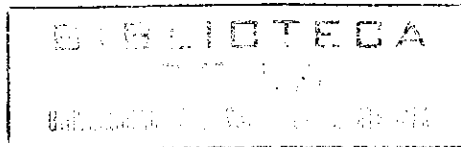


Universidad del Valle de Guatemala  
Facultad de Ciencias y Humanidades

**Automatización de básculas y  
control de tránsito de camiones  
para pesaje múltiple**



Guatemala  
2003



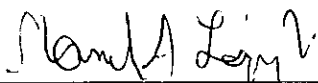
**Automatización de básculas y  
control de tránsito de camiones  
para pesaje múltiple**

Universidad del Valle de Guatemala  
Facultad de Ciencias y Humanidades

**Automatización de básculas y  
control de tránsito de camiones  
para pesaje múltiple**

Trabajo de investigación presentado  
por Jose Galindo Campins  
para optar al grado de Licenciado en Ingeniería Electrónica  
Guatemala  
2003

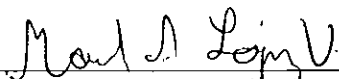
Vo. Bo.



---

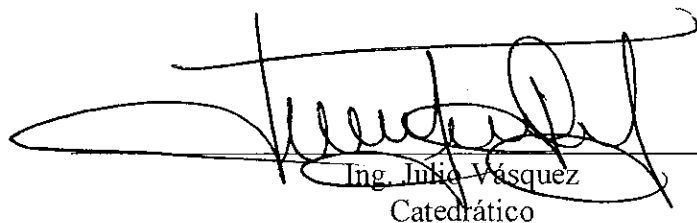
Dr.-Ing. Manuel López Valdez  
Asesor  
Director de Departamento Ing. Electrónica  
Universidad del Valle de Guatemala

Terna examinadora



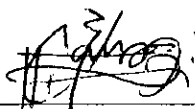
---

Dr.-Ing. Manuel López Valdez  
Director de Departamento Ing. Electrónica  
Universidad del Valle de Guatemala



---

Ing. Julio Vásquez  
Catedrático  
Universidad del Valle de Guatemala



---

Ing. Carlos Esquiet Hernández  
Catedrático  
Universidad del Valle de Guatemala

**Guatemala, 10 de Julio de 2003**

# CONTENIDO

	Página
RESUMEN.....	iv
LISTA DE CUADROS.....	v
LISTA DE GRÁFICOS Y DIAGRAMAS.....	vi
I. INTRODUCCIÓN	
A. CIM – Producción integrada con computadora.....	1
II. OBJETIVOS.....	6
III. DESARROLLO	
A. Método de trabajo.....	7
IV. RESULTADOS OBTENIDOS	
A. Resultados generales y análisis de tiempos.....	20
B. Análisis antes y después de la automatización.....	24
V. CONCLUSIONES Y RECOMENDACIONES	
A. Conclusiones.....	27
B. Recomendaciones.....	28
VI. BIBLIOGRAFÍA .....	29
VII. APENDICE	
A. Código del PIC para conversión de protocolo Wiegand a RS232.....	30
B. Código del programa en Visual Basic.....	38
C. Justificación financiera.....	53

## RESUMEN

El presente trabajo de investigación se enfoca en la utilización de una metodología que integra todas las fases de un proceso el cual da como resultado la automatización completa. En contraste se observa la automatización puntual (como suele realizarse tradicionalmente en las industrias guatemaltecas que implementan procesos automatizados), en la cual se necesita la intervención humana para realizar tareas. Las ventajas más significativas de la implementación de este método es que se reduce el costo y aumenta la productividad, calidad y servicio al cliente.

Se demuestra que es factible automatizar completamente el pesaje de camiones aplicando una combinación de programa y equipo de campo integrados para lograr eliminar errores tipográficos, tiempos de pesaje y sobre todo asegurar un peso exacto.

Se pudo observar que, por medio de utilización de barreras y lectoras de proximidad, combinado con un complejo sistema lógico, se logra controlar el pesaje múltiple de camiones, y a la vez presenta un ahorro a largo plazo.

La aplicación de esta metodología al caso específico de básculas camioneras se describe a continuación y trae beneficios económicos y eficiencia en tiempos del ciclo del proceso tanto así que se pudo observar que los tiempos disminuyeron hasta un 50% a los tiempos anteriores.

