dado a los posibles sensados falsos por el entorno de producción no se lo recomienda, dichos sensados erróneos se podrán eliminar a través de un control estadístico a futuro.

Prueba inicio nuevo paquete

Figura 46.

Ingreso nuevo paquete Raspberrypi1 (emisor1)



```
python3
Archivo Editar Pestañas Ayuda
/ 192.168.1.9 /
FechaFin: 2022/12/27
HoraFin: 12:24:54
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 20.4
-----0-----
<<Pieza # 35 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:54
update
32011,2,12,34
-----0-----
<<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:27:48
```

Nota. Interface grafica en la cual se muestra el inicio de un nuevo paquete, el contador inicia automáticamente una vez ingresado el nuevo paquete. Fuente Autor. Elaborado por Autor.

Figura 47.

Ingreso nuevo paquete Raspberrypi2 (emisor2)

```
python3
Archivo Editar Pestañas Ayuda
/ 192.168.1.6 /
FechaFin: 2022/12/27
HoraFin: 12:24:56
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 20.0
-----0-----
<<Pieza # 35 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:24:57
update
32011,2,12,34
-----0-----
<<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:27:36
```

Nota. Interface grafica en la cual se muestra el inicio de un nuevo paquete, el contador inicia automáticamente una vez ingresado el nuevo paquete. Fuente Autor. Elaborado por Autor.

Figura 48.

Ingreso nuevo paquete Raspberrypi3 (emisor3)

```
python3
Archivo Editar Pestañas Ayuda
/ 192.168.1.5 /
FechaFin: 2022/12/27
HoraFin: 12:25:01
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 22.2
-----0-----
<<Pieza # 35 De 34 >>
Lote: 32011 Paquete: 1 Talla: 10
FechaInicio:2022/12/27
HoraInicio: 12:25:01
update
32011,2,12,34
-----0-----
<<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:27:53
```

Nota. Interface grafica en la cual se muestra el inicio de un nuevo paquete, el contador inicia automáticamente una vez ingresado el nuevo paquete. Fuente Autor. Elaborado por Autor.

Figura 49.

Registro datos nuevo paquete Raspberrypi1 (emisor1)

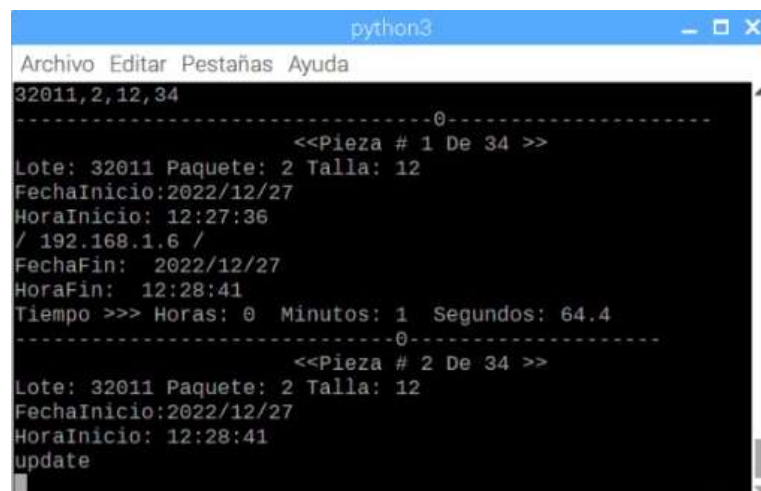


```
python3
Archivo Editar Pestañas Ayuda
32011,2,12,34
-----0-----
<<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:27:48
/ 192.168.1.9 /
FechaFin: 2022/12/27
HoraFin: 12:28:39
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 51.0
-----0-----
<<Pieza # 2 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:28:39
update
```

Nota. Interface grafica en la cual se muestra el registro de los datos de un nuevo paquete. Fuente Autor. Elaborado por Autor.

Figura 50.

Registro datos nuevo paquete Raspberrypi2 (emisor2)



```
python3
Archivo Editar Pestañas Ayuda
32011,2,12,34
-----0-----
<<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:27:36
/ 192.168.1.6 /
FechaFin: 2022/12/27
HoraFin: 12:28:41
Tiempo >>> Horas: 0 Minutos: 1 Segundos: 64.4
-----0-----
<<Pieza # 2 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:28:41
update
```

Nota. Interface grafica en la cual se muestra el registro de los datos de un nuevo paquete. Fuente Autor. Elaborado por Autor.

Figura 51.

Registro datos nuevo paquete Raspberrypi3 (emisor3)

```
python3
Archivo Editar Pestañas Ayuda
32011, 2, 12, 34
-----0-----
<<Pieza # 1 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:27:53
/ 192.168.1.5 /
FechaFin: 2022/12/27
HoraFin: 12:28:44
Tiempo >>> Horas: 0 Minutos: 0 Segundos: 50.6
-----0-----
<<Pieza # 2 De 34 >>
Lote: 32011 Paquete: 2 Talla: 12
FechaInicio:2022/12/27
HoraInicio: 12:28:44
update
```

Nota. Interface grafica en la cual se muestra el registro de los datos de un nuevo paquete. Fuente Autor. Elaborado por Autor.

Visualización de datos finales de simulación

Figura 52.

Visualización de datos finales de simulación Raspberrypi1 (emisor 1)

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
b827abb597a2	192.168.1.9	raspberrypi1	E11111	M11111	02042	32011	2	12	34	1	2022-12-27	12:27:40	51.0	51.00
b827abb597a2	192.168.1.9	raspberrypi3	E11111	M11111	02042	32011	1	10	34	34	2022-12-27	12:28:34	20.4	25.80
b827abb597a2	192.168.1.9	raspberrypi1	E11111	M11111	02042	32011	1	10	34	33	2022-12-27	12:24:23	10.8	26.15
b827abb597a2	192.168.1.9	raspberrypi1	E11111	M11111	02042	32011	1	10	34	32	2022-12-27	12:24:18	6.71	26.65

Figura 53.

Visualización de datos finales de simulación Raspberrypi2 (emisor 2)

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
88:27:eb:1b:f0:07	192.168.1.8	raspberrypi2	822222	M22222	02242	32011	2	12	34	1	2022-12-27	12:27:36	64.4	64.40
88:27:eb:1b:f0:07	192.168.1.8	raspberrypi2	822222	M22222	02242	32011	1	10	34	34	2022-12-27	12:24:36	20.0	64.13
88:27:eb:1b:f0:07	192.168.1.8	raspberrypi2	822222	M22222	02242	32011	1	10	34	33	2022-12-27	12:24:25	11.4	65.47
88:27:eb:1b:f0:07	192.168.1.8	raspberrypi2	822222	M22222	02242	32011	1	10	34	32	2022-12-27	12:24:17	7.6	67.16
88:27:eb:1b:f0:07	192.168.1.8	raspberrypi2	822222	M22222	02242	32011	1	10	34	31	2022-	12:24:11	6.00	69.08

Nota. Visualización de datos base local. Fuente Autor. Elaborado por Autor.

Figura 54.

Visualización de datos finales de simulación Raspberrypi3 (emisor 3)

REGISTRO DE TIEMPOS DE PRODUCCIÓN



Tabla de Códigos de Barra y Tiempos

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
88:27:eb:21:e4:0b	192.168.1.3	raspberrypi3	822222	M22222	02242	32011	2	12	34	1	2022-12-27	12:27:52	90.6	10.40
88:27:eb:21:e4:0b	192.168.1.3	raspberrypi3	822222	M22222	02242	32011	1	10	34	34	2022-12-27	12:24:39	22.2	44.02
88:27:eb:21:e4:0b	192.168.1.3	raspberrypi3	822222	M22222	02242	32011	1	10	34	33	2022-12-27	12:24:31	6.17	65.28
88:27:eb:21:e4:0b	192.168.1.3	raspberrypi3	822222	M22222	02242	32011	1	10	34	32	2022-12-27	12:24:20	10.8	67.87

Nota. Visualización de datos base local. Fuente Autor. Elaborado por Autor

Figura 55.

Visualización de datos Integrados (receptor/integrador) LAN.

REGISTRO DE TIEMPOS DE PRODUCCIÓN INTEGRACION

Tabla de Códigos de Datos Integrados

[Exportar a Excel](#)

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lots	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
88274b31a40b	192.168.1.3	raspberrypi2	033333	M33333	-03342	32011	2	12	34	1	2023-12-27	12:27:53	50.0	50.00
88274b1b9f07	192.168.1.8	raspberrypi2	022222	M22222	-03342	32011	2	12	34	1	2023-12-27	12:27:56	64.4	64.80
88274b3b37e2	192.168.1.8	raspberrypi1	E11111	M11111	-03342	32011	2	12	34	1	2023-12-27	12:27:46	31.0	31.00
88274b31a40b	192.168.1.3	raspberrypi2	033333	M33333	-03342	32011	1	10	34	34	2023-12-27	12:24:39	22.2	64.02
88274b1b9f07	192.168.1.8	raspberrypi2	022222	M22222	-03342	32011	1	10	34	34	2023-12-27	12:24:38	20.0	64.13
88274b3b37e2	192.168.1.8	raspberrypi1	E11111	M11111	-03342	32011	1	10	34	34	2023-12-27	12:24:34	20.4	23.96

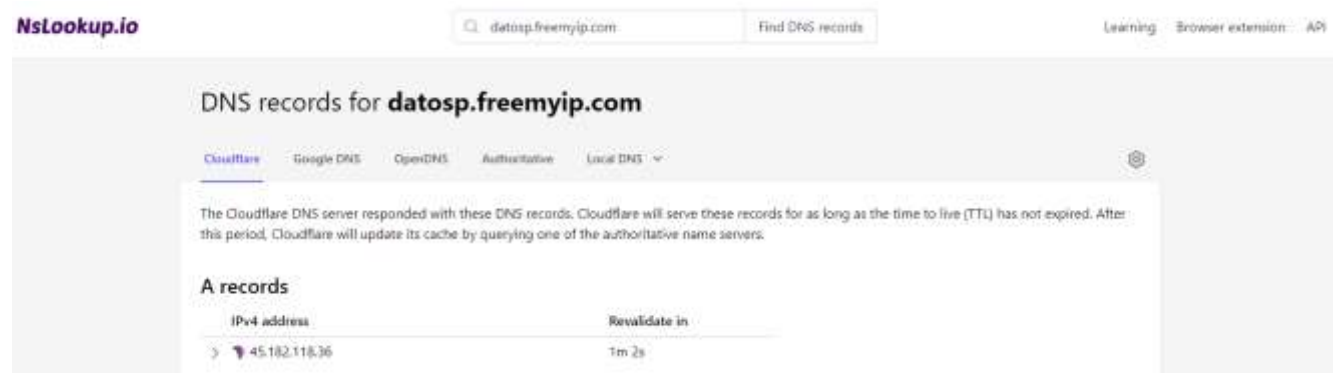
Nota. Fuente Autor. Elaborado por Autor

Como se puede visualizar los datos integrados de los, en este caso, 3 emisores, en la base externa o bien denominada madre, todos coinciden correctamente, verificando así la migración de los datos y la integración de los mismos.

