



SECRETARÍA DE EDUCACIÓN PÚBLICA  
TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE MÉRIDA

**ITM**

**“DESARROLLO DE HERRAMIENTA COMPUTACIONAL  
PARA EL SISTEMA DE RECOLECCIÓN DE DATOS  
PARA LA INDUSTRIA DE MANUFACTURA”**

OPCIÓN I:

**“TESIS”**

PARA OPTAR AL TÍTULO DE:

**MAESTRÍA EN INGENIERÍA**

PRESENTA:

**ING. ALFREDO GABRIEL SOSA BARRERA**

MÉRIDA, YUCATÁN, MÉXICO

2020





"2020, Año de Leonor Vicario, Benemérita Madre de la Patria"

**DEPENDENCIA: DIV. DE EST. DE POSG. E INV.**  
**No. DE OFICIO: X-253/20**

Mérida, Yucatán, **02/septiembre/2020**

**ASUNTO: AUTORIZACIÓN DE IMPRESIÓN**

**C. ALFREDO GABRIEL SOSA BARRERA**  
**PASANTE DE LA MAESTRÍA EN INGENIERÍA**  
**PRESENTE.**

De acuerdo al fallo emitido por su director **Víctor Sandoval Curmina** y la comisión revisora integrada por Carlos Alberto Luján Ramírez, Mario Renán Moreno Sabido y Agustín Flores Novelo, considerando que cubre los requisitos establecidos en el Reglamento de Titulación de los Institutos Tecnológicos le autorizamos la impresión de su trabajo profesional con la TESIS:

**"DESARROLLO DE HERRAMIENTA COMPUTACIONAL PARA EL SISTEMA DE RECOLECCIÓN DE DATOS PARA LA INDUSTRIA DE MANUFACTURA"**

**ATENTAMENTE**  
*Excelencia en Educación Tecnológica*

**HERMILA ANDREA ULIBARRI BENÍTEZ**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

C.p. Archivo  
HAUB/GMR/tja,



**S.E.P.**  
**INSTITUTO TECNOLÓGICO DE MERIDA**  
**DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION**



## **Dedicatoria**

Quisiera dedicarle este documento primeramente a mi esposa María Susana que fue el pilar principal para que pudiera terminar este documento, ya que con su amor y paciencia siempre me dio su apoyo incondicional, supo entender que, a pesar de mi edad, tenía el deseo de seguir estudiando y aprendiendo. A mi hijo Alfredo que, aunque a pesar de su corta edad entendía y sacrificaba tiempo de juego con él, para brindármelo y terminar este proyecto.

A cada uno de mis maestros que me dieron lo más valioso en esta vida, que es su tiempo y conocimiento.

A cada persona que influyó de alguna forma que pudiera terminar este proyecto le doy ¡¡¡¡¡MIL GRACIAS!!!!

## 1 Tabla de contenido

1	Tabla de contenido .....	III
	Lista de Figuras .....	VI
	Lista de tablas.....	VIII
	Lista de ecuaciones .....	VIII
	Resumen:.....	1
1.	Introducción.....	3
1.1.	Antecedentes .....	3
1.2.	Planteamiento del problema.....	7
1.3.	Objetivos .....	8
1.3.1.	Objetivo general.....	8
1.3.2.	Objetivos específicos.....	9
1.4.	Supuestos .....	9
1.5.	Justificación.....	10
2.	Marco Teórico.....	13
2.1.	Introducción.....	13
2.2.	Diferentes tipos de lenguaje.....	13
2.3.	Tipos de datos.....	17

2.3.1.	Datos estructurados.....	17
2.3.2.	Datos no estructurados.....	17
2.4.	Software usado para la búsqueda de información. ....	18
2.4.1.	IBM Hadoop.....	18
2.4.2.	Spark .....	19
2.4.3.	Mapreduce.....	19
2.4.4.	Storm .....	20
2.4.5.	Hive .....	20
2.5.	Norma ANSI_ASQC-Z1.4.....	20
2.6.	Lenguaje G de Labview.....	24
2.6.1.	Constante enumerada .....	24
2.6.2.	El arreglo de colas .....	25
2.6.3.	Uso de error.....	26
2.6.4.	Registro de desplazamiento. ....	26
2.6.5.	Mensajes a usuario.....	27
2.6.6.	Ejecución de sistema mediante instrumentos virtuales. ....	28
3.	Metodología.....	29
3.1.	Introducción.....	29
3.2.	Metas y actividades.....	29
3.2.1.	Infraestructura de la empresa .....	31

3.3.	Selección del lenguaje de programación.....	31
3.4.	Proceso de estudio de tiempos. ....	32
3.5.	Proceso de toma de decisiones. ....	34
3.6.	Origen de los datos .....	35
3.7.	Lector de código 2D .....	38
3.8.	Especificación de archivos de datos .....	39
3.9.	Desarrollo de interfaz de usuario .....	41
3.10.	Diagramas de flujos .....	44
3.10.1.	Configuración de periféricos.....	44
3.10.2.	Uso de colas.....	54
3.10.2.	Manejo y búsqueda de la información.....	56
3.10.3.	Descarga de archivos.....	67
3.10.4.	Búsqueda de un serial.....	68
4.	Resultados.....	71
4.1.	Introducción.....	71
4.2.	Pantalla de usuario.....	71
4.3.	Estudio de tiempos.....	73
4.4.	Piezas malas.....	82
4.5.	Búsqueda de una pieza específica.....	83
5.	Conclusiones .....	87

6	Referencias.....	89
---	------------------	----

### Lista de Figuras

Figura 1.	Distribución de las diferentes plantas.....	4
Figura 2.	Mapeo de servidores.....	5
Figura 3.	Índice TIOBE.....	14
Figura 4.	Gráfica de Índice TIOBE por años [4].....	15
Figura 5.	Constante enumerada.....	25
Figura 6.	Ciclos cuando y para.....	27
Figura 7.	Dialogo de dos botones [12].....	28
Figura 8.	Ejecuta sistema VI [14].....	28
Figura 9.	Hoja de captura de datos. ....	33
Figura 10.	Proceso para toma de acuerdos. ....	34
Figura 11.	Esquema de generación de datos.....	37
Figura 12.	Lector de código de barras 2D. ....	38
Figura 13.	Ejemplo de visualización de datos en una hoja de cálculo.....	39
Figura 14.	Pantallas principales de Labview. ....	41
Figura 15.	Ingreso de información de búsqueda ....	43
Figura 16.	Inicio del diagrama de flujo.....	46
Figura 17.	Paleta de archivos de configuración.....	47
Figura 18.	Programa de lectura del archivo de configuración. ....	53
Figura 19.	Generación de arreglo de colas para transporte de información.....	54



Figura 20. Estados del arreglo de colas. ....	55
Figura 21. Proceso de búsqueda. ....	56
Figura 22. Filtro de cantidad de caracteres. ....	57
Figura 23. Cadena concatenada. ....	58
Figura 24. Anexo y borrado de la lista de hojas viajeras. ....	59
Figura 25. Borrado de la cadena de hojas viajeras. ....	59
Figura 26. Selección de planta. ....	61
Figura 27. Detección de tipo de prueba. ....	62
Figura 28. Verificación de direccionamiento. ....	62
Figura 29. Caso falso de direccionamiento. ....	64
Figura 30. Caso verdadero de direccionamiento. ....	64
Figura 31. Convertir cadena de caracteres en Path. ....	65
Figura 32. Generar carpeta. ....	65
Figura 33. Selección de sobre escritura. ....	66
Figura 34. Copia de archivos. ....	67
Figura 35. Indicador de detección de archivos. ....	67
Figura 36. Búsqueda de un serial. ....	68
Figura 37. Esquemático de búsqueda serial arreglo de colas. ....	68
Figura 38. Ciclo principal buscar una pieza. ....	69
Figura 39. Búsqueda de un serial en un arreglo. ....	70
Figura 40. Pantalla final de usuario. ....	71
Figura 41. Ejemplo de búsqueda de archivo. ....	73
Figura 42. Estudio de tiempo inspección de piezas en forma manual. ....	76
Figura 43. Estudio de tiempo con herramienta. ....	77

Figura 44. Grafica de tiempos de inspección de piezas.....	78
Figura 45. Estudio de tiempo generación de SPC con herramienta.....	79
Figura 46. Estudio de tiempo de generación de SPC manual.....	80
Figura 47. Grafica de tiempo de generación de formato SPC.....	81
Figura 48. Enlistado de piezas malas.....	82
Figura 49. Tiempo de búsqueda manual.....	83
Figura 50. Tiempo de búsqueda con programa.....	84
Figura 51. Grafica de tiempos para búsqueda de una pieza.....	85

### **Lista de tablas**

Tabla 1. Código de letra para muestra [1]. .....	22
Tabla 2. Nivel de rigurosidad [1].....	23
Tabla 3. Descripción de la hoja de toma de tiempos.....	34
Tabla 4. Valores asignados al arreglo de colas de Estados.....	55
Tabla 5. Tipo de pruebas.....	61

### **Lista de ecuaciones**

Ecuación 1. Cálculo de relación de tiempos.....	75
---	----

## Resumen:

En este trabajo se presenta una herramienta computacional capaz de poder estructurar los datos recolectados de las piezas verificados en el área de Producción; esto permitirá la búsqueda de una pieza en específico por medio de un número serial. Este número serial podrá ser ingresado de manera manual o por medio de un escáner, el cual servirá para leer un código datamatrix (2D) que contiene fecha, número de hoja viajera (traveller) y serial de la pieza. El código datamatrix es colocado en cada una de las piezas fabricadas por el área de Producción.

La herramienta permite de igual manera generar un archivo con compatibilidad Excel, denominado SPC, que servirá como referencia ante una auditoría ISO en donde la empresa presenta los resultados como muestra de que se encuentran en conformidad con su proceso. Además, la propuesta presenta una base de estructura de archivos; para que pueda usarse en un módulo anexo, denominado análisis estadísticos de producción, que para efectos de este documento no se presentará en el desarrollo.

Mediante la elaboración del software se presentará un sistema el cual permite la reducción de tiempo en la búsqueda de la información, dando como consecuencia la reducción de tiempo en la toma de decisiones a nivel gerencial; esto es logrado mediante la instalación del programa en las computadoras de los gerentes para que puedan obtener la información de primera mano sin tiempos de espera. También reducir el tiempo de control de calidad al evitar volver a realizar las pruebas

eléctricas a las piezas, además evitando de igual forma una degradación del aislamiento de las piezas.

## 1. Introducción.

### 1.1. Antecedentes

En la industria, un objetivo primordial es la reducción de las mermas y maximización del producto final. Aunque a grandes rasgos es un problema general para todas las empresas, existen muy pocas herramientas que permitan poder hacer este objetivo.

A nivel mundial se presenta el sistema de Industria 4.0 como un nuevo modelo que permite la manipulación de datos, mediante la aplicación de herramientas computarizadas, permitiendo con ello no solo poder reducir mermas en tiempo y productos, sino poder visualizar e inclusive predecir algún problema futuro mediante algoritmos matemáticos.

Actualmente, son muy pocas las empresas a nivel local que tienen estas herramientas a la mano, esto es por su alto costo en desarrollo, no solamente de manera monetaria en forma directa, sino la inversión de horas hombre por parte del ingeniero o los ingenieros a cargo de dicho proyecto.

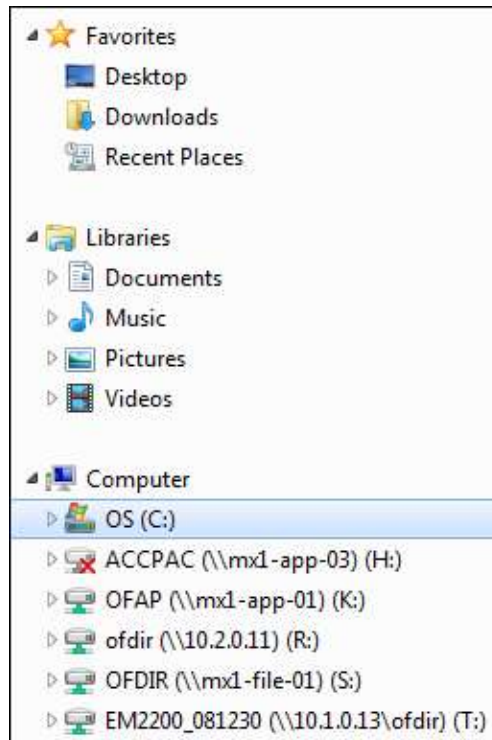
La empresa donde se plantea el proyecto fue fundada en el año de 1991, con oficinas en EUA y México (Yucatán); es una empresa a nivel internacional con 6 plantas distribuidas en 2 continentes; en el continente asiático se encuentra la planta en Xiamen y Keype, en el continente americano se encuentran las plantas 1, 2, 3, 5A, 5B y 5C. Próximamente se creará la planta 5D en México; esta distribución puede visualizarse en la Figura 1.



**Figura 1. Distribución de las diferentes plantas.**

El giro principal de la empresa consiste en la manufactura de componentes electromagnéticos de baja potencia, proceso de estampado y troquelado, moldeado de piezas plásticas para uso en sus procesos productivos; de igual forma cuenta con una planta galvánica para sus piezas de troqueladas que fabrican.

La empresa cuenta con una estructura de red de interconexión entre plantas por medio de una VPN, la cual permite poder visualizar la información que es colocada en los servidores en sus diferentes plantas; estos 3 servidores se encuentran mapeados para los usuarios de México como unidades de disco conocidas como unidad S:/, unidad R:/ y unidad T:/ que corresponden a las plantas México, India y China respectivamente, como se muestra en la Figura 2. Aunque es importante recalcar que cada planta ve su unidad de disco correspondiente como unidad S:/.



**Figura 2. Mapeo de servidores.**

También cuenta con un sistema de recolección de datos que consta a grandes rasgos de máquinas semiautomáticas que son usadas en producción para hacer las pruebas eléctricas a cada pieza que es fabricada en la empresa. Cada pieza se encuentra marcada por medio de un serial único plasmado en un código de barras 2D datamatrix y vinculados a sus respectivos valores eléctricos que genera la máquina al hacer la prueba eléctrica. Se pretende que estos datos sean usados por los superintendentes de las plantas, control de calidad, aseguramiento de calidad y los diferentes gerentes de la empresa.

En la actualidad, la empresa cuenta con datos que radican alrededor de 5 años de antigüedad, por lo que se puede decir que se tiene un vasto universo de información de cada pieza que ha sido fabricada hasta el día de hoy. Ahora bien, no se cuenta

con una herramienta computacional para hacer la búsqueda del registro de una pieza en específico, por lo que cuando es requerida alguna información; esta búsqueda lleva demasiado tiempo y es necesario realizarlo de manera manual por personal que tenga conocimiento de la estructura del sistema de recolección de datos y acceso a éstos.

Por otra parte, control de calidad toma una muestra de cada lote de acuerdo con el estándar ANSI\_ASQC-Z1.4 [1], que especifica el número de las muestras de cada lote a inspeccionar. Estas muestras tienen que ser revisadas al 100% sus parámetros eléctricos y mecánicos; por lo que un lote es aceptado o no dependiendo del nivel de rigor de la prueba y del número de defectos encontrados. Esta búsqueda y verificado de lotes representa un doble trabajo, tiempos muertos y capital humano innecesario que genera costos innecesarios para la empresa.

Otro inconveniente radica principalmente en que, al no contar con un sistema que permita el análisis de datos por parte de los gerentes y superintendentes de producción, éstos no pueden usar la información recolectada por los equipos para tomar las decisiones pertinentes antes de que un cliente presente alguna queja, lo cual provoca que la empresa pierda mucho dinero por reclamaciones o retiro de producto de las bodegas del cliente, inclusive productos en campo (cliente final). Esto no solo representa pérdida de material, sino de costos de transporte, tiempo y principalmente de la preferencia del cliente.

El software que existe actualmente para manejo de datos estadísticos de control son muy costosos para la empresa debido a que se paga una membresía anual por



usuario, y estos, aunque cuentan con herramientas estadísticas poderosas para el manejo de datos, se presenta el inconveniente que los datos recabados por los equipos se encuentran distribuidos en tres archivos, uno constituido por los seriales de solamente piezas buenas, el segundo contiene los seriales de las piezas malas, y el tercer contiene la información de cada vez que es usada la máquina. Como se puede observar, no es posible incluir los tres archivos en una sola base de datos como SQL, debido a la diversidad de productos producidos y pruebas eléctricas que son realizadas en la empresa. Aunque a grandes rasgos son pruebas comunes en la mayoría de los casos, estas presentan cambios en niveles de voltajes, frecuencias y demás parámetros haciendo que aun en una misma familia de piezas se presenten múltiples combinaciones de pruebas. Otro inconveniente radica en que los archivos donde son colocados los resultados de cada prueba son de una extensión .fal, por lo que no son fácilmente utilizados por un software comercial.

## **1.2. Planteamiento del problema**

A pesar de que la empresa cuenta con datos que radican alrededor de 5 años de antigüedad, esta no cuenta con una herramienta computacional para hacer la búsqueda del registro de una pieza en específico. Debido a esto, la búsqueda lleva demasiado tiempo y es necesario realizarlo de manera manual por personal que tenga conocimiento de la estructura del sistema de recolección de datos para tener acceso a éstos. Por consiguiente, la información que les llega a los gerentes es con días de retraso, por lo que las decisiones son tomadas cuando las piezas ya fueron enviadas al cliente. De igual manera como control de calidad, toma una muestra de

cada lote de acuerdo con el estándar ANSI\_ASQC-Z1.4, estas muestras tienen que ser revisadas nuevamente al 100% por control de calidad para aceptar el lote, por lo que les lleva tiempo hacerlo y exponiéndolas nuevamente a pruebas destructivas como alto voltaje y surge test que degradan el aislamiento físico de las piezas producidas.

### **1.3. Objetivos**

Debido a la problemática que se presentó, respecto al tiempos de respuesta, datos generados y pruebas que se realizan, se plantean los siguientes objetivos para el desarrollo del proyecto.

#### **1.3.1. Objetivo general**

Desarrollar una herramienta computacional que permita la búsqueda de información de manera rápida y precisa. Esta herramienta sentará la base para que, mediante un módulo anexo, pueda realizarse un análisis estadístico de acuerdo con los estándares 6 sigmas. Se utilizará el lenguaje G de Labview, ya que contiene librerías incorporadas que permiten una rápida interconexión entre equipos mediante NI Visa, además cuenta con módulos para compatibilidad con Excel. A su vez el lenguaje G permite ver el flujo de datos, que una lista de comandos no es posible visualizarlo. Permitirá de igual manera la búsqueda de información para control de calidad de los valores de las piezas de muestreo para aceptar los lotes sin que tengan que hacer nuevamente la revisión física de los parámetros eléctricos. Generará el formato SPC (Statidistic Process Control ), en un archivo compatible con Excel para resguardo y evidencia ante auditorias ISO (International Organization for Standardization).

### 1.3.2. Objetivos específicos

Para alcanzar el objetivo general se realizarán los siguientes objetivos específicos:

- a) Investigar las diferentes herramientas computacionales mediante la revisión de los lenguajes computacionales más utilizadas en la industria con la finalidad de generar un archivo ejecutable para la distribución en los diferentes departamentos.
- b) Definir las entradas y salidas con base en los procesos de la norma, mediante la aplicación de un diagrama de flujo para visualizar el flujo que debe llevar el proceso.
- c) Desarrollar el algoritmo a través del uso de la herramienta computacional para estructurar la información.
- d) Implementar el programa mediante el compilado del mismo, dentro de la herramienta computacional para la generación del ejecutable y del instalador para la distribución.
- e) Definir pruebas y validación de la herramienta computacional, mediante la aplicación en campo de ésta, para detectar posibles problemas menores que necesiten ser corregidos.

### 1.4. Supuestos

Con el manejo de la herramienta computacional se espera reducir el tiempo de inspección en el departamento de control de calidad. Con esto se tendrá la seguridad de que las piezas inspeccionadas bajo la norma ANSI\_ASQC\_Z1.4

cumplieron al 100% las especificaciones establecidas y no por error humano o negligencia del operador fueran entregadas y empacadas para envío piezas con defecto como producto terminado para cumplir su cantidad estándar.

Además, una herramienta computacional facilita la captura de los resultados de las pruebas hechas a las piezas, evitando cualquier error humano en la generación de los archivos SPC al momento de la captura de datos de las pruebas. Lo cual permitiría disminuir el tiempo de respuesta y toma de decisiones a los problemas presentados en producción.

Se evitará que las piezas sean expuestas y degradadas innecesariamente, mediante la aplicación de una segunda prueba eléctrica que degrade y envejecimiento prematuro del aislamiento interno de las piezas, ya que se contará con el registro de pruebas de producción.

### **1.5. Justificación**

Hoy en día se encuentra en auge a nivel manufactura la industria 4.0, que es a grandes rasgos la recopilación de datos y la manipulación de éstos para la toma de decisiones [2].