## LISTA DE CUADROS

Cuadro	Página
1. Estados de barrera	10
2. Tiempos totales por hora	21
3. Tiempos máximos de pesaje por hora	22
4. Cuenta de tiempos de pesaje por hora	23

# LISTA DE GRÁFICOS Y DIAGRAMAS

Gráficos y Diagramas	Página
1. Diagrama: Estructura de Control	2
2. Diagrama: Planta sin CIM	4
3. Diagrama: Trama de protocolo	7
4. Gráfico: Barrera automática para control de tránsito	9
5. Diagrama: Circuito de control de barrera	10
6. Gráfico: Tarjeta de proximidad	11
7. Diagrama: Componentes de una tarjeta de proximidad	12
8. Gráfico: Entrada a báscula	14
9. Diagrama: Diagrama de flujo del proceso	15
10. Gráfico: Pesaje entre báscula	16
11. Gráfico: Salida de báscula	18
12. Gráfico: Tiempos totales de pesaje por hora	20
13. Gráfico: Máximos tiempos de pesaje por hora	22
14. Cuenta de tiempos de pesaje por hora	23
15. Histograma antes de la automatización	25
16. Histograma después de la automatización	26

# I. INTRODUCCIÓN

## A. CIM – Producción Integrada con Computadora

CIM, por sus siglas en inglés (Computer Integrated Manufacturing), es una metodología que une las tradicionales islas de automatización de producción o proceso, la gente que trabaja con ellos y los sistemas que lo controlan. La implementación de CIM a cualquier nivel requiere la contratación de personal competente y con habilidades de alto nivel, con conocimiento en el área de computación técnica, integraciones de proceso, personal con experiencia en implementación de sistemas electrónicos, control y calidad.

Es una filosofía en la cual las funciones de la organización, desde la definición del producto hasta la disposición del producto final, están diseñadas e integradas para alcanzar metas de manera eficiente y efectiva. La coordinación de funciones se logra a través de la combinación de computadoras, comunicaciones y tecnologías de información.

### 1. Beneficios del CIM

- Menor tiempo para que los productos salgan al mercado
- Mejora de productividad
- Mejor calidad
- Mejor servicio al cliente
- Inventario reducido
- Mayor flexibilidad
- Menor costo total
- Mejor flexibilidad a largo plazo

Una de los conceptos principales es interfazar estándares existentes para permitir la transferencia de datos de maquinaria de proceso a sistemas de información de alto nivel. Una vez implementado, esto ayuda a la integración de calidad y producción, independientemente de la marca del equipo de campo.

**2. Estructura jerárquica de control.** En el CIM, existen múltiples niveles de automatización y control. La pirámide de productividad es concebida como una forma de mostrar la jerarquía de la estructura de control como se muestra en el Diagrama 2.1.

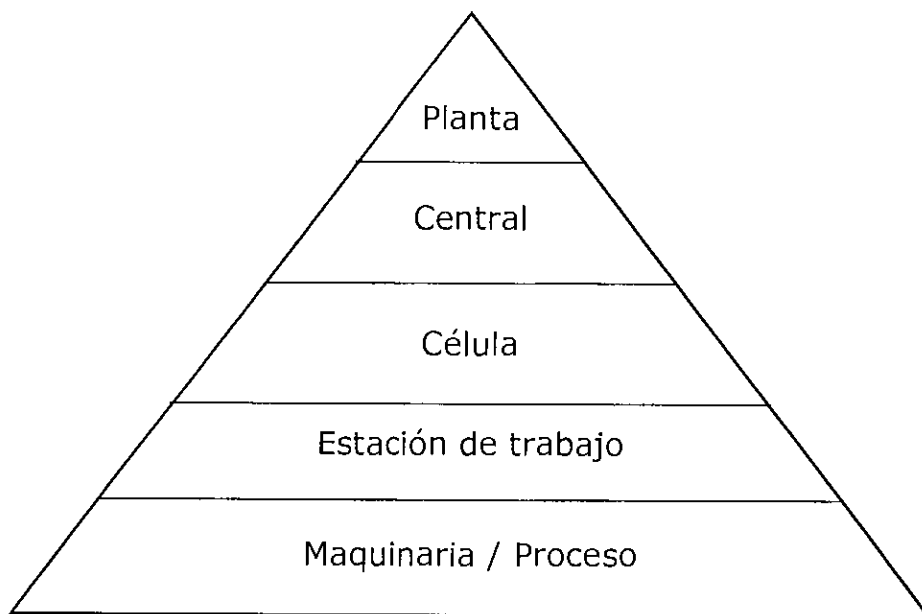


Diagrama 2.1 Estructura de Control

*Esta estructura de control fue introducida por primera vez por CAM-I por sus siglas en inglés (Producción Ayudada por Computadora – Internacional), la cual es una organización no lucrativa en Texas.*

**3. El proceso.** Es la parte más baja de la pirámide. La automatización de un proceso involucra el control de los procesos de producción, tales como maquinaria, prensas, básculas, maquinaria electromecánica, láser y equipo de ensamblaje. En estos procesos, la regulación de las máquinas (que son operados por humanos) son replicados por los mecanismos.

El hardware utilizado en este nivel incluye sensores, dispositivos de poder, dispositivos lógicos e interfaz maquina-humano.

**4. La estación de trabajo.** Hardware y software de computación se utilizan para automatizar estaciones de trabajo que proveen control de bucle cerrado.

En control numérico, esto incluye acceso directo a dibujos CAD (*Computer Aided Design*), plotters en línea, y cualquier control de adaptación que sea necesario. A través de una red de comunicación se une la estación de trabajo con el controlador de célula.