Prueba de conexión DNS para visualización remota

Figura 56.

Prueba enlace dominio con IP dispositivo para acceso en la red



Nota. Fuente Autor. Elaborado por Autor

El sistema de nombres de dominio DNS, traduce los nombres de dominios aptos para lectura humana a direcciones IP aptas para lectura por parte de máquinas, en este caso el dominio momentáneo registrado para el acceso al servidor receptor es “datos.freemyip.com”. Es momentáneo puesto que se lo puede cambiar por uno mas personalizado de acuerdo a cuál requiera la empresa.

Como paso extra hay que establecer el redireccionamiento del puerto 80 esto ya que es el puerto que tiene por defecto el protocolo http de php, funciona como un redireccionamiento de llamada, cuando la red global WAN llame a datos.freemyip.com este devolverá a la IP publica donde se encuentre el servidor receptor y el router deberá dirigir el llamado a la ip LAN que este registrado el servidor **Figura 57.**

Figura 57.

Port forwarding (redireccionamiento de puerto)

The screenshot shows the ZTE F660 web interface for configuring Port Forwarding. The sidebar on the left lists various settings, with 'Port Forwarding' selected. The main configuration area includes the following fields:

- Enable:
- Name:
- Protocol:
- WAN Host Start IP Address:
- WAN Host End IP Address:
- WAN Connection:
- WAN Start Port: (1 ~ 65535)
- WAN End Port: (1 ~ 65535)
- Enable MAC Mapping:
- LAN Host MAC Address:
- LAN Host Start Port: (1 ~ 65535)
- LAN Host End Port: (1 ~ 65535)

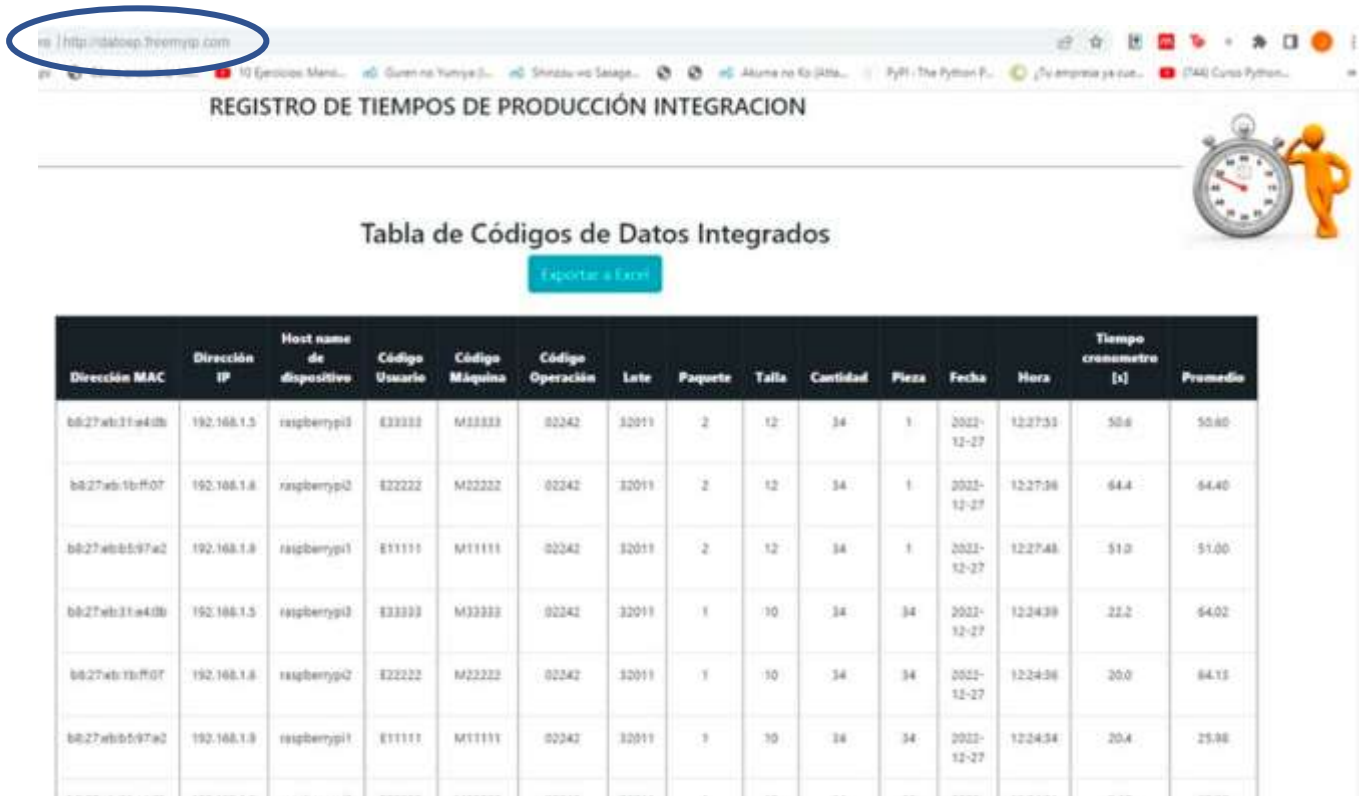
Buttons for 'Modify' and 'Cancel' are located below the form. Below the form is a table summarizing the configuration:

Enable	Name	WAN Host Start IP Address	WAN Start Port	LAN Host Start Port	WAN Connection	Modify	Delete
<input checked="" type="checkbox"/>	DAtos		80	80	CNT_WAN_2		
	TCP AND U		80	80	dc:a6:32:95:c		

Nota. Fuente Autor. Elaborado por Autor

Figura 58.

Visualización de datos integrados, Raspberrypi4 (receptor/integrador) WAN.



REGISTRO DE TIEMPOS DE PRODUCCIÓN INTEGRACION

Tabla de Códigos de Datos Integrados

Exportar a Excel

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
b827ab31e40b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	2	12	34	1	2022-12-27	12:27:53	50.6	50.60
b827ab1b1f07	192.168.1.8	raspberrypi2	E22222	M22222	02342	32011	2	12	34	1	2022-12-27	12:27:56	64.4	64.40
b827ab5597a2	192.168.1.8	raspberrypi1	E11111	M11111	02342	32011	2	12	34	1	2022-12-27	12:27:48	51.0	51.00
b827ab31e40b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	1	10	34	34	2022-12-27	12:24:39	22.2	64.02
b827ab1b1f07	192.168.1.8	raspberrypi2	E22222	M22222	02342	32011	1	10	34	34	2022-12-27	12:24:36	20.0	64.15
b827ab5597a2	192.168.1.8	raspberrypi1	E11111	M11111	02342	32011	1	10	34	34	2022-12-27	12:24:34	20.4	25.96

