

**SEP**

**Tecnológico Nacional De  
México**

**Instituto Tecnológico de Tijuana  
Departamento de Ingeniería Industrial**



**SISTEMA DE CONTROL VISUAL DIGITAL DE CONECTORES Y DIAGRAMAS  
ELÉCTRICOS EN LA EMPRESA GONZÁLEZ MACHINE SHOP**

**Trabajo de tesis presentado por:**

**Bryan Anselmo Solis Magallanes**

**Para obtener el grado de:**

**Maestro en Ingeniería Industrial**

**Director de tesis:**

**Daniel Eduardo Hernández Morales**

**Co-director de tesis:**

**Adrián Rodríguez Aguiñaga**

**Tijuana, Baja California, México.**

**Enero de 2021**



Tijuana Baja California, 18/enero/2021

Asunto: Solicitud de autorización de impresión de tesis

**DRA. YAZMIN MALDONADO ROBLES**  
**JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**  
**PRESENTE**

En lo referente al trabajo de tesis **“SISTEMA DE CONTROL VISUAL DIGITAL DE CONECTORES Y DIAGRAMAS ELÉCTRICOS EN LA EMPRESA GONZÁLEZ MACHINE SHOP”**, presentado por el C. **Bryan Anselmo Solís Magallanes**, alumno del programa de Maestría en Ingeniería Industrial, con número de control **G18212047**; informamos a usted que después de una minuciosa revisión e intercambio de opiniones, los miembros del comité manifiestan **APROBAR LA TESIS**, atendiendo las disposiciones de los Lineamientos para la Operación de Estudios de Posgrado del Tecnológico Nacional de México, por lo que se autoriza al interesado para que proceda a la impresión del mismo.  
Sin más por el momento quedamos a sus órdenes.

**A T E N T A M E N T E**  
**Excelencia en Educación Tecnológica®**  
**Por una Juventud Integrada al Desarrollo de México®**

**DR. DANIEL EDUARDO HERNÁNDEZ MORALES**  
**PRESIDENTE**

**DR. ADRIAN RODRIGUEZ AGUIÑAGA**  
**SECRETARIO**

**DR. LUIS MUÑOZ DELGADO**  
**VOCAL**

**MC. VICTOR RAÚL LÓPEZ LÓPEZ**  
**SUPLENTE**

c.p. Archivo  
c.c.p.Dra. Lilia Cristina Morales Ochoa- Coordinadora de posgrado





**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

Instituto Tecnológico de Tijuana

Tijuana, Baja California,

19/enero/2021

OFICIO No. 005/DEPI/2021

Asunto: Autorización de Impresión de Tesis

**MARIBEL GUERRERO LUIS**  
**JEFA DEL DEPARTAMENTO DE SERVICIOS ESCOLARES**  
**PRESENTE**

En lo referente al trabajo de tesis, "Sistema de control visual digital de conectores y diagramas electrónicos en la empresa González Machine Shop". Presentado por C. Bryan Anselmo Solís Magallanes, alumno de la Maestría en Ingeniería Industrial con número de control G18212047; informo a usted que a solicitud del comité de tutorial, tengo a bien Autorizar la impresión de Tesis, atendiendo las disposiciones de los Lineamientos para la Operación de Estudios de Posgrado del Tecnológico Nacional de México.

Sin más por el momento le envío un cordial saludo.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica-*  
*Por una juventud integrada al desarrollo de México-*

**YAZMIN MALDONADO ROBLES**  
**JEFA DE DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**  
**DIVISIÓN DE ESTUDIOS DE POSGRADO**  
**E INVESTIGACIÓN**

ccp. Archivo  
YMR/lap.



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



**INSTITUTO TECNOLÓGICO DE TIJUANA**



Calzada del Tecnológico 2001 Esq. Av. Chapultepec Chapultepec  
y Calle Guadalupe 200, Fraccionamiento Guadalupe, C.P. 22060,  
Tijuana, Baja California.  
Teléfono: (664) 441-1111 Correo electrónico: [itjt@tecnologico.mx](mailto:itjt@tecnologico.mx)  
[www.itecnologico.mx](http://www.itecnologico.mx)



Tijuana, B.C. a 20/01/2021

### Carta de cesión de derechos

Por medio de la presente, quien suscribe, Ing. Bryan Anselmo Solis Magallanes; alumno(a) del Programa de Maestría en Ingeniería Industrial con número de control G18212047, me permito informarle que declaro mi conformidad para ceder los derechos de reproducir y distribuir copias en su totalidad o en partes de la tesis **Sistema de control visual digital de conectores y diagramas eléctricos en la empresa Gonzalez Machine Shop** al Tecnológico Nacional de México/IT de Tijuana, mencionando la fuente. Esta tesis certifico que es un trabajo original de mi autoría, bajo la dirección del Dr. Daniel Eduardo Hernández Morales y codirección del Dr. Adrian Rodriguez Aguinaga.

Lo anterior con el fin de que sea publicado y difundido por el Tecnológico Nacional de México/IT de Tijuana en los medios que considere convenientes. Firmo la presente en la ciudad de Tijuana a los catorce días del mes de julio del año dos mil veinte.



Ing. Bryan Anselmo Solis Magallanes

## **DEDICATORIA**

### **A mi familia**

El presente proyecto le agradezco a Dios por permitirme llegar a este momento tan importante de mi vida y el cual lo dedico principalmente a mis padres y hermanas los cuales han sido pilares morales y económicos en mi formación profesional lo largo de todos mis logros académicos. Quiero además dedicar este proyecto a mi esposa la cual ha sido base fundamental para darme ánimos, apoyarme en la toma de decisiones que han influido en la continua mejora académica. Especialmente quiero dedicar también este proyecto y logro a mis hijos, mostrándoles que el camino de la educación y la preparación académica continua es importante en sus vidas no solo como reto profesional sino también como puerta de aprendizaje al mar de conocimiento en el cual busquen desarrollarse.

### **A mis amigos**

Les dedico el proyecto a mis compañeros de clase y amigos que me han apoyado con su conocimiento para poder integrar este proyecto de tesis, por su paciencia y palabras de aliento para continuar con mi formación académica.

## **AGRADECIMIENTOS**

### **A mis maestros**

Agradezco con toda sinceridad su paciencia, enseñanzas y tiempo el cual invirtieron en mí creyendo en mi futuro para poder obtener el título de Maestro. Ellos son fundamentales en el proceso de mi formación académica, ya que sin su conocimiento y enseñanzas no hubiera podido llegar a este gran momento de mi vida.

### **A la empresa en que laboro**

La cual me ha brindado la oportunidad de participar en grandes proyectos de mejora continua los cuales se han unido con mis proyectos académicos. Adicional agradezco su tiempo y disponibilidad para poder ejercer este proyecto.

## RESUMEN

La empresa TE CONNECTIVITY dedicada a la fabricación de conectores eléctricos para uso subacuático, su proceso de manufactura pasa por diferentes etapas como lo son soldadura, prueba eléctrica, moldeo, pruebas de presión entre otras. Durante la etapa de preparación para moldeo se presenta el problema de acomodo incorrecto de los cables (*hook up incorrecto*) por esta razón la empresa necesitaba desarrollar una plataforma digital la cual sirva para reducir el error humano en la secuencia errónea de cables dentro de conector durante el proceso de moldeo, lo cual ocasiona que se tenga que retrabajar la pieza. Esto afectaba principalmente a los métricos de la compañía: entregas a tiempo, retrabajos y costo de mano de obra.

TE CONNECTIVITY a manera de dar solución a la problemática de acomodo incorrecto de los cables (*hook up incorrecto*) decidió realizar un proceso de licitación en el cual la empresa Gonzalez Machine Shop dedicada a la manufacturación y automatización de procesos obtuvo la licitación de este proyecto para solucionar la problemática planteada.

Para dar solución a dicha problemática Gonzalez Machine Shop desarrolló un sistema de control visual digital, el cual es un diseño que mediante una computadora Raspberry PI se programa para desplegar diagramas visuales claros, actualizados y sencillos de entender al nivel operador.

Esta mejora pretende dar a la compañía un ahorro de \$25,000 dólares anuales los cuales están distribuidos en mejora de tiempo de operación, reducción de piezas rechazadas y mejora de envíos al cliente. Para lograr estas mejoras se pretenden instalar 5 equipos Raspberry PI 4 con pantallas en sus 5 estaciones de ensamble/moldeo de conectores, dicha área es donde se focaliza el problema que se pretende mejorar con esta propuesta.

## TABLA DE CONTENIDO

RESUMEN	4
CAPÍTULO 1. INTRODUCCIÓN	10
1.1 Planteamiento de problema	15
1.2 Justificación	15
1.3 Hipótesis	16
1.4 Objetivos	16
Objetivo general	16
Objetivos específicos	16
1.5 Alcances	17
1.6 Organización de la tesis	17
CAPÍTULO 2. MARCO TEÓRICO	18
2.1 Antecedentes de la investigación	18
2.2. Usos del Raspberry PI 4	18
2.3 Industria 4.0	19
2.4 MyOpenLab	21
2.5 Ayudas visuales	21
2.6 Herramientas de control visual digital.	22
2.6.1 Tipos de herramientas de control visual.	22
2.7 Lenguaje Python	24
2.8 Gráficos librería Tkinter.	25
2.9 Base de datos MariaDB.	26
2.10 Web PHP y HTML.	27
2.11 Web Bootstrap 4.	27
CAPÍTULO 3. MATERIALES Y MÉTODOS	28



3.1 Descripción del proceso de ensamble de conectores y proceso de moldeo.	28
3.2 Uso de herramientas de control visual digital como solución.	32
3.3 Herramientas de control visual digital como solución.	33
3.4 Implementación de sistema de control visual digital	35
Fase 1. Verificación y registro de datos de afectación por falta de herramienta de control visual digital	36
Fase 2. Selección de números de parte principales de Hook Up incorrecto.	37
Fase 3. Causas de origen de defecto Hook Up incorrecto.	38
Fase 4. Fase de implementación del proyecto.	39
<b>CAPÍTULO 4. RESULTADOS DE LA INVESTIGACIÓN</b>	<b>47</b>
4.1 Resultados primera etapa de implementación en área de moldeo.	48
4.2 Resultados segunda etapa de implementación en área de moldeo.	50
<b>CAPÍTULO 5. CONCLUSIÓN Y TRABAJOS FUTUROS</b>	<b>55</b>
<b>BIBLIOGRAFÍA</b>	<b>56</b>
<b>ANEXOS</b>	<b>57</b>
Anexo 1: Instrucción digital para número de parte MCL-S003-0277	57
Anexo 2: Instrucción digital para número de parte MIN-S006	58
Anexo 4: Instrucción digital para número de parte LP-S015	60
Anexo 5: Instrucción digital para número de parte LP-S049-0018	61
Anexo 6: Código de programación de Registro de usuario	62
Anexo 7: Código de programación de Registro de usuario	65
Anexo 8: Código de pantalla principal	68
Anexo 9: Código de modificación de diagramas	72
Anexo 10: Código de administrador del programa	78

## LISTA DE TABLAS

Tabla 1 Proceso actual de ensamble de conectores de moldeo previo a proceso de moldeo.	11
Tabla 2 Proceso propuesto de ensamble de conectores de moldeo previo a proceso de moldeo.	13
Tabla 3. Costo e impactos negativos por falta de herramienta de control visual digital	15
Tabla 4. Ejemplo de registro diario de defectos y rendimiento por día por número de parte.	34
Tabla 5. Ejemplo de registro semanal de defectos y rendimiento por día por número de parte.	34
Tabla 6. Tabla con los hallazgos por semana de las posibles causas del defecto <i>Hook Up incorrecto</i> .	37
Tabla 7. Rechazos por <i>Hook up</i> incorrecto en principales ofensores.	42
Tabla 8. Resultados preliminares de evaluación de funcionalidad de pruebas de ingeniería, en el cual se obtuvo un 88% de aceptación, considerando que los defectos detectados no son atribuibles a la implementación, la aceptación sería del 100%.	43
Tabla 9. Costo de desperdicio durante la primera implementación.	44
Tabla 10. Ahorro de 84% registrado durante la primera etapa de implementación de proyecto comparado con la situación inicial.	44
Tabla 11. Resultados preliminares de evaluación de funcionalidad de segunda implementación de pruebas de ingeniería, en el cual se obtuvo un 84% de aceptación, considerando que los defectos detectados no son atribuibles a la implementación, la aceptación sería del 100%.	45
Tabla 12. Costo de desperdicio durante la segunda implementación (4 semanas).	46
Tabla 13. Ahorro de 88% registrado durante la segunda etapa de implementación de proyecto comparado con el inicio del proyecto.	46

Tabla 14. Proyección de ahorros totales en caso de implementar sistema en las áreas de moldeo.	48
--	----

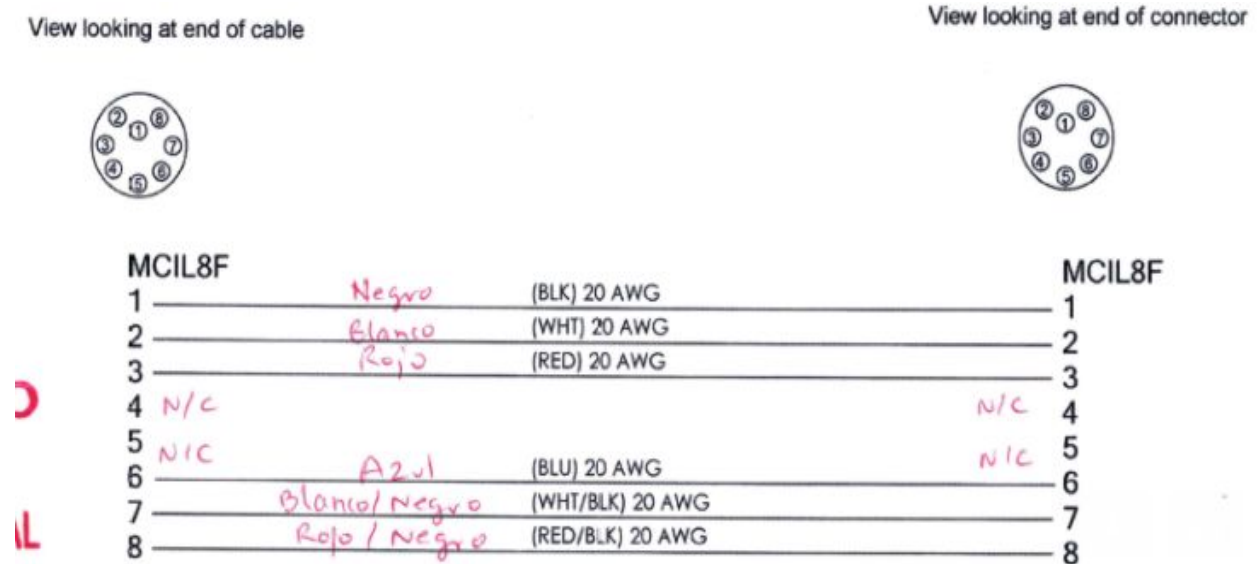
## LISTA DE FIGURAS

Figura 1. Ejemplo de dibujos no claros en el área de producción.	11
Figura 2. Figura ilustrativa de un Raspberry PI 4.	18
Figura 3. Imágenes ilustrativas de herramientas de control visual.	23
Figura 4. Contactos pintados antes del proceso de moldeo.	27
Figura 6. Colocado de pieza en el molde.	28
Figura 7. Molde colocado en la prensa durante el proceso de Moldeo.	28
Figura 8. Pieza Moldeada.	29
Figura 9. <i>Alerta de calidad</i> ocasionada por Hook Up incorrecto.	29
Figura 10. <i>Rechazo</i> ocasionado por Hook Up incorrecto	30
Figura 11. Defectos cosméticos en piezas moldeadas.	30
Figura 12. La gráfica muestra el comportamiento de defecto <i>Hook Up</i> por semana, colocándolo el número de parte MIN-S006 como el principal.	37
Figura 13. Distribución por porcentaje de las razones causantes de <i>Hook Up incorrecto</i> .	38
Figura 14. Dibujo documentado con Hook Up.	39
Figura 15. Dibujo con redlines para actualización en el departamento de diseño.	40
Figura 16. Dibujo con falta de información.	40
Figura 17. Dibujo hecho a mano de Hook Up.	41
Figura 18. Dibujo con notas en inglés.	41
Figura 19. Registro de usuario en sistema	43
Figura 20. Inicio de sesión en sistema.	43
Figura 21. Pantalla de ayuda visual en sistema.	44
Figura 22. Pantalla de modificación de diagramas	45

Figura 23. Pantalla empleados registrados en servidor local administrador.	45
Figura 24. Pantalla diagramas registrados en servidor local administrador.	46
Figura 25. Dashboard con reporte de piezas.	46
Figura 26. Raspberry Pi 4 con pantalla instalada. Imagen de referencia	47
Figura 27. La gráfica comparativa que muestra el comportamiento de <i>Hook up</i> en área de moldeo desde la situación inicial hasta la conclusión de las 2 etapas de mejora realizadas, donde se observa una reducción que representa un 88% de reducción en el defecto.	53
Figura 28. Se muestra el comportamiento del FPY del área de moldeo antes y después de la implementación del sistema Raspberry PI 4, quedando ubicada en 88% lo cual es atribuible a la reducción por <i>hook up</i> .	54

## CAPÍTULO 1. INTRODUCCIÓN

La empresa González Machine Shop ubicada en la colonia Guaycura de la ciudad de Tijuana Baja California, se dedica al maquinado industrial y desarrollo de sistemas de automatización. En esta empresa se llevó a cabo el proyecto de un sistema de control visual digital para la problemática del error de secuencia incorrecta de cables (hook up incorrecto) durante el moldeo de las piezas, este problema ocurre principalmente por que los diagramas electrónicos que se presentan en los dibujos no son claros para el operador lo cual ocasiona que se equivoque, además en los dibujos solo se les proporciona la vista de la pieza de manera frontal y ellos al preparar la pieza para moldear deben voltear esa secuencia ya que ellos la preparan desde una vista trasera de la pieza, esto también ocasiona que los operadores se equivoquen al confundir la vista que se presenta en el dibujo. Para ejemplo ver figura 1.