Para ello se presentan dos vértices principales:

1. Generación de datos.
2. Manipulación de datos para toma de decisiones.

Estos dos vértices deben trabajar en conjunto, a causa de que existe una fuerte dependencia entre ambos; si genero los datos, estos no sirven para tomar decisiones, entonces estos datos son inservibles, en otras palabras, resultan ser trabajo en vano. Ahora bien, si se considera que se tiene el sistema de manipulación para toma de decisiones, pero no se tienen datos para analizar, entonces igualmente esta herramienta es vana e inservible.

La empresa presenta un sistema de recolección de datos, pero no cuenta con una herramienta computacional que permita poder realizar este análisis de datos. Es un proyecto que nació en el año 2013 y que, tras varios años de recolección, se tiene una vasta cantidad de datos que pueden servir para realizar dicha herramienta computacional.

Esta herramienta de igual forma será usada por los siguientes departamentos:

- Producción bajo el resguardo del superintendente de producción.
- Control de calidad, que no necesitará hacer nuevamente todas las pruebas eléctricas, ya que podrá visualizar los resultados de las pruebas hechas en producción, solamente escaneando el código de barras de la pieza y generará su reporte en el formato SPC.
- Gerente de control de calidad, podrá realizar la búsqueda de los datos de las piezas al momento de que un cliente solicite la información de una pieza en específico o de un lote.

Con esto, la empresa tendría una herramienta la cual servirá de gran manera a los diferentes departamentos involucrados, teniendo la información que requiere en un

espacio corto de tiempo; a su vez, permitirá a la empresa poder contar con los datos para hacer análisis y reducir sus problemas mediante toma de decisiones oportunas. Por otra parte, este trabajo presentará las bases para la tesis: “Software de análisis estadístico para la industria manufacturera [3]” realizado por la Ing. Seidy Beatriz Tun Chan.

## 2. Marco Teórico

### 2.1. Introducción

En este capítulo se revisará el estado del arte entre diferentes lenguajes de programación en los cuales se presentan comandos para búsqueda y manejo de información para base de datos, donde se mostrará mediante una tabla la posición que ocupan a nivel mundial como los más usados. De igual manera se presentará una breve descripción de los tipos de datos que actualmente se pueden encontrar de acuerdo a su forma y no a la estructura, sin profundizar en bases de datos, así como revisar cuales son los programas usados por las grandes empresas a nivel mundial para la búsqueda de la información. También se proporcionará una breve descripción de la norma ANSI\_ASQC-Z1.4 usada para la inspección de lotes mediante datos estadísticos; por último, se presentará una breve descripción del lenguaje G de Labview usado en el desarrollo de este proyecto.

### 2.2. Diferentes tipos de lenguaje

Existen varios indicadores del uso de programas para el desarrollo de software; uno de estos indicadores y el cual se utilizará es el TIOBE, el cual es actualizado mes a mes por una empresa británica que su principal tarea es usar un algoritmo que permite buscar datos en los mayores buscadores del mundo.

Un punto para resaltar es que no menciona la complejidad o calidad del lenguaje, sino que solamente hace referencia a que tan usado es el lenguaje, por lo que en las figuras 3 y 4 se muestran los datos encontrados en el índice TIOBE de abril del 2019 [4].

Abr 2019	Abr 2018	Cambio	Lenguaje de programación	Calificaciones	Cambio
1	1		Java	15.035%	-0.74%
2	2		do	14.076%	+ 0.49%
3	3		C ++	8.838%	+ 1.62%
4	4		Pitón	8.166%	+ 2.36%
5	6	▲	Visual Basic .NET	5.795%	+ 0.85%
6	5	▼	DO#	3.515%	-1.75%
7	8	▲	JavaScript	2.507%	-0.99%
8	9	▲	SQL	2.272%	-0.38%
9	7	▼	PHP	2.239%	-1.98%
10	14	▲▲	Lenguaje ensamblador	1.710%	+ 0.05%
11	18	▲▲	C objetivo	1.505%	+ 0.25%
12	17	▲▲	MATLAB	1.285%	-0.17%
13	10	▼	Rubí	1.277%	-0.74%
14	dieciséis	▲	Perl	1.269%	-0.26%
15	11	▼▼	Delphi / Object Pascal	1.264%	-0,70%
dieciséis	12	▼▼	R	1.181%	-0.63%
17	13	▼▼	Visual Basic	1.060%	-0.74%
18	19	▲	lr	1.009%	-0.17%
19	15	▼▼	Rápido	0.978%	-0.56%
20	68	▲▲	Maravilloso	0.932%	+ 0.82%
			.		
			.		
			.		
38	Esquema			0.277%	
39	LabVIEW			0.214%	
40	Julia			0.209%	

**Figura 3. Índice TIOBE.**



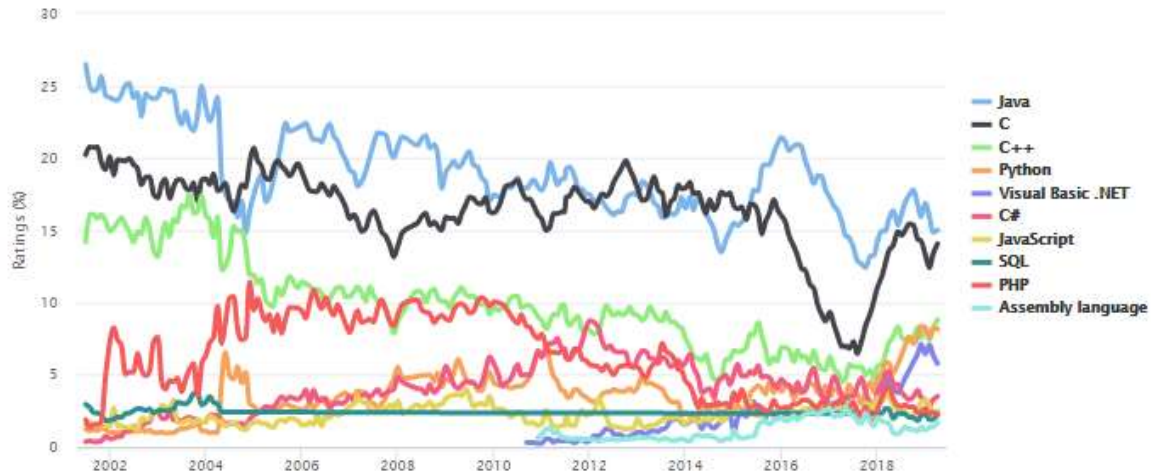


Figura 4. Gráfica de Índice TIOBE por años [4].

Como se muestra en la Figura 3 y Figura 4, los primeros cuatro lenguajes de programación son: JAVA, C, C++ y Python. Estos cuatro lenguajes son realizados a manera de sentencias estructuradas por lo que es necesario tener un amplio conocimiento de programación para llevar a cabo un programa, en comparación de Labview que su programación al ser gráfica se maneja de una manera más intuitiva [5].

Otro punto a por analizar es la facilidad para la realización de cambios en el programa y el tiempo de implementación de éstos. Al trabajar con lenguajes que requieren usar sentencias estructuradas, cualquier variación en el código repercute directamente en las líneas inferiores, aumentando con ello la complejidad en el análisis o mantenimiento del algoritmo. Labview al ser lenguaje G, es fácil la modelación gráfica de los procesos o sentencias, ya que su operación es mediante bloques que permiten visualizar el flujo de los datos entre ellos [5].

En los lenguajes de uso de sentencias, los flujos de datos no son visibles de manera inmediata durante la operación del algoritmo; por lo tanto, se necesita la

implementación de banderas que al final son variables que son integradas en el código incrementando con ello el uso de memoria. Labview tiene una herramienta de depuración interactiva, que permite visualizar el movimiento interactivo de los datos y ver los valores que pasan de una función a otra [5]. Otra de las ventajas que presenta Labview sobre los lenguajes de uso de sentencias es el depurador del compilador; éste no permite la ejecución del programa si presenta un error; de igual manera, especifica el punto exacto en donde se encuentra el problema, en comparación de los otros que solo menciona el problema [5].

Un aspecto importante a considerar es la ejecución en paralelo de las sentencias; aquí los lenguajes estructurados contienen de manera inherente información sobre que parte del código se desea ejecutar en paralelo. En este aspecto, el Labview permite el manejo de datos en forma paralela, aunque estos sean a frecuencias diferentes, por lo que los datos pueden pasar entre ellos sin problemas mediante el uso de colas sin presentar ninguna afectación entre ellos [5]. Otro aspecto a mencionar consiste en el tipo de datos con los que se trabajará, donde al ser datos semiestructurados se observa que los principales programas tienden a volverse complejos con el uso de código para el acceso de estos datos, mientras que Labview permite el acceso a ellos mediante una sentencia [5].

Aunque existen varias utilerías para interrelacionar los primeros 4 programas [6], Labview de igual forma presenta el concepto de nodo de fórmulas (Formula Node) que permite la interacción con fórmulas matemáticas de modo similar al lenguaje C. También contiene el nodo matemático (MathScrip Node) que añade programación textual matemática compatible con síntesis de los archivos con extensión punto m

[5]. Además, el Labview presenta la utilería NI-VISA que permite la conexión y comunicación a dispositivos periféricos como escáner e impresoras, teniendo como herramienta la configuración de los puertos desde el mismo programa.

### **2.3. Tipos de datos**

Existen diferentes formas de clasificación de los datos y manejo de archivos, éstos son:

1. Datos estructurados
2. Datos no estructurados

Para entender esto se realiza una breve descripción de éstos.

#### **2.3.1. Datos estructurados**

Los datos estructurados son todos los archivos que tienen una forma definida en su estructura, que pudiera ser ejemplos las bases de datos SQL, MariaDB, etc. Todos estos datos se encuentran estructurados de tal manera que puedan correlacionarse o manejar relaciones entre los datos [7].

#### **2.3.2. Datos no estructurados**

Los datos no estructurados son aquellos datos no almacenados en una base de datos tradicional. La información no estructurada no puede ser almacenada en estructuras de datos relacionales predefinidas. Como se mencionará posteriormente, estos datos presentan múltiples clasificaciones que para efecto del presente trabajo se van a considerar solamente dos de ellas a los que se denominarán datos no estructurados y datos semiestructurados [7].

Los datos semiestructurados son aquellos datos que no residen de bases de datos relacionales, pero presentan una organización interna que facilita su tratamiento, tales como documentos XML y datos almacenados en bases de datos NoSQL [7].

Los datos no estructurados presentan una división las cuales son datos de tipo texto y no-texto. Los datos de tipo texto son aquellos datos generados en las redes sociales, foros, e-mails, presentaciones Power Point o documentos Word, mientras que los datos de tipo no-texto son ficheros de imágenes jpeg, ficheros de audio mp3 o ficheros de video tipo flash [7].

## **2.4. Software usado para la búsqueda de información.**

### **2.4.1. IBM Hadoop**

IBM es una empresa internacional que cuenta con herramientas computacionales que permiten el manejo de grandes cantidades de datos (Big data), por lo que presenta el software Hadoop, el cual tiene orígenes en el año del 2004. Cuando el ingeniero de software Doug Cutting, que por aquel entonces trabajaba en Google, describe en un documento técnicas para manejar grandes volúmenes de datos, desgranándolos en problemas cada vez más pequeños para hacerlos abordables. Poco después se marchó a Yahoo y allí siguió investigando hasta completar el desarrollo de la plataforma en 2008. El propio buscador utiliza la tecnología para su negocio, así como otras grandes compañías de Internet, como Facebook, Twitter o eBay [8].

### **2.4.2. Spark**

Apache Spark fue creado en la Universidad de Berkeley (California) y es considerado el primer software de código abierto que hace la programación distribuida accesible a los científicos de datos. Al igual que Hadoop, Spark pertenece a Apache Software Foundation [9].

Según explica IBM, Apache Spark es una infraestructura de informática de clúster de código abierto con proceso en memoria para agilizar las aplicaciones de analítica hasta 100 veces más comparado con las tecnologías actuales del mercado. Apache Spark puede reducir la complejidad de la interacción de los datos, aumentar la velocidad de proceso y mejorar las aplicaciones de misión crítica con amplia información útil [9].

Muchos especialistas en Big Data opinan que Spark es una herramienta más avanzada que Hadoop. De hecho, en 2014, Spark batió el Récord Mundial al clasificar 100 terabytes de datos en tan solo 23 minutos, superando así la marca de Hadoop, que se situaba en los 71 minutos. El motor de procesamiento Spark se ha creado para ofrecer velocidad, facilidad de uso y análisis sofisticados [9].

### **2.4.3. Mapreduce**

MapReduce de Hadoop es un marco de software para escribir trabajos que procesan enormes cantidades de datos. La entrada de datos se divide en fragmentos independientes que, a continuación, se procesan en paralelo a través de los nodos del clúster. MapReduce se puede implementar en varios lenguajes, siendo Java la implementación más común [9].

#### **2.4.4. Storm**

Apache Storm es un sistema de cálculo de código abierto, distribuido y con tolerancia a errores que permite procesar datos en tiempo real con Hadoop. Las soluciones de Storm pueden proporcionar también procesamiento de datos garantizado, con la posibilidad de reproducir los datos que no se han procesado correctamente la primera vez [9].

#### **2.4.5. Hive**

Apache Hive es un sistema de almacén de datos para Hadoop, que permite realizar resúmenes de datos, consultas y análisis de datos mediante HiveQL (un lenguaje de consultas similar a SQL). Hive se puede usar para explorar los datos de forma interactiva o para crear trabajos de procesamiento por lotes reutilizables [9].

Hive permite proyectar la estructura del proyecto en datos que en gran medida no están estructurados. Después de definir la estructura, puede usar Hive para consultar esos datos sin conocimiento de Java o MapReduce [9].

### **2.5. Norma ANSI\_ASQC-Z1.4**

La Norma ANSI-ASQC-Z1.4 [10] especifica el procedimiento de toma de muestras en la inspección de un lote por atributos. En ella se establece el tamaño de una muestra de un lote para ser inspeccionado, este tamaño de muestra es establecido por el nivel de rigidez que se necesita. Todas las muestras inician con un nivel de rigidez II que, dependiendo del comportamiento en el tiempo de las muestras tomadas, están irán cambiando a nivel I (menos rígido) o al nivel III (estricto).

A continuación, se presenta un ejemplo para comprender mejor y apreciar cómo es posible seleccionar el nivel de aceptación del lote. Primeramente, se considera un lote de 2000 piezas con un nivel de rigidez de inspección II, como se muestra en la tabla 1. Se puede apreciar que en la primera columna se encuentra denominados los tamaños de lotes, en el ejemplo se especifican 2000 piezas, por lo que se tomará la fila especificada de 1201 a 3200. Si sobre esa fila se mueve hasta la columna marcada como niveles de inspección general (general inspection levels), donde se encuentra la letra "K", esta es la letra denominada para este nivel de inspección en el tamaño de lote.

Si se considera el nivel de proporción para artículos defectuosos igual a 0.1; como primer paso; se tomaría la letra "K" como referencia, recordando que esa letra fue la que se denominó en la tabla anterior. Por lo tanto, la letra "K" en su fila especifica que el tamaño de lote de las piezas de inspección debe ser 125 piezas. Si se sigue sobre la fila "K" al moverse sobre la derecha se encuentran unas columnas especificadas por la proporción de artículos defectuosos que va desde 0.010 hasta 400, nosotros nos enfocaremos en la columna marcada como 0.1. Teniendo las coordenadas "K" y 0.1 podemos ver que la cantidad de piezas para rechazar el lote se de 1 pieza, mientras que para aceptarlo de forma directa es de 0 piezas.

Considerando este ejemplo, solo se vio para el tipo de nivel de inspección normal o tipo II, el mismo proceso es empleado en los otros niveles de inspección refiriéndonos a el tipo I que es inspección reducida y para Tipo III que es inspección rígida.

Lot or batch size	Special inspection levels				General inspection levels		
	S-1	S-2	S-3	S-4	I	II	III
2	A	A	A	A	A	A	B
9	A	A	A	A	A	B	C
16	A	A	B	B	B	C	D
26	A	B	B	C	C	D	E
51	B	B	C	C	C	E	F
91	B	B	C	D	D	F	G
151	B	C	D	E	E	G	H
281	B	C	D	E	F	H	J
501	C	C	E	F	F	I	K
1201	C	D	E	G	G	J	L
3201	C	D	F	H	H	K	M
10001	C	D	F	H	I	L	N
35001	D	E	G	J	L	M	P
150001	D	E	G	J	M	N	Q
500001	D	E	H	K	N	Q	R

Tabla 1. Código de letra para muestra [1].



Sample size code letter	Sample size	Acceptance Quality Limits, $AQLs$ , in Percent Nonconforming Items and Nonconformities per 100 Items (Normal Inspection)																											
		0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000		
A	2	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
B	3	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
C	5	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
D	8	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
E	13	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
F	20	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
G	32	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
H	50	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
J	80	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
K	125	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
L	200	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
M	315	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
N	500	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
P	800	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
Q	1250	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
R	2000	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re

↓ = Use the first sampling plan below the arrow. If sample size equals, or exceeds, lot size, carry out 100 percent inspection.  
 ↓ = Use the first sampling plan above the arrow.  
 Ac = Acceptance number.  
 Re = Rejection number.

Tabla 2. Nivel de rigurosidad [1].

## 2.6. Lenguaje G de Labview

Labview no es un lenguaje de escritura de código por sentencias de sintaxis complejas, más bien es un lenguaje que florece mediante iconos que especifican funciones que son unidos por cables, que representan flujos de datos entre ellos. Esto asemeja en mucho la estructura de componentes electrónicos, que haciendo una analogía se encuentran interconectados por pistas de transportan la corriente que los comunica.

Una bondad más que nos permite el lenguaje G es la posibilidad de visualizar los datos en forma tal que estos avanzan y son usados o generados por los diferentes iconos, así como puntos de prueba que permiten visualizar los datos si el programa estuviese corriendo a una velocidad normal de funcionamiento. Estas bondades pueden ser visualizadas en el funcionamiento de las constantes enumeradas, el arreglo de colas, el uso del error, los registros de desplazamiento, mensajes a usuarios y la ejecución de sistemas mediante instrumentos virtuales.

### 2.6.1. Constante enumerada

La constante enumerada mostrada en la Figura 5 es similar a una constante de anillo , pero no puede personalizar los valores numéricos correspondientes para las etiquetas de cadena en una constante enumerada. El valor numérico siempre es un entero de cero a  $n - 1$ , donde  $n$  es el número de valores en la constante enumerada. A diferencia de la constante de anillo, las etiquetas asociadas con un valor entero son parte del tipo de datos. Si pasa el valor de una constante enumerada a un indicador, LabVIEW muestra la etiqueta de la cadena en lugar del valor numérico.



Figura 5. Constante enumerada.

La constante enumerada solo es visible en el diagrama de bloques y no se puede ver en el panel frontal. Tampoco puede cambiar su valor en tiempo de ejecución, por lo que debe seleccionar un valor de la constante enumerada antes de ser ejecutada el VI. El control de tipo enumerado permite al usuario seleccionar un valor enumerado desde el panel frontal en tiempo de ejecución. La forma de agregar, eliminar o reorganizar valores en constantes enumeradas de la misma manera que lo hace para controles en anillo y enumerados.

Por defecto, la representación numérica de una constante enumerada es un entero sin signo de 16 bits. Puede modificares la representación de una constante enumerada a cualquier tipo de datos entero sin signo, excepto un tipo de datos entero de 64 bits sin signo [11].

### 2.6.2. El arreglo de colas

Un arreglo de colas (Queue) es un buffer de datos tipo FIFO (first in/first out), en donde el primer dato que entra es el primero que sale. Por ejemplo, los clientes esperando ser atendidos en la fila de un restaurant de comida rápida asemejan el comportamiento de datos almacenados dentro de un arreglo de colas. El primer cliente que llega, es el primer cliente que será atendido [12]; este ejemplo muestra la manera de como el dato tiene que es atendido o utilizado dependiendo del orden en que es capturado en la cola, permitiendo una estructura organizada.

También un arreglo de colas es útil cuando es utilizado un patrón de diseño productor/consumidor, donde una porción del código está almacenando datos que serán procesados en otra sección del código. Otra ventaja de usar los arreglos de colas es la diferencia de velocidad que pueden tener el código que genera los datos y el código que procesa los datos, estos dos no necesitan ser la misma. Si el código que procesa es más lento que el código que produce, el arreglo de colas almacenará toda la información y el código productor tendrá que esperar hasta que el código consumidor haya vaciado algún elemento antes de introducir otro a la cola. [12].

### **2.6.3. Uso de error**

El bus es considerado únicamente como un agrupamiento de variables denominado cluster de información el cual contienen dos variables un tipo texto y una variable booleana (verdadero o falso) para transportar información entre los diferentes instrumentos virtuales, pero también puede ser usado como un secuenciador, ya que por principio Labview no permite el uso de un VI sin que se termine la ejecución de uno anterior si estos se encuentran ligados en forma consecutiva. Usando este principio se puede utilizar el bus de error para definir el orden en que los VI se van a ir ejecutando y la secuencia de los mismos, aunque no compartan otro tipo de variable; este método será usado durante la explicación del desarrollo del proyecto, para evitar que los VI generen conflicto o que algunas veces el programa pareciera que funciona y otras no.