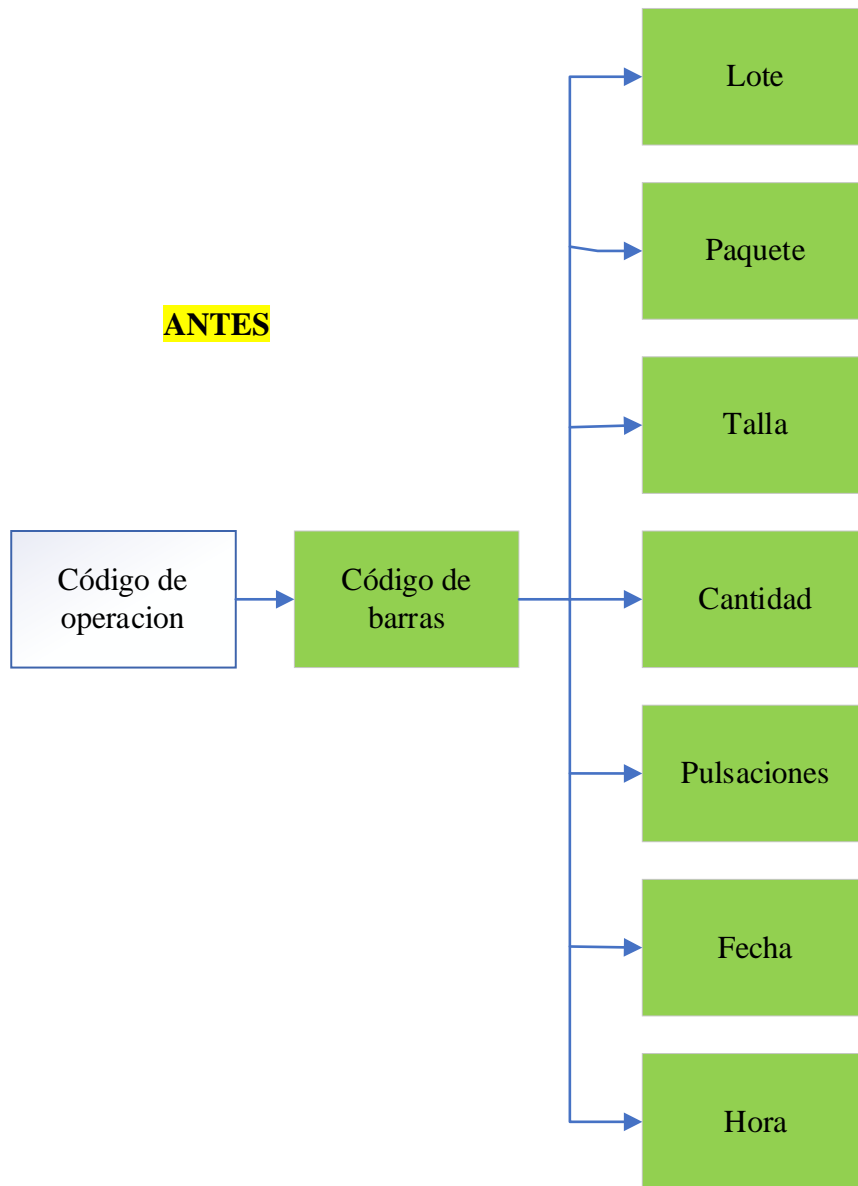
Nota. Fuente Autor. Elaborado por Autor

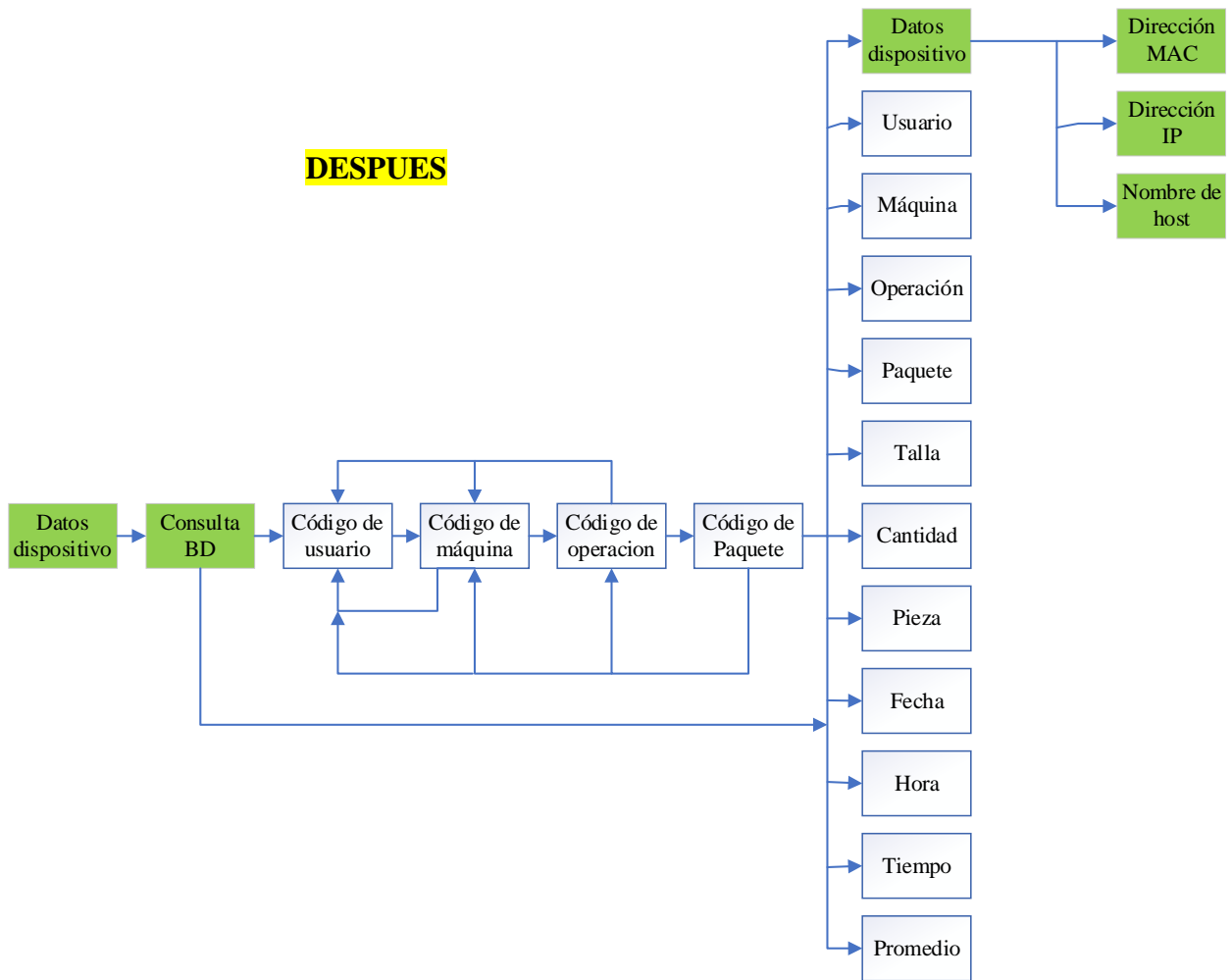
Tomar en cuenta que el enlace por el cual se estableció conexión es diferente al enlace anterior ya que este es llamado por el dominio público en la red global WAN y el otro solo por la dirección IP en la red local LAN.

Resultados esperados.

Figura 59.

Mejora en la gestión del flujo de información.





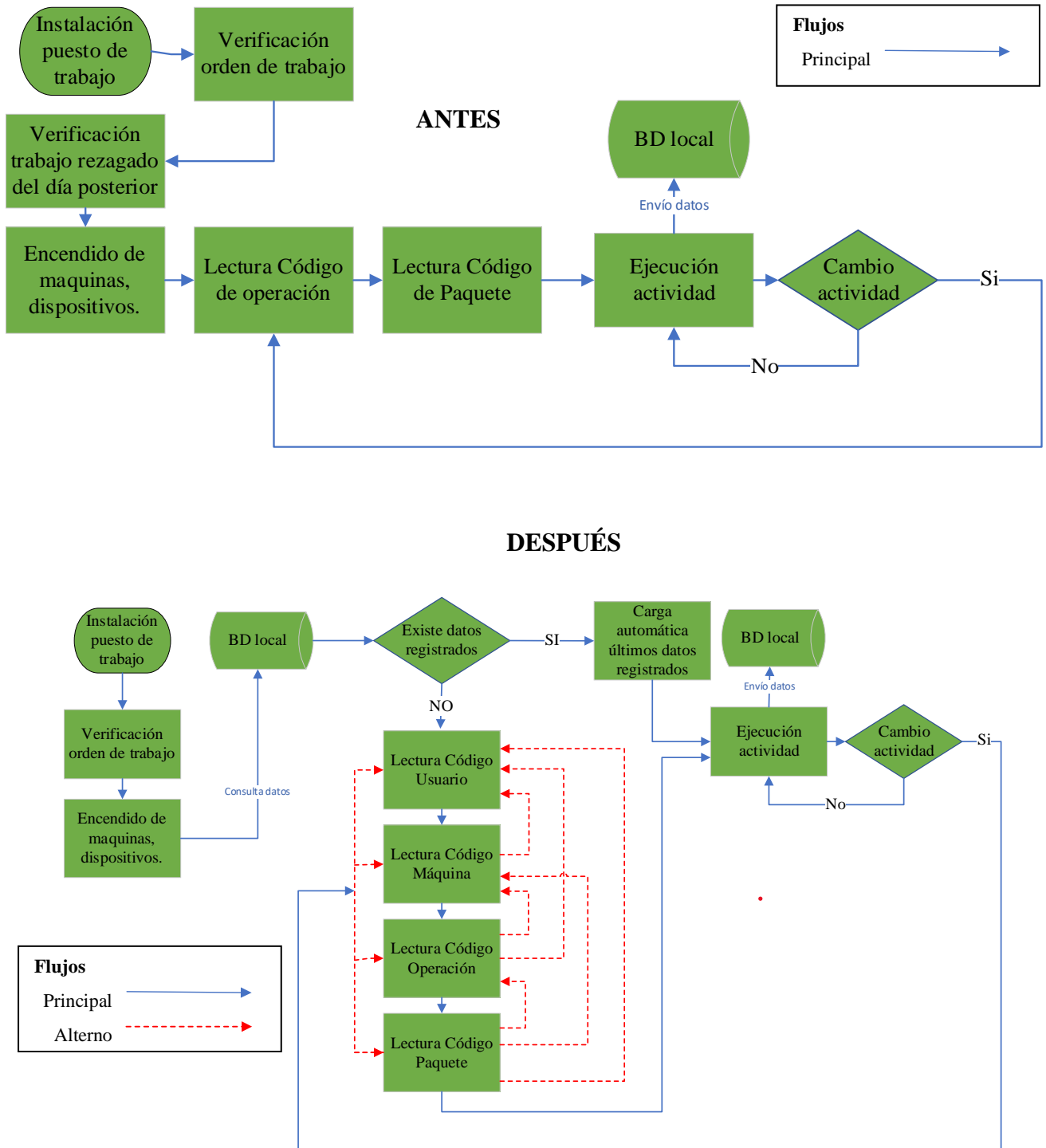
Nota. Fuente Autor. Elaborado por Autor

La manera en la que gestiona la información el dispositivo es más organizada y está acorde a las necesidades de las partes interesadas.

En el antes se puede observar un flujo lineal mientras que en el después la información no depende de un reinicio para cambiar a nuevos datos, su flujo no es en serie presenta un desenvolvimiento en paralelo permitiendo hacer una actualización de los datos independiente.

Figura 60.

Facilidad en la operatividad del sistema.



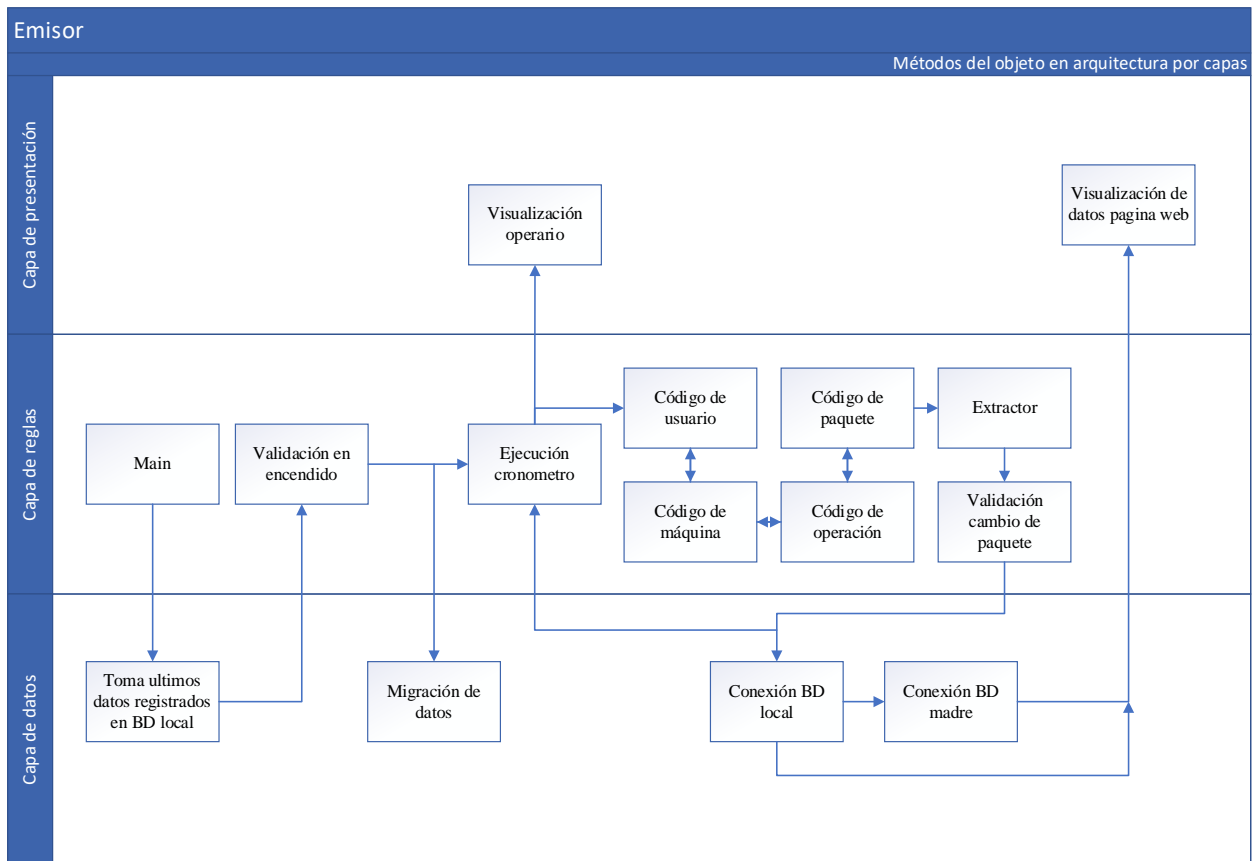
Nota. Fuente Autor. Elaborado por Autor.