**Figura 1. Ejemplo de dibujos no claros en el área de producción.**

Para lograr llegar a la solución final se realizó un estudio de factibilidad donde se planteó primero una lluvia de ideas en conjunto con Ingenieros de proceso, líderes de producción y operadores del ensamble y se llegó a la conclusión de implementar un control visual digital de fácil entendimiento de los operadores ya que los diagramas al

no ser claros, estar en otro idioma y estar en forma de papelería se prestaban para errores de ensamble por parte de los operadores.

Después de dicho análisis se presentaron los posibles escenarios y plataformas con los cuales se podía desarrollar la solución a la problemática, por lo cual se llegó a la conclusión de utilizar un sistema de control visual digital donde se mostrarán los diagramas simplificados y traducidos al español para mayor entendimiento del operador, dichos diagramas se mostrarían en un sistema Raspberry PI 4 con su pantalla, montado en una base de acero 304 con un diseño ergonómico ajustable para los operadores en este proceso.

Con esta implementación se espera reducir drásticamente este modo de falla y evitar las grandes pérdidas económicas, las cuales se despliegan en los siguientes métricos:

- Piezas a la primera FPY (por sus siglas en inglés, First Pass Yield): 8% (de 100% de rechazos)
- Piezas enviadas a tiempo STS (por sus siglas en inglés, Ship to schedule) STS (Piezas enviadas a tiempo): 15% (en riesgo)
- Costo de mala calidad COPQ (por sus siglas en inglés, Cost of poor quality) COPQ : 22% (de 100% reemplazos)
- Costo por afectación total en este año fiscal 2020: \$25,254 dólares

Para dicha problemática previamente mencionada se realizó un listado de las operaciones de proceso, donde se desplegaba paso a paso la secuencia de ensamble, esto para un mayor entendimiento del problema y una lluvia de ideas sólida y eficaz para el problema que se presentaba en la secuencia y ensamble de diagramas eléctricos, dichos pasos se muestran en la tabla 1.

**Tabla 1 Proceso actual de ensamble de conectores de moldeo previo a proceso de moldeo.**

<b>PASOS</b>	<b>OPERACIONES</b>	<b>DESCRIPCIONES</b>
1	Se reciben piezas en la estación de moldeo.	<ul style="list-style-type: none"> <li>Las piezas deben estar previamente soldadas con los contactos y pintadas.</li> </ul>
2	Preparación de moldeo y acomodo de cables.	<ul style="list-style-type: none"> <li>Los cables internos se deben acomodar conforme a la secuencia de colores de acuerdo a lo solicitado por el cliente (cualquier nota, documento o dibujo que ayude a descifrar el requerimiento).</li> </ul>
3	Los cables previamente acomodados se colocan dentro del molde.	<ul style="list-style-type: none"> <li>Cables son acomodados dentro de molde para proceso de moldeo</li> </ul>
4	Proceso de moldeo	<ul style="list-style-type: none"> <li>La temperatura puede ser de 200°F a 300°F.</li> <li>Tiempo desde 30 minutos a 3 horas.</li> <li>Presión de prensa desde 200 psi hasta 500 psi.</li> </ul>
5	Se desmolda pieza	<ul style="list-style-type: none"> <li>Pieza es retirada de molde y se pone enfriamiento por 5 minutos.</li> </ul>

De las operaciones listadas en la Tabla 1, las operaciones 2 y 3 son las más críticas ya que ellas dependen totalmente de la experiencia del operador, provocando posibles errores y defectos de Calidad. Dichas operaciones son completamente manuales y dependen totalmente de la experiencia del operador y que al momento de que se recibe una orden de trabajo en la estación de moldeo se revisa la información

disponible en las hojas viajeras que tiene la orden, dicha información al no ser clara, estar en otro idioma y probablemente se pierde la hoja durante el proceso, son principales factores para un acomodo erróneo de la secuencia de los cables.

Por lo tanto, la mejora planteada en este proyecto consiste en implementar un indicador de control visual, el cual transmita la información clara para el operador poder procesarlo de forma correcta y evitar los errores en el acomodo durante el proceso. Las mejoras obtenidas al lograr implementar este proyecto son:

- Información clara y sencilla.
- Capacidad para reaccionar de forma rápida ante los problemas.
- Garantizar que se cumplan los requisitos de los clientes.
- Informar los datos más relevantes de cada ensamble.
- Detectar desviaciones de los procesos.
- Incrementar y mejorar la comunicación entre distintas áreas.

Dicha plataforma visual-digital mostrará en la pantalla la secuencia de acomodo de los cables dependiendo el tipo de conector y número de parte, esto se pretende mediante el escaneo de un código de barras en la hoja principal de la orden el cual arrojará la información necesaria para que el operador pueda entender de una manera clara y sencilla el orden de los cables, así mismo dando un mejor flujo en la operación de moldeo y logrando un mejor tiempo ciclo de proceso.

La propuesta de la mejora se enlista en la tabla 2, donde muestra a detalle el cambio durante el proceso de acomodo de cables.



Tabla 2 Proceso propuesto de ensamble de conectores de moldeo previo a proceso de moldeo.

<b>PASOS</b>	<b>OPERACIONES</b>	<b>DESCRIPCIONES</b>
1	Se reciben piezas en la estación de moldeo.	<ul style="list-style-type: none"> <li>Las piezas deben estar previamente soldadas con los contactos y pintadas.</li> </ul>
2	Preparación de moldeo y acomodo de cables.	<ul style="list-style-type: none"> <li>Los cables internos se deben acomodar conforme a la secuencia de colores indicada en la pantalla del Raspberry.</li> </ul>
3	Los cables previamente acomodados se colocan dentro del molde.	<ul style="list-style-type: none"> <li>Después de observar información en pantalla, se debe colocar dentro del molde, después de acomodarlos hacer segunda revisión.</li> </ul>
4	Proceso de moldeo	<ul style="list-style-type: none"> <li>La temperatura puede ser de 200°F a 300°F.</li> <li>Tiempo desde 30 minutos a 3 horas.</li> <li>Presión de prensa desde 200 psi hasta 500 psi.</li> </ul>
5	Se desmolda pieza.	Pieza es retirada de molde y se pone enfriamiento por 5 minutos.

### 1.1. Planteamiento de problema

En el proceso de preparación de moldeo se tiene la problemática de la secuencia errónea de cables dentro del conector, este problema se presenta debido a que algunos dibujos no cuentan con el “hook up” o las vistas de moldeo principalmente en las áreas de moldeo.

### 1.2. Justificación

Debido a la falta de una herramienta de control visual digital en las estaciones de preparación de moldeo, se determina implementar el desarrollo de dicha herramienta basada en un sistema sencillo, fácil de entender y operar por el usuario. El cual le permita leer los “hook up” o diagramas de conexión eléctricos, para permitirle ensamblar piezas de manera correcta y moldearlos de tal forma que no generen retrabajo provocado por esta condición ya identificada. La pérdida estimada anualmente es alrededor de \$25,000 dólares, los cuales han impactado el año fiscal 2020 distribuidos en las métricas: piezas buenas a la primera (8%), piezas enviadas a tiempo (15%), rechazos (22%).

Con base en los datos reportados por la compañía se formuló la siguiente tabla (Tabla 3) en la cual se describe por métrico el impacto negativo que sufre la empresa y que por lo tanto se desea mitigar con esta propuesta. Ver tabla 3.

Tabla 3. Costo e impactos negativos por falta de herramienta de control visual digital

<b>COSTO TOTAL DE AFECTACIÓN ANUAL (2020)</b>	<b>PRODUCCIÓN MENSUAL EN PIEZAS EN PROCESO DE MOLDEO</b>	<b>DESPERDICIO MENSUAL EN PORCENTAJE</b>	<b>COSTO DE DESPERDICIO MENSUAL POR FALTA DE HERRAMIENTA DE CONTROL VISUAL DIGITAL</b>
<b>\$25,254</b>	<b>80550</b>	<b>22%</b>	<b>\$2,104</b>

### **1.3. Hipótesis**

El desarrollo de una herramienta interconectada de control visual digital en las estaciones de moldeo permitirá el proceso de moldeo con piezas sin errores por configuración o errores de especificaciones, que actualmente impactan los principales métricos de la compañía. Al plantear una estrategia de desarrollo tecnológico e innovación de proceso, nos lleva a integración de la producción y de los servicios en el diseño, tanto de los productos como de los procesos de producción: a fin de ser capaz de atender órdenes de alta mezcla de números de parte y bajo volumen de dichas piezas que se producen al menos una vez al año, esto no nos permite un mayor entendimiento por parte del operador al momento de comenzar a ordenar los cables conforme a los requisitos del cliente.

### **1.4. Objetivos**

#### **Objetivo general**

Diseñar e implementar un sistema interconectado de control visual digital, este sistema mostrará una ayuda visual electrónica el diagrama de conexión del cliente para que sea fácil de entender.

#### **Objetivos específicos**

- Desarrollar software y hardware que permitan mostrar la información necesaria en la ayuda visual.
- Crear plantilla estándar para el desarrollo de las ayudas visuales.
- Crear ayudas visuales de los 5 principales números de parte ofensores.
- Reducir el defecto en un 80%.
- Reducir el porcentaje de retrabajo y mantener el *Yield* de piezas buenas a la primera por encima del 98%.

### **1.5. Alcances**

El alcance de este proyecto pretende ser aplicado solamente en las áreas de Preparación de Moldeo. Esto con la finalidad de implementar los equipos interconectados donde muestren las ayudas visuales de los cinco principales números de parte ofensores para así poder mejorar el *Yield* de piezas buenas en la primera pasada por encima de un 98.00%.

### **1.6. Organización de la tesis**

El presente proyecto de tesis se desglosa en los siguientes capítulos:

#### **CAPÍTULO 1. INTRODUCCIÓN**

Muestra de manera general la problemática, objetivos, alcance del proyecto, desarrollo de hipótesis, resultados y conclusiones.

#### **CAPÍTULO 2. MARCO TEÓRICO**

Se presenta un estudio del estado del arte, sobre la implementación de ayudas visuales digitales, así como las especificaciones de la plataforma Raspberry Pi, fuentes de información relacionada a tecnicismos del proyecto y conceptos del proyecto a implementar.

#### **CAPÍTULO 3. MATERIALES Y MÉTODOS**

Se detalla la metodología utilizada a la problemática descrita en el capítulo 1, donde además se muestran detalles técnicos utilizados, tablas de descripción y el desarrollo de la implementación del proyecto.

#### **CAPÍTULO 4. RESULTADOS DE LA INVESTIGACIÓN.**

En este capítulo se presentan los escenarios de prueba bajo los cuales fue evaluada la solución propuesta, se describe como fue probada, con cuantas personas, con cuantas partes, en este caso será con una muestra pequeña.

#### **CAPÍTULO 5. Conclusiones y trabajo futuro.**

En este capítulo final se menciona el impacto en los métricos de la empresa y los beneficios que se lograron a través de la implementación del proyecto, así como

también se describen futuras acciones las cuales pueden dar continuidad al proyecto en una fase más amplia donde se puede abarcar más números de parte y áreas que pueden ser beneficiadas con esta mejora.

## **CAPÍTULO 2. MARCO TEÓRICO**

### **2.1 Antecedentes de la investigación**

A Continuación, se muestra parte de la bibliografía consultada para el uso del sistema de Raspberry PI 4, del software MyOpenLab, ayudas visuales, herramientas de control visual y fabrica 4.0 los cuales serán utilizados para la realización de este proyecto.

### **2.2. Usos del Raspberry PI 4**

Este dispositivo es muy popular tanto por su utilidad académica y por las capacidades potentes que tiene a pesar de su tamaño, lo cual lo hace algo bastante atractivo para diferentes tipos de utilidades, desde utilizarlo como servidor o una máquina de juegos, debido a esto y sumado a que es fácil de instalar, tiene un buen almacenamiento (inclusos si necesitas más puede agregarse) y excelente rendimiento su uso hace que el Raspberry Pi 4 tenga una eficacia en procesos. Algunos usos de Raspberry Pi 4:

- Computadora de escritorio.
- Servidor de impresión.
- Centro multimedia.

- Controlador para robots.
- Servidor web o FTP.
- Máquina de juegos retro.
- Procesos donde es necesario utilizar una computadora pequeña.
- Implementación en una celda con ayudas visuales.

Enfocado al proyecto el uso del *Raspberry Pi 4* únicamente se abordó para la integración del desarrollo de un sistema de control visual digital en las celdas. En la Figura 2, se muestra un *Raspberry Pi 4* de referencia.



Figura 2. Figura ilustrativa de un Raspberry PI 4.

### 2.3 Industria 4.0

La Industria 4.0 es una nueva revolución que combina técnicas avanzadas de producción con tecnologías que se integrarán en las organizaciones y las personas. Esta revolución está marcada por la aparición de nuevas tecnologías como la robótica, la analítica, la inteligencia artificial, la nanotecnología y el Internet de las cosas, entre otros. Las organizaciones deben identificar las tecnologías que mejor satisfacen y se adaptan a sus necesidades para invertir en ellas. Si estas no comprenden los cambios y oportunidades que trae consigo la Industria 4.0, corren el riesgo de perder clientes y mercado [1].

Para los líderes tradicionales, este cambio que supone esta nueva revolución industrial proporcionando acceso en tiempo real a los datos transformará la forma en que llevan a cabo sus negocios. La integración digital de la información desde diferentes fuentes y localizaciones permite llevar a cabo negocios en un ciclo continuo. El acceso en tiempo real a la información está impulsado por el flujo de información y acciones entre los mundos físicos y digitales. Esto tiene lugar a través de una serie de pasos conocido como PDP – por sus siglas en inglés “*physical-to-digital-to-physical*”-:

- Del mundo físico a la digital. Se captura la información del mundo físico y se crea un registro digital de la misma.
- De digital a digital. En este paso, la información se comparte y se interpreta utilizando analítica avanzada, análisis de escenarios e inteligencia artificial para descubrir información relevante.
- Del mundo digital al físico. Se aplican algoritmos para traducir las decisiones del mundo digital a datos efectivos, estimulando acciones y cambios en el mundo físico.