### **2.6.4. Registro de desplazamiento.**

El registro de desplazamiento (shift register) es usado en para el transporte de valores, cuando los ciclos repetitivos necesitan ser usados en el siguiente ciclo, en

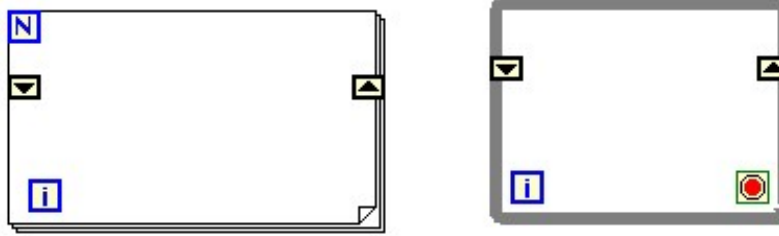


Figura 6. Ciclos cuando y para.

otras palabras, lo que realiza es dar el valor de salida en tiempo  $t-1$ , por ejemplo, si el valor de salida fue un 2, este valor puede ser leído nuevamente en el siguiente ciclo, en el registro de desplazamiento y ser usado nuevamente en el tiempo  $t$ ; este registro de desplazamiento es mostrado en la Figura 6.

#### 2.6.5. Mensajes a usuario.

Es importante determinar los mensajes a usuario que son mostrados al momento en que se encuentre desplegada una información o por el simple hecho de servir de guía para el funcionamiento del programa. Para ello es usado el desplegado de información, Labview cuenta con herramientas de desplegado de mensajes de diferentes tipos: los que sirven para interactuar con el usuario brindándoles la opción de poder seleccionar entre un "SI" o un "NO" (ver Figura 7), hasta los que simplemente solo despliegan una información al usuario.

Este VI permite tener un valor booleano a la salida de este dependiendo del seleccionado por el usuario [13].



Figura 7. Dialogo de dos botones [12].

### 2.6.6. Ejecución de sistema mediante instrumentos virtuales.

Este instrumento virtual ejecuta comandos del sistema operativo en el que se esté usando líneas de comandos, archivos por lotes para Windows y archivos script para Linux y MacOS.

En la Figura 8 pueden verse los valores de entrada y de salida; un punto destacable es que este instrumento virtual es que puede ejecutarse en un modo minimizado, por lo que permite que no se visualice en pantalla mientras se ejecuta; de igual manera tiene una entrada en donde se especifica la dirección de la carpeta en que se ejecutará el comando; el método de funcionamiento es similar el usado en el sistema operativo MS-DOS [14].

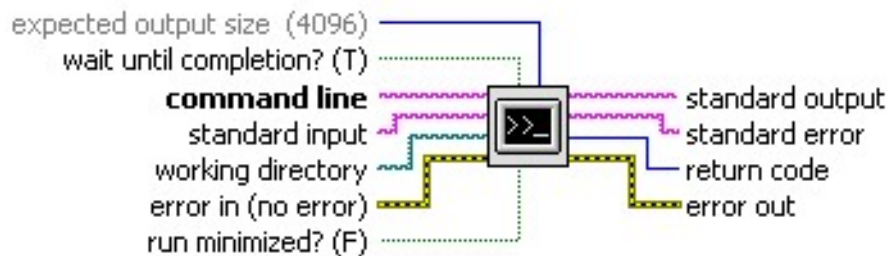


Figura 8. Ejecuta sistema VI [14].

## 3. Metodología

### 3.1. Introducción

En esta sección se presentarán los pasos seguidos en el proyecto, mediante el despliegue de las metas y actividades del proyecto, adjuntando un cronograma y un presupuesto para el funcionamiento de la primera estación. Se presentará de igual forma los criterios de selección del lenguaje de programación, el método que es usado para la generación de los datos de las piezas. También se presentará el desarrollo del proyecto usando una descripción en cada figura para una mejor comprensión del proceso.

### 3.2. Metas y actividades

Se consideran las siguientes metas y actividades para alcanzar el cumplimiento el proyecto:

- Desarrollo de interfaz de usuario en el cual se ingresen los datos para búsqueda. Para ésta se consideraron las siguientes actividades a realizar:
  - Desarrollo de interfaz de usuario.
  - Implementación de escáner.
  - Generación de algoritmo para escáner.
  - Pruebas de algoritmo.
  - Resultados de algoritmo.
- Redacción del documento para presentar a los gerentes de los departamentos involucrados, los diferentes acciones que realizará el software. Aquí se realizará las siguientes actividades:

- Realizar documento especificando detalladamente los requerimientos de los departamentos.
- Programación del algoritmo de búsqueda, para cumplir con este objetivo contamos con las siguientes actividades:
  - Investigación de algoritmos de búsqueda.
  - Generación del algoritmo.
  - Experimentar algoritmo con los datos almacenados.
  - Resultados de búsqueda.
- Generación de estructura para el manejo de datos, permitiendo el cumplimiento de este objetivo contamos con las siguientes actividades:
  - Investigación de estructuras para manejo de datos.
  - Generación de algoritmo para estructurar los datos recabados.
  - Generación de estructura de datos para la implementación de software de estadística, (software generado por Ing. Seidy Beatriz Tun Chan).
- Generación de algoritmo para impresión de datos en pantalla de SPC, para la realización de este objetivo se implementan las siguientes actividades:
  - Investigación de compatibilidad con Excel.
  - Desarrollo de código para compatibilidad con Excel.
  - Prueba de código.
  - Resultados de código.



### **3.2.1. Infraestructura de la empresa**

El precio del diseño de este software es de bajo costo, esto es debido a que la empresa ya cuenta con la infraestructura requerida para desarrollar este proyecto. La empresa cuenta con los servidores instalados en las diferentes plantas, por lo que la comunicación entre estos ya se encuentra establecida. Al contar con una VPN (Virtual Private Network) la cual permite la intercomunicación entre las plantas, cuentan con dos programas internos denominados por sus acrónimos ERP y FSN. Ambos corren bajo la plataforma SQL y fueron realizados en Visual Basic por el departamento de IT, los cuales son unos programas hechos a la medida y necesidades de la empresa. Por lo que realizando el diseño y utilizando la misma infraestructura, es posible generar un software si inversión de equipo, teniendo como gasto principal, única y exclusivamente horas hombre que se utilizarían para realizar dicha labor.

También es importante mencionar que la empresa cuenta con la licencia de Labview profesional, considerando usar este programa con lenguaje G se puede abatir el costo, por lo que no es necesario la adquisición de alguna licencia posterior.

### **3.3. Selección del lenguaje de programación**

Para la realización de este proyecto se usará el lenguaje de programación de Labview; esta decisión es tomada después de realizar el siguiente análisis de los diferentes lenguajes de programación. Aunque se observa en la Figura 3 del Índice TIOBE, que al momento de hacer la investigación este programa no se encuentra entre los primeros 10 de la lista, sino que ocupaba el lugar número 39, no por eso se considera menos apto para realizar la labor de búsqueda de información.

De hecho, aunque el lenguaje de programación se realice en un entorno gráfico, los módulos o VI que éste presenta son muy similares a los de JAVA, permitiendo tener búsquedas, conversiones de variables y lectura de archivos, en forma muy similar.

Otro punto importante a considerar en la selección del programa, es el perfil que presenta el desarrollador de la herramienta computacional. Labview presenta muchas similitudes a programas de diseño y simulación electrónicos, ya que los diferentes VI son unidos mediante líneas de acción que transportan la información similar a un cableado de un simulador.

También se tiene que considerar que la empresa que permite realizar el proyecto, cuenta con una licencia de dicho programa, permitiendo con esto no realizar un gasto más en la adquisición de otra licencia.

Se pretende que también este documento sirva como referencia a estudiantes e investigadores que deseen incursionar en este tipo de lenguaje, que pudiera servir de punto de inicio para el uso de este tipo de lenguaje.

Estos son las razones principales en la selección del programa Labview para la elaboración de la herramienta computacional.

### **3.4. Proceso de estudio de tiempos.**

Para el proceso de estudio de tiempo es utilizado un formato el cual permitirá visualizar y documentar los procesos; en él se escribirá los tiempos tomados a 10 piezas sujeta a estudio de procesos y usando una hoja de cálculo, dará los resultados de forma automática.

Para el proceso se ocupan 11 piezas, las cuales el primero no se tomará en cuenta ya que el tiempo debe tomarse bajo un proceso fluido, por lo que la primera pieza es utilizada como inicio del proceso [18] [19] [20]. En la Figura 9 se muestra la forma de la hoja de captura de datos, se codifica con colores para visualizar cada área de la plantilla.

<b>Study for cc for tester machine</b> Issue by: Date: Machine: FPN: Consumption: Operator numbers: Hour per week: Pieces to produce per week Operator approved:	Dimension of tester machine  <div style="text-align: right; color: white;">Dimension in m</div>																																																																																																																															
<table border="1" style="width: 100%; border-collapse: collapse; background-color: #f2f2f2;"> <thead> <tr> <th rowspan="2" style="width: 30%;">Operation description</th> <th colspan="11">Time loaded operator 1</th> <th rowspan="2">Pieces/hr</th> </tr> <tr> <th>T1</th><th>T2</th><th>T3</th><th>T4</th><th>T5</th><th>T6</th><th>T7</th><th>T8</th><th>T9</th><th>T10</th><th>Average</th> </tr> </thead> <tbody> <tr style="background-color: yellow;"> <td style="height: 20px;"></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> <td rowspan="4" style="text-align: center; vertical-align: middle;">0</td> </tr> <tr style="background-color: yellow;"> <td style="height: 20px;"></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr style="background-color: yellow;"> <td style="height: 20px;"></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr style="background-color: yellow;"> <td style="height: 20px;"></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td style="text-align: right;">Takt time</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="background-color: #2ecc71;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> </tbody> </table>		Operation description	Time loaded operator 1											Pieces/hr	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Average														0																																								Takt time	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																			
Operation description	Time loaded operator 1											Pieces/hr																																																																																																																				
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Average																																																																																																																					
													0																																																																																																																			
Takt time	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																		
<b>Results</b>																																																																																																																																
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Squard meters needed:</td> <td style="width: 10%;">0</td> <td style="width: 10%;">m2</td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> </tr> <tr> <td>Takt time:</td> <td>0 sec</td> <td></td> <td>time not used:</td> <td>0 sec</td> <td></td> <td colspan="10">Comments _____</td> </tr> <tr> <td>Machine time:</td> <td>0 sec</td> <td></td> <td></td> <td>0%</td> <td></td> <td colspan="10">_____</td> </tr> <tr> <td>Factor funtion:</td> <td>10%</td> <td></td> <td></td> <td></td> <td></td> <td colspan="10">_____</td> </tr> <tr> <td>CC time:</td> <td>0 sec</td> <td></td> <td>eces per week:</td> <td>0</td> <td></td> <td colspan="10">_____</td> </tr> <tr> <td>Piece per (takt time):</td> <td>0 each</td> <td></td> <td>Extra hours:</td> <td>0</td> <td></td> <td colspan="10">_____</td> </tr> <tr> <td>Piece per (Machine time):</td> <td>0 each</td> <td></td> <td>achine needed:</td> <td>0</td> <td></td> <td colspan="10">_____</td> </tr> <tr> <td>Piece per (CC time):</td> <td>0 each</td> <td></td> <td></td> <td></td> <td></td> <td colspan="10">_____</td> </tr> </table>		Squard meters needed:	0	m2													Takt time:	0 sec		time not used:	0 sec		Comments _____										Machine time:	0 sec			0%		_____										Factor funtion:	10%					_____										CC time:	0 sec		eces per week:	0		_____										Piece per (takt time):	0 each		Extra hours:	0		_____										Piece per (Machine time):	0 each		achine needed:	0		_____										Piece per (CC time):	0 each					_____									
Squard meters needed:	0	m2																																																																																																																														
Takt time:	0 sec		time not used:	0 sec		Comments _____																																																																																																																										
Machine time:	0 sec			0%		_____																																																																																																																										
Factor funtion:	10%					_____																																																																																																																										
CC time:	0 sec		eces per week:	0		_____																																																																																																																										
Piece per (takt time):	0 each		Extra hours:	0		_____																																																																																																																										
Piece per (Machine time):	0 each		achine needed:	0		_____																																																																																																																										
Piece per (CC time):	0 each					_____																																																																																																																										

Figura 9. Hoja de captura de datos.

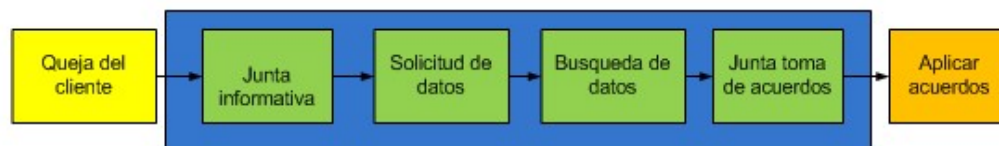
Color	Descripción
Área de descripción del proceso, consumo, persona que realizó el tiempo.	
Dibujo esquemático de dimensiones del área necesaria para el proceso.	
Pasos del proceso	
Valores de tiempo por pasos	
Valor promedio de tiempos	
Tiempo de cada proceso	

**Tabla 3. Descripción de la hoja de toma de tiempos.**

Tabla 3 se muestra la descripción de cada zona del formato de toma de tiempos; los valores permiten ver el comportamiento del proceso que se desea monitorear, que en nuestro caso sirve como métrica para determinar el comportamiento entre dos procesos; estos procesos serán: antes de implementar la herramienta computacional y después de implementar la herramienta computacional.

### 3.5. Proceso de toma de decisiones.

Uno de los puntos planteados en la problemática es el tiempo de respuesta entre que se recibe una queja del cliente y el tiempo que se aplican los acuerdos; éstos pueden tardar entre 3 a 4 días en el mejor de los casos. El proceso que se muestra en la Figura 10 describe las etapas que son usadas actualmente para tener una respuesta y acuerdo para la resolución de una queja del cliente.



**Figura 10. Proceso para toma de acuerdos.**

El proceso comienza recibiendo una queja o una alerta de parte del cliente; en esta queja el cliente proporciona los seriales que presentan algún problema, se procede a realizar una junta en donde se presenta a los representantes de los departamentos involucrados, las quejas o alertas del cliente. A continuación, se procede a la solicitud de los datos al departamento correspondiente y éste procede a la búsqueda de los datos de los seriales en los archivos de resultados.

Una vez encontrados los datos se les manda a los representantes de los departamentos involucrados y se programa una segunda junta donde se presentarán los datos encontrados y se tomarán los caminos para darle solución al problema dado por el cliente. Este esquema se repite en cada proceso de queja o advertencia que presentan los clientes.

### **3.6. Origen de los datos**

Los datos que se usarán son generados por máquinas semi-automáticas que ligan un número serial con los resultados de las pruebas eléctricas de cada pieza fabricada en la empresa. Este número serial está conformado de acuerdo al siguiente código TTTYWWSSSS, en donde:

- TTTTT se refiere al número de hoja viajera el cual es único para cada lote fabricado; este número tiene vigencia hasta el siguiente año, también se encuentra ligado directamente al nombre asignado por la empresa a la pieza realizada.
- YYWW hace mención a la fecha de fabricación de la pieza, YY determina el código del año de fabricación en dos dígitos, por ejemplo 2013 sería 13 los

primeros dos caracteres. WW se refiere a la semana de fabricación, por los que si se tiene por ejemplo el código 1305, se refiere a la semana 05 del año 2013.

- SSSSS que hace referencia al serial de la pieza en el número de hoja viajera asignado, por ejemplo, si se tiene el código 12345130500001, sería el serial 00001 de la semana 05 del año 2013 asignado a la hoja viajera 12345.

Se hace mención a esto, debido a que en cada máquina de prueba es necesario realizar el escaneo previo de la pieza para hermanar los resultados obtenidos a el serial de cada una. En la Figura 11 se muestra el proceso de generación de datos por cada máquina de prueba.

# Proceso de generación de datos en las líneas de producción

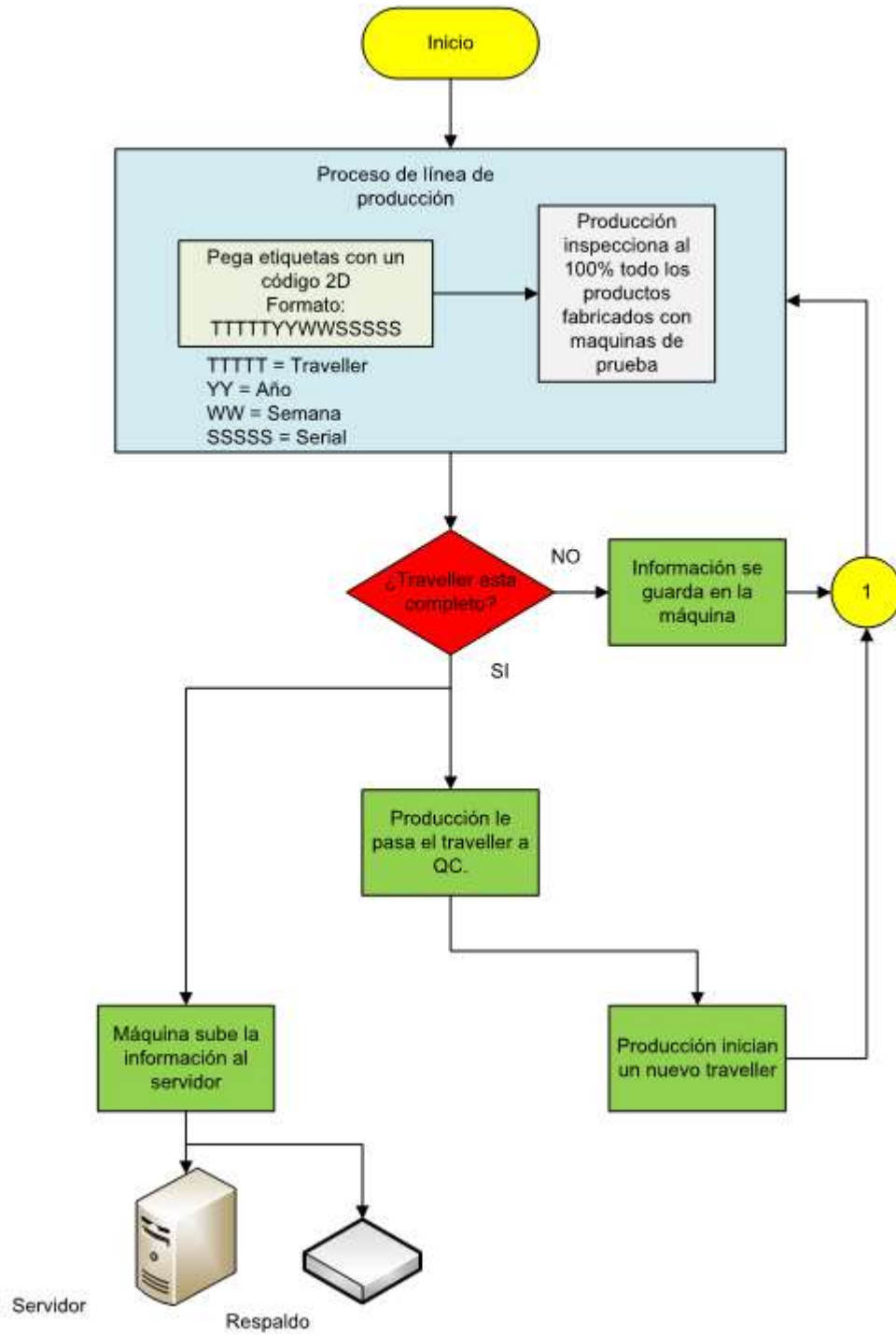


Figura 11. Esquema de generación de datos.

Es importante recalcar que la información de una hoja viajera se mantiene en la máquina de prueba y solamente es subido al servidor hasta que el número de piezas buenas asignada a la hoja viajera sea cumplida. Una vez cumplida esta condición la información estará disponible para la búsqueda de la herramienta computacional.

### 3.7. Lector de código 2D

Para el proyecto fue seleccionado el lector de la marca Honeywell [15] modelo 1400G2D-2USB-1 (ver la Figura 12). El lector cuenta con dos características principales para su selección: un driver emulador de puerto serial, que permite la interconexión rápido con NI-VISA aunque su conexión física sea USB y como segunda característica el amplio repertorio de códigos para escaneo que presenta.



Figura 12. Lector de código de barras 2D.



### 3.8. Especificación de archivos de datos

Los archivos en donde se encuentran los datos son una combinación de archivos CSV (comma-separated values) y archivos de textos. Esto es debido a que se deseaba en un principio una compatibilidad con programa de hojas de cálculo como Excel o LibreOffice. En la Figura 13 se muestra como es visualizado el archivo mediante una hoja de cálculo.

Con el motivo de realizar una búsqueda de una pieza o usar plantillas para el cálculo de datos estadísticos, en un principio esto fue fácil ya que no se tenía implementado en todas las piezas fabricadas, pero con el tiempo y el aumento de la necesidad de llevar reportes, se volvió prácticamente imposible realizar esto, por tiempo y debido a la manipulación del volumen de la información.