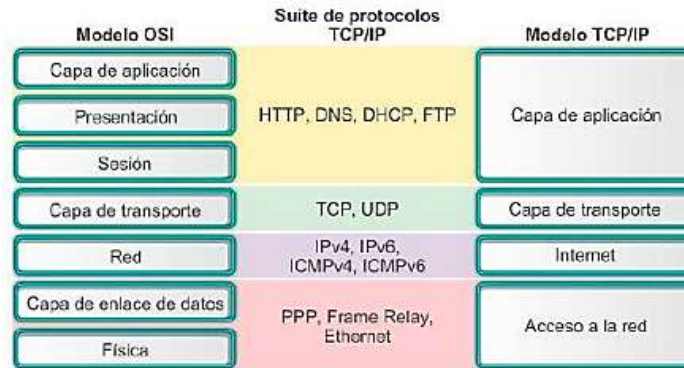
Como se puede evidenciar la diferencia en la comparativa a manera de operatividad, el leer todos los datos siempre que se desee cambiar una actividad, significaba un problema operativo con el trabajador.

Al realizar los cambios se obtiene una mejora en la operatividad puesto que el hecho de leer solo los datos necesarios evita un reproceso innecesario en dicha actividad.

Figura 61.

Inclusión de IoT.





http://idatasep.themyp.com

REGISTRO DE TIEMPOS DE PRODUCCIÓN INTEGRACION



Tabla de Códigos de Datos Integrados

Exportar a Excel

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
bb27ab31e40b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	2	12	34	1	2022-12-27	12:27:53	50.6	50.60
bb27ab1b1ff07	192.168.1.8	raspberrypi2	E22222	M22222	02342	32011	2	12	34	1	2022-12-27	12:27:56	64.4	64.40
bb27ab5597a2	192.168.1.8	raspberrypi1	E11111	M11111	02342	32011	2	12	34	1	2022-12-27	12:27:48	51.0	51.00
bb27ab31e40b	192.168.1.3	raspberrypi3	E33333	M33333	02342	32011	1	10	34	34	2022-12-27	12:24:39	22.2	64.02
bb27ab1b1ff07	192.168.1.8	raspberrypi2	E22222	M22222	02342	32011	1	10	34	34	2022-12-27	12:24:36	20.0	64.13
bb27ab5597a2	192.168.1.8	raspberrypi1	E11111	M11111	02342	32011	1	10	34	34	2022-12-27	12:24:34	20.4	25.98

Nota. Implementación estructura IoT a programación del sistema. Fuente (Madakam et al., 2015).

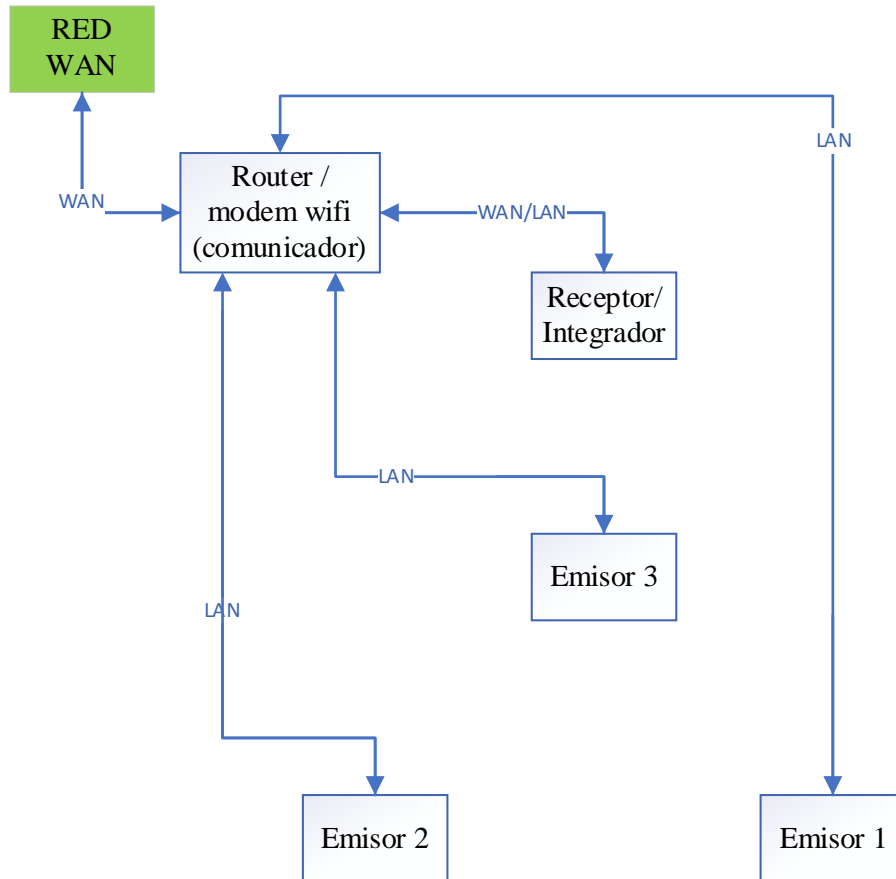
Elaborado por el Autor.

La arquitectura de programación utilizada basa su estructura en lo que establece la Unión Internacional de Telecomunicaciones (UIT). Al ser por capas y optar por una programación orientada a objetos se obtiene un entorno de programación organizado.

El anterior sistema no posee esta estructura. Además, se puede acceder a los datos integrados desde cualquier parte del mundo vía red global o WAN.

Figura 62.

Datos integrados de varios equipos en un solo punto.



Nota. Fuente Autor. Elaborado por el Autor

A diferencia del anterior sistema, el presente posee un punto de integración el cual mantiene comunicación permanente con los equipos denominados emisores los cuales envían la información mediante IoT, en la red local para ser visualizado en el receptor por la red pública.

Datos a la espera de ser procesados/analizados.

- En el **Anexo 1** se visualiza Los datos obtenidos que se integran en una sola base de datos lo cual da paso a la posibilidad de realizar un tratamiento de datos a futuro a través de un framework u otro método de tratamiento de datos.

Figura 63.

Cambio periférico de entrada.

ANTES



DESPUES



Nota. Fuente Autor. Elaborado por Autor

El periférico de entrada para pulsaciones anterior, al ser un pulsador necesitaba el contacto físico y fuerza del operario para establecer su accionar, sin mencionar que al momento de realizar la evaluación dicho componente ya presento fallas en sus pulsaciones producto del accionar constante. La ventaja al tener un sensor en su lugar es principalmente la facilidad que presenta en su ejecución ya que no necesita el contacto físico, además de ventajas ergonómicas, como evitar fatiga en la muñeca ya que el proceso es repetitivo y dar un mejor posicionamiento nominal a la misma es primordial, evitando desviación cubital y radial.

Vista interna del Sistema final **Anexo 2** y **Anexo 3**.

Tabla 23.

Cronograma sugerido para implementación

Actividad	Tiempo		Responsables
	Días	Horas día	
Instalación y operatividad	2	2	Investigador
			Técnico de Planta
Toma de muestras para evaluación	7	8	Investigador
			Técnico de Planta
			Operador de Planta
Tratamiento de Resultados	7	2	Investigador
			Coordinador de planta

Nota. Se actualizan los valores partiendo del trabajo propuesto por (Cuyachamín, 2022). Elaborado por Autor.

El presente cronograma es una sugerencia a tomar en cuenta para el momento en el que se desee implementar el sistema y realizar una la validación de los datos, teniendo como número de días máximos como horas recomendables para la ejecución de las actividades de acuerdo a la disponibilidad de la planta, como se realizó en (Maldonado, 2022)

Análisis de costos.

Tabla 24.

Costos referenciales de elementos utilizados.

Elemento	Cantidad	Valor U. (\$)	Valor total. (\$)
Lector Código Barras	3	32	96
Sensor	3	3	9
Pantalla LCD táctil para Raspberry Pi 3 B	3	28	84
Raspberry Pi 3 B	3	68	204
Raspberry Pi 4	1	110	110
Carcasa de Raspberry Pi 3 B	3	11	33
MicroSD 32G	4	22	88
Conductor eléctrico	8 metros	0,25	2
	Total		626

Nota. Se actualizan los valores partiendo del trabajo propuesto por (Cuyachamín, 2022). Elaborado por Autor.

En el año 2022 la patente de raspberry fue abierta como libre uso por lo cual se lanzaron al mercado muchos microcontroladores nuevos, a futuro se debería realizar una ponderación de relación de componentes nueva para escoger la más optable.

Tabla 25.*Costos implementación en la empresa (fijos)*

RUBRO\EMPLEA DO	Gerencia	Analista Calidad	Técnico de planta	Operario	Coordinador	TOTAL
Salario Mínimo Vital (2023)	450	450	450	450	450
Sueldo	3000	600	800	450	550	5400
IESS Patronal (11,35%)	340,5	68,1	90,8	51,1	62,4	612,9
13	250	50	66,7	37,5	45,8	450
14	37,5	37,5	37,5	37,5	37,5	187,5
FR	250	50	66,7	37,5	45,8	450
Vacaciones	125	25	33,3	18,8	22,9	225
Desahucio	62,5	12,5	16,7	9,4	11,5	112,6
TRANSPORTE	0	0	0	0	0	0
Total, Mensual	4065,5	843,1	1111,6	641,7	776	7437,9
Personal	3	3	1	12	6
Total	12196,5	2529,3	1111,6	7700,4	4656	14875,9

Nota. Fuente Royaltex. Elaborado por el Autor.