Los impactos de la Industria 4.0 son en los siguientes niveles: en grandes ecosistemas, a nivel organizacional y a nivel individual (en empleados y clientes):

- Ecosistemas. Además del cambio en el que las empresas operan y en la producción de bienes, la Industria 4.0 afecta a todos los agentes del ecosistema (los proveedores, los clientes, los inversores, etc). Estas tecnologías permiten interacciones entre cada uno de ellos.
- Organizaciones. La capacidad de ajustarse y aprender de los datos en tiempo real puede hacer que las organizaciones sean más receptivas, proactivas y predictivas. Esto también permite a la organización reducir sus riesgos en materia de productividad.
- Individuos. Por ejemplo, para los empleados puede significar un cambio en el trabajo que van a realizar, mientras que para los clientes significa una mayor

personalización en los productos y servicios que satisfagan mejor sus necesidades de cada uno de ellos.

## **2.4 MyOpenLab**

Es un software que integra elementos para comunicarse con las plataformas más utilizadas entre las que podemos encontrar Arduino, Raspberry PI, Android y dispositivos con Comunicación Serial [2].

Este software va agregando interfaces a medida que los usuarios realizan sugerencias en los canales de comunicación disponibles que tienen, por lo tanto, se encuentra en constante actualización y se implementan las necesidades de los usuarios.

El proyecto fue iniciado por Carmelo Daniel Salafia ,él fue quien desarrolló el proyecto durante varios años, siendo el desarrollador principal hasta mediados del año 2017 cuando decidió transferir el desarrollo del proyecto al Ingeniero Javier Velásquez, quien es actualmente el desarrollador principal. La distribución del software cuenta con varios ejemplos del uso de los elementos para facilitar a los usuarios el entendimiento de la lógica de este programa y hacer más fácil su uso.

Los usuarios de este software no necesariamente tienen que ser programadores avanzados, pero deben tener un nivel básico de los conceptos de programación para poder hacer un uso adecuado del mismo.

MyOpenLab se desarrolló en Java y Netbeans, por lo tanto, se puede ejecutar en diferentes plataformas, siempre y cuando no haya dependencias de algún tipo hardware.

## **2.5 Ayudas visuales**

Las ayudas visuales incluyen una amplia variedad ya sea para máquinas o tareas específicas y ayudan a mejorar la eficiencia y efectividad de la interacción humana con el equipo. Estas ayudas visuales sirven como recordatorios de advertencias: Comunican configuración de procesos o procedimientos, ayudan a exponer partes ocultas, lo que facilita la inspección del equipo, reducen de manera significativa el tiempo de capacitación, al colocar la información importante en el equipo donde se



necesita y se reduce el tiempo invertido en resolución de problemas con el uso de estas ayudas visuales [3].

## **2.6 Herramientas de control visual digital.**

El control visual es una metodología que puede ser aplicada de diferentes maneras. Es cualquier dispositivo o sistema que permita representar, en un solo lugar, información de utilidad sobre el proceso. Con tan solo una mirada podemos saber cómo está funcionando nuestro proceso y si se está desviando con respecto a un patrón preestablecido. Nos muestra información sobre el estado del flujo de trabajo, inventarios, desempeño de un trabajador o un equipo de trabajo. Por esto, esta herramienta sirve para agregar valor a nuestros procesos productivos. Teniendo una imagen clara y sencilla del estado del proceso podemos detectar con mayor facilidad los problemas, y atacarlos a tiempo, evitando caer en retrabajos o pérdidas.

Una herramienta de control visual debe permitir que cualquier persona que observe la información plasmada en ella sea capaz de interpretarla. Esto permite, a un gerente recorrer la planta y saber cómo están funcionando los procesos, los trabajadores y las máquinas involucradas con solo ver la herramienta de control visual.

Un ejemplo de un sistema de control visual son las 5s. Mantener todo limpio y ordenado facilita la detección de cualquier problema. Un lugar para cada cosa y cada cosa en su lugar. Ni más, ni menos. Es muy fácil que los problemas saltan a la luz ya que quedan expuestos por alejarse del orden establecido. Algunas de las maneras de utilizar control visual en 5S: sombrear la ubicación de las herramientas, marcar la posición en que debe encontrarse el operador en su área de trabajo, o tener una fotografía de cómo debería ser idealmente el espacio de trabajo [4].

### **2.6.1 Tipos de herramientas de control visual.**

- Raspberry Pi, es una computadora de tamaño compacto la cual se puede conectar a su televisor o monitor para ser utilizado, así como también se pueden conectar diversos periféricos. Es una placa que soporta varios componentes a la vez, de tamaño pequeño y capaz de ser utilizado en muchas de las cosas que

una PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos [5].

- Andon: Tiene ese nombre debido a las tradicionales 'lámparas japonesas'. Un andon es un tablero de tipo luminoso en el que aparecen los estados de las máquinas en un solo lugar, facilitando la identificación de los problemas. Son tableros, en donde pueden aparecer luces de diferentes colores. Todos deben ser capaces de comprender la información que se muestra en estos, la cual debe ser actualizada constantemente.
- Kanban: Sistema de tarjetas que se utiliza para indicar necesidades de producción en los procesos, permitiendo el funcionamiento adecuado y fluido que requiere un sistema
- Reporte A3: Consiste en una única hoja de tamaño A3 en donde se grafican y exponen todos los datos de mayor índice crítico, ejemplo: indicadores, gráficas de control o análisis de problemas. Puede utilizarse en una presentación para exponer los datos más relevantes y detectar de manera rápida y sencilla errores, o tendencias
- Cuadro de Mando Integral (BSC, por Balanced Scorecard): Similar al Reporte A3, pero sin limitaciones de tamaño, esquema de presentación o formato, un BSC permite representar todos los indicadores de importancia para conocer el estado del proceso o la organización. Este nos brinda información sumamente útil para determinar si las acciones concretas llevadas a cabo y la tendencia están alineadas con la estrategia de la organización.
- Obeya: Obeya puede traducirse del japonés como «gran habitación». Y es una gran habitación en donde se muestran diversas herramientas de control visual, las cuales son necesarias para comprender el estado actual y real de los procesos en función de los estándares definidos, tales como: indicadores de desempeño, comparación con la competencia, indicadores financieros, agenda de actividades y cualquier otra información que pueda ser necesaria para la toma de decisiones trascendentales. Para ejemplos vea la figura 3.



Figura 3. Imágenes ilustrativas de herramientas de control visual.

## 2.7 Lenguaje Python

Este lenguaje de programación interpretado, multiplataforma permite que el código del programador identifique lo traduzca y lo ejecute a la vez, de la misma forma también es capaz de interpretar a la vez varios paradigmas de programación tales como programación orientada a objetos, programación imperativa y programación funcional. Python en los últimos años ha crecido y su popularidad se ha incrementado notablemente. Este lenguaje de programación se caracteriza por ser simple, versátil y rápido para el desarrollo de programas. Python es un lenguaje de scripting independiente de una plataforma y sirve para programar aplicaciones en Windows, así como también páginas web. Este lenguaje necesita compilar el poder ejecutarlo el programa, esto le da la ventaja en rapidez de desarrollo, pero también le da inconvenientes como una menor velocidad [6].

Su reciente popularidad se debe a varias razones como:

- Contiene muchas librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar bastantes tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad para la creación de los programas.
- La cantidad de plataformas en las que se puede desarrollar, algunos ejemplos :Unix, Windows, OS/2, Mac entre otros.

## Características del lenguaje

- Propósito general: Se pueden crear todo tipo de programas.
- Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos.
- Interpretado: No se debe compilar el código antes de su ejecución, lo cual lo hace más rápido (En realidad sí que se realiza una compilación, pero esta se realiza de manera oculta para el programador).
- Interactivo: Python dispone de un intérprete por la línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- Funciones y librerías: Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos.

### **2.8 Gráficos librería Tkinter.**

Las interfaces gráficas o GUI son medios visuales, mucho más amigables que una terminal de texto, a través de las cuales los usuarios pueden interactuar y realizar tareas más fácilmente. Hoy en día la interfaz gráfica de usuario se ha convertido en algo muy común desde los sistemas operativos (SO) hasta en las aplicaciones de software. Estas interfaces son tan fáciles de usar que hasta usuarios con pocos o nulos conocimientos de informática pueden aprender a utilizar aplicaciones de distintos tipos como: procesamiento de textos, finanzas, inventario, diseño, ilustraciones, etc [7].

¿Qué es Tkinter?

Un conjunto de herramientas GUI del tipo Tcl/Tk (Tcl: Tool Command Language), esta librería les proporciona a las aplicaciones de Python gama de usos muy variables que van desde aplicaciones web hasta redes y más.

¿Por qué utilizar Tkinter?

Esta no es la única librería especializada en la creación de interfaces gráficas para Python, algunas de las más utilizadas son wxPython, PyQt y PyGtk, todas con ventajas y desventajas. Entre las ventajas que caracterizan a Tkinter, es que viene instalado junto a Python en casi todas las plataformas, otra cosa que es importante mencionar es que su sintaxis es clara, es muy fácil de aprender y cuenta con documentación completa.

## **2.9 Base de datos MariaDB.**

MariaDB, es una variante de MySQL que se caracteriza por ser un software libre y asegura dos cosas muy importantes: la primera es mantener la compatibilidad con MySQL, esto significa que MariaDB pueda usarse como reemplazo de MySQL sin necesidad de ningún costo de migración al otro software, y que la Comunidad pueda seguir actualizando el sistema gestor de base de datos, sin que sea condicionado por las estrategias comerciales de algún tipo [8].

¿Cuándo usar MariaDB?

MariaDB se puede usar en cualquier lugar donde antes se usaba MySQL. Al ser un sistema compatible, cualquier otro software que antes utilizará MySQL es capaz de seguir funcionando igualmente con MariaDB. Se puede usar MariaDB en cualquier proyecto de nueva creación, así como intercambiar MySQL por MariaDB en prácticamente todos los proyectos que puedan estar ya en producción.

¿Por qué usar MariaDB?

MariaDB ha implementado diversas mejoras y nuevas funcionalidades. Las mejoras muchas veces afectan directamente al rendimiento o permiten optimizar mejor las bases de datos, por lo cual es un programa que se actualiza constantemente. La compatibilidad es tal que muchas veces el uso de MariaDB en lugar de MySQL es transparente para desarrolladores o administradores de sistemas.

## **2.10 Web PHP y HTML.**

PHP: Es un lenguaje "script" que se ejecuta del lado del servidor y que se debe de instalar en el servidor para poder utilizarlo, el código escrito en PHP se ejecuta en el servidor y luego muestra la salida, normalmente desplegada a los usuarios usando el lenguaje HTML.

Algunas de las operaciones que permite en el servidor son :

- Acceso a bases de datos.
- Lectura/escritura de archivos.
- Manejo de fechas.
- Operaciones complejas con texto manipulación/modificación de imágenes entre otras muchas cosas.

HTML, es considerado un lenguaje de marcado, éste código es el responsable y encargado principal de ejecutar una descripción acerca de los contenidos que aparecen como textos y sobre la estructura de este, es simplemente un conjunto de etiquetas de código que le indican a los navegadores cómo desplegar texto, poner texto en negritas, cursiva, acomodar el texto en tablas, y claro, permite colocar enlaces a otras páginas o documentos [9].

#### **2.11 Web Bootstrap 4.**

Bootstrap 4 es la última versión de Bootstrap, que cuenta con el framework de CSS, HTML y JavaScript, lo cual permite el desarrollo de webs que se ajustan a cualquier resolución esto quiere decir, que el sitio web se adapta automáticamente al tamaño de una PC, una tablet u otro dispositivo. A esta técnica de diseño y desarrollo se conoce como diseño adaptativo [10].

## **CAPÍTULO 3. MATERIALES Y MÉTODOS**

### 3.1 Descripción del proceso de ensamble de conectores y proceso de moldeo.

El proceso de ensamble de conectores comienza con la entrega de las piezas a moldear previamente soldadas y pintadas en la celda (ver figura 4), para proceder con el proceso de moldeo el cual se describe continuación.



**Figura 4. Contactos pintados antes del proceso de moldeo.**

1. Se toma orden y se revisa material a procesar para tratar de interpretar el requerimiento del cliente.
2. Se toma la pieza y los cables se deben acomodar de acuerdo al Hook Up indicado en la hoja de instrucción o en el dibujo (si se cuenta con ellos), teniendo en cuenta que deben acomodarlo de forma inversa a como viene en el dibujo (siendo esta parte un error recurrente al acomodarlos en la vista frontal que plantea el dibujo), en caso de no tener se debe consultar con ingeniería para definir el código de colores que se utilizará en dicho número de parte, en la figura 5, se observa la pieza con acomodo de acuerdo al código de colores correspondiente.



**Figura 5. Acomodo de cables de acuerdo a Hook Up.**

3. Se coloca la pieza en el molde, se agrega plástico a la prensa y en partes del molde para ayudar a llenar durante el proceso de inyección. Tal como se aprecia en la figura 6.



**Figura 6. Colocado de pieza en el molde.**

4. Se coloca el molde en la prensa, y se configura la presión, temperatura y el tiempo de moldeo, dependiendo la pieza que se vaya a moldear, tal como se muestra en la figura 7.



**Figura 7. Molde colocado en la prensa durante el proceso de Moldeo.**

5. Posterior al moldeo, se procede a retirar el molde de la prensa y abrirlo para retirar las piezas y se dejan enfriar 5 min, como se muestra en la Figura 8.





Figura 8. Pieza Moldeada.


- La pieza pasa a prueba eléctrica para su revisión, asegurando que la pieza fue moldeada con el Hook Up correcto, para poder seguir con el proceso siguiente ya sea Pruebas de presión, accesorios o empaque dependiendo el modelo que se esté trabajando.

Errores típicos por deficiencias en el proceso de ensamble y moldeo.

Se define como *Hook up incorrecto* a cualquier número de parte que no haya sido moldeada de acuerdo al código de colores solicitado por el cliente, esto provoca que se necesite hacer un retrabajo o en casos más graves que se vuelva scrap al no poder rescatar los contactos o ya no cumpla con la longitud que se indica en las especificaciones, tal como se aprecia en las figura 9 y figura 10.

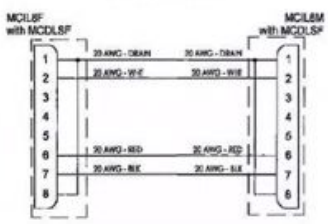


Figura 9. Alerta de calidad ocasionada por Hook Up incorrecto.

 <b>ALERTA DE CALIDAD</b>							
Numero de Parte	MC-S005-0828	RMA	12781	Cliente	SEACON / BRANTNER & ASSOCIATES	# Alerta	AD-485
Descripción:	(PRE1)11116485-S MCIL8F+MCDLSF,7.02FT SMT-003CF,MCIL8M+MCDLSF DWG# 11116485-S					Forma	SM272496
Defecto/Problema	CG-027 COLORES CAMBIADO/HOOK-UP INCORRECTO					Área	RM
Piezas Rej:	2	Muestras Disponible	NO				
Vigencia del:	11-Dec-20	al	17-Dec-20	Comentarios:			
						Elaboró:	Nestor G.


  

**Correcto:**



**CG-027 COLORES CAMBIADO/HOOK-UP INCORRECTO**

**Incorrecto:**



MCIL8F to MCIL 8M  
Pin 1+8 to Pin 1+2  
Pin 2 to Pin 8  
Pin 3 to Pin 7

Fotos de Cliente

Figura 10. Rechazo ocasionado por Hook Up incorrecto

Defectos cosméticos.

Todo defecto visible a simple vista o bajo microscopio que afecta cosméticamente una pieza para los clientes, como se puede apreciar en la figura 11, en la que se muestran defectos ocasionados por el operador al quitar el mandril cuando la pieza sale de la prensa, en la segunda imagen se muestra un pin dañado, en la tercer y cuarta se muestra una pieza dañada durante el proceso de limpieza de la pieza moldeada .