FPN: XXXXXX										
This File was created on:11/24/2017 03:57:58 p. m.										
Traveller: 1748_47076_01										
[TESTS]										
SERIAL	RESULT	INDUCTANC	DCR	ACR	PHASE	TURNS	DATE	HOUR	OPERATOR	TEMPERATURE
[DATA]										
47076174800531	31	1.16E+02	1.95E+01	5.07E+03	1.00E+00	2.98E+01	11/24/2017	#####	4230	
47076174800530	31	9.11E+01	1.95E+01	5.04E+03	1.00E+00	2.94E+01	11/24/2017	#####	4230	
47076174800529	31	1.28E+02	1.95E+01	5.09E+03	1.00E+00	2.99E+01	11/24/2017	#####	4230	
47076174800025	31	1.03E+02	1.91E+01	5.06E+03	1.00E+00	2.80E+01	11/25/2017	#####	4230	
47076174800022	31	1.25E+02	1.95E+01	5.06E+03	1.00E+00	2.84E+01	11/25/2017	#####	4230	
47076174800019	31	1.30E+02	1.94E+01	5.05E+03	1.00E+00	2.91E+01	11/25/2017	#####	4230	
47076174800016	31	1.75E+02	1.93E+01	5.08E+03	1.00E+00	2.85E+01	11/25/2017	#####	4230	
47076174800013	31	1.48E+02	1.93E+01	5.09E+03	1.00E+00	2.75E+01	11/25/2017	#####	4230	
47076174800010	31	1.51E+02	1.93E+01	5.07E+03	1.00E+00	2.86E+01	11/25/2017	#####	4230	
47076174800007	31	1.02E+02	1.94E+01	5.05E+03	1.00E+00	2.90E+01	11/25/2017	#####	4230	
47076174800004	31	1.03E+02	1.93E+01	5.04E+03	1.00E+00	2.84E+01	11/25/2017	#####	4230	
47076174800001	31	1.50E+02	1.89E+01	5.06E+03	1.00E+00	2.90E+01	11/25/2017	#####	4230	
47076174800070	31	1.03E+02	1.93E+01	5.06E+03	1.00E+00	2.95E+01	11/25/2017	#####	4230	
47076174800067	31	1.32E+02	1.96E+01	5.06E+03	1.00E+00	2.84E+01	11/25/2017	#####	4230	

Figura 13. Ejemplo de visualización de datos en una hoja de cálculo.

El archivo inicia con un encabezado en donde se especifica el modelo de la pieza; a continuación, se especifica la fecha y hora de creación del mismo y la especificación de la hoja viajera.

FPN: XXXXXX

This File was created on: 11/24/2017 3:57:59 PM

Traveller: 1748\_47076\_01

A continuación, se presenta una sección llamada "TESTS" en donde se enlistarán los encabezados de las pruebas realizadas a este modelo de pieza; estos se encuentran separados por un TAB; nótese que inicia con SERIAL para el serial que le corresponden las pruebas eléctricas.

[TESTS]

SERIAL/RESULT/INDUCTANCE/DCR/ACR/PHASE/TURNS/DATE/HOUR/OPERATOR/TEMPERATURE

[DATA]

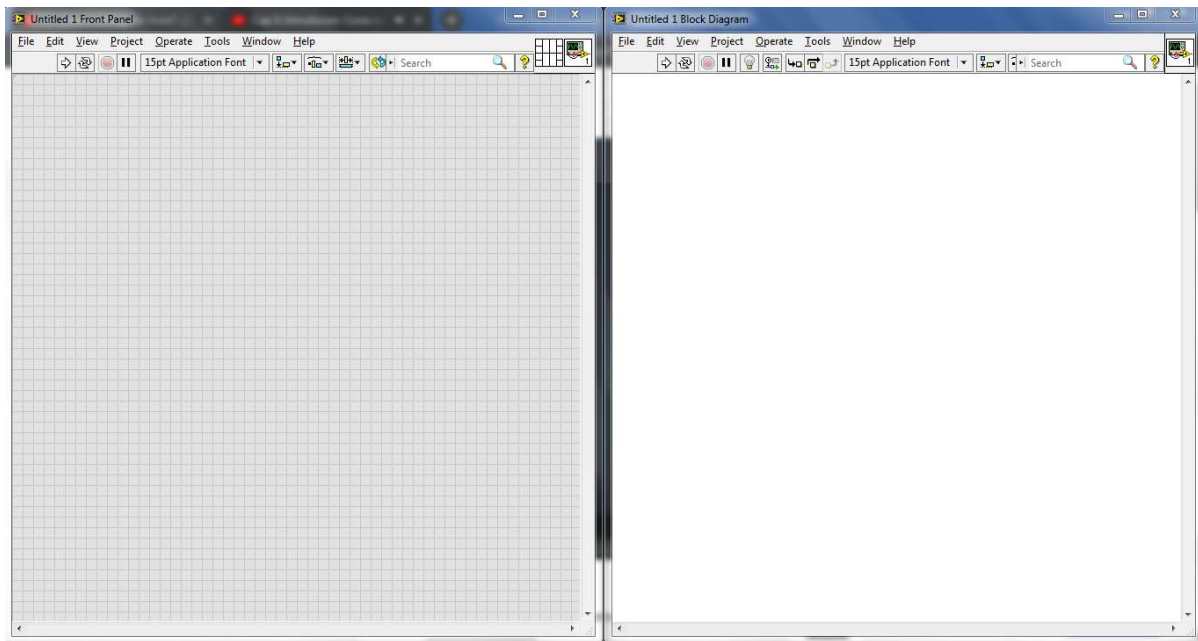
47076174800531/31/116.27E+0/19.46E+0/5.07E+3/1.00E+0/29.81E+0/11/24/2017/3:58:18PM/4230

Se ha cambiado el TAB por / para reducir espacio y visualizarlo mejor.

### 3.9. Desarrollo de interfaz de usuario

Unas de las razones principales en las que se consideró utilizar el programa de Labview es debido la generación de la interface de usuario; esta es generada al mismo tiempo en que es desarrollado el algoritmo de programación.

En la Figura 14 se muestran las dos pantallas principales de Labview; la de la izquierda es el panel frontal; en este panel serán colocados todos los indicadores y controles con los cuales podrá interactuar el usuario, mientras que en la pantalla de la derecha estarán los indicadores y controles colocados como iconos, para la interconexión e interacción con el programa. Como inicio el usuario necesitará ingresar información, la cual se enumera:



**Figura 14. Pantallas principales de Labview.**

1. El código de la parte a buscar parte
2. Número de hoja viajera o traveller.
3. El código de la fecha o Date code.
4. Tipo de prueba o Test type.
5. Número de inspector.
6. La planta en que se requiere hacer la búsqueda.

Quedando como el ejemplo mostrado en la Figura 15. Es importante recalcar que los campos en fondo blanco son los controles en donde el usuario ingresará los datos de búsqueda, mientras que los campos en tono gris son indicadores, en donde se visualizará la información ya ingresada.

Se ha anexado igual manera dos controles binarios representados como botones denominados como: "Add" el cual activará la rutina para ingreso y acomodo de la información para que pueda hacer la búsqueda, el botón "Delete" borra el último registro anexado en lista, el botón "Update files" descarga los archivos de registro para que pueda realizarse la búsqueda de manera local, para que se realice con mayor velocidad la búsqueda de la información.

En los campos denominados "List of traveller", se coloca la información que se vaya anexando; nótese que estos campos son indicadores, por los que no podrá ser modificado por el usuario en forma directa, más bien estos se modificarán por medio de la interacción con los demás controles.

**Figura 15. Ingreso de información de búsqueda**

En párrafos anteriores se mencionó la descarga de archivos, esto es debido a la estructura de los archivos, los datos que se tienen ahí son datos semiestructurados, es decir tienen definida una estructura, pero no tanto para ser manejada en una base de datos, debido a la diversidad de los tipos de productos que se realizan en la empresa.

Una pregunta muy importante es ¿porque se descargan y no se usan los archivos que se encuentran en la red?, esto es simple, ya que al no estar comprendidos en una base de datos no se tiene un sistema de regulación de manejo de archivo, si por algún motivo dos o más usuarios piden la misma información al mismo archivo al mismo tiempo, produciría un error de lectura, en cambio, si se copian los archivos al disco duro local no presentará tal conflicto, esto es a que solamente el usuario local estará requiriendo la información.

Otro motivo por el cual no se usan los archivos estando en la red, es por la carga que representa al ancho de banda de la red durante el tiempo y búsqueda de los archivos, produciendo picos de carga de uso y necesitando más ancho de banda si la carga de usuarios es mucha. Aunque este problema no es exclusivamente debido a la carga por software, sino que intervienen otros factores de hardware que limitan el uso de la información.

Estos dos son los principales problemas si se enfocamos a una red local, pero si por medio de una VPN se hace la comunicación con otros servidores, entonces se estaría consumiendo ancho de banda saliente de la planta que impacta directamente en precios de renta de equipo y costos de servicio.

Teniendo en cuenta las variables que son usadas entonces se procede con una parte primordial en el desarrollo de toda herramienta computacional y que a veces son pasados por alto el cual es: el diagrama de flujo.

### **3.10. Diagramas de flujos**

En esta sección se presentará los diferentes conceptos utilizados en la elaboración de la herramienta computacional mediante la configuración de periféricos, uso de colas, manejo y búsqueda de información, descarga de archivos y búsqueda de un serial.

#### **3.10.1. Configuración de periféricos**

En esta sección del documento se considerará que el lector se encuentra familiarizado con el concepto de diagrama de flujo, pero es importante recalcar que este sirve para tener una guía de cómo funcionará el programa, visualizando y

evaluando de manera externa el funcionamiento del mismo antes de sumergirse en el inicio de la programación.

Una vez definida las variables que se usarán, se empezará con el desarrollo de la herramienta computacional y para ello se considerará el siguiente diagrama de flujo de la Figura 16.

Es importante recordar que la herramienta computacional no solamente tendrá el teclado o el pad numérico para el ingreso de la información, sino que tendrá un periférico más que será el escáner 2D, esto es para realizar la búsqueda de las piezas de forma rápida y eficiente, reduciendo el tiempo que se necesitaría para teclear la información comprendida en cada código 2D de cada pieza.

En vista de lo anterior es indispensable configurar cada uno de los periféricos y decirle al programa en que puerto se encuentra para obtener los datos de él, pero en este momento se presenta el primer problema, ¿qué ocurre si el puerto asignado en cada máquina no es el mismo?, esto es un gran problema ya que, si se le asigna por default un puerto, las demás máquinas que usen la herramienta computacional no tiene asignado el mismo puerto al periférico, por lo que se necesitará llamar o asignar el número de puerto por máquina.

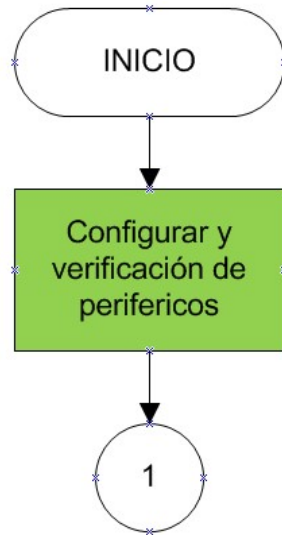


Figura 16. Inicio del diagrama de flujo.

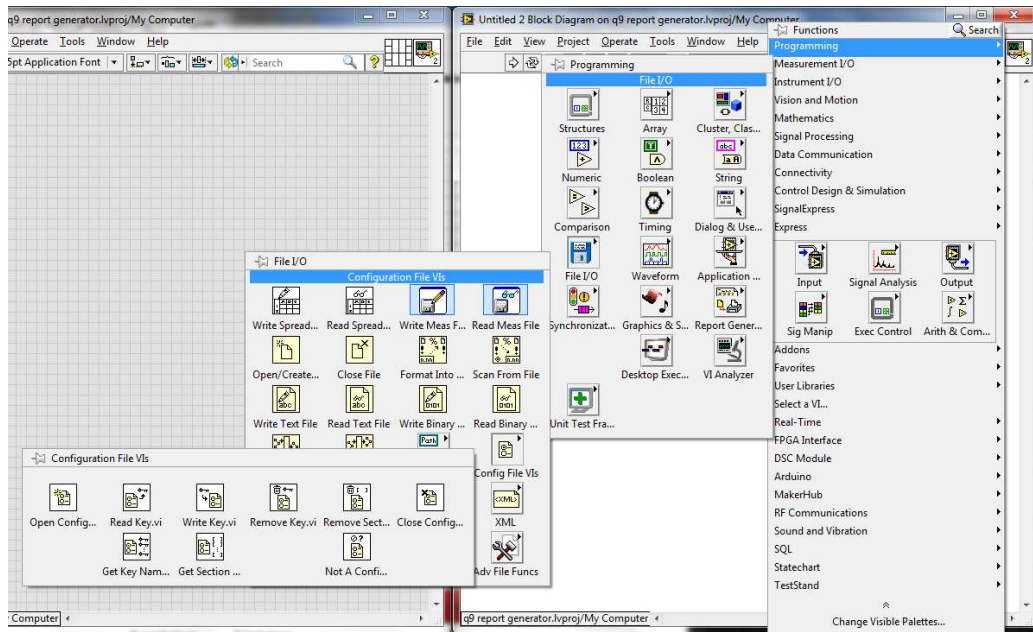
Para solucionar este problema son utilizadas las utilerías colocadas en programming > Files I/O > Configuration files VIs [16], puede verse esta ubicación en la Figura 17. Para el uso de estos rutinas o VIs, hay que generar un archivo estructurado de la siguiente forma:

```

[VISA]
COM ="COM4"
BAUDRATE = 9600
  
```

Sección de variables a leer.  
 Llave o key a ser leída, especifica el puerto del escáner.  
 Llave o key a ser leída, especifica la velocidad de comunicación del puerto





**Figura 17. Paleta de archivos de configuración.**

Estos archivos no se encuentran limitados a una sola sección; estos pueden contener múltiples secciones con el mismo nombre en las llaves (key) de cada sección. Éste sirve para determinar no solamente para que configuración de los periféricos, sino también para determinar los accesos a las diferentes plantas, de manera de restringir el acceso a las plantas y el acceso a las máquinas.

Para ello se generará 3 secciones una denominada “MACHINES” que es donde se colocará el acceso a las máquinas, la otra será “VISA” donde se establecerá la configuración con el escáner y la última se llama “SERVER”, donde se colocará la dirección de los servidores.

[MACHINES]

```
TEST1 = "mx1-getester06\Labview_Soft\Testing_x"
```

```
TEST2 = "MX1-LANDYS6\Labview_Soft\Testing_x"
```

TEST3 = "mx1-p1sample01\Labview\_Soft\Testing\_x"  
TEST4 = "mx1-p1tester01\Labview\_Soft\Testing\_x"  
TEST5 = "mx1-p1tester02\Labview\_Soft\Testing\_x"  
TEST6 = "mx1-p1tester03\Labview\_Soft\Testing\_x"  
TEST7 = "mx1-p1tester04\Labview\_Soft\Testing\_x"  
TEST8 = "mx1-p1tester05\Labview\_Soft\Testing\_x"  
TEST9 = "mx1-p1tester06\Labview\_Soft\Testing\_x"  
TEST10 = "mx1-p1tester07\Labview\_Soft\Testing\_x"  
TEST11 = "mx1-p1tester08\Labview\_Soft\Testing\_x"  
TEST12 = "mx1-p1tester09\Labview\_Soft\Testing\_x"  
TEST13 = "mx1-p1tester10\Labview\_Soft\Testing\_x"  
TEST14 = "mx1-p1tester13\Labview\_Soft\Testing\_x"  
TEST15 = "mx1-p1tester14\Labview\_Soft\Testing\_x"  
TEST16 = "mx1-p1tester15\Labview\_Soft\Testing\_x"  
TEST17 = "mx1-p1tester16\Labview\_Soft\Testing\_x"  
TEST18 = "mx1-p1tester17\Labview\_Soft\Testing\_x"  
TEST19 = "mx1-p1tester18\Labview\_Soft\Testing\_x"  
TEST20 = "mx1-p1tester19\Labview\_Soft\Testing\_x"  
TEST21 = "mx1-p2test-01\Labview\_Soft\Testing\_x"  
TEST22 = "mx1-p2test-02\Labview\_Soft\Testing\_x"  
TEST23 = "mx1-p2test-03\Labview\_Soft\Testing\_x"  
TEST24 = "mx1-p2test-05\Labview\_Soft\Testing\_x"  
TEST25 = "mx1-p2tester01\Labview\_Soft\Testing\_x"  
TEST26 = "MX1-P2TESTER08\Labview\_Soft\Testing\_x"

TEST27 = "MX1-P5TESTER01\Labview\_Soft\Testing\_x"

TEST28 = "MX1-P5TESTER02\Labview\_Soft\Testing\_x"

TEST29 = "MX1-P5TESTER04\Labview\_Soft\Testing\_x"

TEST30 = "mx1-p5tester16\Labview\_Soft\Testing\_x"

TEST31 = "MX1-TESTER04\Labview\_Soft\Testing\_x"

-----

TEST32 = "beita\Labview\_Soft\Testing\_x"

TEST33 = "cn1-class7-t1\Labview\_Soft\Testing\_x"

TEST34 = "cn1-class7-t2\Labview\_Soft\Testing\_x"

TEST35 = "cn1-class10-t1\Labview\_Soft\Testing\_x"

TEST36 = "cn1-class11-AS3XXX\Labview\_Soft\Testing\_x"

TEST37 = "cn1-class11-DPG10\Labview\_Soft\Testing\_x"

TEST38 = "cn1-class11-t1\Labview\_Soft\Testing\_x"

TEST39 = "cn1-class11-t2\Labview\_Soft\Testing\_x"

TEST40 = "cn1-class11-t3\Labview\_Soft\Testing\_x"

TEST41 = "cn1-class11-t4\Labview\_Soft\Testing\_x"

TEST42 = "cn1-class11-t8\Labview\_Soft\Testing\_x"

TEST43 = "cn1-class12-t1\Labview\_Soft\Testing\_x"

TEST44 = "cn1-class13-DPG10\Labview\_Soft\Testing\_x"

TEST45 = "cn1-class13-t1\Labview\_Soft\Testing\_x"

TEST46 = "cn1-class13-t2\Labview\_Soft\Testing\_x"

TEST47 = "cn1-class13-t3\Labview\_Soft\Testing\_x"

TEST48 = "cn1-class13-t4\Labview\_Soft\Testing\_x"

TEST49 = "cn1-class16-t1\Labview\_Soft\Testing\_x"

TEST50 = "cn1-class16-t2\Labview\_Soft\Testing\_x"  
TEST51 = "cn1-class16-t3\Labview\_Soft\Testing\_x"  
TEST52 = "cn1-class16-t4\Labview\_Soft\Testing\_x"  
TEST53 = "cn1-class18-t1\Labview\_Soft\Testing\_x"  
TEST54 = "cn1-class18-t2\Labview\_Soft\Testing\_x"  
TEST55 = "cn1-class21-t1\Labview\_Soft\Testing\_x"  
TEST56 = "cn1-class23-t2\Labview\_Soft\Testing\_x"  
TEST57 = "cn1-class27-t1\Labview\_Soft\Testing\_x"  
TEST58 = "cn1-class27-t2\Labview\_Soft\Testing\_x"  
TEST59 = "cn1-class30-t1\Labview\_Soft\Testing\_x"  
TEST60 = "cn1-class30-t2\Labview\_Soft\Testing\_x"  
TEST61 = "cn1-class31-t1\Labview\_Soft\Testing\_x"  
TEST62 = "cn1-class32-DPG10\Labview\_Soft\Testing\_x"  
TEST63 = "cn1-class32-t3\Labview\_Soft\Testing\_x"  
TEST64 = "cn1-class33-t1\Labview\_Soft\Testing\_x"  
TEST65 = "cn1-class33-t2\Labview\_Soft\Testing\_x"  
TEST66 = "cn1-class33-t3\Labview\_Soft\Testing\_x"  
TEST67 = "cn1-class33-t4\Labview\_Soft\Testing\_x"  
TEST68 = "cn1-class33-t5\Labview\_Soft\Testing\_x"  
TEST69 = "cn1-class33-t6\Labview\_Soft\Testing\_x"  
TEST70 = "cn1-class34-t1\Labview\_Soft\Testing\_x"  
TEST71 = "cn1-class35-t1\Labview\_Soft\Testing\_x"  
TEST72 = "cn1-class35-t2\Labview\_Soft\Testing\_x"  
TEST73 = "cn1-class42-t1\Labview\_Soft\Testing\_x"