Al considerar una implementación total es crucial considerar factores importantes de todos los involucrados que intervienen para la operatividad del sistema, para lo cual se realiza la presente cotización tomando en cuenta todos los costos fijos de los principales factores involucrados en el periodo de un mes.

Para determinar el costo total de la implementación del sistema se divide el total de un mes de los costos fijos obtenidos para el total de horas necesaria del cronograma de implementación y validación sugerido, al mismo se le suma el costo de componentes del sistema y se tiene como resultado un estimado del costo real de implantación del sistema en la empresa. **Tabla 26 .**

Tabla 26.

Costo total de implementación del sistema

Costo fijo por mes	Costo fijo por día	Costo fijo por hora
\$14875,9	\$495,86	\$61,98
Total, de horas sugeridas para implementación en la empresa		74 h
Costo fijo total por horas de los involucrados necesarios		\$4.586,52
Costo de componentes del sistema		\$626
Total, implementación		\$5.212,52

Nota. Fuente Autor. Elaborado por el Autor

Como se puede observar el mayor costo se lleva las partes involucradas en la implantación del sistema, puesto que se toma el tiempo de los mismos para la correcta implementación del sistema propuesto.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones.

- Se recopiló información del dispositivo inicial y requerimientos de los principales usuarios, mediante una inspección cualitativa obteniendo la ficha descriptiva (Tabla 7) que ayudo a determinar los atributos técnicos del sistema inicial siendo; lógica y lenguaje de programación, periférico de entrada, periférico de salida, conectividad, estructura de visualización de datos, procesamiento, tipo de energía que ocupa el sistema, almacenamiento de datos, tamaño de datos, entre los principales atributos clasificados de forma general. Mediante una entrevista que se realizó objetivamente a las partes que intervienen en la toma de decisiones y ejecución del proceso siendo el operador del prototipo, jefe de mecánica, jefe de ingeniería, investigador se obtuvo las principales especificaciones demandadas por estas partes interesadas: eficacia en la adquisición de datos, eficiencia y ergonomía en la pulsación para envío de datos, fiabilidad en los datos recolectados, integración en más de un punto de la línea de producción, visualización remota de los datos en cualquier parte del mundo, interface gráfica amigable para el operario, eficiencia operativa en la visualización de datos en la pantalla, autonomía, inicio en los últimos datos registrados, plug and play (rápido y fácil de usar) (Tabla 8).
- Se analizó los atributos y especificaciones encontradas a través de una calificación por las partes interesadas (Tabla 9). Obteniendo las especificaciones 1, 3, 4, 9, 10 de la encuesta realizada (Anexo 4) como la mayor importancia obteniendo una calificación de 5 puntos, donde 5 es la mayor y uno la menor; una vez obtenido la ponderación se realizó la matriz QFD o despliegue de función de calidad en una primera etapa (Figura 20) la cual compara

que atributo incide más con respecto a las especificaciones demandadas, el cual dio como resultado la lógica de programación y el procesamiento del sistema con una valoración de 3 puntos en su importancia (Tabla 10) siendo los atributos más altos con respecto al resto, el puntaje óptimo a lograr es una puntuación de 1; se ordena los atributos influyentes (Tabla 11) para tener una mejor visualización del orden de importancia, se identifica las oportunidades de mejora a realizar en el diseño del prototipo de acuerdo a la realidad del mismo y sus atributos influyentes (Tabla 13), siendo la gestión del flujo de información, el entorno de programación, inclusión de IoT, operatividad de los dispositivos, integrar datos en un solo punto, datos a la espera de ser procesados/analizados, mejor periférico de entrada entre las principales oportunidades de mejora en los atributos del sistema.

- Se diseñó un sistema de adquisición de tiempos de producción mejorando principalmente el orden del flujo de datos (Figura 22) con respecto al anterior (Figura 21), para luego ir al análisis de la operatividad en un puesto de trabajo (Figura 23) para pasar a una mejora (Figura 24) en el cual se puede evidenciar que la operación tiene caminos alternos, los cuales evitan al operario el ingreso total de datos al momento de cambiar una actividad. Se diseñó el sistema de comunicación por el medio en el cual se desarrollará el mismo (Figura 25), basado en la arquitectura y tipo de programación seleccionada (Tabla 16) y (Tabla 17) se obtiene el diagrama del entorno de programación a realizar que incorpora una arquitectura de programación por capas mediante la aplicación del Internet de las Cosas (Figura 26); luego se desarrolla el pseudocódigo (Tabla 12) en el cual se basa la programación final (Anexo 2), programa principal y (Anexo 3) programación visualización web, donde se logra disponer de datos en línea de los procesos de manufactura que se muestran en: Figura 52, Figura 53, Figura 54 y Figura 55; finalmente, el procesamiento de

datos toma por totalidad la programación principal el hardware del receptor, de esta manera permite ingresar un número n de emisores de acuerdo a la capacidad del receptor del sistema que es 2 GB de RAM y 32 GB de MSD

Recomendaciones.

- Mantener un constante registro sobre la nueva información que se añada al sistema en caso de actualizar el mismo.
- Realizar un segundo QFD con el fin de dar continuidad al despliegue de la función calidad, para identificar los nuevos atributos principales, hasta lograr que resultado sea el mínimo 1 en toda la función de calidad.
- Se recomienda realizar un estudio de validación de datos para corroborar el nuevo diseño con respecto al antiguo tal y como se realizó en (Maldonado, 2022); si bien el nuevo periférico de entrada mejora el censado de datos, es recomendable realizar una comparación entre los diferentes sensores que existen en el mercado, puesto que el sensor infrarrojo funciona por el principio de la refracción de luz y el entorno de producción en el que se encuentra puede ocurrir censados falsos, ocasionados por la tela en especial de tonalidad clara, un sensor touch podría mejorar ya que su principio de funcionamiento es diferente y no depende de la luz. Es necesario realizar la programación para el tratamiento de datos en el dispositivo receptor, este con el fin de analizar, predecir, comparar y dar reportes útiles en general a la empresa para aprovechar el procesamiento y utilidad que puede brindar el sistema mas no solo como adquisición sino también como tratamiento de datos; si se requiere añadir cualquier elemento que necesite energía externa, se recomienda realizar el diseño de una placa de alimentación tanto para 3v como 5v tomando en cuenta la capacidad de la misma; analizar la viabilidad de utilizar otro tipo de hardware en el receptor si se aumenta la capacidad o de acuerdo a la

función y el tratamiento de datos a realizar. La Raspberrypi 4 está perfectamente equipada para procesamientos que no excedan la 2GB de uso de memoria RAM y 32 GB de MSD.

Aunque IoT tiene abundantes beneficios, existen algunas fallas en el nivel de implementación y gobernanza de IoT. Las observaciones clave en la literatura son que, no existe una definición estándar en todo el mundo, se requieren estandarizaciones universales en el nivel arquitectónico, las tecnologías varían de un proveedor a otro, por lo que debe ser interoperable y para una mejor gobernanza, se necesita construir protocolos estándar.

BIBLIOGRAFÍA

Bahillo, L. (2022, May 16). Historia de Internet: ¿cómo nació y cuál fue su evolución? Marketing 4 Ecommerce - Tu Revista de Marketing Online Para E-Commerce.

<https://marketing4ecommerce.net/historia-de-internet/#:~:text=Es%20el%20a%C3%B1o%201983%20el,nombre%20de%20%C3%BAnicamente%20%C2%ABInternet%C2%BB.>

Industria Manufacturera - Concepto, tipos, ejemplos y características. (2022). Concepto.

<https://concepto.de/industria-manufacturera/>

Banco central del Ecuador. (2022). *Información Estadística.* Bce.fin.ec.

<https://www.bce.fin.ec/index.php/informacioneconomica>

La importancia de la calidad de los datos en las empresas. (2020, November 25). Artículos Y Tendencias Sobre Soluciones Tecnológicas; Artículos y tendencias sobre soluciones tecnológicas.

<https://itblogsogeti.com/2020/11/25/la-importancia-de-la-calidad-de-los-datos-en-las-empresas/>

Guagliano, C. (2019). *Programación en Python II: Programación orientada a objetos.* RedUsers.

Abd Rahman, M. S. bin, Mohamad, E., & Abdul Rahman, A. A. bin. (2021). Development of IoT—enabled data analytics enhance decision support system for lean manufacturing process improvement. *Concurrent Engineering Research and Applications*, 29(3), 208-220.

<https://doi.org/10.1177/1063293X20987911>

Al-Turjman, F., Zahmatkesh, H., & Shahroze, R. (2022). An overview of security and privacy in smart cities' IoT communications. *Transactions on Emerging Telecommunications Technologies*, 33(3).

<https://doi.org/10.1002/ett.3677>

Chapman, S. N. (s. f.). *Planificación y control de la producción.* www.pearsoneducacion.net

Cuyachamín. (2022). *DISEÑO DE UN SISTEMA DE COMUNICACIÓN PARA ADQUISICIÓN DE*

DATOS EN UNA EMPRESA TEXTIL UBICADA EN LA CIUDAD DE QUITO.