**5. La célula.** El controlador de célula supervisa el trabajo de las estaciones de trabajo. Almacena todos los programas de proceso e información que necesitan las estaciones de trabajo. El controlador de célula verifica el proceso y la configuración de los programas. La célula es la encargada de controlar el flujo de material en el proceso. Recolecta información de sensores y estaciones que son utilizados con fines estadísticos dentro de la empresa, y también de recolección instantánea de errores y soluciones a los mismos. Esta comunicación se logra generalmente a través de redes ethernet.

En muchas aplicaciones, el robot de célula maneja todos los materiales. También hace los cambios de herramientas, extracción de chips, inspección de herramientas, etc. También es capaz de iniciar procesos de emergencia.

**6. La central.** La computadora central generalmente incluye un arreglo de computadoras en redundancia con base de datos relacional. Muchas operaciones de la célula se procesan directamente en la central, disminuyendo así la carga a la misma. En la central se realizan la creación de nuevos programas de robots de célula.

Las decisiones de control que se llevan a cabo incluyen variación de secuencias, balanceo de líneas, programaciones de producción, control de inventario y programaciones de mantenimiento.

El Centro de Producción consiste en una colección de células integradas al sistema de manejo de la planta. La central es un sistema automático de almacenamiento y búsqueda de información.

7. **Planta.** Los ingredientes básicos para una planta automatizada incluye todo el hardware y software en el centro de producción, además de las siguientes funcionalidades:

- Base de datos de ingeniería y producción
- Planeación de capacidades
- Control de procesos de bajo nivel
- Control estadístico de calidad
- Simulación por computadora

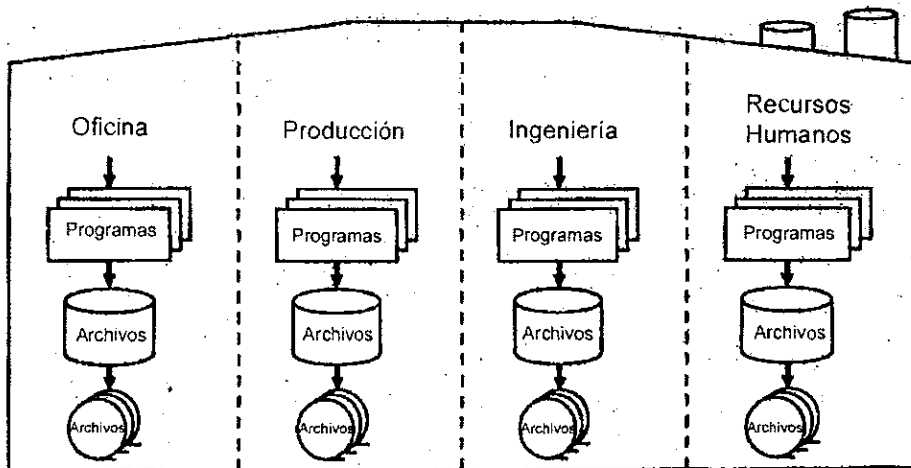


Diagrama 2.2 Planta sin CIM

En una planta sin CIM, como se muestra en el Diagrama 2.2, las diferentes áreas están aisladas y los sistemas de información no están centralizados, por lo que presenta una barrera de información que impide la integración de la planta. Esto es lo que se pretende eliminar cuando se utiliza el CIM en la producción.

Actualmente, en Guatemala se pueden ver muchas situaciones en la industria en la que se realizan automatizaciones puntuales, las cuales necesitan de la intervención humana para realizar

el resto de la automatización. Son pocos los casos en los que se logra un CIM completo, que integre la automatización del equipo y a la vez flujos y comunicaciones.

En el caso específico de las básculas de camiones, se observan en muchos casos programas aislados que solamente solucionan la comunicación con la controladora de la báscula (nivel de proceso), pero un usuario debe ingresar datos y demás para poder asociar ese peso a un camión en específico, es por ello que este trabajo de investigación se enfoca en la automatización completa de la báscula.

- A nivel de proceso se capturan los datos del equipo de campo, lectoras y control de barreras de acceso
- Se tiene una computadora dedicada que funciona como la Estación de Trabajo y la célula, la cual está encargada de la captura, proceso y alimentación de datos al servidor. También está encargada de la lógica del programa y flujo de información.
- A nivel de Central, se maneja una base de datos redundante central en la cual se almacenan los datos recopilados por la báscula para su reutilización y lectura dentro y fuera de la planta.

## II. OBJETIVOS

Los objetivos generales de este trabajo de investigación son los siguientes:

1. Diseñar un sistema para controlar el flujo de camiones dentro de la báscula y así evitar pesajes dobles, pesajes incompletos, etc.
2. Mejorar el aspecto de seguridad en el pesaje de camiones para evitar lecturas erróneas o errores humanos.

Los objetivos específicos de este trabajo de graduación son los siguientes:

1. Interfazar una computadora al sistema de báscula para poder capturar el peso actual.
2. Mejorar el tiempo de pesaje de camiones comparado con el sistema de automatización puntual.
3. Integrar la información de pesaje a un sistema de información central para poder ser consultada dentro y fuera de la planta.
4. Crear un sistema de identificación único que se pueda utilizar dentro de toda la planta para todos los procesos que se llevan a cabo dentro de ella. Debe ser seguro, único y difícil de duplicar.