TEST74 = "cn1-class42-t2\Labview\_Soft\Testing\_x"  
TEST75 = "cn1-class42-t3\Labview\_Soft\Testing\_x"  
TEST76 = "cn1-class43-t1\Labview\_Soft\Testing\_x"  
TEST77 = "cn1-class44-t2\Labview\_Soft\Testing\_x"  
TEST78 = "cn1-class46-t1\Labview\_Soft\Testing\_x"  
TEST79 = "cn1-kinberlyx\Labview\_Soft\Testing\_x"  
TEST80 = "cn1-pmeeting\Labview\_Soft\Testing\_x"  
TEST81 = "cn1-pqc3\Labview\_Soft\Testing\_x"  
TEST82 = "cn1-public\Labview\_Soft\Testing\_x"  
TEST83 = "GREDER\Labview\_Soft\Testing\_x"  
TEST84 = "HYT\Labview\_Soft\Testing\_x"  
TEST85 = "JIANQI\Labview\_Soft\Testing\_x"  
TEST86 = "WELTER\Labview\_Soft\Testing\_x"  
TEST87 = "WODE\Labview\_Soft\Testing\_x"  
TEST88 = "YOUDES\Labview\_Soft\Testing\_x"  
TEST89 = "ZONYUE\Labview\_Soft\Testing\_x"  
-----  
TEST90 = "in4-SHORT-ORDER\Labview\_Soft\Testing\_x"  
TEST91 = "in4-test1\Labview\_Soft\Testing\_x"  
TEST92 = "in4-test2\Labview\_Soft\Testing\_x"  
TEST93 = "in4-test3\Labview\_Soft\Testing\_x"  
TEST94 = "in4-test4\Labview\_Soft\Testing\_x"  
TEST95 = "in4-test5\Labview\_Soft\Testing\_x"  
TEST96 = "in4-test6\Labview\_Soft\Testing\_x"

TEST97 = "in4-test7\Labview\_Soft\Testing\_x"  
TEST98 = "in4-test8\Labview\_Soft\Testing\_x"  
TEST99 = "in4-test9\Labview\_Soft\Testing\_x"  
TEST100 = "in4-test10\Labview\_Soft\Testing\_x"  
TEST101 = "in4-test11\Labview\_Soft\Testing\_x"  
TEST102 = "in4-test12\Labview\_Soft\Testing\_x"  
TEST103 = "in4-test13\Labview\_Soft\Testing\_x"  
TEST104 = "in4-test14\Labview\_Soft\Testing\_x"  
TEST105 = "in4-test15\Labview\_Soft\Testing\_x"  
TEST106 = "in4-test16\Labview\_Soft\Testing\_x"  
TEST107 = "in4-test17\Labview\_Soft\Testing\_x"  
TEST108 = "in4-test18\Labview\_Soft\Testing\_x"  
TEST109 = "mx1-p5tester23\Labview\_Soft\Testing\_x"  
TEST110 = "mx1-p5tester25\Labview\_Soft\Testing\_x"

[VISA]

COM ="COM4"

BAUDRATE = 9600

[SERVER]

ADDRESSFPN = "FPN"

SERVER="S: \EM2200\_081230\DATA\_COLLECTION"

SERVER1="T:\DATA\_COLLECTION"

SERVER2="R:\EM2200\_081230\DATA\_COLLECTION"

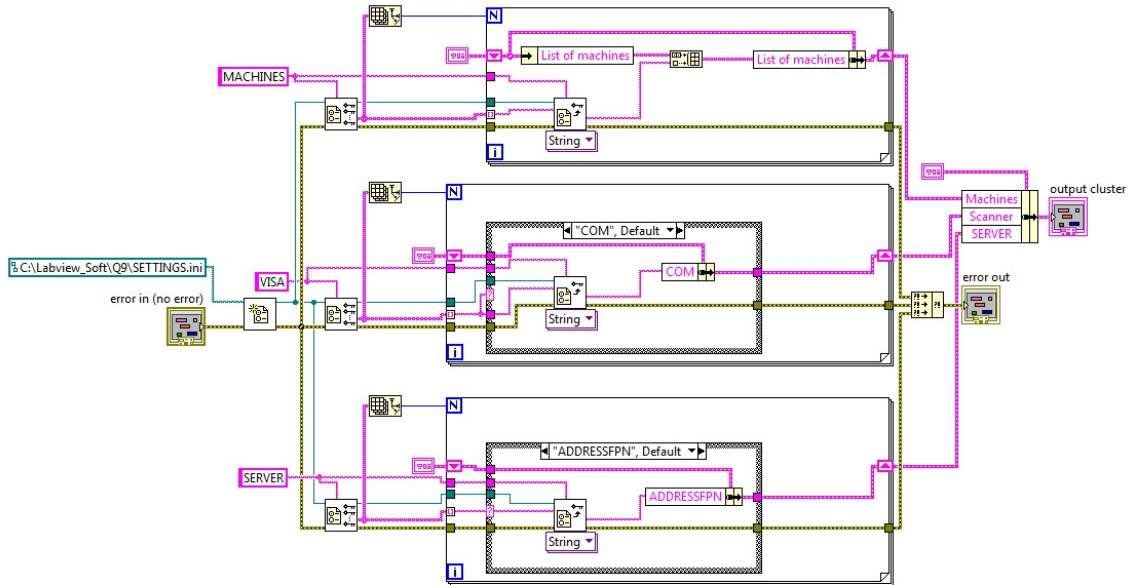


Figura 18. Programa de lectura del archivo de configuración.

Para efecto de prueba se cambiará la key SERVER de la sección SERVER por F:\P1\ para extraer toda la información de un disco duro externo, aunque se dejará debajo de esta la dirección de la ubicación en la red de la empresa.

¿Por qué dividirlo tanto y no simplemente llamar a las máquinas simplemente?, el motivo de esto es simple, en una empresa no todos tienen que tener los mismos accesos, por ejemplo una persona encargada de revisar una planta solamente, no necesita conocer ni tener acceso a lo que se encuentre pasando en otra planta, o un en una planta de otro país, en contraparte un gerente de alto mando si necesita saberlo y tener acceso a todas las plantas, por lo que se usa el archivo de esta forma para tener un mejor control de acceso.

En la Figura 18 se presenta la forma en que la lectura de las KEY se lleva a cabo, obsérvese que son tres procesos paralelos que leen el mismo archivo, permitiendo que la lectura se realice más rápido.

### 3.10.2. Uso de colas

Debido a la característica de los arreglos de colas (Queue) se pueden usar como medio de puente entre dos secciones del programa que se encuentren corriendo en forma paralela, para que una sea la generadora de información y la otra parte sea la ejecutora de los comandos, esto se puede ver en la Figura 19.

El arreglo de colas generado es del tipo entero por ello el programa lo denomina de color azul, con esto se puede hacer que el programa no se quede esperando a que se presione un botón se active, sino que más bien sea el botón que active la sección del programa reduciendo con ello el tiempo y consumo de recursos de procesador.

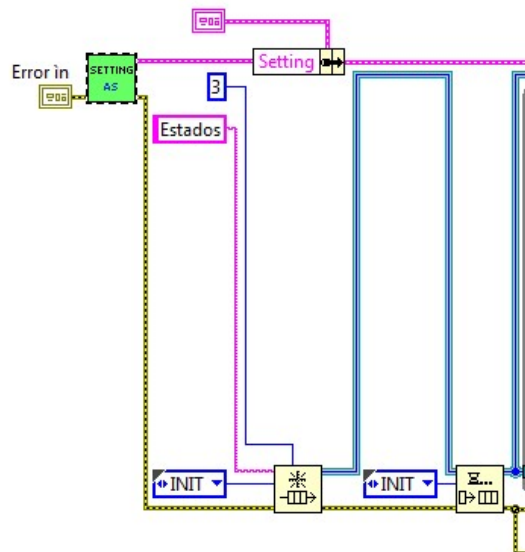


Figura 19. Generación de arreglo de colas para transporte de información.



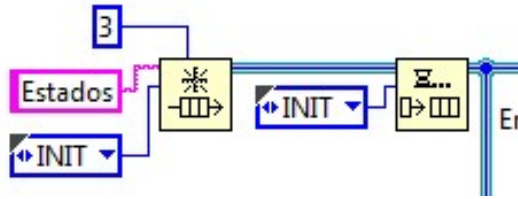


Figura 20.Estados del arreglo de colas.

Después de generar el arreglo de colas dándole un nombre y definiendo el tipo de datos que transportará, se establece el tamaño del buffer que en este caso serían 3, para que a continuación se establezca el primer dato que transportará el cual será "0", representado con la palabra "INIT"; esto puede ser visualizado en la Figura 20.

Como se vislumbró, aunque el arreglo de colas tiene ya asignado nombre y tamaño del buffer de datos, falta definir el tipo de dato que transportará, esto se hace con una variable enumerada. A esta variable enumerada se le asignan los nombres mostrados en la Tabla 4.

Nombre	Valor
INIT	0
ERROR	1
CLOSE	2
UPDATE	3
FINDPIECE	4
DOWNLOAD	5
PRINTSPC	6
FULLSPC	7
NGPRINT	8
ADD	9
PRINT PCS	10
QA	11

Tabla 4. Valores asignados al arreglo de colas de Estados.

### 3.10.2. Manejo y búsqueda de la información

Una vez pasada la etapa de configuración se pasa a lo que sería el manejo de la interface; para ello es necesario que el usuario ingrese la información mostrada en la Figura 15.

El arreglo de colas de estado está en valor "0" o INIT, como inicio se ingresan uno a uno los datos de las hojas viajeras, ¿Por qué no solamente se ingresan las piezas escaneándolas y extrayendo la información?, como esta información será usado por el departamento de control de calidad, es necesario tener un control estricto de los lotes que se está inspeccionando y si solamente se escanean las piezas para extraer información, no se sabría si el lote de la pieza ya fue inspeccionado o no, o si producción por error uso esa pieza para completar un lote. Se recuerda que el uso de software o herramientas computacionales permite la reducción de errores humanos o la detección de los mismos.

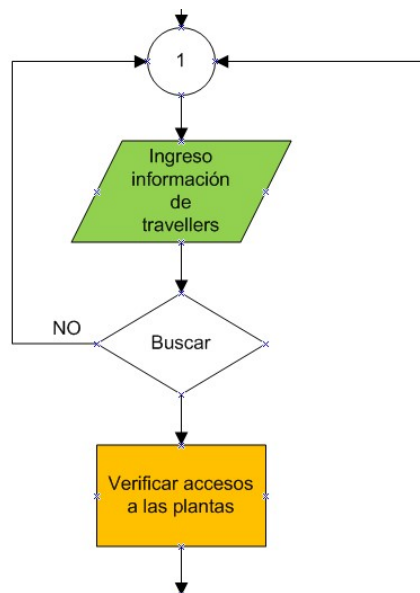


Figura 21. Proceso de búsqueda.

El usuario irá ingresando uno a uno la información de las hojas viajeras y presionando el botón “Add” para anexarlos en un arreglo de cadena de caracteres de una dimensión; para ello el software debe filtrarlo ya que como se mencionó anteriormente, se tiene cierto número de caracteres para cada sección de los seriales de las piezas y, por ende, se tiene igual cierto número de caracteres para cada campo.

Para ello se estableció que el programa pueda determinar si los campos tienen una cantidad determinada de caracteres, que serían: 5 para las hojas viajeras y 2 para los tipos de prueba Figura 22. En caso de no cumplirlo se anexarían “0” a la información hasta completar la cantidad necesaria para cada campo.

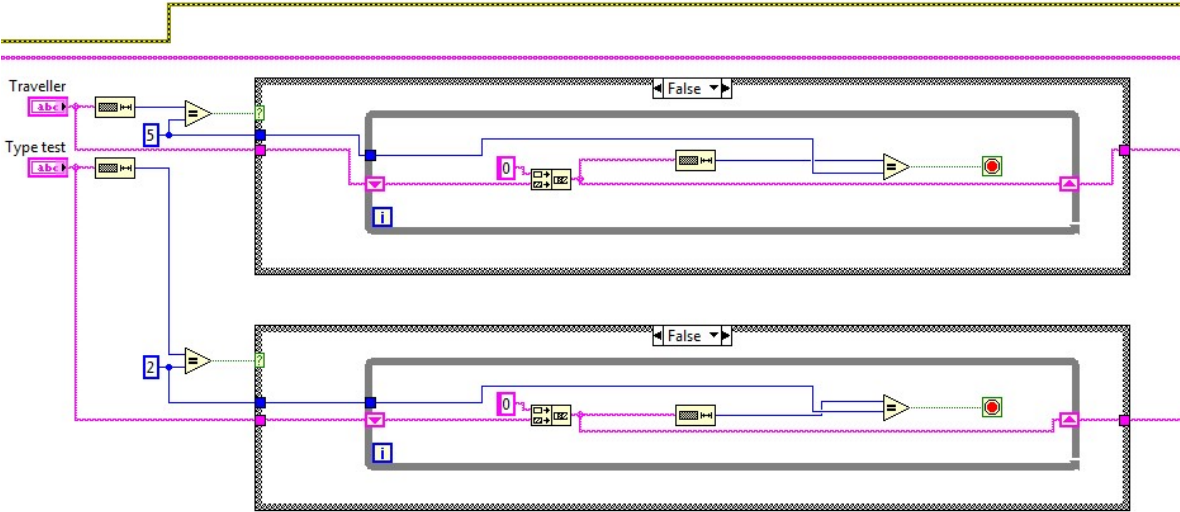


Figura 22. Filtro de cantidad de caracteres.

Nótese que la línea de error no se encuentra pasando por ellos; esto es para optimizar el tiempo de respuesta y permitir que se ejecuten ambos al mismo tiempo sin tener que esperar que uno termine para ejecutarse el siguiente, usando como referencia el proceso mencionado en el capítulo anterior como uso de error.

En la Figura 23 puede observarse que la salida de estas estructuras es conectada a una cadena concatenada; esto es para generar un arreglo donde se estructurará el nombre de los archivos para la realización de la búsqueda. En la Figura 24 se muestra cómo se encuentra interconectada la salida de la cadena concatenada y la lista de hojas viajeras. Nótese que se usa un nodo de propiedades con salida de los valores de la tabla, y mientras que el botón de borrado (Delete) no es presionado, se irán agregando en el arreglo. Esto se hará sucesivamente cada vez que se anexe una hoja viajera, mientras que si es presionado el botón de borrado se eliminará la última información del arreglo de la lista de hojas viajeras.

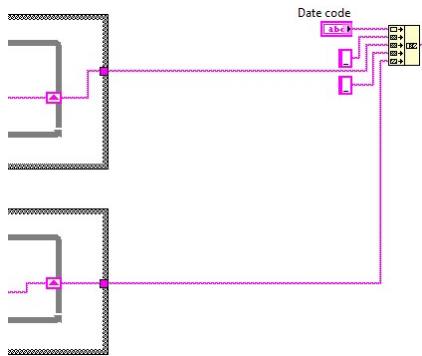


Figura 23. Cadena concatenada.

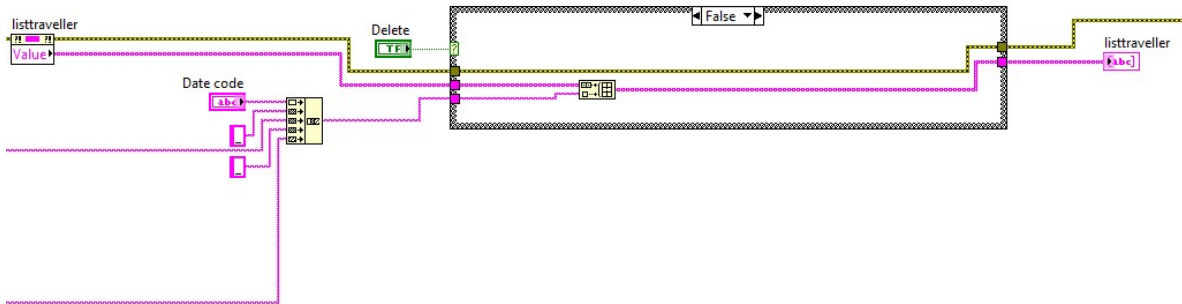


Figura 24. Anexo y borrado de la lista de hojas viajeras.

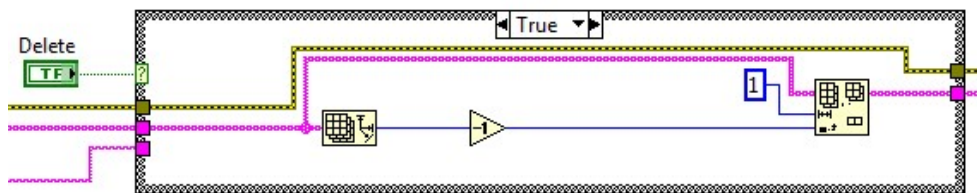


Figura 25. Borrado de la cadena de hojas viajeras.

En la Figura 25 se nota como cuando el botón de borrado es presionado; en otras palabras, es verdadero (true), la estructura cambia y activa la otra parte del caso, que lo que realiza es extraer el último dato del arreglo.

Una vez que se tiene la lista de hojas viajeras, se procede a descargarlas de los servidores; el motivo de esto es que si se intenta hacer la búsqueda de información usando la red, éste consume un ancho de banda lo que a lenta el flujo de la información; esto posiblemente no sea un conflicto grave si solamente se tuviera un ancho de banda asignado el proyecto, pero cuando es compartido con múltiples sistemas, el consumo del ancho de banda es crítico, pudiendo tener entre 3 y 5 segundos por archivo.

Esto se ve aún más cuando se trabaja con los archivos de las otras plantas del extranjero, dando tiempos de entre 45 y 50 seg, mientras que si se copian los archivos la búsqueda da un tiempo de entre 8 y 12 seg.

Si se considera que se tiene ya completa y cargada la lista de hojas viajeras, ahora se tiene que iniciar el proceso de copia de archivo antes mencionado; primeramente, lo que se tiene que verificar es que todos los archivos corresponden a un mismo tipo de prueba, ya que si son de pruebas diferentes los datos que se lean se encontrarán revueltos y la cantidad de datos no coincidirán perdiendo la estructura y control de los mismos; para ello se verifica que todos las hojas viajera cargados tengan el mismo prefijo de tipo de prueba.

En el arreglo de strings llamada lista de hojas viajeras (list of travellers) mostrada en la Figura 15, se denota que el nombre de cada uno se encuentra de la siguiente forma, YYWW\_TTTTT\_ZZ; nótese que en este caso se tienen los caracteres “ZZ”; estos hacen alusión al tipo de prueba que se está verificando; estos tipos se encuentran mostrados en la Tabla 5.

En la Figura 15 se muestra una sección en donde se selecciona la planta maquiladora en la que se desea hacer la búsqueda, se especifica MX para la planta México, In para la planta de India y CN para la planta de China. Este selector activa la estructura de casos el cual extrae valores del cluster de setting dependiendo del valor seleccionado.

Tipo de prueba	Caracteres
Alto voltaje	HP
Surge test	ST
Inductancia, capacitancia y resistencia	O1
Inductancia con corriente aplicada	DP

Tabla 5. Tipo de pruebas.

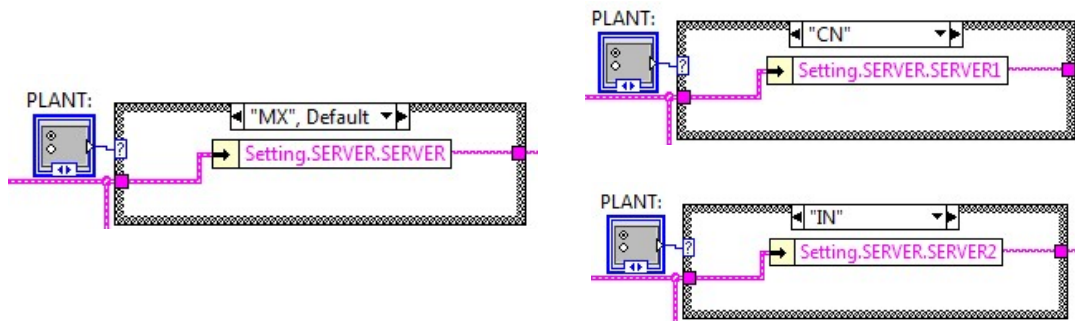


Figura 26. Selección de planta.

Esta selección de planta es debido a que cada planta por año tiene el mismo número de hoja viajera, por ejemplo, en la planta de México se tiene la hoja viajera 12345, pero en la planta china tendrán asignado la misma hoja viajera, pero a otro número de pieza, por eso es muy importante la decisión de la planta a la que se desea extraer la información.

En la Figura 26 se puede ver cómo es extraído la información de la planta dependiendo de cuál es seleccionada; es importante recordar que esta información viene del archivo settings.ini, así que, si se desea que solamente tengan acceso a uno de ellos, simplemente con no colocar la información del servidor en la planta que corresponde, éste no tendrá acceso a estos datos.

En la Figura 27 se ve un ciclo para este ciclo extraer cada uno de las cadenas de caracteres; es importante recordar que la entrada de esto es un arreglo de clúster y que en estos clústeres se encuentra contenida una cadena de caracteres y un dato booleano, por lo que al usar este ciclo solamente extrae uno por uno estos datos del clúster, y luego solamente se extrae del clúster cada cadena.

Es importante recordar de igual forma que la cadena que se extrae es del formato YYWW\_TTTT\_ZZ, por lo que se necesita eliminar todo y extraer solamente ZZ, por lo que esto se realiza con el cortador de cadena (Split string) y se realiza la comparación con respecto a dato anterior eso se hace haciendo un registro de desplazamiento (shift register) en la salida; una vez que todos los valores se han verificado, se elimina el primero, ya que la comparación de este siempre dará falso, debido a que se compara un carácter en blanco con respecto al primer tipo de prueba.