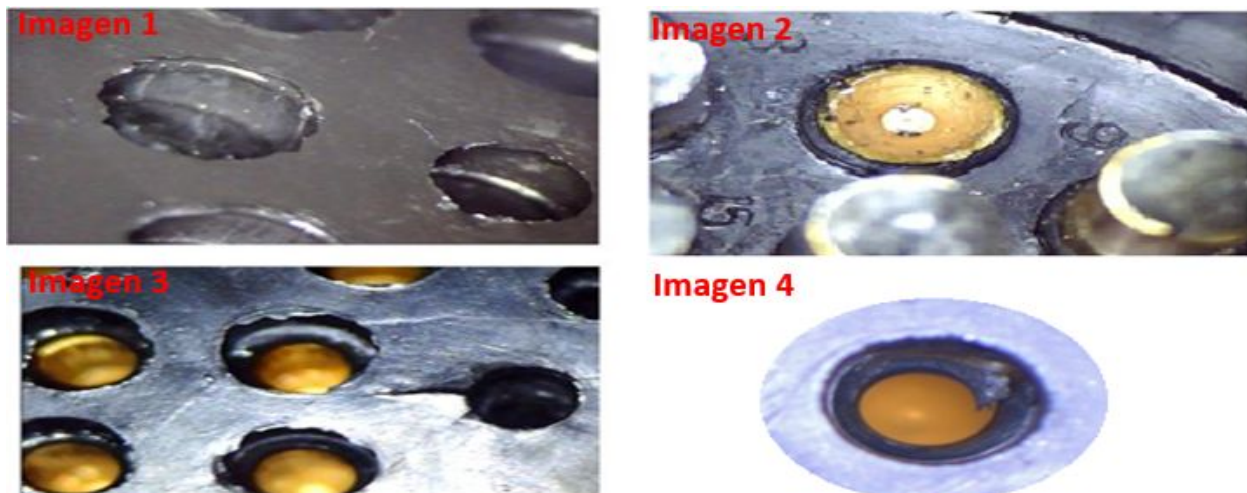


Figura 11. Defectos cosméticos en piezas moldeadas.

### **3.2 Uso de herramientas de control visual digital como solución.**

La opción de utilizar herramientas de control visual digital se volvió la forma más rápida y sencilla de dar solución a este problema por lo tanto se tomó la decisión del desarrollo e implementación de este sistema, esto también es sustentado con otras empresas que en lugar de utilizar algún tipo de papelería para el traslado de sus órdenes o procesamiento de las mismas utilizan sistemas digitales, que por medio de un escaneo en un lector código de barras en las piezas o la orden despliega sobre una pantalla la información necesaria para continuar la operación sin problemas.

Dichos sistemas también pueden ser implementados en otros procesos dentro de la empresa debido a que comparten la misma problemática de una toma de decisión por parte del operador con base a una información que no es clara para él, las áreas que comparten dichas similitudes se describen a continuación:

- Soldadura: área en la cual se reciben las órdenes para soldar los contactos en los cables que ya se envían preformados al área de moldeo, el problema es que la información de secuencia de colores para el tipo de conector no es clara para el operador.
- Epóxico: proceso en el cual se moldean con una resina los conectores y previamente se pre-forman con la secuencia del diagrama eléctrico que requiere el cliente. La falta de información clara también es una toma de decisión que se deja a la experiencia del operador, convirtiéndose en el mismo problema descrito.
- Empaque: la selección de accesorios, detallado final y colocación de etiquetas es un problema constante ya que en ocasiones se envían piezas sin la información solicitada en etiquetas por el cliente o los accesorios que complementan el conector.

Visto de esta forma, en esta empresa se tiene áreas de oportunidad en diferentes áreas las cuales pueden dar solución con propuesta descrita para el proceso de preparación de moldeo. Por lo tanto, al desarrollar este sistema en conjunto con la plantilla que sea entendible para el operador, este debe ser sencillo y claro que permita a cualquier persona que observe la información mostrada en ella sea capaz de interpretarla. Esto

permite, por ejemplo, a un gerente recorrer la planta y saber cómo están funcionando los procesos, los trabajadores y las máquinas involucradas (de un solo vistazo).

Así que siguiendo con el principio de un control visual sencillo y fácil de interpretar se buscó la opción de sistemas de cómputo que cumplieran esos requisitos, al ser para una operación única y sencilla por cada estación, se decidió utilizar un sistema Raspberry Pi 4 con pantalla y lector de código de barras, dicho sistema al ser pequeño y sencillo nos brinda la versatilidad que requiere el operador para poder acceder a la información necesaria fácilmente.

Este sistema se integrará como un aditamento a lo que es la estación de moldeo y será una herramienta fundamental para el operador, dicha herramienta no modifica las propiedades del producto ni puntos críticos del proceso, por lo tanto, la validación por parte del equipo de ingeniería se concentró en la parte del sistema.

Como principios de sistemas visuales al no estar nada implementado similar en la empresa se tomaron conceptos como sistemas de Fabrica Visual y Ayudas visuales digitales, los cuales son conceptos de manufactura esbelta que mencionan que los recursos visuales deben colocarse en el punto donde se necesitan, esto quiere decir que cada operador debe tener a la mano la información clara de lo que va hacer y deben ser estándares visuales que refuerzan la operación que está a punto de hacer.

Por lo tanto, se puede afirmar que la propuesta del desarrollo de un sistema de control visual digital tiene bases y fundamentos de lo que son herramientas de manufactura esbelta, ya que dichas herramientas su principal enfoque es la eliminación de desperdicios, que en este caso son varios métricos que impactan de manera negativa a la empresa.

### **3.3 Herramientas de control visual digital como solución.**

Aplicación de Raspberry PI 4 : Raspberry Pi, es una computadora de tamaño compacto la cual se puede conectar a su televisor o monitor, así como también se pueden conectar diversos periféricos. Es una placa que soporta varios componentes a la vez, de tamaño pequeño y capaz de ser utilizado en muchas de las cosas que una PC de escritorio hace.

Software de MyOpenLab: Es un software de desarrollo fácil de usar basado en el uso de elementos gráficos, cada uno tiene diferentes propiedades, formas de configuración y una función específica. Estos componentes se pueden unir entre sí fácilmente para implementar de una manera sencilla la parte gráfica y lógica de una aplicación, en la cual la programación se programa por bloques.

### **3.4. Materiales por utilizar durante la implementación.**

A manera de dar una propuesta como posible solución se realizó un estudio de datos históricos del problema y análisis de las diferentes piezas que llegaban a cada estación de moldeo, se identificaron los principales números de parte que contribuyen por lo menos con el 80% del total de los defectos ocasionados por una mala conexión eléctrica.

Después de dicho análisis se realizó una lluvia de ideas en conjunto con el ingeniero del área, jefe de grupo y operador, esto para revisar realmente cual era la causa raíz del por qué las conexiones eléctricas no se realizaban conforme a los requerimientos del cliente. Por lo tanto, se llegó a la conclusión que la información para realizar la conexión previa al proceso de moldeo no era clara y en ocasiones no estaba disponible, ya que se extraviaba las hojas físicas durante el proceso.

Para dar una propuesta de solución eficaz se plantearon varias propuestas para evaluar, una de ellas fue la redirección de los números parte complejos a una sola estación, pero se descartó debido a la mezcla de números de parte y volumen no era factible concentrarlos en una sola estación.

Al descartarse esa opción se realizó una segunda propuesta la cual era más radical y eficaz ya que en ella no importaría el tipo de producto que fuera a procesar el operador ni tampoco dependería de la expertis del mismo.

La segunda propuesta era el implementar un sistema interconectado de cómputo el cual mostrará a manera de ayuda visual los requerimientos de conexión solicitados por el cliente, después de evaluar opciones de cómputo sencillas y a un precio accesible, se llegó a la conclusión de utilizar equipos Raspberry PI 4.

Este ordenador de un tamaño pequeño permite instalarlo en la estación de moldeo sin generar una aglomeración de equipos y materiales por estación de trabajo, adicional esta mini computadora se utilizará con su pantalla de 7 pulgadas que es táctil y por lo tanto la ergonomía ni el espacio de la estación de trabajo se verán afectados por la instalación del sistema.

Para poder hacer una estación prototipo fue necesario utilizar los siguientes componentes:

- **Raspberry PI 4:** Minicomputadora que controlara las funciones del sistema para desplegar la ayuda visual en la pantalla. \$125.90 US
- **Pantalla táctil Raspberry PI 4:** Pantalla utilizada para mostrar la información de la ayuda visual. \$86.38 US
- **Lector código de barras:** Este lector escanea el código de barras de la orden de producción a procesar para poder mostrar la información. \$30.23 US
- **Lenguaje Python:** Este lenguaje de programación ayudará a desarrollar la lógica para la funcionalidad del sistema.
- **Base de acero inoxidable:** Base de acero inoxidable 304 para la sujeción de todo el sistema Raspberry. \$19.99 US

Después de instalar la estación prototipo se detectó que se necesitaba un pedestal ajustable dependiendo la altura del operador, esto para poder cumplir con los requerimientos de ergonomía. Posteriormente a la evaluación de la modificación de la estación se logró concluir satisfactoriamente lo que era la instalación física de manera exitosa, ya que con esto se podría replicar en las diferentes estaciones de moldeo.

Partes como cables, conectores o todos los insumos que se necesitan para lograr la instalación fueron proporcionados por González machine shop para la instalación dentro de la empresa.

### **3.5. Implementación de sistema de control visual digital**

Para el desarrollo de esta herramienta fue fundamental realizar un análisis de datos con base al historial de problemas relacionados con este modo de falla, por lo tanto, se

llegó a la conclusión de cuáles eran los principales métricos afectados de la empresa como los son piezas rechazadas, retrabajos, tiempo ciclo de operación alto y entregas a tiempo fuera de lo acordado. Al ser nosotros una empresa subcontratada se realizó el análisis financiero donde se colocó el costo beneficio del desarrollo del proyecto, esto para convencer a la gerencia del costo de la inversión de la implementación del proyecto en todas las estaciones de moldeo.

Para poder concretar este proyecto de manera exitosa fue necesario hacer la labor de investigación e implementación en 4 secciones, las cuales se describen a continuación:

### 3.6.Fase 1. Verificación y registro de datos de afectación por falta de herramienta de control visual digital

El registro de información para recabar datos para así poder validar la magnitud del problema es una operación importante y se realizó por medio del departamento de ingeniería. El cual consistía en la validación de proceso físicamente en piso de producción aunado con el desarrollo de un formato de registro de problemas de proceso que completaba durante el turno personal del área de moldeo.

Dicho registro por parte del operador consistía en llenar el formato dependiendo el defecto y el tipo de conector, dicho monitoreo de proceso se realizó de manera exitosa durante 2 semanas y el resultado se registró en la siguiente tabla.

Tabla 4. Ejemplo de registro diario de defectos y rendimiento por día por número de parte.

Enero 27														
Defecto	MC-S040-0021	MCL-S003-0277	MC-S003-0069	MC-S006-0670	MIN-S006	MC-S004-0482	MCL-S003-0022	LP-S049-0018	LP-S015	HRN-S0166	MC-S002-0156	MC-S062-0114	MC-S060-0193	MC-S006-0595
Hook Up incorrecto	20	25	16	13	45	15	30	8	6	10	12	8	12	9
Total de rechazos	20	25	16	13	45	15	30	8	6	10	12	8	12	9
Produccion Total	185	150	200	125	177	190	210	65	54	195	158	98	100	112
Rendimiento	89.19%	83.33%	92.00%	89.60%	74.58%	92.11%	85.71%	87.69%	89.89%	94.87%	92.41%	91.84%	88.00%	91.96%

Tabla 5. Ejemplo de registro semanal de defectos y rendimiento por día por número de parte.

Semana 16														
Defecto	MC-S040-0021	MCL-S003-0277	MC-S003-0069	MC-S006-0670	MIN-S006	MC-S004-0482	MCL-S003-0022	LP-S049-0018	LP-S015	HRN-S0166	MC-S002-0156	MC-S062-0114	MC-S060-0193	MC-S006-0595
Hook Up incorrecto	20	25	16	13	45	15	30	8	6	10	12	8	12	9
Total de rechazos	124	143	105	72	245	83	173	44	39	51	79	41	75	65
Produccion Total	1105	925	1155	735	1045	1117	1289	375	302	1059	924	532	589	625
Rendimiento	88.78%	84.54%	90.91%	90.20%	76.56%	92.57%	86.58%	88.27%	87.09%	95.18%	95.45%	92.29%	87.27%	89.60%

### 3.7. Fase 2. Selección de números de parte principales de Hook Up incorrecto.

De acuerdo a los registros diarios y semanales se definió como números de parte principales lo siguientes:

- MIN-S006.
- MCL-S003-0277.
- LP-S015.
- MC-S060-0193.
- LP-S049-0018.

Estos intervienen en el bajo rendimiento del área; estos números de parte que de acuerdo al registro de la semana 16 a la semana 20 del año 2020, se posicionaron en los primeros 5 lugares de la producción afectada con este defecto. Ver figura 12, de comportamiento semanal de números de parte por hook up incorrecto.



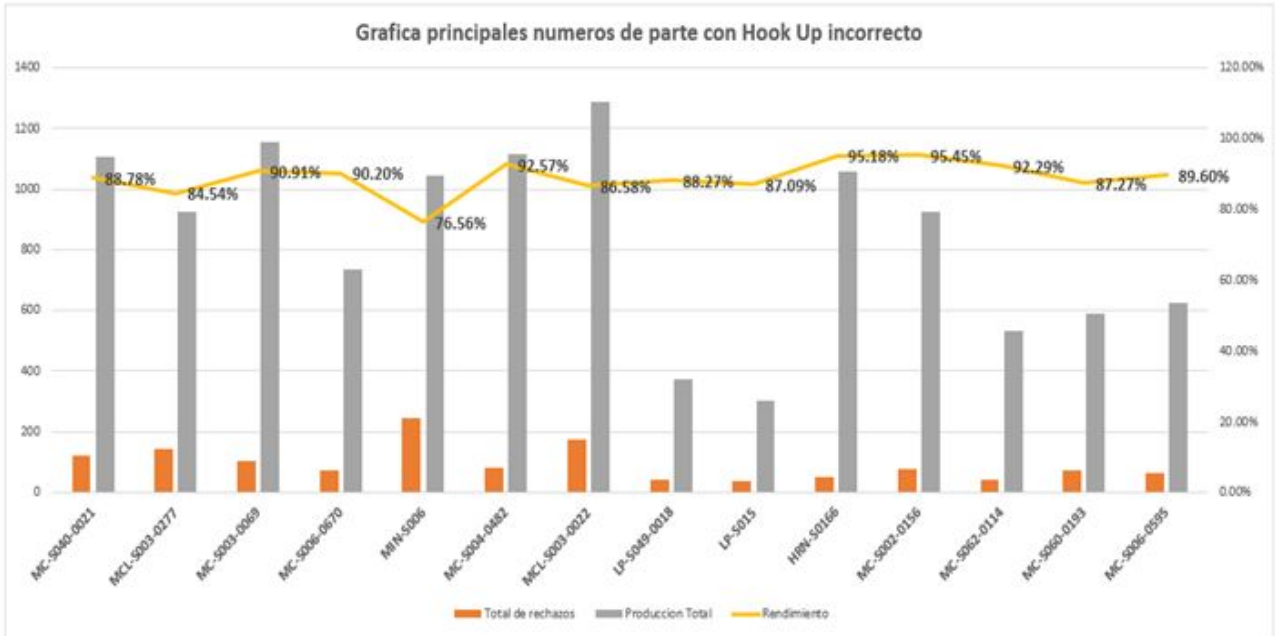


Figura 12. La gráfica muestra el comportamiento de defecto *Hook Up* por semana, colocándolo el número de parte MIN-S006 como el principal.

### 3.8. Fase 3. Causas de origen de defecto *Hook Up incorrecto*.

De acuerdo al estudio realizado para determinar las causas de los *Hook Up incorrectos* se detectó que las principales fuentes que propician dicho defecto son:

- Los dibujos no son claros.
- No hay dibujos.
- Los dibujos no cuentan con *Hook Up*.
- Los dibujos están en inglés.

Todas estas causas hacen que el personal tenga que ir a buscar el dibujo, al ingeniero a cargo del área o incluso hacerlo en base a su experiencia lo cual ocasiona bastante pérdida de tiempo, retrabajos y rechazos impactando drásticamente los métricos. Para mayor detalle de los hallazgos por semana se describe en la tabla 6 y figura 13.

Tabla 6. Tabla con los hallazgos por semana de las posibles causas del defecto *Hook Up incorrecto*.