### III. DESARROLLO

#### A. Método de Trabajo

Para lograr el objetivo, se pretende hacer un estudio completo del ciclo de los camiones, datos que se necesitan almacenar, requisitos para una buena estabilidad, y control con tarjetas de proximidad.

1. **Definiciones.** Para poder comprender mejor la metodología de trabajo, se enumeran unas definiciones que son esenciales para el proyecto.

a. Microcontrolador. Es un circuito integrado programable con puertos de entrada y salida que pueden ser utilizados en muchas aplicaciones de la industria debido a que internamente poseen memoria RAM, memoria EEPROM y ROM para almacenar el programa. Su programación se hace sencilla debido a sus instrucciones en lenguaje de máquina y sus protocolos integrados, como el serial y el I<sup>2</sup>C.

Además, el microcontrolador trabaja a velocidades relativamente altas (hasta 40MHz), lo que lo hace ideal para aplicaciones donde se necesita un rápido acceso o proceso de información.

b. Protocolo Wiegand. El protocolo Wiegand proximidad (de 26 bits) es un estándar en la comunicación que consta de dos líneas de datos, una exclusivamente para representar ceros, y la otra para representar unos en una trama digital. Consta de 26 bits, los cuales se dividen de la siguiente manera:



Diagrama 1 Trama de Protocolo

en donde I indica el número de estación asociado a esa tarjeta, luego un BIT de paridad del número de estación, seguido por 16 bits que indican el número de la tarjeta y por último un BIT de paridad del número de la tarjeta.

c. Protocolo serial. El protocolo serial, en el más simple de los casos, utiliza tres líneas, una de transmisión de datos (TX), otra de recepción de datos (RX) y una de tierra (GND). Sincronizando los parámetros entre el receptor y el transmisor se pueden lograr transmisiones desde los 300bps hasta los 57,600bps, con o sin paridad, con 8 ó 9 bits de datos y con control de hardware para el flujo de datos. Para ello se manejan voltajes CMOS, representando el 1 como -12VDC y el 0 como +12VDC.

d. Puerto paralelo de la computadora. El puerto paralelo de la computadora consiste en un arreglo de 8 bits bi-direccionales que pueden ser leídos o escritos desde un programa. Cada BIT puede ser manejado independientemente de los otros y se manejan voltajes TTL de 0-5VDC con poca corriente.

2. **Secciones del proyecto.** Teniendo ya una idea general de los términos anteriores, se procede a la descripción de la metodología que se divide en cinco secciones:

- **Control de barreras:** Consiste en colocar barreras de acceso a la entrada y a la salida de la báscula para controlar el tránsito de los camiones a pesar. Para ello se utilizará un circuito de conmutación electrónica y puerto paralelo de la computadora. Esto resolvería el aspecto de control de flujo y disminución de errores.

- **Control de acceso:** Consiste en colocar, configurar y leer la información de tarjetas de proximidad o algún otro tipo de método de identificación única. Para ello se utiliza un microcontrolador que convierte la información de un protocolo estándar de proximidad llamado Wiegand a protocolo serial para ser procesado por la computadora. Con esto se resuelve el aspecto de identificación dentro de la planta, además de ser muy seguro y difícil de duplicar. En el apéndice A se presenta el Código del PIC para conversión de protocolo Wiegand a RS-232.

- **Control de báscula:** Consiste en recopilar la información que provee la báscula a la computadora y poder descifrar la información que esta provee. Esta información se utiliza en el siguiente paso para almacenarla a la base de datos.

- **Bases de datos:** Consiste en poder recuperar y guardar información a un camión específico para luego tomar decisiones sobre esta información y poder consultarla dentro y fuera de la planta.

- **Control lógico:** Es la parte del proyecto que incluye el algoritmo lógico de estados y que provee la seguridad con operaciones y el manejo de todo el hardware involucrado. Para ello se elaborará un software específico.

- a. **Control de Barreras.** Para este caso en específico, se pretende utilizar barreras tales como se muestran en el Gráfico 2.1. Las mismas constan de dos modos de programación, el primero que trabaja transmitiéndole un pulso único para abrir y luego esperar otro pulso (que proviene de un detector de metales) cuando el vehículo pasa. En el caso específico de los camiones, este modo no es útil, debido a los ejes del camión. Para contrarrestar este problema, se

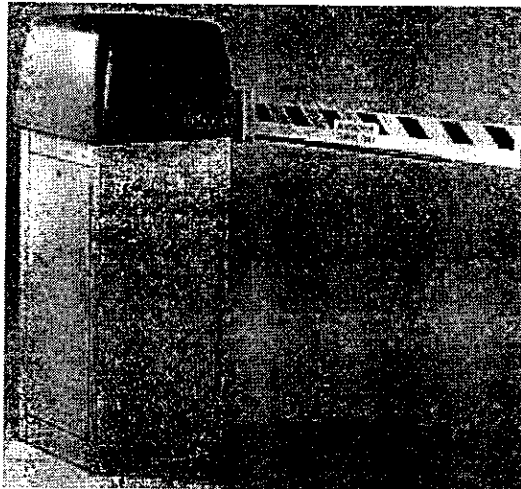


Grafico 2.1 Barrera automática para control de tránsito

desea utilizar el modo 2 de programación, el cual consiste en que si se le mantiene un pulso continuo a la barrera, ésta se levanta, y al quitar el pulso, ella baja. Con esto se logra el objetivo

deseado, ya que es posible mantener un pulso por medio de un control por *switches* magnéticos controlados por el puerto paralelo.