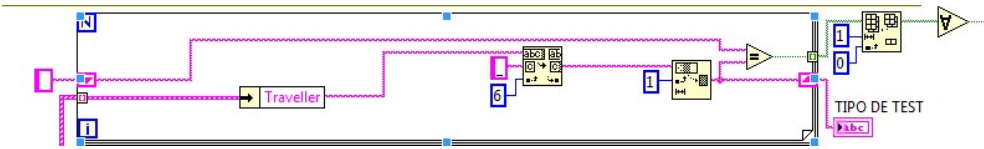


Figura 27. Detección de tipo de prueba.

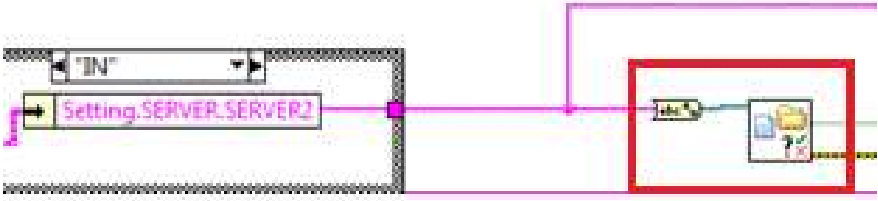


Figura 28. Verificación de direccionamiento.



Mientras tanto, el programa verifica la existencia de la dirección asignada en el archivo setting (ver Figura 28); se ha enmarcado en rojo para que pueda ser visualizado; el primero convierte la cadena de caracteres a path, mientras que el siguiente verifica la existencia de la dirección especificada por el path (éste devuelve un verdadero o falso dependiendo sea el caso de que si existe o no la dirección).

Luego entrará en un case que dependiendo de que, si el resultado anterior fue un falso, desplegará un mensaje en pantalla; si fue verdadero, entonces se procederá a hacer las copias de los archivos (ver Figura 29). Si el valor es verdadero o en otras palabras existe la dirección dada, entonces procede a convertir todos los nombres de la máquina en un path agregándoles la dirección del servidor (ver Figura 30).

En el subVI con el nombre de machine list convierte el arreglo de cadena de caracteres en un arreglo de cadena de path agregándole al inicio de la cadena la dirección del servidor como se muestra en la Figura 31. Puede verse como es uno por uno agregado la sección del direccionamiento de los servidores de cada planta. A continuación, se presenta el segundo subVI llamado copy files; en este se toma primeramente el arreglo generado en el sub VI anterior y es verificado la existencia de las carpetas en donde se realizará la copia de los archivos; en caso de que no exista, es generada en la dirección c:\Q9; esto se hace mediante el proceso de la Figura 32.

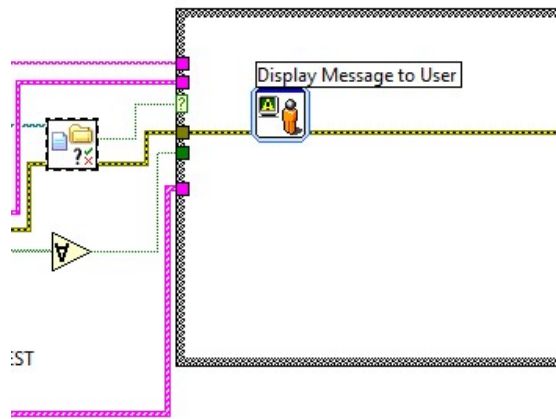


Figura 29. Caso falso de direccionamiento.

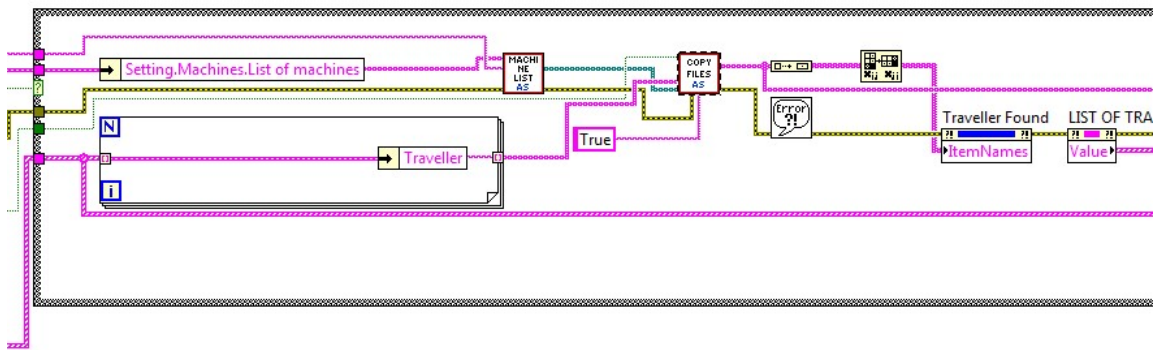


Figura 30. Caso verdadero de direccionamiento.

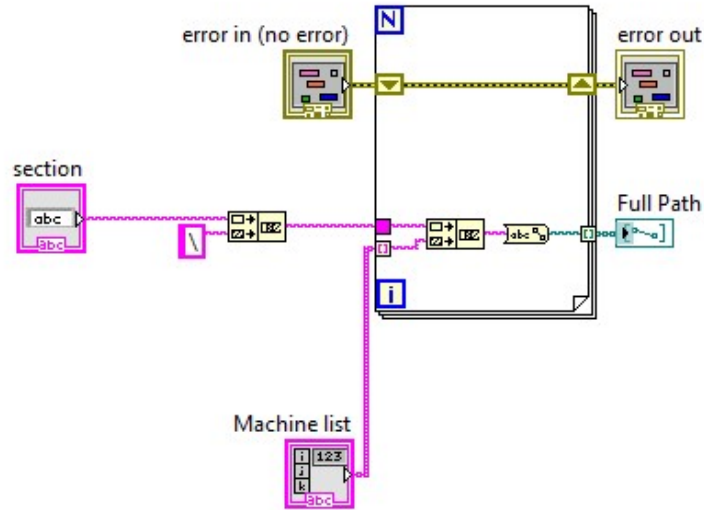


Figura 31. Convertir cadena de caracteres en Path.

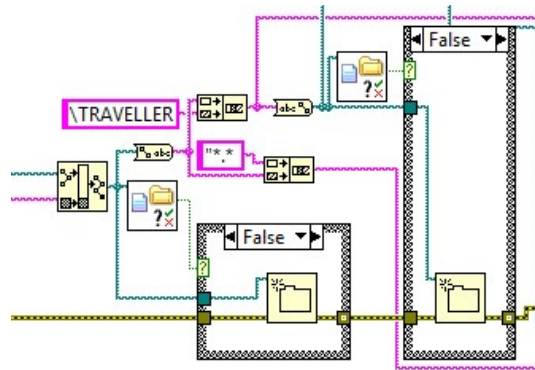


Figura 32. Generar carpeta.

Ahora porque hacer este paso, primeramente, ya que al iniciar el programa no se sabe si es la primera vez que corre en la máquina y por ende tenga ya generados las carpetas en donde se resguardará la copia de los datos generados por producción. Si este fuera el caso entonces las carpetas se generarían antes de que se copien los archivos de datos; en caso contrario si ya existen entonces no se ejecuta ninguna acción y solamente se pasa a que sean copiados los archivos.

En este punto ocurren dos casos importantes; el primero sería que los datos ya fueron revisados previamente y necesitan realizar la consulta una segunda vez; en este caso se le tiene que indicar al usuario que existen datos previos guardados y si desea sobre escribir el archivo; en este caso se le coloca un dialogo con dos botones, en el que el usuario puede decidir si desea sobre escribirlo o no. En la Figura 33 se muestra el Vi que sirve para ello.

La información desplegada en la pantalla es una cadena de caracteres concatenado, esto es decir que es agregado una tras otra, pero ésta se encuentra dividida en tres partes: la primera indica al usuario “El traveller “, la segunda parte anexa el nombre del archivo, y como última parte se muestra existe ¿desea sobre escribirlo?”; esto es con la intención que el usuario determine la sobre escritura de un archivo previo consultado.

Obsérvese que se encuentra ubicado en una estructura caso, por lo que esto ocurre únicamente si el archivo ya existe previamente; sino no ejecuta esto y el software continúa normalmente.

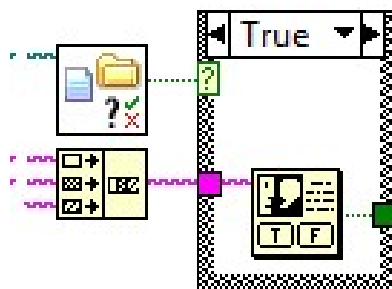


Figura 33. Selección de sobre escritura.

### 3.10.3. Descarga de archivos

Para ello se usará el ejecutor de sistemas (sistema exec) mostrado en la Figura 34, el cuál es básicamente un ejecutor de comandos en MS-DOS; para ello lo primero que se tiene que colocar es “cmd /c “ y por medio de un arreglo concatenado colocar la función que se necesita; en este caso se utilizará la función copy del sistema operativo MS-DOS, después se especifica la ubicación del archivo para ser copiado, y luego de destino hacia donde se copiará.

Una vez realizado esto el subVI se prosigue a una etapa de indicación de detección de los archivos; esto es para indicarle al usuario que fue encontrado o no un archivo.

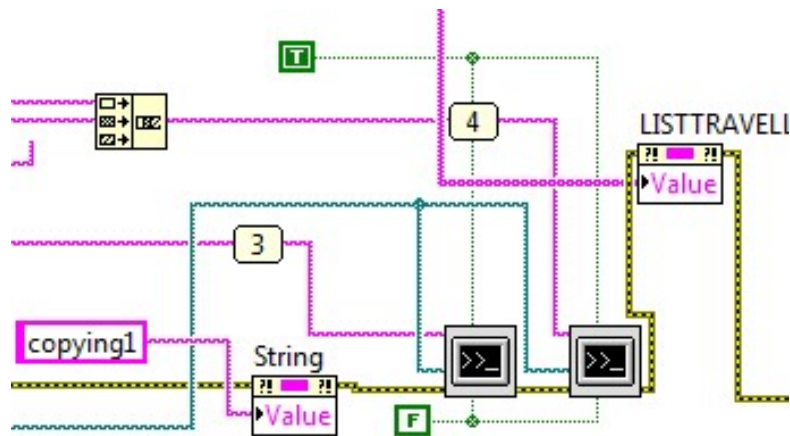


Figura 34. Copia de archivos.

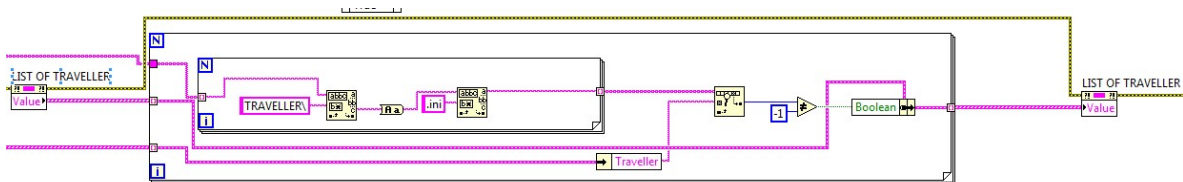


Figura 35. Indicador de detección de archivos.

Esta parte del programa realiza una búsqueda de uno por uno del arreglo de archivos copiados y los compara con la lista de archivos que se necesitan ser encontrados; si el nombre aparece es marcado con un valor booleano verdadero; en caso contrario si no es encontrado será marcado con un falso. En la Figura 35 se muestra como es realizada dicha acción.

### 3.10.4. Búsqueda de un serial

Para realizar la búsqueda de un serial de una pieza es necesario primeramente correr la descarga de los archivos y que se encuentre desplegado en pantalla. Es escrito manualmente el número del serial que se desea buscar, luego es presionado el botón de “buscar una pieza” (fin done piece) especificado en inglés.

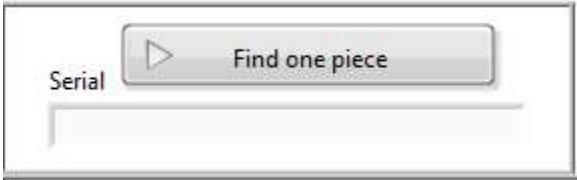


Figura 36. Búsqueda de un serial.

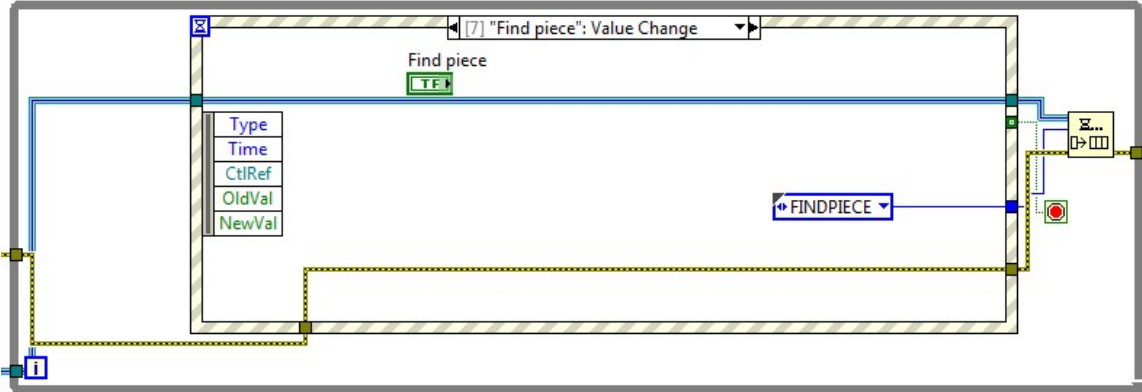


Figura 37. Esquemático de búsqueda serial arreglo de colas.

Este botón manda el valor FINDPIECE al arreglo de colas de estados mostrado en la Figura 37, haciendo que el ciclo principal se corra el subprograma, búsqueda de piezas.

En la Figura 38 puede verse que se inicia mandando el código de las piezas a ser buscada y el número serial, dentro de esta subrutina, se encuentra un buscador de que es marcado en un cuadro rojo; este buscador localiza una cadena de caracteres y devuelve la ubicación de la casilla en un numero entero. Una vez que termina esta subrutina es colocado un carácter en blanco para borrar del control serial el valor buscado.

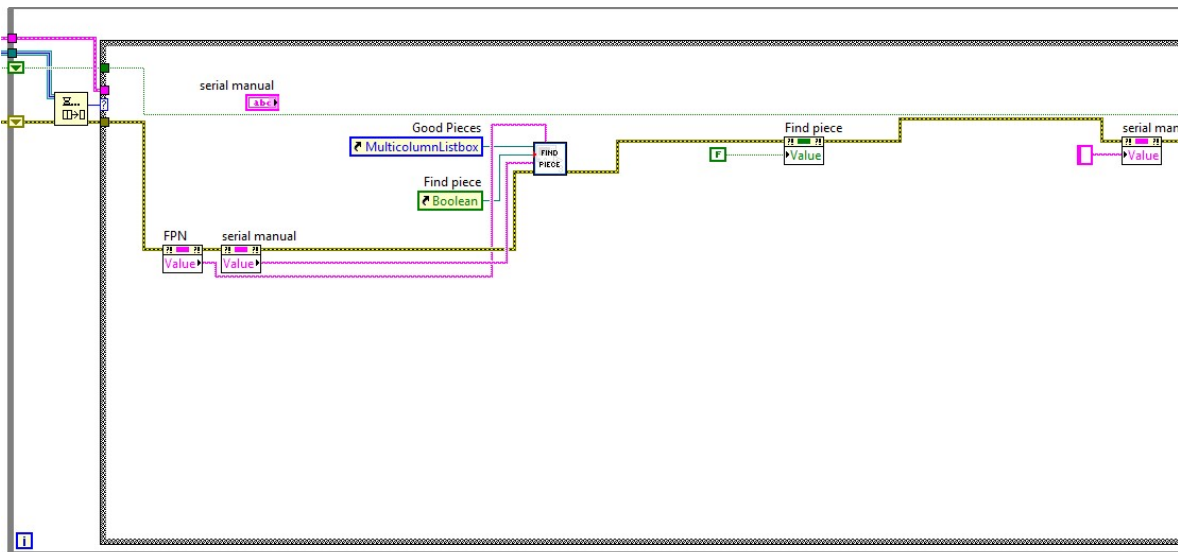


Figura 38. Ciclo principal buscar una pieza.

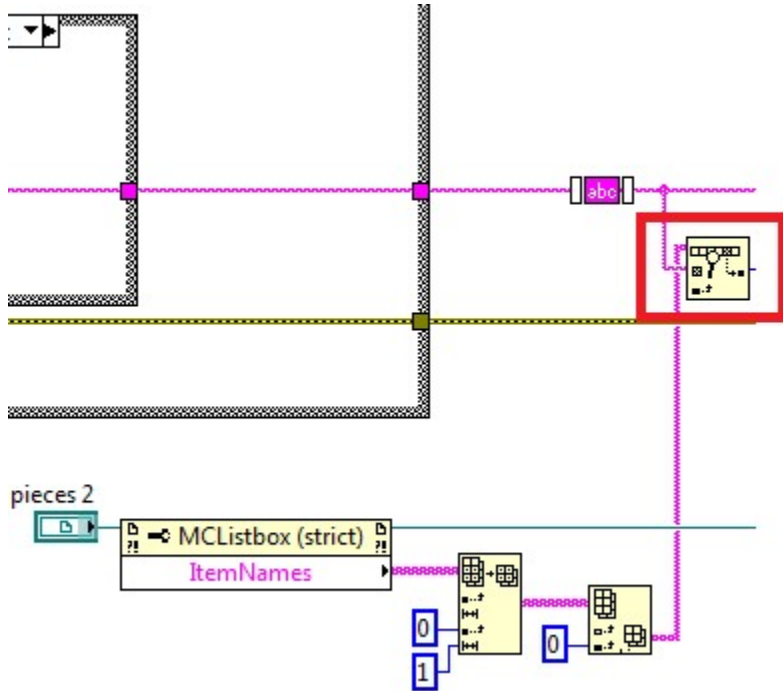


Figura 39. Búsqueda de un serial en un arreglo.



## 4. Resultados

### 4.1. Introducción

En este capítulo se presentarán los resultados obtenidos mediante estudios de tiempo de los dos procesos (manual y mediante herramienta computacional) y la comparativa de estos, permitiendo con ello vislumbrar de una mejor forma la diferencia marcada que existe entre ambos procesos. Para ello se presentarán los estudios de tiempo realizados en campo mediante el formato presentado en la Figura 9 con el propósito de ilustrar el método utilizado para ello.

### 4.2. Pantalla de usuario

The screenshot displays the 'Q9 Report Generator ver 2.0' interface. It features a control panel on the left with input fields for 'FPN', 'DATE CODE', 'TRAVELLER', 'TEST TYPE', 'INSP', and 'PLANT' (with radio buttons for MX, CN, IN). Below these are 'Add', 'Delete', and 'Update files' buttons. The middle-left section includes 'Order' (FILTERING/MANUAL), 'Sampling', and 'Sampling Size' options, along with 'Print report', 'Print SPC', and 'Display report' buttons. The bottom-left section has a 'Serial' field, 'Find one piece' button, and 'NG Print' button. The right side contains two tables: 'Good Pieces' and 'Bad Pieces'. The bottom-right section has 'Reset' and 'End program' buttons, each with an 'OK' button. A status bar at the bottom left shows 'Lot Qty', 'Pieces Tested', 'Good pieces', 'Bad pieces', '% good pieces', and '% bad pieces', all currently at 0.

Figura 40. Pantalla final de usuario.

En la Figura 40 puede verse la pantalla final de usuario, el cual es colocado uno por uno los números de hoja viajera, contiene unos indicadores que permiten la visualización más clara de los resultados y saber si fue o no encontrado una hoja viajera mediante el indicador en color violeta junto a cada número de hoja viajera.

Si los datos de la hoja viajera se encuentran ya terminada y subida en el sistema, el indicador se prenderá, pero en caso contrario si la hoja viajera no se encuentra terminada o archivos no se encuentran en el sistema; este indicador permanecerá apagado, permitiendo con ello saber cuál hoja viajera puede ser inspeccionada y verificar lo ocurrido con la no encontrada.

En la Figura 41 se ve como la hoja viajera 1834\_29841\_01 no se encuentra presente, por lo que el indicador se mantiene apagado. Además, tiene un contador de piezas permitiendo con esto que el usuario pueda visualizar de manera rápida la cantidad de piezas.