Posibles causas de Hook Up incorrecto	Hallazgos por semana				Total	Porcentaje
	Semana 16	Semana 17	Semana 18	Semana 19		
El Dibujo no es claro	9	16	8	4	37	28.03%
No hay dibujo	8	3	6	10	27	20.45%
Dibujo sin Hook Up	15	9	7	11	42	31.82%
Dibujo en ingles	5	6	9	6	26	19.70%

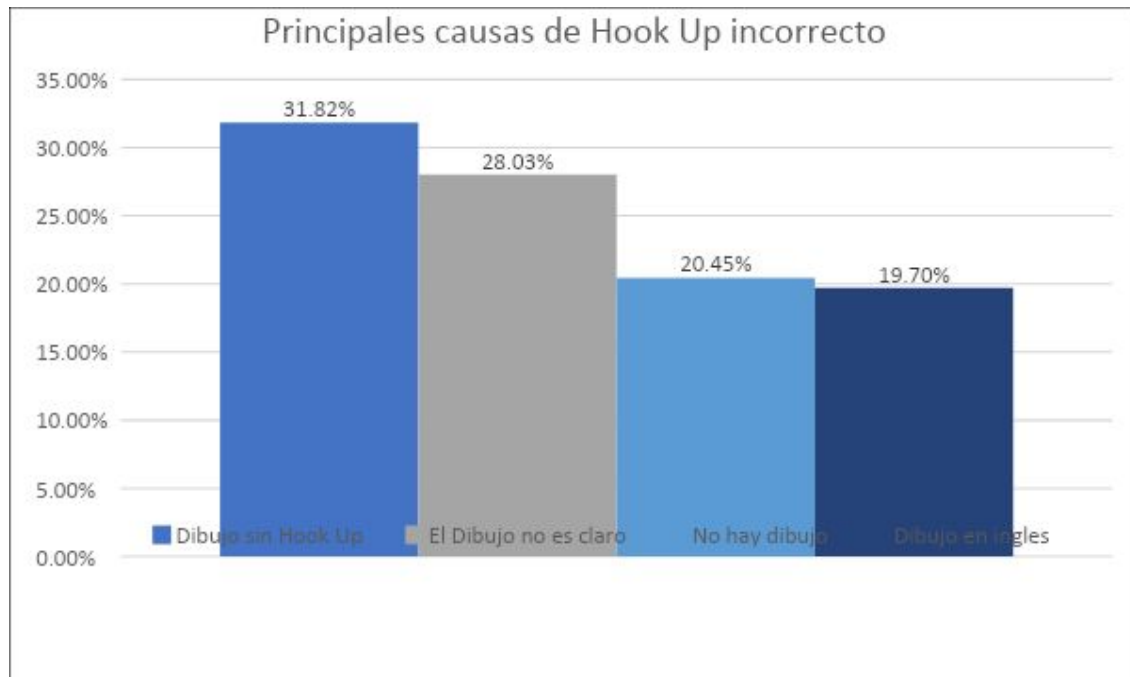


Figura 13. Distribución por porcentaje de las razones causantes de *Hook Up incorrecto*.

### 3.9. Fase 4. Fase de implementación del proyecto.

Sabiendo las principales razones de defecto fue necesario realizar una serie de actividades las cuales nos llevaron a la reducción del defecto descritas a continuación.

- **Documentación de dibujos con Hook Up:** Enfocando la revisión en la detección de dibujos que no cuentan con el Hook Up especificado, se actualizaron dichos dibujos para poder incluirlos en las ayudas digitales. Ver figura 14.
- **Actualización de dibujos no claros:** La actualización consistió en revisar los dibujos en los cuales los operadores se equivocaban constantemente, agregando las notas necesarias o vistas, en esta parte apoyándonos con el departamento de diseño. Detalles en la Figura 15 y Figura 16.

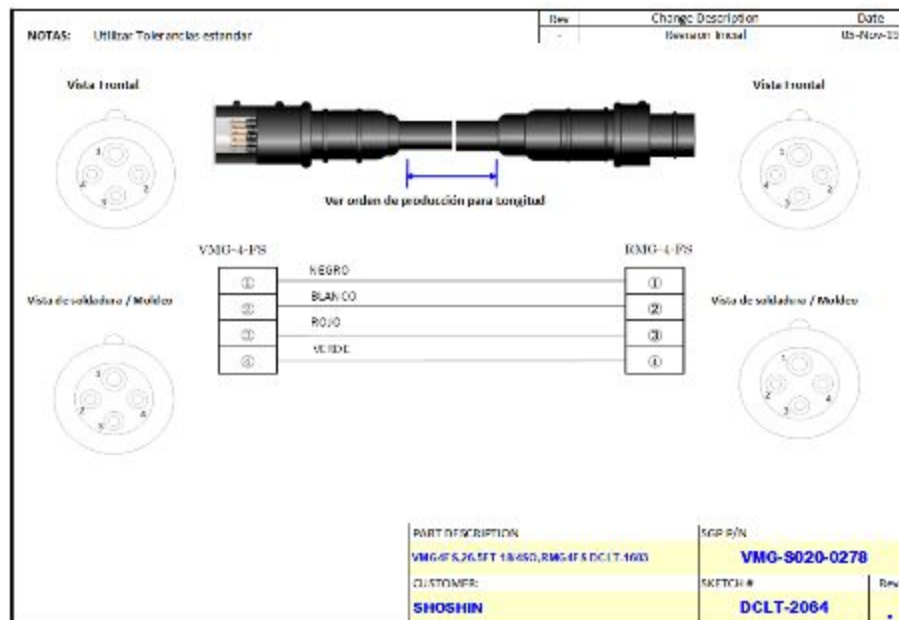


Figura 14. Dibujo documentado con Hook Up.



- **No hay dibujo:** En este caso al ser números de parte de una misma familia, en algunos casos se omiten los dibujos para tal número de parte y se usan dibujos similares e incluso dibujos hechos a mano lo cual puede provocar errores, tal como se aprecia en la figura 17.

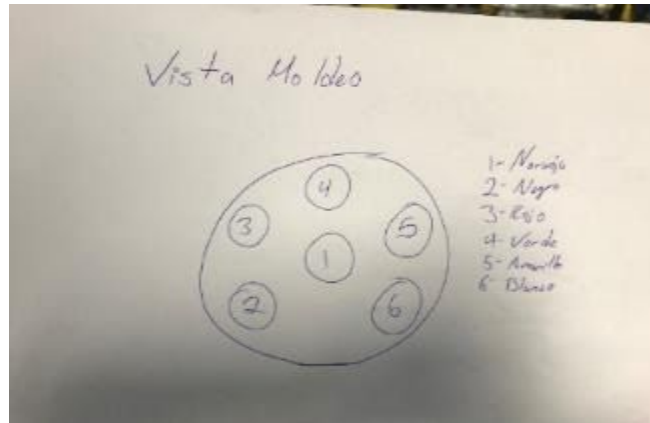


Figura 17. Dibujo hecho a mano de Hook Up.

- **Dibujos en inglés:** En este caso la mayoría de los dibujos están en inglés, pero se hacen las respectivas notas, ayudas visuales o Traveler's para que el operador pueda hacer la pieza sin problema, pero hay algunos que no cuentan con nada de esto y además la información que viene es muy extensa, tal como se aprecia en la Figura 18.

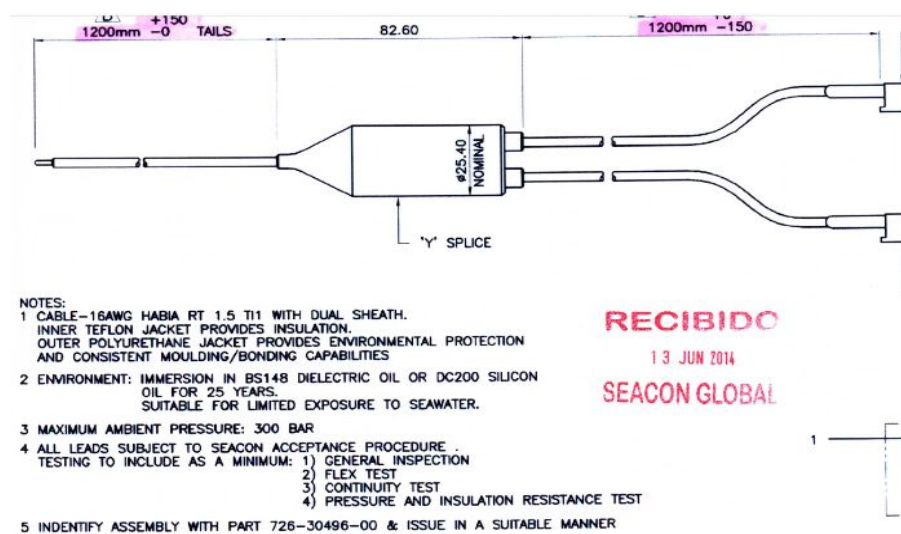


Figura 18. Dibujo notas en inglés.

- **Programación de sistema Rasperry Pi 4:** En la sección de programación se muestra el desarrollo del código en Python de cada parte del programa, empezando con el registro del usuario, en esta parte se puede crear el usuario que va a ingresar al sistema, solamente con introducir su nombre, número de empleado y pulsando en el botón de registrar. Para ver código completo ver anexo 6.

The image shows a window titled "Registro" with a green header bar containing the text "Introduzca sus datos". Below the header, there are two input fields: "Nombre" and "Numero de empleado". At the bottom of the form, there are two buttons: a green "Registrarse" button and a blue "Volver" button.

Figura 19. Registro de usuario en sistema.

Una vez que se ha registrado el usuario, ahora puede iniciar sesión introduciendo su usuario y contraseña, a continuación, se muestran las ventanas del programa y del código. Figura 20 de imagen de inicio de sesión en el sistema.

The image shows a window titled "Inicio" with a green header bar containing the text "Inicie sesion o registrese". Below the header, there are two input fields: "Introduce tu usuario" and "Introduce tu contraseña". At the bottom of the form, there is a green "Iniciar sesion" button.

Figura 20. Inicio de sesión en sistema.

Una vez que se ha iniciado sesión con el usuario, se muestra la pantalla donde se escanea el código de la orden y automáticamente aparece en pantalla la ayuda visual correspondiente al número de parte de la orden, a continuación se muestran las ventanas del programa y del código. Ver figura 21 de pantalla de ayuda visual en sistema.



Figura 21. Pantalla de ayuda visual en sistema.

En la pantalla de modificación de diagramas, se pueden agregar nuevos diagramas, modificarlos y eliminarlos, solo agregando el número de parte, la familia a la que pertenece, la versión y cargando el diagrama correspondiente, a continuación se muestran las ventanas del programa y del código. Ver figura 22 imagen de modificación de diagramas en el sistema.

Modificación de diagramas

Numero de parte

Familia

Version

Nombre del archivo con extension, ejemplo(Archivo.png)

Para agregar o editar un diagrama es necesario que llene todos los campos

Para borrar un diagrama solo se necesita el numero de parte

Volver

Agregar

Editar

Borrar

**Figura 22. Pantalla de modificación de diagramas**

En la pantalla del servidor local se puede revisar que empleados ya cuentan con un usuario, los diagramas que están dados de alta y también las horas en las que los empleados han entrado al sistema, a continuación se muestran las ventanas del programa y del código. Ver figura 23 pantalla de empleados en sistema.

Opciones para filtrar contenido

Nombre   Nombre de empleado   Numero   Numero de empleado   Filtrar

Empleados registrados

ID Empleado	Nombre	Numero
14	Maria	DB
15	Maria	DB
16	ass	wss
17		
18		
19	PY_VAR0	PY_VAR1
20	PY_VAR0	PY_VAR1
21	PY_VAR0	PY_VAR1
22	PY_VAR2	PY_VAR3
23	sad	PY_VAR1
24	angel	15211309

**Figura 23. Pantalla empleados registrados en servidor local administrador.**



Opciones para filtrar contenido						
Parte	Numero de parte	Familia	Familia o area	Version	Version del diagrama	Filtrar
Diagramas registrados						
ID Diagrama	Numero de parte	Familia o area	Version	Ruta de imagen		
1	15211309	TLoZ	1.0	Diagramas/Link		
2	15211308	TLoZ	1.0	Diagramas/Link.png		
3	15211307	Halo	1.2	Diagramas/Link.png		
8	LP-S015	RUBBER MOLDING	1	Diagramas/LP-S015.png		
9	LP-S049-0018	RUBBER MOLDING	1	Diagramas/LP-S049-0018.png		
10	MCL-S003-0277	RUBBER MOLDING	1	Diagramas/MCL-S003-0277.png		
11	MC-S060-0193	RUBBER MOLDING	1	Diagramas/MC-S060-0193.png		
12	MIN-S006	RUBBER MOLDING	1	Diagramas/MIN-S006.png		

Figura 24. Pantalla diagramas registrados en servidor local administrador.

En la pantalla del dashboard se puede visualizar la gráfica que indica el Yield, las piezas aceptadas, las rechazadas y el total de la producción de cada número de parte, esto con la finalidad de poder medir el desempeño por cada número de parte y orden procesada en el área de moldeo



Figura 25. Dashboard con reporte de piezas

- **Instalación de sistema Rasperry Pi 4:** En la sección de implementación se corrieron cincuenta piezas de prueba monitoreadas por el ingeniero de proceso, producción y equipo de calidad, posterior a esta evaluación previa y de acuerdo a los resultados se decidió dejar funcionando el equipo durante cuatro semanas continuas. Ver figura 26.



Figura 26. Rasperry Pi 4 con pantalla instalada. Imagen de referencia

## **CAPÍTULO 4. RESULTADOS DE LA INVESTIGACIÓN**

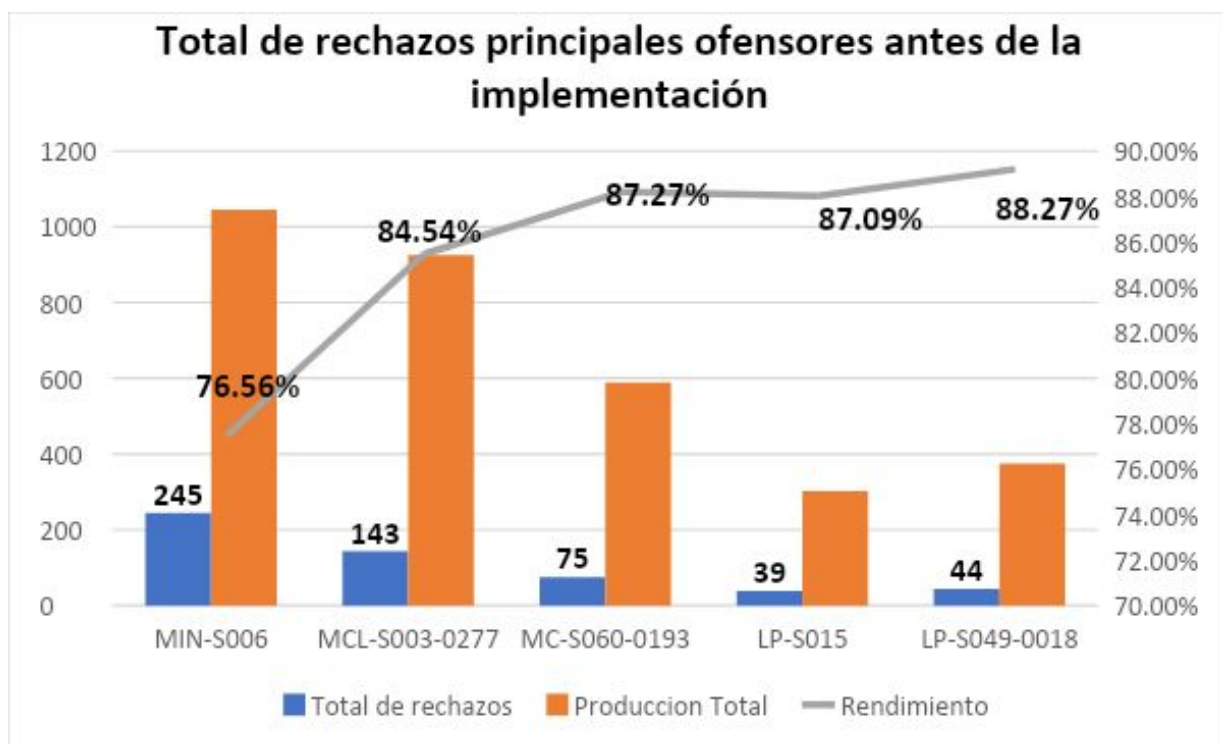
De acuerdo a los objetivos y alcances planteados del proyecto, se llevó a cabo la implementación del proyecto a manera de prueba y como prototipo en el área de moldeado designada para la evaluación, donde fue necesario monitorear y validar la funcionalidad. Describiendo a continuación los resultados enfocados a un solo equipo.