El estado esperado para bajar la barrera luego que el camión haya pasado es cuando el peso es igual a cero, lo cual indica la ausencia de camión en la báscula.

El circuito de control para la barrera el cual consiste de un transistor NPN en modo de *switch* que a su vez activa la bobina de un *relay* para aislar los voltajes del puerto paralelo y la barrera. Además, se utiliza una resistencia de  $100\Omega$  a la entrada del transistor para protegerlo de cualquier variación del puerto paralelo, como se muestra a continuación.

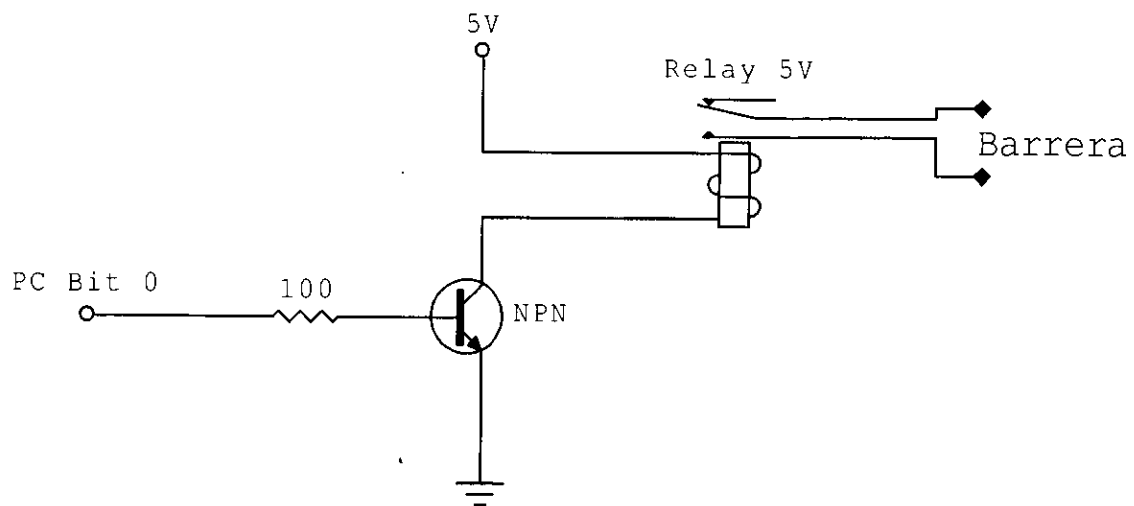


Diagrama 2.1 Circuito de Control de Barrera

En el cuadro siguiente se describen los bits que se deben mandar a la barrera para subirla o bajarla respectivamente.

Estado de bit paralelo	Estado de la barrera
1	Arriba
0	Abajo

Cuadro 2.1 Estados de Barrera

b. Control de acceso. Para el control de acceso, se utilizaron lectoras marca HID de proximidad. Estas lectoras emiten señales RF de bajo nivel las cuales son captadas por la antena de las tarjetas de proximidad (Gráfica 2.2) y la utilizan como fuente de poder, por lo que las tarjetas no necesitan de energía interna para su operación.

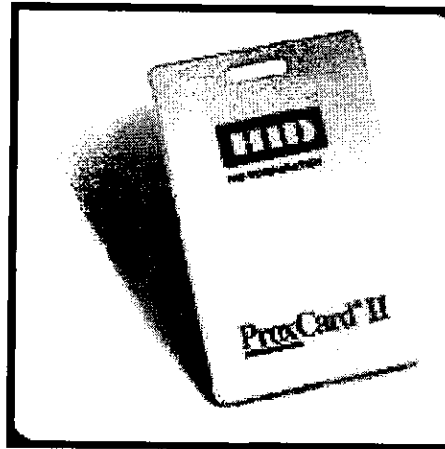


Gráfico 2.2 Tarjeta de Proximidad

Esta corriente se utiliza para alimentar el micro-controlador interno de la tarjeta, el cual transmite el número único que se le ha programado. El proceso toma unos cuantos milisegundos. Este número único está relacionado, por medio de una base de datos, a un camión específico, por lo que no se necesita ningún dato adicional del transporte en todo el proceso. El protocolo de comunicación de la lectora es un estándar llamado *Wiegand*, el cual se originó a partir del estándar de banda magnética. La trama de comunicación consta de dos líneas, una que transmite los unos y otra que transmite los ceros para formar una trama digital de 26 *bits*. El reloj no es necesario en este transporte de datos, debido a que el microcontrolador solamente detecta el dato en cambio de la línea de alto a bajo.