Flores Diana. (2022). *UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA FACULTAD DE INGENIERÍA, INDUSTRIA Y PRODUCCIÓN.*

Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications, 03(05)*, 164-173.
<https://doi.org/10.4236/jcc.2015.35021>

Maldonado. (2022). *UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA FACULTAD DE INGENIERÍA Y TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN.*

Qué importancia tienen los datos en su producción. (s. f.). Recuperado 2 de noviembre de 2022, a partir de <https://elproductorporcino.com/leerEntrada/num/966>

Recopilar datos en fábricas mejora la eficiencia | COPA-DATA. (s. f.). Recuperado 2 de noviembre de 2022, a partir de <https://www.copadata.com/es/industrias/smart-factory/smart-factory-insights/recopilacion-datos-fabrica-mejora-eficiencia/>

SEDANO ALBERTO. (2021). *UNIVERSIDAD PERUANA LOS ANDES FACULTAD DE INGENIERIA ESCUELA PROFESIONAL DE INGENIERIA INDUSTRIAL.*

Sistemas de adquisición de datos de producción (PDA) con Plant iT y brewmaxx. (s. f.). Recuperado 2 de noviembre de 2022, a partir de <https://www.proleit.es/soluciones/adquisicion-de-datos-de-produccion-pda/>

ANEXOS

Anexo 1. Datos integrados por IoT

Dirección MAC	Dirección IP	Host name de dispositivo	Código Usuario	Código Máquina	Código Operación	Lote	Paquete	Talla	Cantidad	Pieza	Fecha	Hora	Tiempo cronometro [s]	Promedio
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	2	12	34	1	27/12/2022	12:27:48	51	51
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	34	27/12/2022	12:24:34	20,4	25,98
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	33	27/12/2022	12:24:23	10,9	26,15
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	32	27/12/2022	12:24:16	6,71	26,63
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	31	27/12/2022	12:24:10	6,43	27,27
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	30	27/12/2022	12:23:58	11,4	27,97
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	29	27/12/2022	12:23:54	3,76	28,54
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	28	27/12/2022	12:23:50	4,05	29,42
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	27	27/12/2022	12:23:30	20,5	30,36
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	26	27/12/2022	12:23:23	6,34	30,74
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	25	27/12/2022	12:23:17	6,72	31,72
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	24	27/12/2022	12:23:09	7,47	32,76
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	23	27/12/2022	12:22:17	51,7	33,86
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	22	27/12/2022	12:22:07	10,3	33,05
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	21	27/12/2022	12:21:54	13,1	34,13
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	20	27/12/2022	12:21:41	12,7	35,19
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	19	27/12/2022	12:21:30	10,6	36,37

b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	18	27/12/2022	12:21:19	11,1	37,8
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	17	27/12/2022	12:21:10	9,39	39,37
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	16	27/12/2022	12:20:43	27,2	41,24
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	15	27/12/2022	12:20:35	7,97	42,18
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	14	27/12/2022	12:20:25	9,05	44,62
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	13	27/12/2022	12:20:17	8,34	47,36
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	12	27/12/2022	12:20:10	6,78	50,61
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	11	27/12/2022	12:20:03	6,96	54,6
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	10	27/12/2022	12:19:58	5,56	59,36
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	9	27/12/2022	12:19:51	7,03	65,34
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	8	27/12/2022	12:19:40	11	72,63
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	7	27/12/2022	12:19:33	6,22	81,43
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	6	27/12/2022	12:19:27	6,7	93,97
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	5	27/12/2022	12:19:15	11,9	111,42
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	4	27/12/2022	12:19:01	14,2	136,3
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	3	27/12/2022	12:13:35	325	177
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	2	27/12/2022	12:04:56	100	103
b8:27:eb:b5:97:e2	192,168,1,9	raspberrypi1	E11111	M11111	2242	32011	1	10	34	1	27/12/2022	12:03:10	106	106
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	2	12	34	1	27/12/2022	12:27:36	64,4	64,4
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	34	27/12/2022	12:24:36	20	64,13
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	33	27/12/2022	12:24:25	11,4	65,47

b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	32	27/12/2022	12:24:17	7,6	67,16
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	31	27/12/2022	12:24:11	6,03	69,08
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	30	27/12/2022	12:24:06	4,74	71,18
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	29	27/12/2022	12:24:02	4,6	73,47
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	28	27/12/2022	12:23:56	5,99	75,93
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	27	27/12/2022	12:23:51	4,45	78,52
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	26	27/12/2022	12:23:24	26,8	81,37
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	25	27/12/2022	12:23:18	6	83,56
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	24	27/12/2022	12:23:11	7,4	86,79
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	23	27/12/2022	12:22:20	51,2	90,24
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	22	27/12/2022	12:22:10	9,96	92,01
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	21	27/12/2022	12:21:58	11,7	95,92
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	20	27/12/2022	12:21:45	13	100,13
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	19	27/12/2022	12:21:33	11,6	104,72
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	18	27/12/2022	12:21:23	10,2	109,89
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	17	27/12/2022	12:21:12	10,8	115,76
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	16	27/12/2022	12:20:45	27,3	122,32
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	15	27/12/2022	12:20:37	7,96	128,65
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	14	27/12/2022	12:20:28	8,72	137,27
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	13	27/12/2022	12:20:19	8,85	147,16
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	12	27/12/2022	12:20:12	6,89	158,69

b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	11	27/12/2022	12:20:05	7,51	172,49
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	10	27/12/2022	12:19:59	5,55	188,98
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	9	27/12/2022	12:19:53	6,07	209,36
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	8	27/12/2022	12:19:42	11,3	234,78
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	7	27/12/2022	12:19:35	6,58	266,7
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	6	27/12/2022	12:19:28	7	310,06
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	5	27/12/2022	12:19:21	7,13	370,67
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	4	27/12/2022	12:19:02	18,9	461,55
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	3	27/12/2022	11:51:52	1629	609,1
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	2	27/12/2022	12:04:57	101	99,15
b8:27:eb:1b:ff:07	192,168,1,6	raspberrypi2	E22222	M22222	2242	32011	1	10	34	1	27/12/2022	12:03:20	97,3	97,3
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	2	12	34	1	27/12/2022	12:27:53	50,6	50,6
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	34	27/12/2022	12:24:39	22,2	64,02
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	33	27/12/2022	12:24:31	8,17	65,28
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	32	27/12/2022	12:24:20	10,8	67,07
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	31	27/12/2022	12:23:41	38,6	68,88
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	30	27/12/2022	12:23:33	7,95	69,89
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	29	27/12/2022	12:23:32	0,8	72,03
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	28	27/12/2022	12:23:28	4,69	74,57
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	27	27/12/2022	12:23:26	1,2	77,16
b8:27:eb:31:e4:0b	192,168,1,5	raspberrypi3	E33333	M33333	2242	32011	1	10	34	26	27/12/2022	12:23:23	2,99	80,08

b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	25	27/12/20 22	12:23: 21	2,01	83,17
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	24	27/12/20 22	12:23: 14	7,5	86,55
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	23	27/12/20 22	12:22: 23	51	89,98
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	22	27/12/20 22	12:22: 13	9,67	91,76
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	21	27/12/20 22	12:22: 03	10,1	95,67
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	20	27/12/20 22	12:21: 48	14,6	99,94
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	19	27/12/20 22	12:21: 36	11,8	104,44
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	18	27/12/20 22	12:21: 26	10	109,58
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	17	27/12/20 22	12:21: 15	10,7	115,44
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	16	27/12/20 22	12:21: 06	9,55	121,99
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	15	27/12/20 22	12:20: 39	26,7	129,48
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	14	27/12/20 22	12:20: 31	8,2	136,82
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	13	27/12/20 22	12:20: 22	9,02	146,72
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	12	27/12/20 22	12:20: 15	7,1	158,19
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	11	27/12/20 22	12:20: 07	7,81	171,93
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	10	27/12/20 22	12:20: 01	5,6	188,34
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	9	27/12/20 22	12:19: 55	5,83	208,64
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	8	27/12/20 22	12:19: 45	9,87	233,99
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	7	27/12/20 22	12:19: 37	8,08	266,01
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	6	27/12/20 22	12:19: 30	7,01	309
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr y pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	5	27/12/20 22	12:19: 18	12	369,4

b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	4	27/12/20 22	12:19: 06	12	458,75
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	3	27/12/20 22	11:51: 18	1668	607,67
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	2	27/12/20 22	12:05: 08	93,3	77,5
b8:27:eb:31:e 4:0b	192,168, 1,5	raspberr pi3	E3333 3	M3333 3	2242	3201 1	1	10	34	1	27/12/20 22	12:04: 06	61,7	61,7

Anexo 2. Código del sistema de adquisición de tiempos de producción en Python

```
listaDatos = []

class Emisor (object):
    def __init__(self, _DirMAC, _DirIP, _NombreHost):
        self.MAC = _DirMAC
        self.IP = _DirIP
        self.HostName = _NombreHost

#-----0-----
----
#Metodos:
##Funcion para almacenamiento de datos en MySQL
def run_query(query=''):
    import pymysql
    import pymysql.cursors
    credenciales=[DB_HOST,DB_USER,DB_PASS,DB_NAME]

    conectar=pymysql.connect(*credenciales) #conexion DB
    cursor=conectar.cursor()                #cursor
    try:
        cursor.execute(query)                #Ejecuta una consulta SQL
        conectar.commit()                    #finaliza transaccion
    except:
        conectar.rollback()
    #data = None
    #result=cursor.fetchone()
    #print (result)
    finally:
        #cursor.close()
        conectar.close()

#-----
def run_queryr(query=''):
    import pymysql
    import pymysql.cursors
    credenciales=[DB_HOST,DB_USER,DB_PASS,DB_NAME]

    conectar=pymysql.connect(*credenciales) #conexion DB
    cursor=conectar.cursor()                #cursor
    try:
        cursor.execute(query)                #Ejecuta una consulta SQL
        #conectar.commit()                    #finaliza transaccion
    #data = None
```



```

        result1=cursor.fetchone()
    except:
        print("Encountred error while retrieving data from database")
    #cursor.close()
    finally:
        conectar.close()
        return (result1)
#-----
DB_HOST='localhost'
DB_USER='root'
DB_PASS='password'
DB_NAME='Registro_Tiempo'
#-----
def run_queryrM(query=''):
    import pymysql
    import pymysql.cursors
    DB_HOST='raspberrypi4.local'
    DB_PORT='3306'
    DB_USER='admin'
    DB_PASS='Datos'
    DB_NAME='Madre'
    credenciales=[DB_HOST,DB_USER,DB_PASS,DB_NAME]

    conectar=pymysql.connect(*credenciales) #conexion DB
    cursor=conectar.cursor()              #cursor
    try:
        cursor.execute(query)              #Ejecuta una consulta SQL
    #conectar.commit()                    #finaliza transaccion
    #data = None
        result1=cursor.fetchone()
    except:
        print("Encountred error while retrieving data from database")
    #cursor.close()
    finally:
        conectar.close()
        return (result1)
#-----
def run_queryr1f(query=''):
    import pymysql
    import pymysql.cursors
    DB_HOST='localhost'
    DB_USER='root'
    DB_PASS='password'
    DB_NAME='Registro_Tiempo'
    credenciales=[DB_HOST,DB_USER,DB_PASS,DB_NAME]