Cabe mencionar, y debido al resultado obtenido, el proyecto es justificable para la implementación en el área de moldeo, para el cual se muestra una proyección tentativa de ahorros e inversión necesaria para dicha implementación.

#### 4.1 Resultados primera etapa de implementación en área de moldeo.

De acuerdo a lo descrito en la metodología, la implementación fue realizada en el área de moldeo, donde se buscó la disminución de rechazos por *Hook up incorrecto*. Aunado a esto se instaló el sistema de Raspberry Pi 4 con el lector de código de barras para el escaneo de las formas. Para ver detalle de la situación inicial de rechazos por *Hook up incorrecto* en principales ofensores ver tabla 7.

Tabla 7. Rechazos por *Hook up incorrecto* en principales ofensores.



Información de evaluaciones y cálculos de ahorros en base a la primera etapa de implementación ver Tabla 8 y 9.

Tabla 8. Resultados preliminares de evaluación de funcionalidad de pruebas de ingeniería, en el cual se obtuvo un 88% de aceptación, considerando que los defectos detectados no son atribuibles a la implementación, la aceptación sería del 100%.

No.	INSPECCIÓN DESPUÉS DE MOLDEO	INSPECCIÓN FINAL	No.	INSPECCIÓN DESPUÉS DE MOLDEO	INSPECCIÓN FINAL
1	PIEZA OK	BUENO	26	PIEZA OK	BUENO
2	PIEZA OK	BUENO	27	PIEZA OK	BUENO
3	PIEZA OK	BUENO	28	PIEZA OK	BUENO
4	PIEZA OK	BUENO	29	PIEZA OK	BUENO
5	PIEZA OK	BUENO	30	PIEZA CON DEFECTO	MARCA MOLDE
6	PIEZA OK	BUENO	31	PIEZA OK	BUENO
7	PIEZA CON DEFECTO	BURBUJA	32	PIEZA OK	BUENO
8	PIEZA OK	BUENO	33	PIEZA OK	BUENO
9	PIEZA OK	BUENO	34	PIEZA OK	BUENO
10	PIEZA OK	BUENO	35	PIEZA CON DEFECTO	BURBUJA
11	PIEZA OK	BUENO	36	PIEZA CON DEFECTO	BUENO
12	PIEZA OK	BUENO	37	PIEZA OK	BUENO
13	PIEZA OK	BUENO	38	PIEZA OK	BUENO
14	PIEZA OK	BUENO	39	PIEZA OK	BUENO
15	PIEZA OK	BUENO	40	PIEZA OK	BUENO
16	PIEZA OK	BUENO	41	PIEZA OK	BUENO
17	PIEZA OK	BUENO	42	PIEZA CON DEFECTO	CORTO LONGITUD
18	PIEZA OK	BUENO	43	PIEZA OK	BUENO
19	PIEZA OK	BUENO	44	PIEZA OK	BUENO
20	PIEZA CON DEFECTO	BURBUJA	45	PIEZA OK	BUENO
21	PIEZA OK	BUENO	46	PIEZA OK	BUENO
22	PIEZA OK	BUENO	47	PIEZA OK	BUENO
23	PIEZA OK	BUENO	48	PIEZA OK	BUENO
24	PIEZA OK	BUENO	49	PIEZA OK	BUENO
25	PIEZA OK	BUENO	50	PIEZA CON DEFECTO	BURBUJA

Tabla 9. Costo de desperdicio durante la primera implementación.

PRODUCCIÓN MENSUAL PIEZAS EN MOLDEO	DESPERDICIO MENSUAL EN PORCENTAJE (MOLDEO)	COSTO DESPERDICIO MENSUAL POR HOOK UP (USD)	COSTO ANUAL DESPERDICIO HOOK UP (USD)
80500	16%	\$336.64	\$4,039.68

Tabla 10. Ahorro de 84% registrado durante la primera etapa de implementación de proyecto comparado con la situación inicial.

	<b>COSTO DE DESPERDICIO MENSUAL POR <i>Hook up</i> (USD)</b>	<b>COSTO DE DESPERDICIO ANUAL POR <i>Hook up</i> (USD)</b>
<b>COSTO INICIAL POR DESPERDICIO DE <i>Hook up</i> ANTES DE LA PRIMERA ETAPA DEL PROYECTO</b>	<b>\$2,104</b>	<b>\$25,254</b>
<b>COSTO POR DESPERDICIO DE <i>Hook up</i> DURANTE LA PRIMERA ETAPA DEL PROYECTO</b>	<b>\$336.64</b>	<b>\$4,039.68</b>
<b>AHORROS</b>	<b>\$1,767.36</b>	<b>\$ 21,214.32</b>
<b>PORCENTAJE DE AHORRO DURANTE LA PRIMERA ETAPA</b>	<b>84%</b>	<b>84%</b>

#### 4.2 Resultados segunda etapa de implementación en área de moldeo.

Con el ahorro registrado por la primera etapa de la implementación se decidió continuar con una segunda implementación dejando correr las piezas por 4 semanas. Para ver detalle de la evaluación y cálculos de ahorro por reducción de *Hook Up* ver tabla 11 y tabla 12.

Tabla 11. Resultados preliminares de evaluación de funcionalidad de segunda implementación de pruebas de ingeniería, en el cual se obtuvo un 84% de aceptación, considerando que los defectos detectados no son atribuibles a la implementación, la aceptación sería del 100%.

No.	INSPECCIÓN DESPUÉS DE MOLDEO	INSPECCIÓN FINAL	No.	INSPECCIÓN DESPUÉS DE MOLDEO	INSPECCIÓN FINAL
1	PIEZA OK	BUENO	26	PIEZA OK	BUENO
2	PIEZA OK	BUENO	27	PIEZA OK	BUENO
3	PIEZA OK	BUENO	28	PIEZA OK	BUENO
4	PIEZA OK	BUENO	29	PIEZA OK	BUENO
5	PIEZA OK	BUENO	30	PIEZA CON DEFECTO	MARCA MOLDE
6	PIEZA OK	BUENO	31	PIEZA OK	BUENO
7	PIEZA CON DEFECTO	BURBUJA	32	PIEZA OK	BUENO
8	PIEZA OK	BUENO	33	PIEZA OK	BUENO
9	PIEZA OK	BUENO	34	PIEZA OK	BUENO
10	PIEZA OK	BUENO	35	PIEZA CON DEFECTO	BURBUJA
11	PIEZA OK	BUENO	36	PIEZA CON DEFECTO	BUENO
12	PIEZA OK	BUENO	37	PIEZA OK	BUENO
13	PIEZA OK	BUENO	38	PIEZA OK	BUENO
14	PIEZA CON DEFECTO	CORTO LONGITUD	39	PIEZA OK	BUENO
15	PIEZA OK	BUENO	40	PIEZA OK	BUENO
16	PIEZA OK	BUENO	41	PIEZA OK	BUENO
17	PIEZA OK	BUENO	42	PIEZA CON DEFECTO	CORTO LONGITUD
18	PIEZA OK	BUENO	43	PIEZA OK	BUENO
19	PIEZA OK	BUENO	44	PIEZA OK	BUENO
20	PIEZA CON DEFECTO	BURBUJA	45	PIEZA OK	BUENO
21	PIEZA OK	BUENO	46	PIEZA OK	BUENO
22	PIEZA OK	BUENO	47	PIEZA OK	BUENO
23	PIEZA OK	BUENO	48	PIEZA OK	BUENO
24	PIEZA OK	BUENO	49	PIEZA OK	BUENO
25	PIEZA CON DEFECTO	BURBUJA	50	PIEZA CON DEFECTO	BURBUJA

Tabla 12. Costo de desperdicio durante la segunda implementación (4 semanas).

PRODUCCIÓN MENSUAL PIEZAS EN MOLDEO	DESPERDICIO MENSUAL EN PORCENTAJE (MOLDEO)	COSTO DESPERDICIO MENSUAL POR HOOK UP (USD)	COSTO ANUAL DESPERDICIO HOOK UP (USD)
79500	12%	\$252.48	\$3029.76

Tabla 13. Ahorro de 88% registrado durante la segunda etapa de implementación de proyecto comparado con el inicio del proyecto.

	<b>COSTO DE DESPERDICIO MENSUAL POR <i>Hook up</i> (USD)</b>	<b>COSTO DE DESPERDICIO ANUAL POR <i>Hook up</i> (USD)</b>
<b>COSTO INICIAL POR DESPERDICIO DE <i>Hook up</i> ANTES DE LA PRIMERA ETAPA DEL PROYECTO</b>	<b>\$2,104</b>	<b>\$25,254</b>
<b>COSTO POR DESPERDICIO DE <i>Hook up</i> DURANTE LA PRIMERA ETAPA DEL PROYECTO</b>	<b>\$252.48</b>	<b>\$3029.76</b>
<b>AHORROS</b>	<b>\$1,851.52</b>	<b>\$ 22,224.24</b>
<b>PORCENTAJE DE AHORRO DURANTE LA PRIMERA ETAPA</b>	<b>88%</b>	<b>88%</b>

A continuación, se muestra la gráfica del porcentaje de *Hook up* y del *FPY* antes y después de la implementación, así como la proyección de ahorros totales planteados para este año. Ver figura 27 y figura 28.

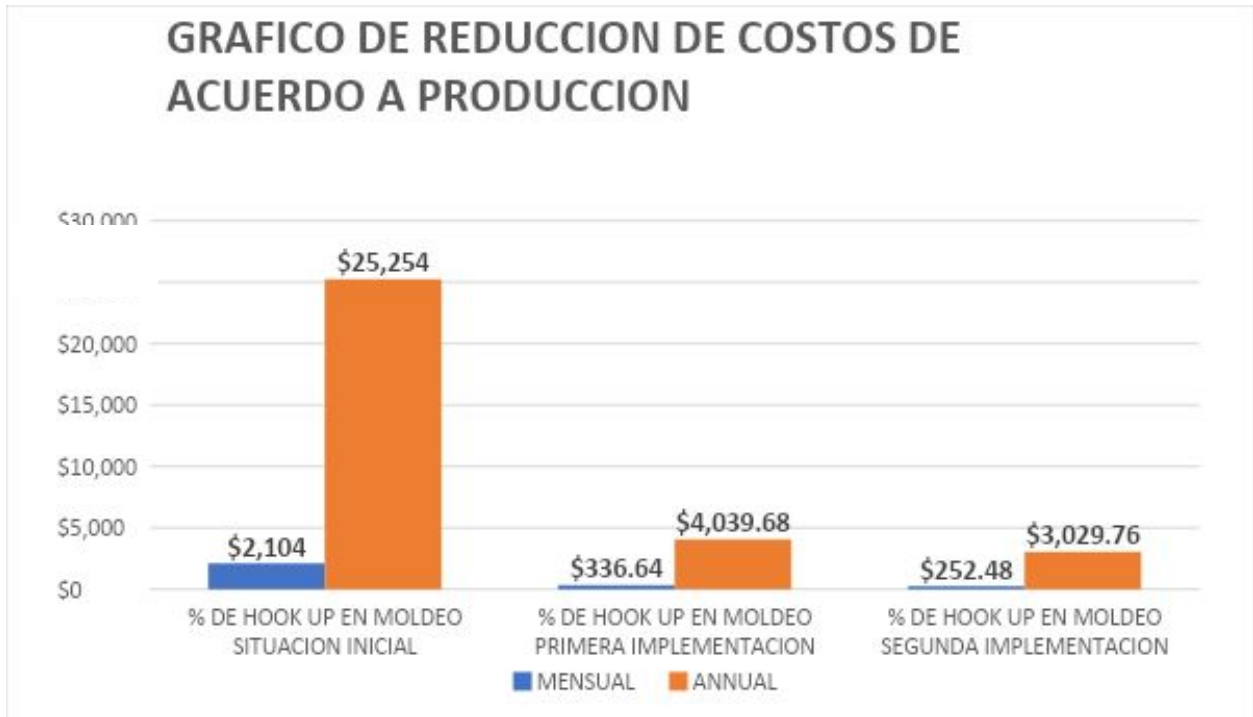


Figura 27. La gráfica comparativa que muestra el comportamiento de *Hook up* en área de moldeo desde la situación inicial hasta la conclusión de las 2 etapas de mejora realizadas, donde se observa una reducción que representa un 88% de reducción en el defecto.



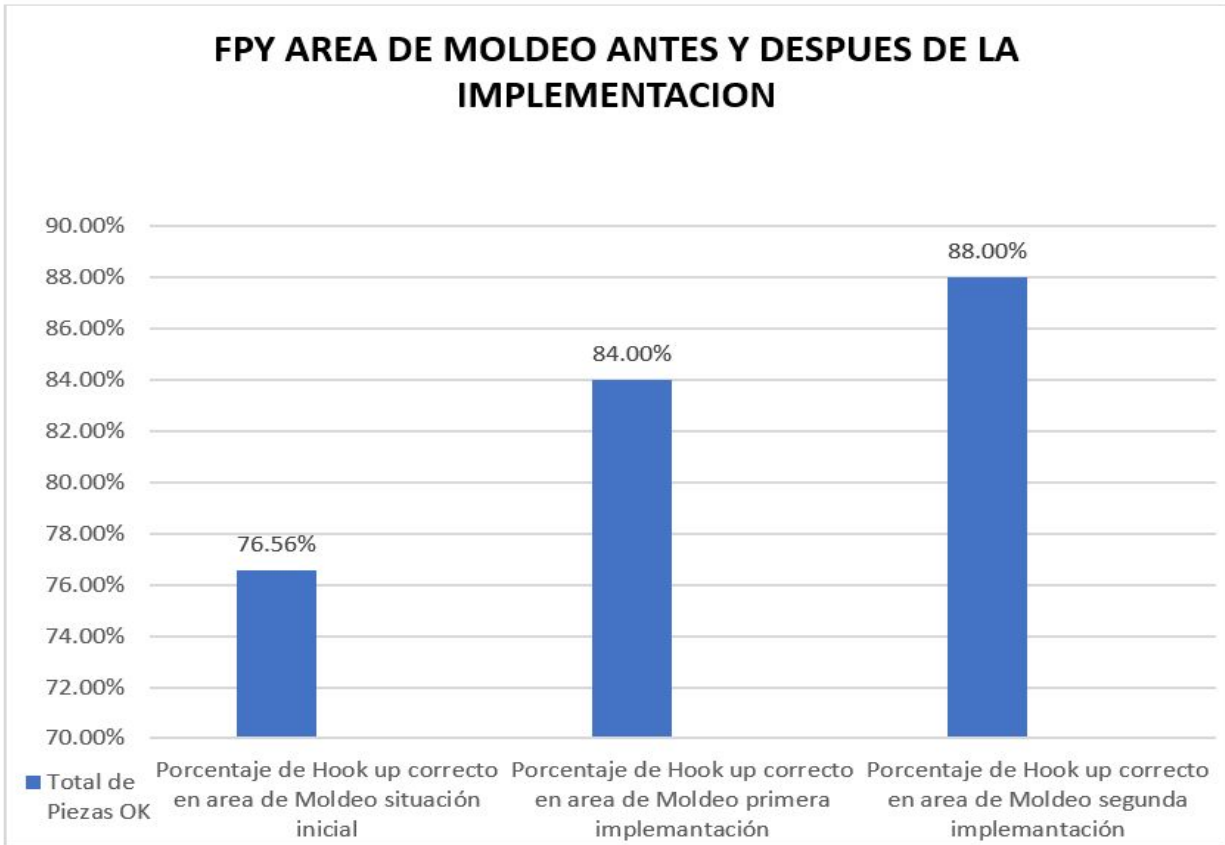


Figura 28. Se muestra el comportamiento del FPY del área de moldeo antes y después de la implementación del sistema Raspberry PI 4, quedando ubicada en 88% lo cual es atribuible a la reducción por hook up.