Debido a la complejidad de la señal, es necesario incorporar un circuito de conversión de protocolo *Wiegand* a protocolo RS-232, el cual es fácilmente interpretable por una computadora.

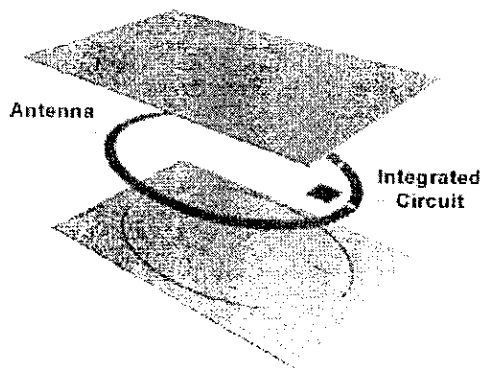


Diagrama 2.2 Componentes de una Tarjeta de Proximidad

Para ello se utilizó un microcontrolador de la serie PIC-16F877 por la facilidad de incluir el protocolo serial y poseer terminales de interrupción en cambio, las cuales se ajustan perfectamente a la señal de Wiegand. El programa básicamente espera a recibir un cambio en la señal de dos de sus terminales, y al recibirlo, inmediatamente reporta a la computadora ese cambio para que la misma pueda decodificarlo.

c. Control de báscula. Las básculas que se pretendían automatizar son marca Metler Toledo especiales para pesaje de camiones. Funcionan con un protocolo propietario por lo que necesitan de una terminal de decodificación propia. Dicha terminal posee dos salidas, una paralela y otra serial. La más conveniente es la serial, debido a su simpleza de recepción y su largo alcance.

De igual manera, la báscula cuenta con dos modos de trabajo, uno de ellos es la impresión por demanda, en la cual la PC le envía un carácter y la báscula responde con el peso actual y otros parámetros, y el otro modo es el modo continuo, en el cual la báscula continuamente transmite el peso actual a la PC seguido de un carácter 13 (enter), un carácter 10 (fin de línea) y un carácter de borrar pantalla.

Después de analizar la señal y obtener esos resultados, se procedió a desglosar la señal en la PC para desechar los datos que no interesan y obtener el peso actual. Para lograr esto se creó un procedimiento en software el cual puede ser llamado en cualquier momento del programa para recibir el peso actual.

d. Base de datos. Es el paso equivalente a la Central en la CIE. En ella se encuentran todos los datos necesarios para la identificación, almacenamiento y bitácora de lo que está sucediendo en el proceso. En este caso específico se utilizó la base de datos de Oracle para almacenar la información. Esta base de datos tiene la ventaja de ser muy rápida y robusta para aplicaciones industriales. La misma está instalada en unos servidores de datos redundantes tanto en disco duro, como en cpu y fuentes de poder. Con ello se garantiza la seguridad de los datos.

Se utiliza mucho durante el proceso de pesaje, ya que en ella se almacena la información del piloto, el número de tarjeta que le corresponde, la placa, la licencia, el peso bruto y tara teórico (que proviene del despachador), etc.

Luego, en las básculas, esta información es recobrada con propósitos de seguridad y de estado actual de un camión específico. Con ello se comparan diferencias muy grandes en pesos, tarjetas inválidas, dobles pesajes, etc. Y además facilita la consulta de datos en toda la planta y a través de Internet.

También se incluye una bitácora local en la PC, donde se almacena cada uno de los pasos que sucedieron en la báscula, cada cambio de estado y las mediciones promedio.

Se cuenta además con una tabla especial en donde se almacenan los trece pesos que se toman en cuenta para almacenar el peso final y además, el tiempo que se tarda en realizar dicha operación. Con ello se pueden hacer estadísticas de rendimientos.

e. Control lógico. Es la parte esencial del trabajo, la cual consiste en la integración de todos los elementos involucrados en la automatización, la toma de decisiones, reporte de errores y control de equipo de campo. Es la interfaz entre el equipo y la central. Es equivalente en CIE a la estación de trabajo y al mismo tiempo a la Célula.

Para el desarrollo del software, se utilizó el lenguaje Visual Basic .NET, el cual ofrece mucha versatilidad y a la vez escalabilidad para nuevas aplicaciones.

Se necesitaron desarrollar procedimientos para el control de los puertos seriales y el puerto paralelo de la PC, ya que son indispensables para el funcionamiento de la estación de trabajo.

Debido a que se pretende no necesitar de personal dentro de la báscula, el sistema tiene que ser suficientemente inteligente para manejar todas las situaciones que puedan presentarse.

A continuación se presente el diagrama de flujo del proceso y se pueden apreciar todas las decisiones involucradas en el.