```

```

conectar=pymysql.connect(*credenciales) #conexion DB
cursor=conectar.cursor()                #cursor
try:
    cursor.execute(query)                 #Ejecuta una consulta SQL
#conectar.commit()                       #finaliza transaccion
#data = None
    result1=cursor.fetchall()
except:
    print("Encountred error while retrieving data from database")
#cursor.close()
finally:
    conectar.close()
    return (result1)
#-----
#Metdo 1: Funcion para tomar y validar el código de usuario
def TomaCodigoUsuario ():
    global Usuario
    Usuario = input("Ingrese codigo de Usuario: ")
    #Extrae primer caracter de usuario y lo almacena en firstValidacion
    firstValidacion = Usuario[:1]
    #Extrae lal longitud del string y lo almacena en secondValidacion
    secondValidacion = len(Usuario)
    #Sentencia ejecuta solo si firstValidacion=="E" and secondValidacion == "6"
    if firstValidacion=="E" and secondValidacion == 6:
        TomaCodigoMaquina ()
    else:
        print ("Ingresa unCodigo de Usuario Valido")
        TomaCodigoUsuario()
#-----0-----
----
#Metdo 2: Funcion para tomar y validar el código de Maquina
def TomaCodigoMaquina ():
    global Maquina
    Maquina = input("Ingrese codigo de Maquina: ")
    #Extrae primer caracter de usuario y lo almacena en firstValidacion
    firstValidacion = Maquina[:1]
    #Extrae lal longitud del string y lo almacena en secondValidacion
    secondValidacion = len(Maquina)
    #Sentencia ejecuta solo si firstValidacion=="E" and secondValidacion == "6"
    if firstValidacion=="E" and secondValidacion == 6:
        global Usuario
        Usuario=Maquina
        TomaCodigoMaquina()
    elif firstValidacion=="M" and secondValidacion == 6:

```

```

        TomaCodigoOperacion()
    else:
        print ("Ingresa unCodigo de Maquina Valido")
        TomaCodigoMaquina ()
#-----0-----
----
#Metdo 3: Funcion para tomar y validar el código de operacion
def TomaCodigoOperacion ():
    global Operacion
    Operacion = input("Ingresa codigo de Operación: ")
    #Extrae primer caracter de usuario y lo almacena en firstValidacion
    firstValidacion = Operacion[:1]
    #Extrae la longitud del string y lo almacena en secondValidacion
    secondValidacion = len(Operacion)
    #Sentencia ejecuta solo si firstValidacion=="E" and secondValidacion == "6"
    if firstValidacion=="E" and secondValidacion == 6:
        global Usuario
        Usuario=Operacion
        TomaCodigoMaquina()
    elif firstValidacion=="M" and secondValidacion == 6:
        global Maquina
        Maquina=Operacion
        TomaCodigoOperacion()
    elif secondValidacion == 5:
        TomaCodigoPaquete ()
    else:
        print ("Ingresa unCodigo de Operacion Valido")
        TomaCodigoOperacion()
#-----0-----
----
#Metdo 4: Funcion para tomar y validar el código de Paquete
def TomaCodigoPaquete ():
    global Paquete
    Paquete = input("Ingresa codigo de Paquete: ")
    #Extrae primer caracter de usuario y lo almacena en firstValidacion
    firstValidacion = Paquete[:1]
    #Extrae la longitud del string y lo almacena en secondValidacion
    secondValidacion = len(Paquete)
    # Busca si existe "," en el string y lo almacena en la variable thirdValidacion
    thirdValidacion = Paquete.find(",")
    #Sentencia ejecuta solo si firstValidacion=="E" and secondValidacion == "6"
    if firstValidacion=="E" and secondValidacion == 6:
        global Usuario
        Usuario=Paquete
        TomaCodigoMaquina()

```

```

elif firstValidacion== "M" and secondValidacion == 6:
    global Maquina
    Maquina=Paquete
    TomaCodigoOperacion()
elif thirdValidacion != -1:
    separador()
    TomaTiempoPiezasCambio ()
elif secondValidacion == 5:
    global Operacion
    Operacion = Paquete
    TomaCodigoPaquete ()

else:
    print ("Ingresa unCodigo de Paquete Valido")
    TomaCodigoPaquete()
#-----0-----
----
#Metdo 4.1: Funcion para extraer lote, paquete, talla, cantidad del codigo de Paquete
def separador():
    lc=len(Paquete)
    #separador(Paquete)
    #Ejecuta la funcion reciclada separador
    global lote, paquete,talla,cant
    indice = Paquete.find(",")
    lote = Paquete[0 : indice]
    #print("Lote: ",lote)

    c2 = Paquete[indice + 1 : ]
    indice = c2.find(",")
    paquete = c2[0 : indice]
    #print("Paquete: ",paquete)

    c3 = c2[indice + 1 :]
    indice = c3.find(",")
    talla = c3[0 : indice]
    #print("Talla: ",talla)

    c4 = c3[indice + 1 :]
    cant = c4[0 :]
    #print("Cantidad: ",cant)
#-----
#def FuncionUpdate():
    #import pymysql
    #import pymysql.cursors
    #import time

```

```

#-----
def Vconexion ():
    import pymysql
    import time
    conexion =0
    DB_HOST='raspberrypi4.local'
    DB_PORT='3306'
    DB_USER='admin'
    DB_PASS='Datos'
    DB_NAME='Madre'
    credenciales=[DB_HOST,DB_USER,DB_PASS,DB_NAME]
    #time.sleep(1)
    while conexion==0:
        time.sleep(1)
        try:
            conectar=pymysql.connect(*credenciales)
            #print("conectado")
            conectar.close()
        except:
            #print("no conectado")
            continue
        else:
            queryId = ("SELECT id FROM TPulsaciones ORDER BY id DESC LIMIT 1")
            lastID = run_queryr(queryId) ##pasar a numero
            if lastID == None:
                continue
            else:
                lastId = list(lastID)
                idL = lastId[0]

                queryMadre = ("SELECT id FROM Integracion WHERE Mac= '%s' AND id=
%s"%(MAC, idL)
                lastm = run_queryrM(queryMadre) ##pasar a numero

                if lastm == None:
                    print("update")

                queryMadre = ("SELECT id FROM Integracion WHERE Mac= '%s' ORDER
BY id DESC LIMIT 1")%(MAC)
                lastm = run_queryrM(queryMadre) ##pasar a numero
                #insertar en base local lastm

                if lastm == None:
                    idM = 0

```

```

else:
    lastListM =list( lastm)
    idM = lastListM[0]

    queryLocalFull = ("SELECT * FROM TPulsaciones WHERE Mac= '%s' AND
id > %s")%(MAC, idM)
    lastLFull = run_queryr1f(queryLocalFull) ##pasar a numero

    fullPulsaciones = list(lastLFull)

    fullPulsacionesSplit = list()
    length = 100

    for i in range(0, len(fullPulsaciones), length):
        fullPulsacionesSplit.append(fullPulsaciones[i:i+length])

    #print(fullPulsacionesSplit)

    for pulsaciones in fullPulsacionesSplit:

        import pymysql
        import pymysql.cursors
        DB_HOST='raspberrypi4.local'
        DB_PORT='3306'
        DB_USER='admin'
        DB_PASS='Datos'
        DB_NAME='Madre'
        credenciales=[DB_HOST,DB_USER,DB_PASS,DB_NAME]

        conectar=pymysql.connect(*credenciales) #conexion DB

        with conectar.cursor() as cursor:
            try:
                cursor.executemany("INSERT INTO
Integracion(CodigoOperacion,CodigoBarras,Lote,Paquete,Talla,Cantidad,Pulsaciones,Hora
,Promedio,id,FechaAc,Mac,Ip,Hostname,Usuario,Maquina,TiempoCronometro)
VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", pulsaciones )
                conectar.commit()
            except:
                conectar.rollback
            finally:
                conectar.close()

#.....

```

```

#-----0-----
----
#Metdo 5: Funcion para tomar ultimos datos y guardar en variables de los datos
registrados de la ultima fila de mysql
def TomaUltimosDatosRegistrados():
    import pymysql
    import pymysql.cursors
    import time
    from threading import Thread
    #Recoge Utimos datos de la ultima fila de la base de datos mysql y los almacena
    en el mismo orden en el que se enviaron con las mismas variables
    global Usuario, Maquina, Operacion, Paquete, lote, paquete, talla, cant, Tiempo, Pieza, idL

    queryLast = "SELECT Usuario, Maquina, CodigoOperacion, CodigoBarras, Lote,
Paquete, Talla, Cantidad, TiempoCronometro, FechaAc, pulsaciones, Hora, id FROM
TPulsaciones ORDER BY id DESC LIMIT 1"
    lastL = run_queryr(queryLast) ##pasar a numero
    if lastL == None:
        def integracion():
            Vconexion()
            hilo = Thread(target=integracion)
            hilo.start()
            TomaCodigoUsuario ()
    else:
        lastList =list( lastL)
        #print(lastList)
        Usuario = lastList[0]
        Maquina = lastList[1]
        Operacion= lastList[2]
        Paquete= lastList[3]
        lote= lastList[4]
        paquete=lastList[5]
        talla=lastList[6]
        cant=lastList[7]
        Tiempo = lastList[8]
        Pieza = lastList[10]
        idL = lastList[12]
        time.sleep(1)
        def integracion():
            Vconexion()
            hilo = Thread(target=integracion)
            hilo.start()
            TomaTiempoPiezasEncendido()

```

```

#-----0-----
----
#Metdo 6: Funcion para tomar y validar tiempos de producción
def TomaTiempoPiezasEncendido ():
    global contador
    contador = 0

    if Pieza == 0:
        contador = 1
        Print("No existe piezas pendientes porfavor ingresa nuevo paquete")
    else:
        contador = Pieza +1
        print("Continuacion.....")
        #print()
        #print("Ultimo Paquete registrado:")
        print("Ultima pieza registrada:>>>",Pieza,"De",cant)
        print("Lote:",lote,"/", "Paquete:",paquete,"/", "Talla:",talla,"/", "Cantidad:",
cant,)
        #print("Ultima pieza registrada:>>>",Pieza)
        #print()
        print("-----")
        print("Siguiete Pieza.....")
        #print("
                                <<Pieza #",contador,">>")
        print("
                                <<Pieza #",contador,"De",cant,">>")
        print("Lote:",lote,"Paquete:",paquete,"Talla:",talla)
        #contador=contador+1
        EjecucionCronometro()

def TomaTiempoPiezasCambio ():
    global contador
    contador = 1
    print("-----0-----")
    #print("
                                <<Pieza #",contador,">>")
    print("
                                <<Pieza #",contador,"De",cant,">>")
    print("Lote:",lote,"Paquete:",paquete,"Talla:",talla)

    #contador = contador +1
    EjecucionCronometro()

#ejecuta programa Validacion para romper bucle de toma de tiempo
def EjecucionCronometro ():
    import RPi.GPIO as GPIO
    import pymysql
    import pymysql.cursors