Tabla 14. Proyección de ahorros totales en caso de implementar sistema en las áreas de moldeo.

	COSTO DE DESPERDICIO MENSUAL POR Hook up (USD) ÁREA DE MOLDEO	COSTO DE DESPERDICIO ANUAL POR Hook up (USD) ÁREA DE MOLDEO
<b>SITUACIÓN INICIAL</b>	<b>\$2,104</b>	<b>\$25,254</b>
<b>SITUACIÓN DESPUÉS DE IMPLEMENTACION DE SISTEMA RASPBERRY PI</b>	<b>\$252.48</b>	<b>\$3029.76</b>
<b>AHORROS TOTALES</b>	<b>\$1,851.52</b>	<b>\$ 22,224.24</b>

## CAPÍTULO 5. CONCLUSIÓN Y TRABAJOS FUTUROS

La implementación resultó exitosa en la celda que se designó para realizar la mejora, con lo cual se logró obtener una reducción de 88% por el defecto de *Hook up incorrecto* en el área de moldeo. Con el presente proyecto realizado se logró cumplir la reducción de retrabajo y aumentó el Yield de piezas buenas a la primera.

Este 88% representa un ahorro mensual en defecto *hook up incorrecto* de **\$1851.52 USD**, de manera mensual y de **\$22,224.24 USD** anuales, únicamente en la celda implementada. Por el lado el rendimiento general en *Yield* se incrementó a un 98.5%, considerando únicamente el incremento por reducción de *hook up incorrecto* de 88%.

Lo cual se podría llevar a cabo únicamente con una inversión de **\$262.5 USD** por el costo del equipo o **\$1312.5 USD** en total por 5 equipos, ya que al ser exitosa la implementación en la celda asignada se buscará implementar en otras celdas, para poder erradicar este problema de forma completa en todas las celdas afectadas.

Este proyecto implementado en la celda representa un incremento significativo y ahorro que contribuye a la optimización y mejora de la empresa con la aplicación de las herramientas de ingeniería, dando oportunidad a realizar la réplica en el resto de las estaciones de moldeo y adicional el evaluar esta mejora para poder implementarse en otras áreas como lo es Soldadura y el proceso de Epóxico.

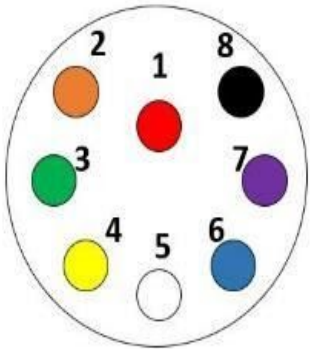
## Bibliografía

- [1] J. G. Arrieta Posada, Herramientas de Producción. Ayudas para el mejoramiento de los procesos productivos, Colombia: Atmire, 2011.
- [2] M. E. RODAL , Industria 4.0 Conceptos, tecnologías habilitadoras y retos, Madrid: Pirámide, 2020.
- [3] J. V. Robinson, "MyOpenLab," MyOpenLab Project, Colombia, 2018.
- [4] L. Socconini, Lean Manufacturing. Paso a Paso, Barcelona: Marge Books, 2019.
- [5] A.F, "¿Qué es Raspberry Pi y para qué sirve?," ABC, 21 JULIO 2013.
- [6] E. Bahit, Introducción al lenguaje Python, Buenos Aires, Argentina: Creative Commons Atribución 4.0, 2011.
- [7] B. A. Meier, Python GUI Programming, Birmingham: Packt Publishing Ltd., 2017.
- [8] J. K., "Introducción a MariaDB," Internacional de Creative Commons, California, 2015.
- [9] J. E. Pérez, "Introducción a HTML," Creative Commons, California, 2008.
- [10] J. Lett, Bootstrap Reference Guide, California: Bootstrap Creative, 2013.
- [11] J. V. Robinson, "2018," MyOpenLab, Colombia, 2018.

# ANEXOS

## Anexo 1: Instrucción digital para número de parte MCL-S003-0277

VISTA DE MOLDEO

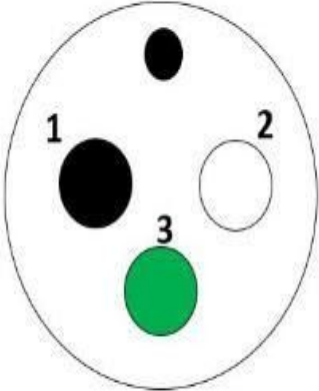


### INSTRUCCIÓN DIGITAL

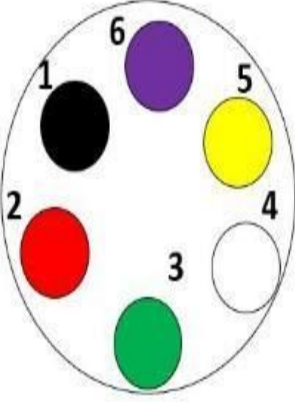
MCL-S003-0277	
POSICION	COLOR
1	ROJO
2	NARANJA
3	VERDE
4	AMARILLO
5	BLANCO
6	AZUL
7	MORADO
8	NEGRO

Documentos de Referencia:	Área : RUBBER MOLDING	
<input checked="" type="checkbox"/> DCLT/DWGN#: 107-13-6100-01 REV.A	No. de Control Instrucción Digital:	Rev:
<input type="checkbox"/> JPP: N/A		
<input type="checkbox"/> IT: N/A	Descripción:	<b>MCL-S003-0277</b>
<input type="checkbox"/> OTROS:		

**Anexo 2: Instrucción digital para número de parte MIN-S006**

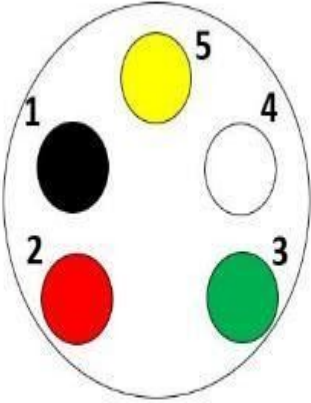
INSTRUCCIÓN DIGITAL																			
<p>VISTA DE MOLDEO</p> 	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #4F81BD; color: white;"> <th colspan="2">MIN-S006</th> </tr> <tr style="background-color: #D9E1F2;"> <th>POSICION</th> <th>COLOR</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">NEGRO</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">BLANCO</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">VERDE</td> </tr> </tbody> </table>	MIN-S006		POSICION	COLOR	1	NEGRO	2	BLANCO	3	VERDE								
MIN-S006																			
POSICION	COLOR																		
1	NEGRO																		
2	BLANCO																		
3	VERDE																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Documentos de Referencia:</td> <td colspan="2">Área : RUBBER MOLDING</td> </tr> <tr> <td><input checked="" type="checkbox"/> DCLT/DWG#:</td> <td style="width: 30%;">No. de Control Instrucción Digital:</td> <td style="width: 20%;">Rev:</td> </tr> <tr> <td style="text-align: center;"><b>7036-460 REV A</b></td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/> JPP: N/A</td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/> IT: N/A</td> <td>Descripción:</td> <td style="text-align: center; vertical-align: middle;"><b>MIN-S006</b></td> </tr> <tr> <td><input type="checkbox"/> OTROS:</td> <td></td> <td></td> </tr> </table>	Documentos de Referencia:	Área : RUBBER MOLDING		<input checked="" type="checkbox"/> DCLT/DWG#:	No. de Control Instrucción Digital:	Rev:	<b>7036-460 REV A</b>			<input type="checkbox"/> JPP: N/A			<input type="checkbox"/> IT: N/A	Descripción:	<b>MIN-S006</b>	<input type="checkbox"/> OTROS:			
Documentos de Referencia:	Área : RUBBER MOLDING																		
<input checked="" type="checkbox"/> DCLT/DWG#:	No. de Control Instrucción Digital:	Rev:																	
<b>7036-460 REV A</b>																			
<input type="checkbox"/> JPP: N/A																			
<input type="checkbox"/> IT: N/A	Descripción:	<b>MIN-S006</b>																	
<input type="checkbox"/> OTROS:																			

**Anexo 3: Instrucción digital para número de parte MC-S060-0193**

INSTRUCCIÓN DIGITAL																	
<p>VISTA DE MOLDEO</p> 	<table border="1"> <thead> <tr> <th colspan="2">MC-S060-0193</th> </tr> <tr> <th>POSICION</th> <th>COLOR</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>NEGRO</td> </tr> <tr> <td>2</td> <td>ROJO</td> </tr> <tr> <td>3</td> <td>VERDE</td> </tr> <tr> <td>4</td> <td>BLANCO</td> </tr> <tr> <td>5</td> <td>AMARILLO</td> </tr> <tr> <td>6</td> <td>MORADO</td> </tr> </tbody> </table>	MC-S060-0193		POSICION	COLOR	1	NEGRO	2	ROJO	3	VERDE	4	BLANCO	5	AMARILLO	6	MORADO
MC-S060-0193																	
POSICION	COLOR																
1	NEGRO																
2	ROJO																
3	VERDE																
4	BLANCO																
5	AMARILLO																
6	MORADO																
<p>Documentos de Referencia:</p> <p><input checked="" type="checkbox"/> DCLT/DWG#: <b>64-3-004 REV G</b></p> <p><input type="checkbox"/> IPP: N/A</p> <p><input type="checkbox"/> IT: N/A</p> <p><input type="checkbox"/> OTROS:</p>	<p>Área : RUBBER MOLDING</p> <table border="1"> <tr> <td>No. de Control Instrucción Digital:</td> <td>Rev:</td> </tr> <tr> <td> </td> <td> </td> </tr> </table> <p>Descripción: <b>MC-S060-0193</b></p>	No. de Control Instrucción Digital:	Rev:														
No. de Control Instrucción Digital:	Rev:																

**Anexo 4: Instrucción digital para número de parte LP-S015**

VISTA DE MOLDEO



**INSTRUCCIÓN DIGITAL**

**LP-S015**

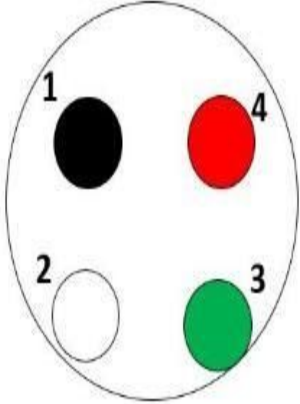
POSICION	COLOR
1	NEGRO
2	BLANCO
3	ROJO
4	VERDE
5	NARANJA

Documentos de Referencia:	Área : RUBBER MOLDING	
<input checked="" type="checkbox"/> DCLT/DWGR:	No. de Control Instrucción Digital:	Rev:
<b>7449-406 REV-</b>		
<input type="checkbox"/> IPP: N/A		
<input type="checkbox"/> IT: N/A	Descripción:	<b>LP-S015</b>
<input type="checkbox"/> OTROS:		

**Anexo 5: Instrucción digital para número de parte LP-S049-0018**

VISTA DE MOLDEO



**INSTRUCCIÓN DIGITAL**

**LP-S049-0018**

POSICION	COLOR
1	NEGRO
2	BLANCO
3	VERDE
4	ROJO

Documentos de Referencia:	Área : RUBBER MOLDING	
<input checked="" type="checkbox"/> DCLT/DWG#:	No. de Control Instrucción Digital:	Rev:
64-3-001 REV.E		
<input type="checkbox"/> IPP: N/A	Descripción:  <b>LP-S049-0018</b>	
<input type="checkbox"/> IT: N/A		
<input type="checkbox"/> OTROS:		



## Anexo 6: Código de programación de Registro de usuario

```
from tkinter import *
from tkinter import messagebox
import Administrador
import mariadb
import tkinter as tk
import os
import sys

# Connect to MariaDB Platform

class Registro(tk.Toplevel):
    def __init__(self, parent):

        super().__init__(parent)
        self.parent=parent
        self.title("Registro")
        self.geometry("350x350+0+0")
        self.protocol("WM_DELETE_WINDOW", self.volver)

    try:
        self.conn = mariadb.connect(
            user="admin",
            password="1309",
            host="localhost",
            port=3306,
            database="Visuales"

        except mariadb.Error as e:
            print(f"Error connecting to MariaDB Platform: {e}")
            sys.exit(1)

# Get Cursor
```

```
self.cur = self.conn.cursor()
```

```
global nombre
```

```
global numero
```

```
global entrada_nombre
```

```
global entrada_numero
```

```
self.nombre=StringVar()
```

```
self.numero=StringVar()
```

```
Label(self, text="Introduzca sus datos", bg="LightGreen", width="300", height="2", font=("Calibri", 20)).pack()
```

```
Label(self, text="").pack()
```

```
etiqueta_nombre=Label(self, text="Nombre", font=("Calibri", 15))
```

```
etiqueta_nombre.pack()
```

```
entrada_nombre=Entry(self, textvariable=self.nombre, font=("Calibri", 15))
```

```
entrada_nombre.pack()
```

```
entrada_nombre.focus()
```

```
Label(self, text="").pack()
```

```
etiqueta_numero= Label(self, text="Numero de empleado", font=("Calibri", 15))
```

```
etiqueta_numero.pack()
```

```
entrada_numero=Entry(self, textvariable=self.numero, font=("Calibri", 15))
```

```
entrada_numero.pack()
```

```
Label(self, text="").pack()
```

```
Button(self, text="Registrarse", width=10, height=1, bg="LightGreen", command=self.registro, font=("Calibri", 15)).pack()
```

```
Label(self, text="").pack()
```

```

Button(self, text="Volver", width=5, height=1, bg="SteelBlue1", command=self.volver, font=("Calibri", 15)).pack()
self.parent.withdraw()

def volver(self):
    Administrador.Administrador(self.parent)
    self.destroy()
    self.conn.close()

def registro(self):
    try:
        contador=0
        self.cur.execute("SELECT NumEmpleado FROM Empleado WHERE NumEmpleado=?", (self.numero.get(),))
        for NumEmpleado in self.cur:
            contador=contador+1
        if contador>=1:
            messagebox.showinfo(message="Ya existe un usuario con dicho numero", title="Numero existente")
            entrada_numero.delete(0, 'end')
            entrada_numero.focus()
        else:
            if str(self.nombre.get()).isspace() ==False and str(self.nombre.get())!=" and
str(self.numero.get()).isspace()==False and str(self.numero.get())!=":
                self.cur.execute("INSERT INTO Empleado (Nombre,NumEmpleado) VALUES (?, ?)",
(str(self.nombre.get()),str(self.numero.get())))
                messagebox.showinfo(message="Se ha registrado con exito", title="Exito")
                entrada_nombre.delete(0, 'end')
                entrada_numero.delete(0, 'end')
                entrada_nombre.focus()
            else:
                messagebox.showinfo(message="Ingrese su nombre o numero de empleado", title="Error")
                entrada_nombre.focus()
    except mariadb.Error as e:
        print(f"Error: {e}")
self.conn.commit()