### 3. Descripción del flujo

- a. Inicio del programa (Estado “Esperando 0”).
- b. La báscula reporta el peso al programa.
- c. Si no es cero, sigue tomando pesos hasta que se cumpla la condición (paso b)
- d. Cuando se cumple la condición indica que la báscula ha estabilizado su peso en cero, por lo que ya puede ingresar un camión. Se abre la barrera de entrada y se cierra la de salida.



Grafico 3.1 Entrada a Báscula

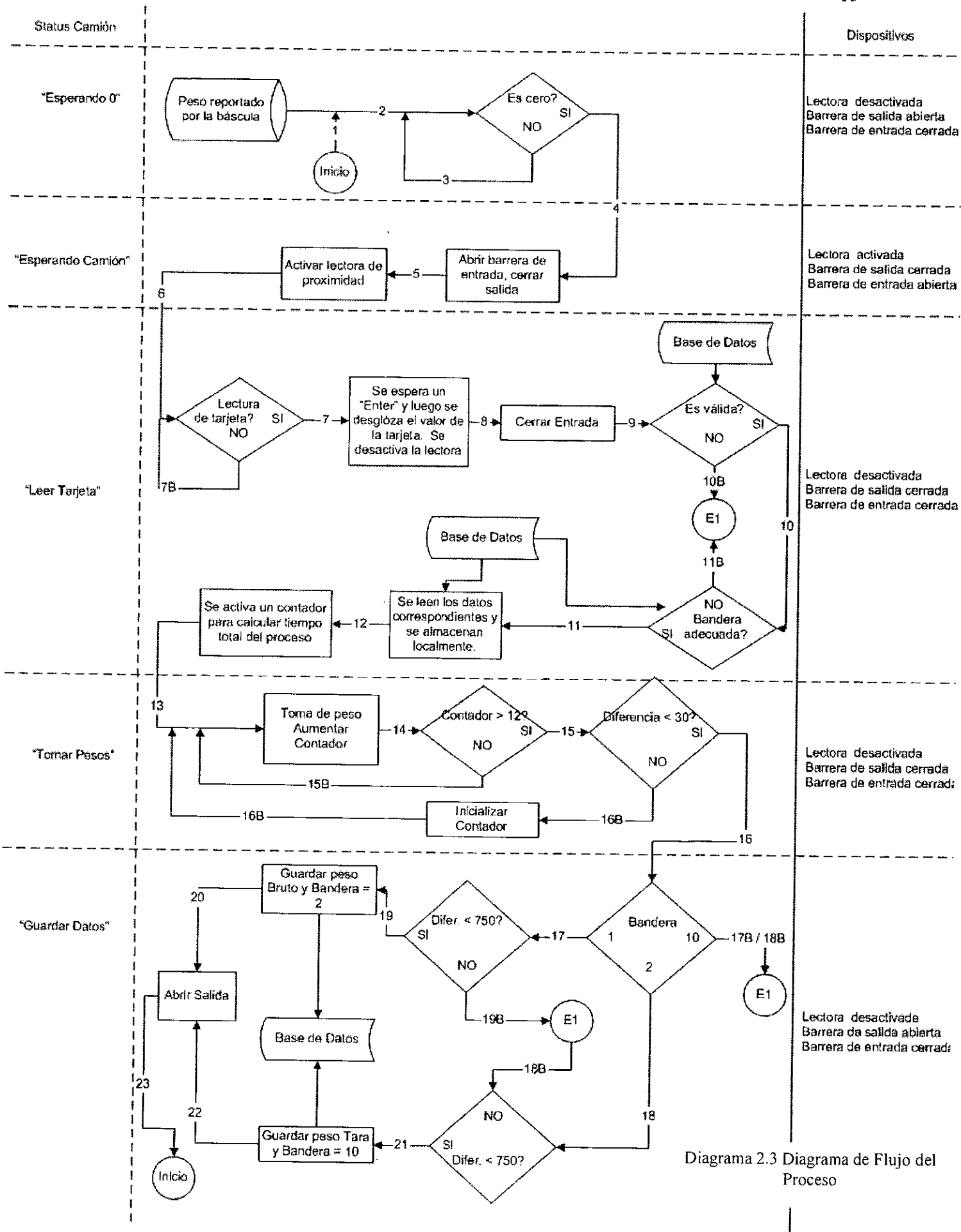


Diagrama 2.3 Diagrama de Flujo del Proceso

- e. Se activa la lectora de proximidad (Estado “Esperando Camión”).
- f. Se espera la lectura de la tarjeta (Estado “Leer Tarjeta”).
- g. Decisión para lectura de tarjeta.
  - 1) Si se recibe una lectura, se analiza, se busca el *enter*, se desglosa y se obtiene el número de la tarjeta. Inmediatamente se desactiva la lectora. (se procede a paso h)
  - 2) Si no se recibe lectura, no se hace nada (se regresa a paso f)
- h. Se cierra la barrera de entrada.



Grafico 3.2 Pesaje entre Bascula

- i. Se toma la decisión si la tarjeta es válida.
  - 1) Si es válida, se procede a chequear el estado del camión, por medio de la bandera (se procede a paso j)
  - 2) Si no es válida, se llega a un error fatal y el camión queda encerrado hasta resolver el problema.