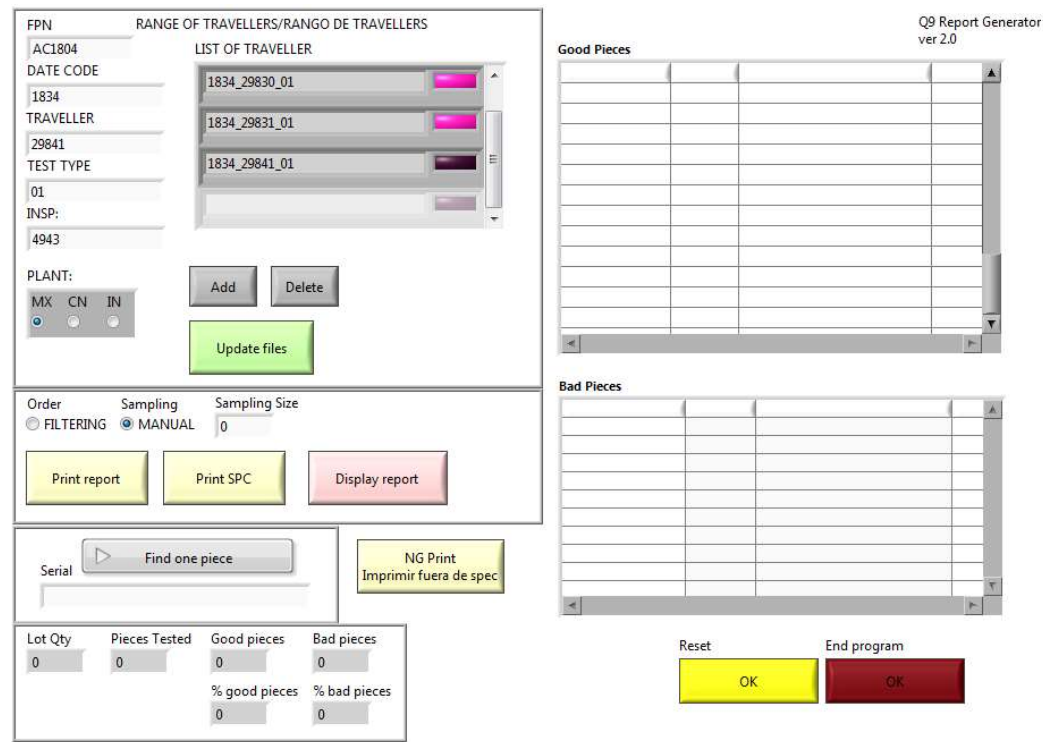


Figura 41. Ejemplo de búsqueda de archivo.

### 4.3. Estudio de tiempos

Los estudios de tiempo son una herramienta utilizada en la ingeniería industrial que consiste en la toma de tiempo de cierta actividad repetitiva de un operador; el proceso es simple únicamente es necesario un cronómetro, papel y lápiz.

El procedimiento para la implementación de la herramienta es el siguiente: colocar el operador de tal forma en cómo se realizaría la operación en la línea de producción o en control de calidad, luego seleccionar una cantidad significativa de piezas que en nuestro caso usaremos 11, pero solamente se tomara el dato de 10 piezas; el motivo es simple, como el estudio de tiempo debe realizarse midiendo el flujo continuo de la operación realizada, entonces la primera pieza no representa ser parte del flujo, ya que esta primera pieza es el inicio del flujo continuo, sino sería hasta la segunda pieza que se iniciaría el cronómetro [18] [19] [20].

Luego se pasan las piezas restantes por el proceso a estudiar y el cronómetro es detenido hasta que las piezas terminan de pasar en la última actividad seleccionada. Durante todo este tiempo es anotado el tiempo que tarda cada pieza en pasar por los pasos del proceso, luego es plasmado en una hoja para hacer los cálculos de tiempo mostrado en la Figura 9.

Para el proceso de inspección se toma una muestra del lote entregado por producción ya inspeccionadas al 100%, dando con esto lugar a que se tengan todos los registros de las piezas a ser inspeccionadas, aunque se realizó el proceso de forma manual y usando la herramienta computacional para hacer la comparativa.

Para ello se especifica el tamaño de la muestra de acuerdo con la norma ANSI\_ASQC-Z1.4 y al grado de severidad de la misma, luego se colocan las piezas sobre la mesa y se realizan los siguientes pasos:

1. Tomar la pieza.
2. Conectar la pieza al equipo de prueba.
3. Realizar la prueba la pieza.
4. Desconectar la pieza.
5. Colocar de nuevo la pieza en la mesa.

Con esto se tiene definido el proceso el cual se usará para la métrica.

El cálculo de la relación entre ambos procesos es realizado de acuerdo a la siguiente ecuación.

$$Rp = \frac{T_{pm} - T_{np}}{T_{np}}$$

**Ecuación 1. Cálculo de relación de tiempos.**

Donde:

Rp es la relación entre procesos.

Tpm es el tiempo promedio manual.

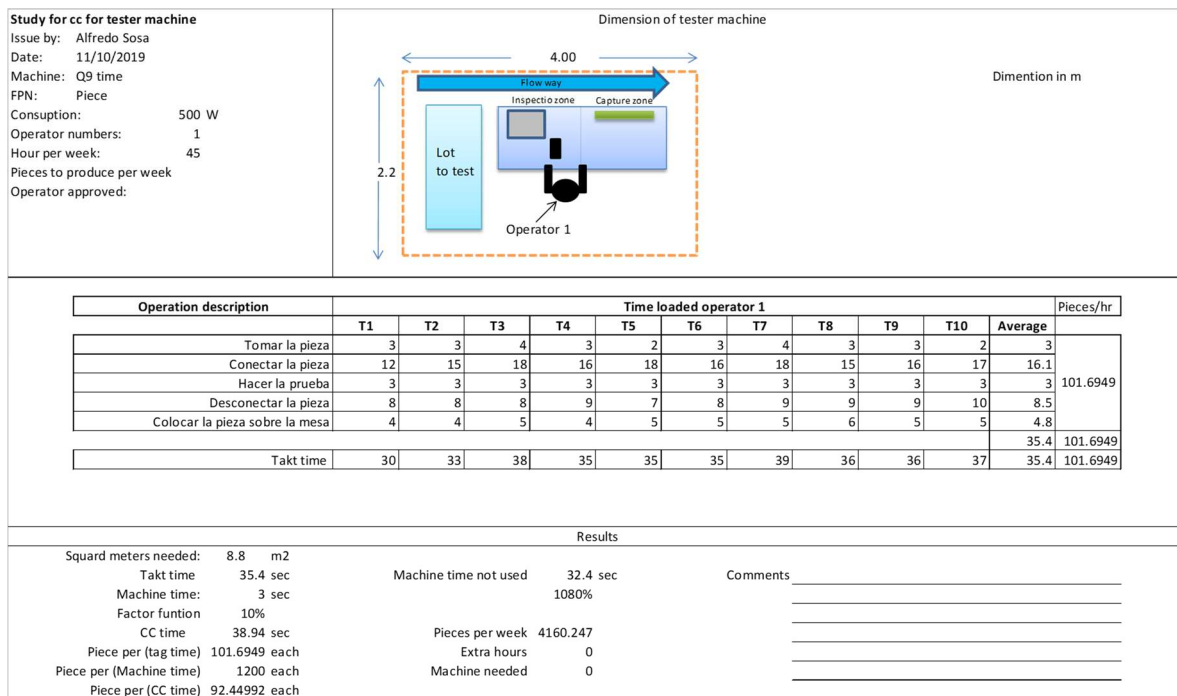
Tnp es el tiempo de nuevo proceso.

También se usará como métrica el takt time [17] que se define como el tiempo que transcurre entre el inicio de una unidad en producción y el inicio de la siguiente unidad en producción. Esto servirá principalmente hacer la comparativa de los procesos anterior y el nuevo proceso con la herramienta computacional de manera gráfica, ya que al tener ambos procesos diferentes pasos no es posible hacer la comparación directa entre ellos; los datos serán tomados de los estudios de tiempo en la fila denominada como takt time.

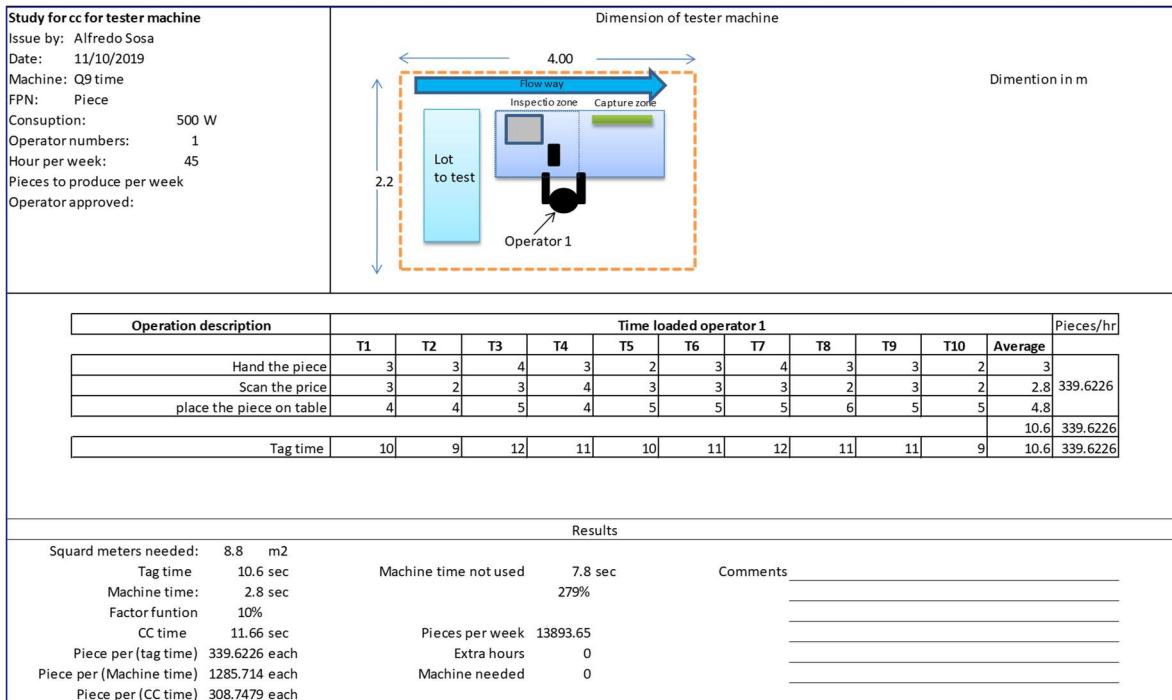
Se mostrarán ambos procesos en la misma grafica con la intención de visualizar de una manera clara la diferencia entre los dos procesos, permitiendo observar la forma de afectación y como se diferencia entre ambos procesos; esto es importante ya que mostrará como un proceso es estandarizado al visualizarse una gráfica en donde la variación de la pendiente es casi nula, sin importar la modificación y cantidad en las operaciones individuales del proceso; esto se plasma en el nuevo proceso usando la herramienta computacional.

En la Figura 42 y Figura 43 se presentan los tiempos tomados en el proceso manual y con la herramienta respectivamente, usando al mismo operador pero usando los dos procesos, permitiendo con ello evaluar de una mejor forma.

Mediante la Ecuación 1 se puede hacer el cálculo y ver como el tiempo del proceso manual es 2.33 veces más que el proceso con la herramienta. Significando un incremento en el número de piezas inspeccionadas que va de 101.69 piezas por hora a 339.62 piezas por hora, teniendo una diferencia de 237.93 de piezas.

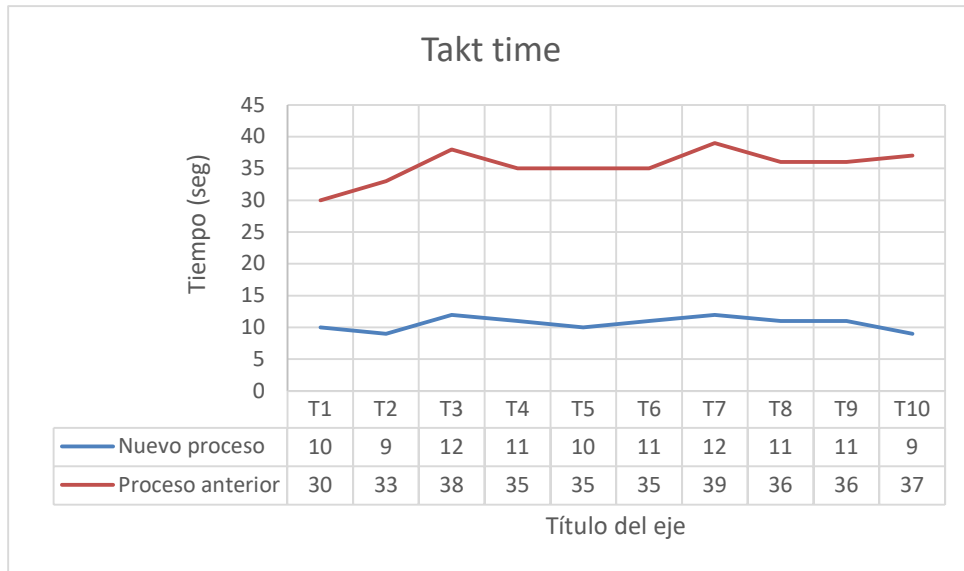


**Figura 42. Estudio de tiempo inspección de piezas en forma manual.**



**Figura 43. Estudio de tiempo con herramienta.**

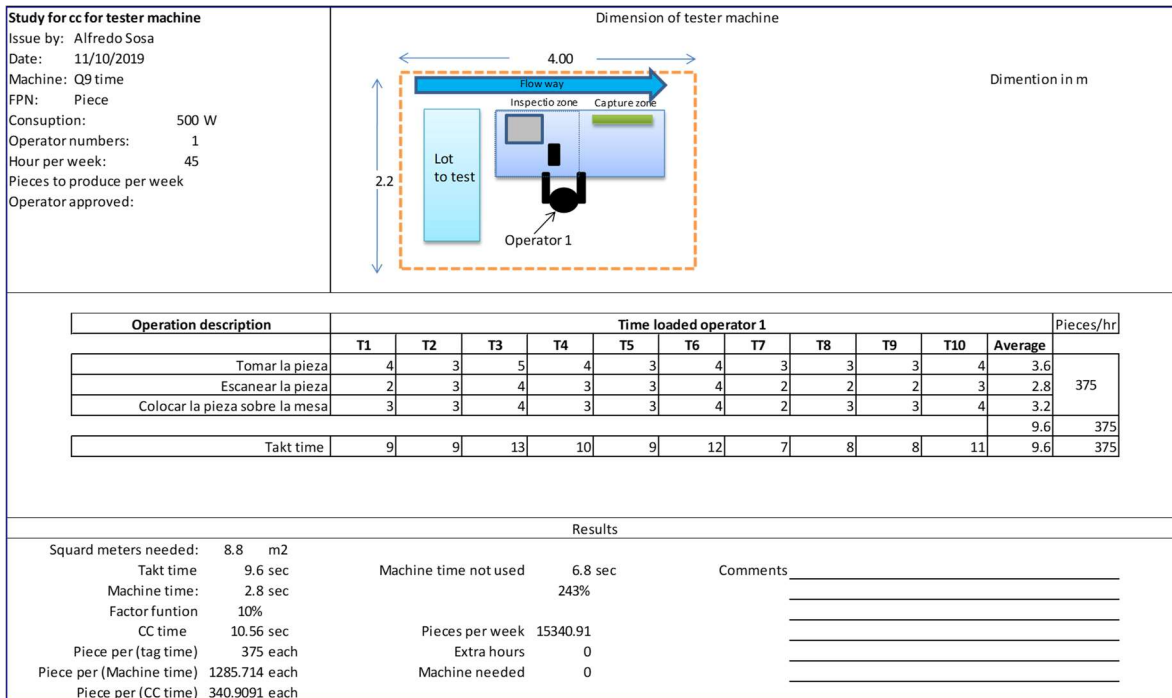
En piezas a la semana tomando un tiempo de 48 horas, sería un incremento 11,420.64 piezas. Hay que tomar en cuenta que esta cantidad es para piezas de un modelo que solo se realiza 2 conexiones, por lo que el tiempo de conexión para piezas con más terminales se incrementará en el proceso manual dando como resultado una diferencia marcada mayor en ambos procesos. Con lo mencionado anteriormente es considerado el tiempo de conexión de dos terminales que es el vértice más cercano entre ambos procesos.



**Figura 44. Grafica de tiempos de inspección de piezas.**

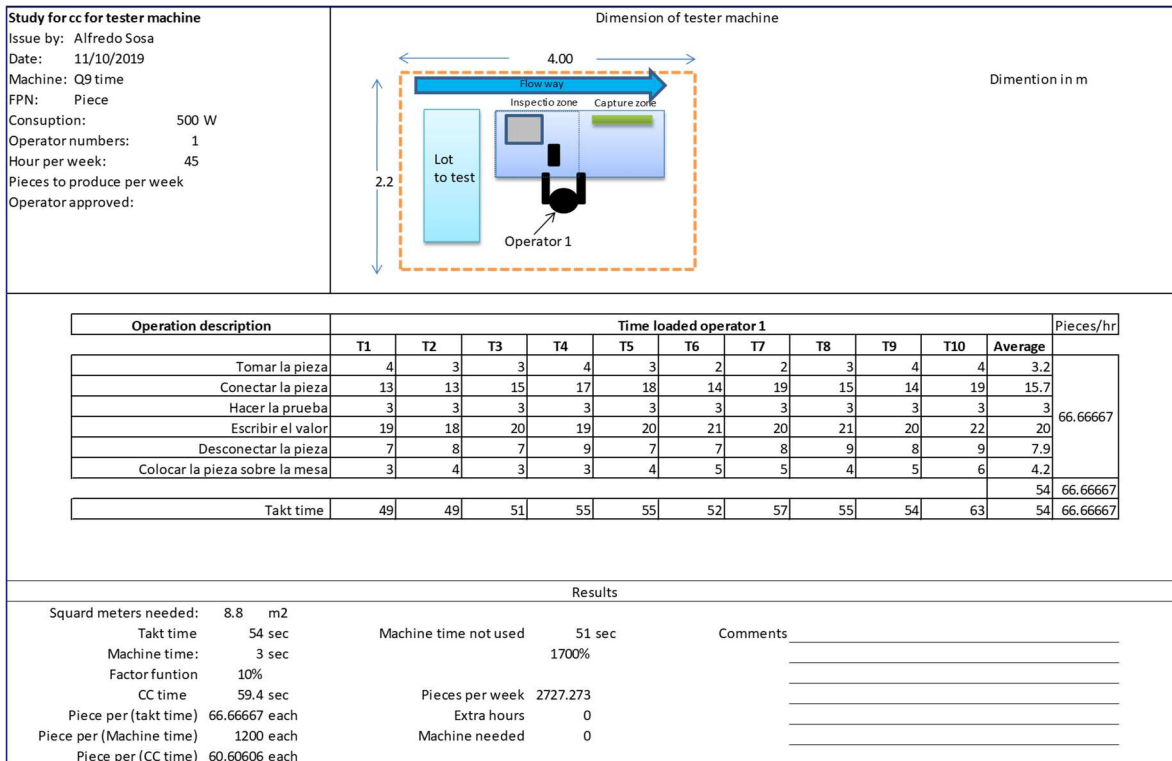
En la Figura 44 se muestran ambos procesos en donde se puede ver la variabilidad que presenta el proceso anterior en comparación con el proceso nuevo; es importante recordar que en el proceso anterior se emplearon piezas en donde solamente se tiene dos terminales para conexión, teniendo con ello el mejor escenario para el proceso anterior; en caso de que se tuvieran piezas con más de dos terminales, los tiempos serían mayores y más variantes; ahora se observará el proceso nuevo con la herramienta computacional; al no necesitar conectar ninguna terminal, este proceso será más estable y estándar por lo que las variaciones en el tiempo serán mínimas además que la afectación por el número de terminales de la pieza no tendrá ningún efecto en el proceso.





**Figura 45. Estudio de tiempo generación de SPC con herramienta.**

En el proceso de generación de datos de SPC se puede ver en las Figura 45 y Figura 46 que existe un disminución igual del tiempo; en el proceso manual se tiene 54 segundos de generación del documento, mientras que en el proceso con la herramienta el tiempo se reduce hasta 9.6 segundos, representando un 4.62 veces en la reducción del tiempo de generación del documento.