```



```

import datetime
import time
from threading import Thread
from curses import ascii
global Sensor,CodigoNuevo
CodigoNuevo=None
#CodigoNuevo=None
#CodigoNuevo1 = None
#Define variables del pin GPIO
Sensor=32      #GND 30
#Apagar=36     #GND 34
#Configurar GPIO con el modelo BOARD
GPIO.setmode(GPIO.BOARD)
#GPIO.setup(Apagado, GPIO.IN)
GPIO.setup(Sensor, GPIO.IN)
#GPIO.setup(Apagar, GPIO.IN)
tiempo=time.strftime('%Y/%m/%d-%H:%M:%S')
print("FechaInicio:",end="")
fecha = str(tiempo[0:10]) #print (fecha)
print (fecha)
hora = str(tiempo[11:19]) #print (hora)
print ("HoraInicio:",hora)
def paro():
    global CodigoNuevo
    CodigoNuevo1 = input()
    if CodigoNuevo1 == Paquete:
        print("*YA ESTAS EN ESE PAQUETE*")
        paro()
    if CodigoNuevo1 == Operacion:
        print("*YA ESTAS EN ESA OPERACION*"),)
        paro()
    if CodigoNuevo1 == Maquina:
        print("*YA ESTAS EN ESA MAQUINA*")
        paro()
    if CodigoNuevo1 == Usuario:
        print("*UN GUSTO SALUDARTE NUEVAMENTE*")
        paro()
    else:
        #global CodigoNuevo
        CodigoNuevo=CodigoNuevo1
        time.sleep(0.3)

hilo = Thread(target=paro)
hilo.start()
inicio = time.time()

```



```

tiempo=time.strftime('%Y/%m/%d-%H:%M:%S')
global fecha,incio, hora
print("FechaInicio:",end="")
fecha = str(tiempo[0:10]) #print (fecha)
print (fecha)
hora = str(tiempo[11:19]) #print (hora)
print ("HoraInicio:",hora)
global contador
#Inicia cronometro
inicio = time.time()
while lec_Sensor == 0:
    #time.sleep(0.3)
    lec_Sensor = GPIO.input(Sensor)
else:
    time.sleep(0.3)
    continue

else:
    continue

#Validacion para actualizar las variables Usuario, Maquina, Operacion o Paquete
firstValidacion=""
secondValidacion=0
thirdValidacion= False
#"Extrae primer caracter de Pieza y lo almacena en firstValidacion"
firstValidacion = CodigoNuevo[:1]
#Extraer la longitud de Usuario y almacenalo en secondValidacion
secondValidacion = len(CodigoNuevo)
#Buscar si existe "," y almacenalo en thirdValidacion
thirdValidacion = CodigoNuevo.find(",")
#Sentencia ejecuta solo si firstValidacion = "E" and secondValidacion=6
if firstValidacion=="E" and secondValidacion == 6:
    global Usuario
    Usuario = CodigoNuevo
    TomaCodigoMaquina()
elif firstValidacion=="M" and secondValidacion == 6:
    global Maquina
    Maquina = CodigoNuevo
    TomaCodigoOperacion()
elif thirdValidacion != -1:
    global Paquete
    Paquete = CodigoNuevo
    separador()
    TomaTiempoPiezasCambio ()
elif secondValidacion == 5:

```

```

        global Operacion
        Operacion = CodigoNuevo
        TomaCodigoPaquete()
    else:
        #Imprime ("Ingresa unCodigo de Paquete Valido")
        print("Ingresa unCodigo Valido")
        EjecucionCronometro()

def main():
    #Importar librerias necesarias (Modulos)
    import datetime
    import socket
    import getmac
    #Encerar Variables
    global MAC
    global IP
    global HostName
    MAC= 0
    IP=0
    HostName=""
    #Define Variables
    MAC= getmac.get_mac_address()
    Ip= socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    Ip.connect(("8.8.8.8", 80))
    IP = Ip.getsockname()[0]
    HostName= socket.gethostname()
    print("-----")
    print("Datos de Dispositivo >>>")
    print("Ip:",IP,end="")
    print("/ Hostname:",HostName)
    print("Mac:",MAC)
    print("-----")
    ObjRaspberry = Emisor(MAC, IP, HostName)
    listaDatos.append(ObjRaspberry)
    TomaUltimosDatosRegistrados()

if __name__ == '__main__':
    #main()
    main()

```

Anexo 3. Código visualización web (emisor)

```
<?php
$conexion=mysqli_connect("localhost","root","password");
mysqli_select_db($conexion,"Registro_Tiempo");

$sql="SELECT * FROM TPulsaciones ORDER BY id DESC";
$resultado=mysqli_query($conexion,$sql)or die(mysql_error());
$celdas=array();
while($rows=mysqli_fetch_assoc($resultado)){
    $celdas[]=$rows;
}
mysqli_close($conexion);

if(isset($_POST["export_data"])){
/*Se establecen los encabezados para que el navegador interprete que descargará un
archivo de Excel.*/

header('Content-type: application/vnd.ms-excel;charset=utf-8');
header("Content-Disposition: attachment; filename=Datos_Pi2.xls");
header("Pragma: no-cache");
header("Expires: 0");

/*Se construye una tabla HTML*/
$print = "<table>";
$print .= "<tr>";
$print .= "<td>";
$print .= "Valor 1";
$print .= "</td>";
$print .= "<td>";
$print .= "Valor 2";
$print .= "</td>";
$print .= "<td>";
$print .= "Total";
$print .= "</td>";
$print .= "</tr>";
$print .= "<tr>";
$print .= "<td>";
$print .= "10";
$print .= "</td>";
$print .= "<td>";
$print .= "20";
$print .= "</td>";
$print .= "<td>";
```

```

$print .= "=suma(A2:B2)";
$print .= "</td>";
$print .= "</tr>";
$print .= "</table>";

echo $print;

}
?>

<!DOCTYPE html>
<html>
<title>Tiempos de Procesamiento</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aowXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
</head>
<body>

<h3><font size=5><center><h3><font size=5><center>REGISTRO DE TIEMPOS DE
PRODUCCIÓN</center></font></h3>
</center></font></h3>
<br><hr align="center" size=5 noshade><br>
<div id="div1" style="width:500px; height:10px; overflow:auto;"></div>

<div class="container">
  <h3><center>Tabla de Códigos de Barra y Tiempos</center></h3>
  <div class="well-sm col-sm-12" style="text-align:center !important;">
    <div class="btn-group pull-right">
      <form action="<?php echo $_SERVER["PHP_SELF"]; ?>" method="post">
        <button type="submit" id="export_data" name='export_data'
          value="Export to excel" class="btn btn-info">Exportar a Excel</button>
      </form>
    </div>
  </div>
  <br>

  <table class="table" align="center" border="1" cellspacing="1" cellpadding="1"
style="font-size:10pt" >
    <thead class="thead-dark">
      <tr>
        <th width="80"><div align="center"><b>Dirección MAC</b></font></div></th>
        <th width="80"><div align="center"><b>Dirección IP</b></font></div></th>

```

```

        <th width="80"><div align="center"><b>Host name de
dispositivo</b></font></div></th>
        <th width="80"><div align="center"><b>Código
Usuario</b></font></div></th>
        <th width="80"><div align="center"><b>Código
Máquina</b></font></div></th>
        <th width="80"><div align="center"><b>Código
Operación</b></font></div></th>
        <th width="80"><div align="center"><b>Lote</b></font></div></th>
        <th width="80"><div align="center"><b>Paquete</b></font></div></th>
        <th width="80"><div align="center"><b>Talla</b></font></div></th>
        <th width="80"><div align="center"><b>Cantidad</b></font></div></th>
        <th width="80"><div align="center"><b>Pieza</b></font></div></th>
        <th width="80"><div align="center"><b>Fecha</b></font></div></th>
        <th width="80"><div align="center"><b>Hora</b></font></div></th>
        <th width="80"><div align="center"><b>Tiempo cronometro
[s]</b></font></div></th>
        <th width="80"><div align="center"><b>Promedio</b></font></div></th>

    </tr>
</thead>
<tbody>
<?php foreach($celdas as $celda){?>
    <tr>
        <td><div align="center"><?php echo $celda ['Mac'];?></div></td>
        <td><div align="center"><?php echo $celda ['Ip'];?></div></td>
        <td><div align="center"><?php echo $celda
['Hostname'];?></div></td>
        <td><div align="center"><?php echo $celda
['Usuario'];?></div></td>
        <td><div align="center"><?php echo $celda
['Maquina'];?></div></td>
        <td><div align="center"><?php echo $celda
['CodigoOperacion'];?></div></td>
        <td><div align="center"><?php echo $celda ['Lote'];?></div></td>
        <td><div align="center"><?php echo $celda
['Paquete'];?></div></td>
        <td><div align="center"><?php echo $celda ['Talla'];?></div></td>
        <td><div align="center"><?php echo $celda
['Cantidad'];?></div></td>
        <td><div align="center"><?php echo $celda
['Pulsaciones'];?></div></td>
        <td><div align="center"><?php echo $celda
['FechaAc'];?></div></td>
        <td><div align="center"><?php echo $celda ['Hora'];?></div></td>

```

```
        <td><div align="center"><?php echo $celda
['TiempoCronometro'];?></div></td>
        <td><div align="center"><?php echo $celda
['Promedio'];?></div></td>
    </tr>
    <?php } ?>
</tbody>
</table>
</div>

</table>
</body>
</html>
```


Requerimientos del sistema

El presente formulario está diseñado para realizar una priorización a los requerimientos mencionados por las partes interesadas.

1. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Eficacia en la adquisición de datos en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

2. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Eficiencia y ergonomía en la pulsación para envío de datos, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

3. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Fiabilidad en los datos recolectados, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

4. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Integración en más de un punto de la línea de producción, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

5. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Visualización remota de los datos en cualquier parte del mundo, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

6. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Interface gráfica amigable para el operario, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

7. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Eficiencia operativa en la visualización de datos en la pantalla, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

8. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Autonomía en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

9. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted la Inicio en los últimos datos registrados, en el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5

10. En una escala donde 1 es no prioritario y 5 prioritario. Indique por favor.
¿Qué grado de importancia es para usted que sea plug and play (rápido y fácil de usar) el sistema de adquisición proporcionado en la empresa?

1 2 3 4 5