```

## Anexo 7: Código de programación de Registro de usuario

```
from tkinter import *
import Registro
import Sistema
import Modificar
import Administrador
from tkinter import messagebox
import mariadb
import sys
from Interfaz import Interfaz_Main
import tkinter as tk
import os

class Main(tk.Frame):
    #Ventana principal

    def __init__(self, parent, *args, **kwargs):
        super().__init__(parent, *args, **kwargs)
        self.parent=parent

        global nombre
        global numero
        global entrada_login_nombre
        global entrada_login_numero

        self.nombre=StringVar()
        self.numero=StringVar()
        self.empleadoid=IntVar()
        self.parent.title("Inicio")

        Label(self, text="Inicie sesion o registrese", bg="LightGreen", width="300", height="2", font=("Calibri", 20)).pack()

        Label(self, text="").pack()

        Label(self, text="Introduce tu usuario", font=("Calibri", 15)).pack()
```

```
entrada_login_nombre=Entry(self, textvariable=self.nombre, font=("Calibri", 15))
```

```
entrada_login_nombre.pack()
```

```
entrada_login_nombre.focus()
```

```
Label(self, text="").pack()
```

```
Label(self, text="Introduce tu contraseña", font=("Calibri", 15)).pack()
```

```
entrada_login_numero=Entry(self, textvariable=self.numero, font=("Calibri", 15))
```

```
entrada_login_numero.pack()
```

```
Label(self, text="").pack()
```

```
Button(self, text="Iniciar sesion", bg="LightGreen", command=self.valores_sistema, font=("Calibri", 15)).pack()
```

```
Label(self, text="").pack()
```

```
ex=Interfaz_Main()
```

```
def valores_sistema(self):
```

```
    if str(self.nombre.get())=="admin" and str(self.numero.get())=="1521":
```

```
        messagebox.showinfo(message="Has iniciado sesion como administrador", title="Inicio exitoso")
```

```
        Administrador.Administrador(self.parent)
```

```
        entrada_login_numero.delete(0, 'end')
```

```
        entrada_login_nombre.delete(0, 'end')
```

```
        entrada_login_nombre.focus()
```

```
    else:
```

```
        try:
```

```
            self.conn = mariadb.connect(
```

```
                user="admin",
```

```
                password="1309",
```

```
                host="localhost",
```

```
                port=3306,
```

```
                database="Visuales"
```

```

    )
except mariadb.Error as e:
    print(f"Error connecting to MariaDB Platform: {e}")
    sys.exit(1)

# Get Cursor
self.cur = self.conn.cursor()
try:

    self.contador=0

    self.cur.execute("SELECT EmpleadoID, NumEmpleado, Nombre FROM Empleado WHERE
NumEmpleado=? AND Nombre=?", (self.numero.get(), self.nombre.get()))

    for EmpleadoID, NumEmpleado, Nombre in self.cur:
        self.contador=self.contador+1
    if self.contador>=1:
        self.empleadoid= EmpleadoID
        messagebox.showinfo(message="Has iniciado sesion", title="Inicio exitoso")
        Sistema.Sistema(self.parent, self.nombre.get(), self.numero.get(), self.empleadoid )
        entrada_login_numero.delete(0, 'end')
        entrada_login_nombre.delete(0, 'end')
        entrada_login_nombre.focus()

        self.conn.close()
    else:
        messagebox.showinfo(message="El usuario no existe, porfavor registrese", title="Usuario inexistente")
        entrada_login_nombre.focus()
except mariadb.Error as e:
    print(f"Error: {e}")

if __name__=="__main__":
    root=tk.Tk()
    Main(root).pack(side="top", fill="both", expand=True)
    root.mainloop()

```

## Anexo 8: Código de pantalla principal

```
from tkinter import *
import tkinter as tk
import mariadb
import time
import sys

import os

class Sistema(tk.Toplevel):
    def __init__(self, parent, nombre, numero, empleadoid):
        super().__init__(parent)
        self.nombre=tk.StringVar()
        self.numero=tk.StringVar()
        self.id=tk.StringVar()
        self.nombre.set(nombre)
        self.numero.set(numero)
        self.id.set(empleadoid)
        self.parent=parent
        self.title("Sistema de apoyo")
        self.geometry("1000x530+0+0")
        self.protocol("WM_DELETE_WINDOW", self.volver)
        self.codigo=StringVar()
        self.version=StringVar()
        self.familia=StringVar()
        self.ruta=StringVar()
        self.parte=StringVar()
        self.diagramaid=StringVar()
        self.tiempo = StringVar()

        global entrada_codigo

        self.label_timestamp=Label(self, textvariable=self.tiempo, font=("Calibri", 15))
```

```
self.label_timestamp.pack()
```

```
self.update_time()
```

```
self.label_nombre=Label(self, textvariable=self.nombre, font=("Calibri", 18))
```

```
self.label_nombre.place(x=0,y=0)
```

```
self.label_numero=Label(self, textvariable=self.numero, font=("Calibri", 15))
```

```
self.label_numero.place(x=0,y=25)
```

```
self.label_familia=Label(self, textvariable=self.familia, font=("Calibri", 15))
```

```
self.label_familia.place(x=160,y=25)
```

```
self.label_version=Label(self, textvariable=self.version, font=("Calibri", 15))
```

```
self.label_version.place(x=450,y=25)
```

```
self.label_parte=Label(self, textvariable=self.parte, font=("Calibri", 15))
```

```
self.label_parte.place(x=600,y=25)
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```

```
Label(self, text="").pack()
```



```

Label(self, text="").pack()
Label(self, text="").pack()
Label(self, text="").pack()
Label(self, text="Escanee codigo", font=("Calibri", 15)).pack()
entrada_codigo=Entry(self, textvariable=self.codigo, font=("Calibri", 15))
entrada_codigo.pack()
entrada_codigo.focus()
Label(self, text="").pack()
Button(self, text="Mostrar diagrama", bg="LightGreen", command=self.diagrama, font=("Calibri", 15)).pack()
Button(self, text="Cerrar sesion", bg="SteelBlue1", command=self.volver, font=("Calibri", 15)).place(relx=0.85,
rely=0.02)

self.parent.withdraw()
def volver(self):
    self.parent.deiconify()
    self.destroy()
def update_time(self):
    self.ts = time.localtime()
    self.readable = time.strftime("%H:%M:%S %d/%m/%Y",self.ts)
    self.tiempo.set(self.readable)
    self.label_timestamp.after(1000, self.update_time)

def diagrama(self):
    self.mostrar_imagen=Label()
    self.imagen=PhotoImage()
    try:
        self.conn = mariadb.connect(
            user="admin",
            password="1309",
            host="localhost",
            port=3306,
            database="Visuales"
        )
    except mariadb.Error as e:
        print(f"Error connecting to MariaDB Platform: {e}")

```

```

        sys.exit(1)
self.mostrar_imagen.config(image="")
self.familia.set(f"")
self.version.set(f"")
self.parte.set(f"")
# Get Cursor
self.cur = self.conn.cursor()
self.contador=0

        self.cur.execute("SELECT DiagramaID, RutaImagen, Familia, Version, NumeroParte FROM Diagrama WHERE
NumeroParte=?",(self.codigo.get(), ))
for DiagramaID, RutaImagen, Familia, Version, NumeroParte in self.cur:
    self.ruta=RutaImagen
    self.contador=self.contador+1
if self.contador>=1:
    self.diagramaid.set(DiagramaID)
    self.familia.set(f"Familia: {Familia}")
    self.version.set(f"Version: {Version}")
    self.parte.set(f"Número de parte: {NumeroParte}")
    try:
        self.imagen.config(file=self.ruta)
        self.mostrar_imagen=Label(self, image=self.imagen).place(relx=0.5, rely=0.42, anchor=CENTER)
    except:
        messagebox.showinfo(message="La ruta de imagen no es correcta", title="Error")
        self.cur.execute("INSERT INTO Log (Fecha, FKEmpleadoID, FKDiagramaID) VALUES (default, ?, ?)",
(int(str(self.id.get())), int(str(self.diagramaid.get()))))
        self.conn.commit()
else:
    messagebox.showinfo(message="El código no existe", title="Error")
self.ruta=""
entrada_codigo.delete(0, 'end')
entrada_codigo.focus()
self.conn.close()

```

## Anexo 9: Código de modificación de diagramas

```
from tkinter import *
from tkinter import messagebox
import mariadb
import Administrador
import tkinter as tk
import os
import sys

# Connect to MariaDB Platform

class Modificar(tk.Toplevel):
    def __init__(self, parent):

        super().__init__(parent)
        self.parent=parent
        self.title("Modificacion de diagramas")
        self.geometry("1000x520+0+0")
        self.protocol("WM_DELETE_WINDOW", self.volver)

    try:
        self.conn = mariadb.connect(
            user="admin",
            password="1309",
            host="localhost",
            port=3306,
            database="Visuales"
        )
    except mariadb.Error as e:
        print(f"Error connecting to MariaDB Platform: {e}")
        sys.exit(1)

    # Get Cursor
    self.cur = self.conn.cursor()
```

```
global entrada_version
global entrada_numero
global entrada_familia
global entrada_ruta
self.numero=StringVar()
self.familia=StringVar()
self.version=StringVar()
self.ruta=StringVar()
```

```
Label(self, text="").pack()
```

```
etiqueta_numero=Label(self, text="Numero de parte", font=("Calibri", 15))
etiqueta_numero.pack()
```

```
entrada_numero=Entry(self, textvariable=self.numero, font=("Calibri", 15))
entrada_numero.pack()
entrada_numero.focus()
```

```
etiqueta_familia= Label(self, text="Familia", font=("Calibri", 15))
etiqueta_familia.pack()
```

```
entrada_familia=Entry(self, textvariable=self.familia, font=("Calibri", 15))
entrada_familia.pack()
```

```
etiqueta_version= Label(self, text="Version", font=("Calibri", 15))
etiqueta_version.pack()
```

```
entrada_version=Entry(self, textvariable=self.version, font=("Calibri", 15))
entrada_version.pack()
```

```
etiqueta_ruta= Label(self, text="Nombre del archivo con extension, ejemplo(Archivo.png)", font=("Calibri", 15))
etiqueta_ruta.pack()
```

```

entrada_ruta=Entry(self, textvariable=self.ruta, font=("Calibri", 15))
entrada_ruta.pack()

Label(self, text="").pack()

Label(self, text="Para agregar o editar un diagrama es necesario que llene todos los campos",fg="Blue",
font=("Calibri", 15)).pack()

Button(self, text="Agregar", width=10, height=1, bg="LightGreen", command=self.agregar, font=("Calibri",
15)).pack()

Label(self, text="").pack()

Button(self, text="Editar", width=10, height=1, bg="gold2", command=self.editar, font=("Calibri", 15)).pack()

Label(self, text="").pack()

Label(self, text="Para borrar un diagrama solo se necesita el numero de parte",fg="Blue", font=("Calibri",
15)).pack()

Button(self, text="Borrar", width=10, height=1, bg="coral1", command=self.borrar, font=("Calibri", 15)).pack()

Label(self, text="").pack()

Button(self, text="Volver", width=5, height=1, bg="SteelBlue1", command=self.volver, font=("Calibri",
15)).place(relx=.914, rely=.015)

self.parent.withdraw()

def volver(self):
    Administrador.Administrador(self.parent)
    self.destroy()
    self.conn.close()

def agregar(self):
    try:

```

```

contador=0

self.cur.execute("SELECT NumeroParte FROM Diagrama WHERE NumeroParte=?", (self.numero.get(),))

for NumeroParte in self.cur:

    contador=contador+1

if contador>=1:

    messagebox.showinfo(message="Ya existe un diagrama con dicho número de parte", title="Número
existente")

    entrada_numero.delete(0, 'end')

    entrada_numero.focus()

else:

    if str(self.familia.get()).isspace() ==False and str(self.familia.get())!=" and
str(self.numero.get()).isspace()=False and str(self.numero.get())!=" and str(self.version.get()).isspace()=False and
str(self.version.get())!=" and str(self.ruta.get()).isspace()=False and str(self.ruta.get())!=":

        self.cur.execute("INSERT INTO Diagrama (Familia,Rutalmagen,Versión,NumeroParte) VALUES (?, ?, ?,
?)", (str(self.familia.get()),"Diagramas/"+str(self.ruta.get()), str(self.version.get()), str(self.numero.get())))

        messagebox.showinfo(message="Se agrego el diagrama con exito", title="Exito")

        entrada_version.delete(0, 'end')

        entrada_familia.delete(0, 'end')

        entrada_ruta.delete(0, 'end')

        entrada_numero.delete(0, 'end')

        entrada_numero.focus()

    else:

        messagebox.showinfo(message="Asegúrese que lleno todos los campos", title="Error")

        entrada_numero.focus()

except mariadb.Error as e:

    print(f"Error: {e}")

self.conn.commit()

def editar(self):

    try:

        contador=0

        self.cur.execute("SELECT NumeroParte FROM Diagrama WHERE NumeroParte=?", (self.numero.get(),))

        for NumeroParte in self.cur:

            contador=contador+1

```

```

if contador<1:
    messagebox.showinfo(message="No existe un diagrama con dicho número de parte", title="Error")
    entrada_numero.delete(0, 'end')
    entrada_numero.focus()
else:
    if str(self.familia.get()).isspace() ==False and str(self.familia.get())!=" and
str(self.version.get()).isspace()==False and str(self.version.get())!=" and str(self.ruta.get()).isspace()=False and
str(self.ruta.get())!=":
        self.cur.execute("UPDATE Diagrama SET Familia=?, Version=?, Rutalmagen=? WHERE
NumeroParte=?", (str(self.familia.get()),str(self.version.get()),"Diagramas/"+str(self.ruta.get()), str(self.numero.get())))
        messagebox.showinfo(message="Se modificó el diagrama con éxito", title="Exit")
        entrada_version.delete(0, 'end')
        entrada_familia.delete(0, 'end')
        entrada_ruta.delete(0, 'end')
        entrada_numero.delete(0, 'end')
        entrada_numero.focus()
    else:
        messagebox.showinfo(message="Asegúrese que lleno todos los campos", title="Error")
        entrada_numero.focus()
except mariadb.Error as e:
    print(f"Error: {e}")
self.conn.commit()

def borrar(self):
    try:
        contador=0
        self.cur.execute("SELECT NumeroParte FROM Diagrama WHERE NumeroParte=?", (self.numero.get(),))
        for NumeroParte in self.cur:
            contador=contador+1
        if contador<1:
            messagebox.showinfo(message="No existe un diagrama con dicho numero de parte", title="Numero
existente")
            entrada_numero.delete(0, 'end')
            entrada_numero.focus()

```

```
else:
    if str(self.numero.get()).isspace() == False and str(self.numero.get())!="":
        self.cur.execute("DELETE FROM Diagrama WHERE NumeroParte=?", (str(self.numero.get()),))
        messagebox.showinfo(message="Se borro el diagrama con exito", title="Exito")
        entrada_version.delete(0, 'end')
        entrada_familia.delete(0, 'end')
        entrada_ruta.delete(0, 'end')
        entrada_numero.delete(0, 'end')
        entrada_numero.focus()
    else:
        messagebox.showinfo(message="Asegurese que lleno el campo >Numero<", title="Error")
        entrada_numero.focus()
except mariadb.Error as e:
    print(f"Error: {e}")
self.conn.commit()
```



## Anexo 10: Código de administrador del programa

```
from tkinter import *
from tkinter import messagebox
import Modificar
import csv
import Registro
import mariadb
import tkinter as tk
import os
import time
import sys

# Connect to MariaDB Platform

class Administrador(tk.Toplevel):
    def __init__(self, parent):
        super().__init__(parent)
        self.parent=parent
        self.title("Administrador")
        self.geometry("500x400+0+0")
        self.protocol("WM_DELETE_WINDOW", self.volver)
        Label(self, text="").pack()
        Button(self, text="Registro", bg="LightGoldenrod1", command=self.valores_registro, font=("Calibri", 15)).pack()
        Label(self, text="").pack()
        Button(self, text="Modificar diagramas", bg="CadetBlue1", command=self.valores_modificar, font=("Calibri", 15)).pack()
        Label(self, text="").pack()
        Button(self, text="Descargar log", bg="wheat1", command=self.valores_log, font=("Calibri", 15)).pack()
        Button(self, text="Volver", width=5, height=1, bg="SteelBlue1", command=self.volver, font=("Calibri", 15)).place(relx=.83, rely=.015)
        self.parent.withdraw()

    def valores_registro(self):
```

```

Registro.Registro(self.parent)
self.destroy()

def valores_modificar(self):
    Modificar.Modificar(self.parent)
    self.destroy()

def valores_log(self):
    self.QUERY='SELECT * FROM Log;'
    try:
        self.conn = mariadb.connect(
            user="admin",
            password="1309",
            host="localhost",
            port=3306,
            database="Visuales"
        )
    except mariadb.Error as e:
        print(f"Error connecting to MariaDB Platform: {e}")
        sys.exit(1)
    self.cur = self.conn.cursor()
    self.cur.execute(self.QUERY)
    self.result=self.cur.fetchall()
    self.ts = time.localtime()
    self.readable = time.strftime("%H:%M:%S %d-%m-%Y",self.ts)
    c = csv.writer(open('Logs/'+self.readable+'.csv', 'w'))
    for x in self.result:
        c.writerow(x)
    messagebox.showinfo(message="Se ha creado el archivo en la carpeta Logs", title="Exito")
    self.conn.close()

def volver(self):
    self.parent.deiconify()
    self.destroy()

```