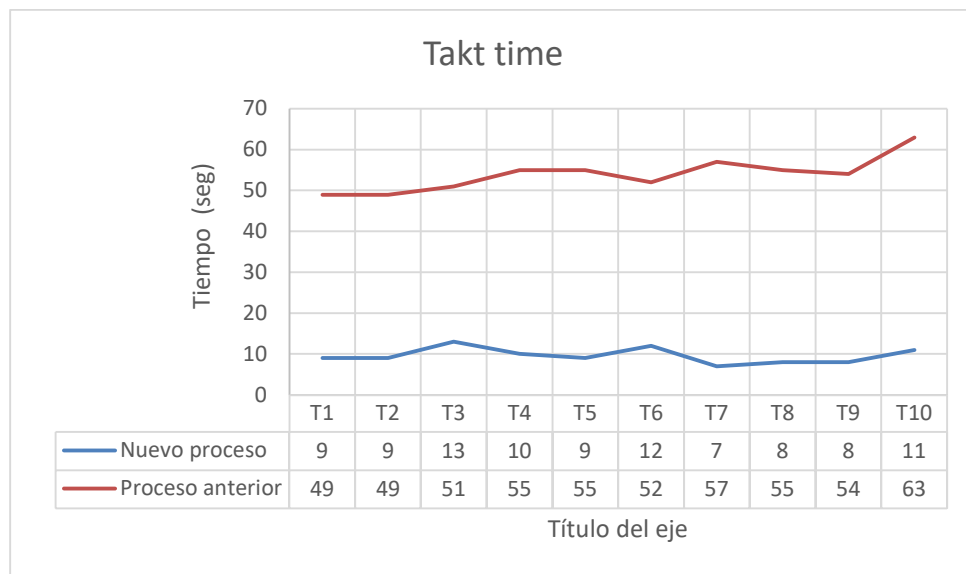
**Figura 46. Estudio de tiempo de generación de SPC manual**

Ahora bien, esto es relativo debido a que los lotes de las piezas varían dependiendo de la norma ANSI ASQC Z1.4, por lo que estos tiempos solo determinan una referencia comparativa para lotes de igual tamaño. También un factor importante que incrementa el tiempo manual es la cantidad de valores que tienen que ser anotados; en este estudio se consideró solamente que es anotado dos valores, incrementándose drásticamente si en el proceso manual se tiene más de un valor a anotar. Mientras que en proceso con la herramienta computacional no sufriría modificaciones.

También se puede estar seguro de que el 100% de los datos colocados en el documento no presentan variaciones o errores por una mala captura de los datos,

ya que los datos son sacados de los archivos generados por los equipos de pruebas y en ningún momento son manipulados por los usuarios.

En la Figura 47 es mostrada la gráfica de tiempo para la generación de formato SPC; en este caso es observable que la diferencia de tiempos es alta entre ambos procesos; en este caso el proceso anterior tiene una tendencia hacia el incremento a esto es conocido como fatiga de proceso, debido a que el usuario al estar teniendo procesos repetitivos de conexión y desconexión, realización de prueba, toma de datos y escritura de mismos tienden a distraerse o a reducir el ritmo de trabajo, mientras que en el nuevo proceso, la tendencia se mantiene debido a que mucho del proceso la maquina realiza todo el trabajo y el usuario solamente tiene pocas interacciones.



**Figura 47. Grafica de tiempo de generación de formato SPC.**

#### 4.4. Piezas malas

En la Figura 48 es mostrada la ubicación de la tabla de piezas malas; estas sirven para que control de calidad pueda exigirle a producción la entrega de estas piezas o el serial correspondiente a estas piezas. En esta lista se desplegarán las piezas malas, así como los valores que tuvo al momento de las pruebas.

Evitando con ello que las piezas malas checadas en los equipos de producción sean empacadas y enviadas al cliente. Con esto se tendrá la certeza que del 100% de las piezas que han sido probadas en producción, saldrán de la planta sin tener defectos eléctricos.

FPN RANGE OF TRAVELLERS/RANGO DE TRAVELLERS

DATE CODE

TRAVELLER

TEST TYPE

INSP:

PLANT:  
MX CN IN

Add Delete

Update files

Order Filtering Sampling Sampling Size  
FILTERING MANUAL 0

Print report Print SPC Display report

Serial Find one piece

NG Print  
Imprimir fuera de spec

Lot Qty	Pieces Tested	Good pieces	Bad pieces
0	0	0	0
		% good pieces	% bad pieces
		0	0

Q9 Report Genera ver 2.0

Good Pieces

Bad Pieces

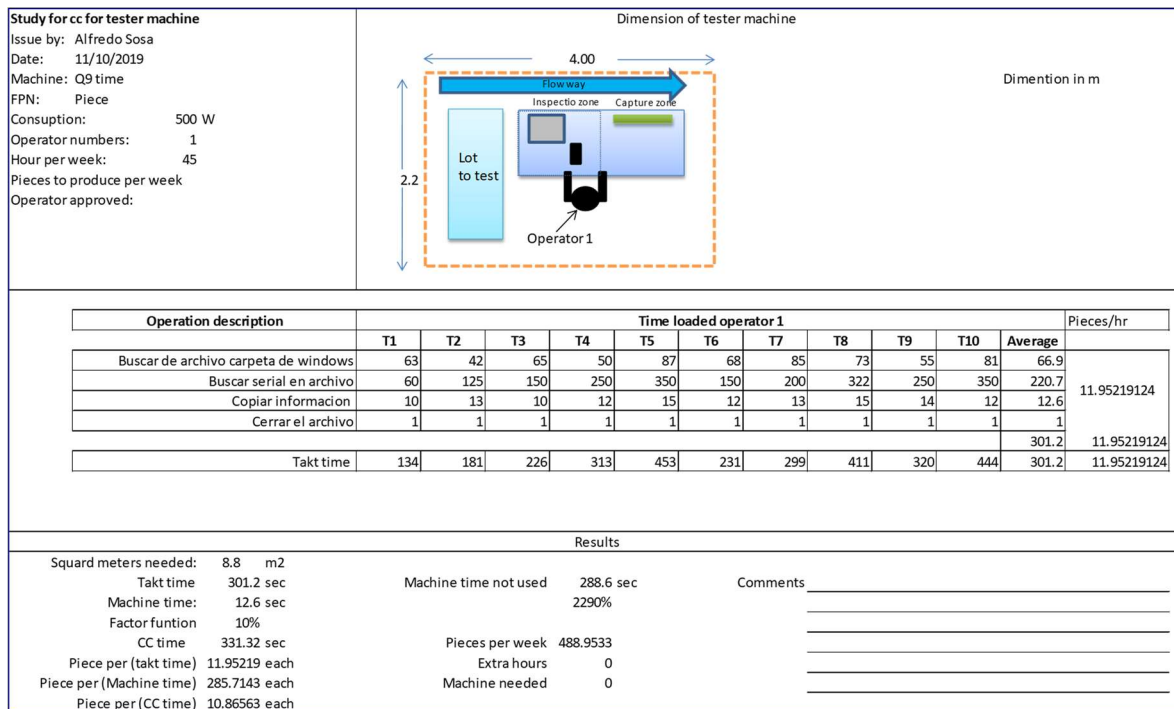
Reset OK End program OK

Figura 48. Enlistado de piezas malas.

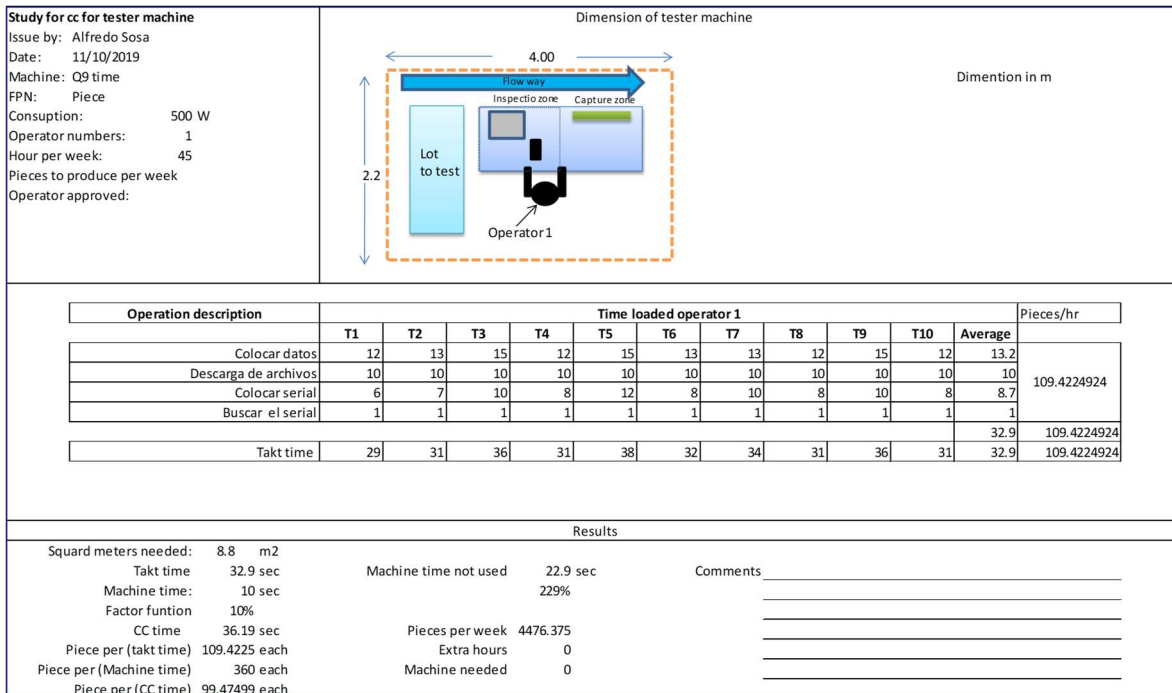
#### 4.5. Búsqueda de una pieza específica.

La búsqueda de la información en forma manual tiene una característica que no son constantes los tiempos de ya que la búsqueda depende directamente de la ubicación del archivo y del tamaño del mismo.

En la Figura 49 puede observarse que el tiempo de búsqueda del explorador de Windows es variable, mientras que la búsqueda de información de una pieza en específico es menos variable pero más elevado. El tiempo promedio por pieza es de 301 segundos, mientras que el valor mínimo es de 134 segundos y el máximo es de 444 segundos.



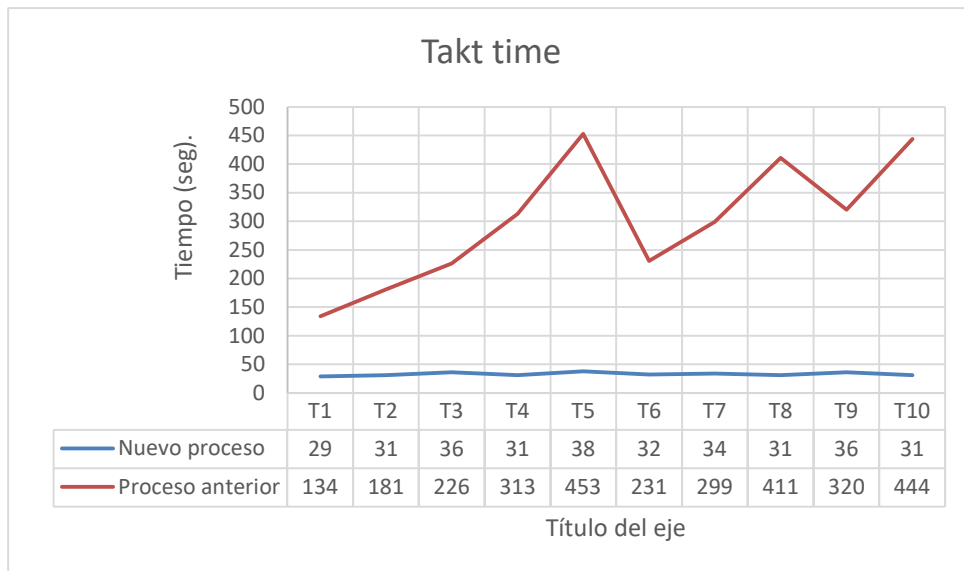
**Figura 49. Tiempo de búsqueda manual.**



**Figura 50. Tiempo de búsqueda con programa.**

En la Figura 50 puede verse el tiempo que se necesita para hacer la búsqueda de la información de una pieza; el tiempo promedio es de 32.9 segundos, mientras que el tiempo máximo es de 38 segundos y el mínimo es de 29 segundos.

Si se consideran los tiempos promedio para hacer el cálculo de mejora de tiempo usando la Ecuación 1, se puede notar que existe una mejora de 8.15; esto significa que existe una diferencia de 97.47 piezas por hora más con herramienta computacional, que haciéndolo de forma manual.



**Figura 51. Grafica de tiempos para búsqueda de una pieza.**

La Figura 51 muestra la gráfica de tiempos de ambos procesos (anterior y nuevo); cómo se puede apreciar, existe una gran variabilidad en el proceso anterior; esto es a causa de la naturaleza del proceso, la búsqueda de información depende la velocidad de búsqueda del archivo por parte del sistema operativo usado, para que después el usuario pueda ingresar al archivo y buscar la información de la pieza, por lo que hacer eso por cada una de las piezas representa un trabajo complicado y meticuloso, siendo el usuario responsable directo de la correcta selección de la información y con ello propenso a errores.

En la gráfica del nuevo proceso se puede notar como presenta una tendencia más suave y estándar, sin presentar picos, esto debido a que la búsqueda de la información se realiza por medio de la herramienta computacional, el usuario no tiene que estar solicitando al sistema operativo la búsqueda del archivo y luego ingresar a éste; en el nuevo proceso estos pasos lo realiza la herramienta

computacional reduciendo el tiempo de captura de información, reduciendo la variabilidad de los tiempos drásticamente, y permitiendo como se mencionó, establecer un estándar en el tiempo con variaciones mínimas en tiempo de proceso.



## 5. Conclusiones

Es innegable el hecho de que la aparición de la industria 4.0 en los procesos de manufactura permite la realización de mejoras en los procesos; en el caso específico de este proyecto presentado se utilizaron dos pilares que es la estructura de datos y el análisis de éstos.

Mediante el estudio de diferentes lenguajes de programación se pudo ver que aunque Labview no se encuentra colocado entre los primeros lenguajes utilizados en el mercado actualmente, presenta herramientas poderosas las cuales permiten la realización de proyectos con un grado alto de complejidad. Por el perfil que presento debido a mi especialidad de electrónica, este programa en lenguaje G similar a los simuladores de dispositivos eléctricos representa una de las mejores opciones, ya que muchas veces es de forma intuitiva la implementación de los instrumentos virtuales.

Fue muy importante definir primeramente las variables de entrada y salida, ya que, al tener una amplia gama de valores eléctricos, número de partes, seriales, códigos, nombre de archivos, hojas viajeras, etc., reduciendo en gran manera el proceso y verlo primeramente como una caja negra en la cual se ingresan variables y se obtiene un resultado, o visto de otra manera, un camino a seguir.

Considerando el proceso de la implementación del proyecto, se tiene que tomar en un alto grado el factor humano y muchas veces la renuencia del personal a utilizar los nuevos esquemas, se hace mención de este punto debido a que, aunque no es un factor que depende de la herramienta computacional ni de su desempeño, el

factor humano repercute directamente en el tiempo y es variable dependiendo del individuo e incluso del estado de ánimo del mismo. Por lo que muchas veces pueden existir barreras que una vez eliminadas y comprendida la mejora que tienen disminuyendo la carga de trabajo, este nuevo proceso es aceptado y muchas veces elogiado por los mismos usuarios.

Con referencia a los tiempos y la disminución de estos hacen reflejo en como el uso de herramientas computacionales a la medida de las necesidades permiten a las empresas una mejora directa en sus procesos e impactan significativamente en sus insumos de capital.

Con relación a la mano de obra, éste permite que con la misma cantidad de personal poder realizar un trabajo más rápido, eficiente y disminuyendo el error, que muchas veces repercute en el peor lugar de la cadena de suministros, que es el cliente final.

Como último punto de beneficio del proyecto es el renombre de la empresa, ya que, al disminuir errores con el cliente, éste permite tener mejores proyectos futuros y con ello mejores ganancias, que muchas veces no son considerados en la realización de estos proyectos.

De manera personal y profesional este proyecto me ha permitido poder desempeñarme en un ámbito emergente, ya que me ha permitido tener la experiencia de colaborar con múltiples departamentos y personal con amplia experiencia en sus diferentes áreas, desde operadores hasta niveles gerenciales y ver de cerca cada caso.

## 6 Referencias

- [1] A. S. F. QUALITY, SAMPLING PROCEDURES AND TABLES FOR INSPECTION, Milwaukee: American Society for Quality, 2003.
- [2] G. SCHUETZ, «Modern Machine Shop Mexico,» 02 Octubre 2017. [En línea]. Available: <https://www.mms-mexico.com/columnas/recolecci%C3%B3n-de-datos-para-la-industria-40>. [Último acceso: 28 Septiembre 2018].
- [3] I. S. B. T. Chan, «Software de análisis estadístico para la industria manufacturera,» Mérida, Yucatán, 2019.
- [4] TIOBE, "TIOBE Index for April 2019," TIOBE, [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Accessed 28 Abril 2019].
- [5] N. I. Corporation, «Los Beneficios de la Programación Gráfica en LabVIEW,» National Instruments Corporation, 19 Marzo 2019. [En línea]. Available: <http://www.ni.com/es-mx/innovations/white-papers/13/benefits-of-programming-graphically-in-ni-labview.html>. [Último acceso: 8 Mayo 2019].
- [6] B. Rasin, «IntegratingPythonWithOtherLanguages,» Phyton TM, 28 Marzo 2019. [En línea]. Available:

<https://wiki.python.org/moin/IntegratingPythonWithOtherLanguages>. [Último acceso: 8 Mayo 2019].

[7] Juan\_Vidal, «Big data: Gestión de datos no estructurados,» 18 Octubre 2018. [En línea]. Available: <http://www.dataprix.com/blog-it/big-data/big-data-gestion-datos-no-estructurados>. [Último acceso: 18 Octubre 2018].

[8] P. G. Bejerano, «Hadoop, la plataforma open source que lidera big data,» 18 Octubre 2018. [En línea]. Available: <https://blogthinkbig.com/hadoop-open-source-big-data>. [Último acceso: 18 Octubre 2018].

[9] C. González, «Herramientas para big data más utilizadas,» 20 Octubre 2018. [En línea]. Available: <https://www.emagister.com/blog/herramientas-big-data-mas-utilizadas/>. [Último acceso: 20 Octubre 2018].

[10] A. N. STANDARD, «SAMPLING PROCEDURES AND TABLES FOR INSPECTION BY ATTRIBUTE,» de *SAMPLING PROCEDURES AND TABLES FOR INSPECTION BY ATTRIBUTE*, Milwaukee, Wisconsin, American Society for Quality, 2003, p. 96.

[11] N. Instruments, «Enum Constant,» [En línea]. Available: [https://zone.ni.com/reference/en-XX/help/371361R-01/glang/enum\\_constant/](https://zone.ni.com/reference/en-XX/help/371361R-01/glang/enum_constant/). [Último acceso: 03 Julio 2019].

[12] N. Instruments, «¿Qué es un Queue?,» National Instruments, [En línea]. Available:

<http://digital.ni.com/public.nsf/allkb/6B63645C1217827786257D50007540F2>.

[Último acceso: 05 Agosto 2019].

[13] N. Instruments, «Función de diálogo de dos botones,» [En línea]. Available:

[http://zone.ni.com/reference/en-XX/help/371361R-](http://zone.ni.com/reference/en-XX/help/371361R-01/glang/two_button_dialog_box/)

[01/glang/two\\_button\\_dialog\\_box/](http://zone.ni.com/reference/en-XX/help/371361R-01/glang/two_button_dialog_box/). [Último acceso: 10 08 2019].

[14] N. Instruments, «System Exec VI,» [En línea]. Available:

[https://zone.ni.com/reference/en-XX/help/371361R-01/glang/system\\_exec/](https://zone.ni.com/reference/en-XX/help/371361R-01/glang/system_exec/).

[Último acceso: 10 08 2019].

[15] Honeywell, «Voyager 1400g - Honeywell AIDC,» [En línea]. Available:

[https://www.honeywellaidc.com/es-ES/-/media/es-es/data-sheets/voyager-](https://www.honeywellaidc.com/es-ES/-/media/es-es/data-sheets/voyager-1400g-general-duty-scanner-data-sheet-es-es.pdf)

[1400g-general-duty-scanner-data-sheet-es-es.pdf](https://www.honeywellaidc.com/es-ES/-/media/es-es/data-sheets/voyager-1400g-general-duty-scanner-data-sheet-es-es.pdf). [Último acceso: 12 10

2018].

[16] N. Instruments, «Using Configuration File VIs,» [En línea]. Available:

[https://zone.ni.com/reference/en-XX/help/371361R-](https://zone.ni.com/reference/en-XX/help/371361R-01/lvhowto/using_configuration_filevi/)

[01/lvhowto/using\\_configuration\\_filevi/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvhowto/using_configuration_filevi/). [Último acceso: 11 Junio 2019].

[17] V. Y. P. Dr, «¿Qué es Takt Time? ¿Se puede aplicar en la construcción?,»

Universitat Politècnica de València, 4 Septiembre 2014. [En línea]. Available:

[https://victoryepes.blogs.upv.es/2014/09/04/que-es-takt-time-se-puede-](https://victoryepes.blogs.upv.es/2014/09/04/que-es-takt-time-se-puede-aplicar-en-la-construccion/)

[aplicar-en-la-construccion/](https://victoryepes.blogs.upv.es/2014/09/04/que-es-takt-time-se-puede-aplicar-en-la-construccion/). [Último acceso: 21 Diciembre 2019].

[18] H. G. Pulido, Ciudad total y productividad, Mexico: Mc Graw Hill, 2010.

[19] D. H. Besterfield, Control de calidad, Naucalpan de Juárez, Estado de México:  
Pearson Educación de México, S.A. de C.V., 2009.

[20] H. G. P. y. R. d. I. V. Salazar, Control estadístico de la calidad y seis sigmas,  
Mexico, D.F.: Mc Graw Hill, 2013.