- j. Es correcta la bandera (estado actual).
  - 1) Si es correcta, significa que este camión todavía le falta pesar peso bruto o peso tara (se procede al paso k)
  - 2) Si no es correcta significa que el sistema registra que este camión ya pesó tara y bruto, por lo que no debería estar pesando de nuevo. Se llega a un error fatal y el camión queda encerrado.
- k. Debido a que es correcta la bandera, se leen los datos del camión actual de la base de datos.
- l. Se activa un contador para calcular el tiempo total de proceso.
- m. Se toma el peso y se aumenta un contador global.
- n. Control si ya se tomaron 12 pesos.
  - 1) Si ya se tomaron 12 pesos se procede al paso o
  - 2) Si no se han tomado 12 pesos, se regresa al paso m para tomar el resto de los pesos, los cuales son almacenados temporalmente en la memoria de la computadora.
- o. Se toma el peso menor y el peso mayor de los 12 pesos y se restan y se les saca el valor absoluto, luego se compara si la diferencia entre ellos es menor a 30kg.
  - 1) Si es menor a 30kg, se procede al paso p
  - 2) Si es mayor a 30kg, se reinician los contadores y se regresa al paso m
- p. Si la bandera es 1, se procede a capturar el peso de la báscula y compararlo con el peso teórico. Luego se compara la diferencia entre ellos.
  - 1) Si es menor que 750kg, se procede al paso s
  - 2) Si es mayor que 750kg, se llega a un error fatal por mucha diferencia entre pesos teóricos y real.
- q. Si la bandera es 2, se procede a capturar el peso de la báscula y compararlo con el peso teórico. Luego se compara la diferencia entre ellos.

- 1) Si es menor que 750kg, se procede al paso u
  - 2) Si es mayor que 750kg, se llega a un error fatal por mucha diferencia entre pesos teóricos y real.
- r. Si la bandera es 10, esto indica que el camión no debería estar en báscula, debido a que ya tiene registros de peso bruto y peso tara, lo cual genera un error fatal debido a mal estado del camión.
  - s. Se almacena el resultado del peso bruto en la base de datos y se le actualiza la bandera para que valga 2.
  - t. Se abre la barrera de salida para que el camión se retire de la báscula. (ir al paso v).
  - u. Se almacena el resultado del peso tara en la base de datos y se le actualiza la bandera para que valga 10.
  - v. Se abre la barrera de salida para que el camión se retire de la báscula como se muestra en el grafico 3.3.

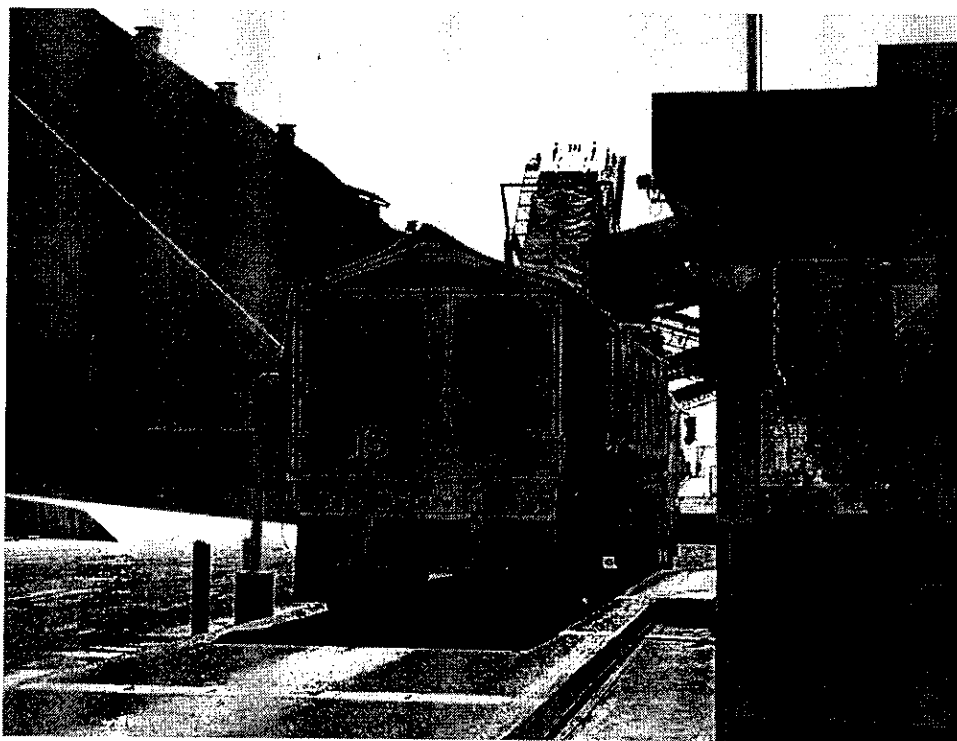


Grafico 3.3 Salida de Báscula

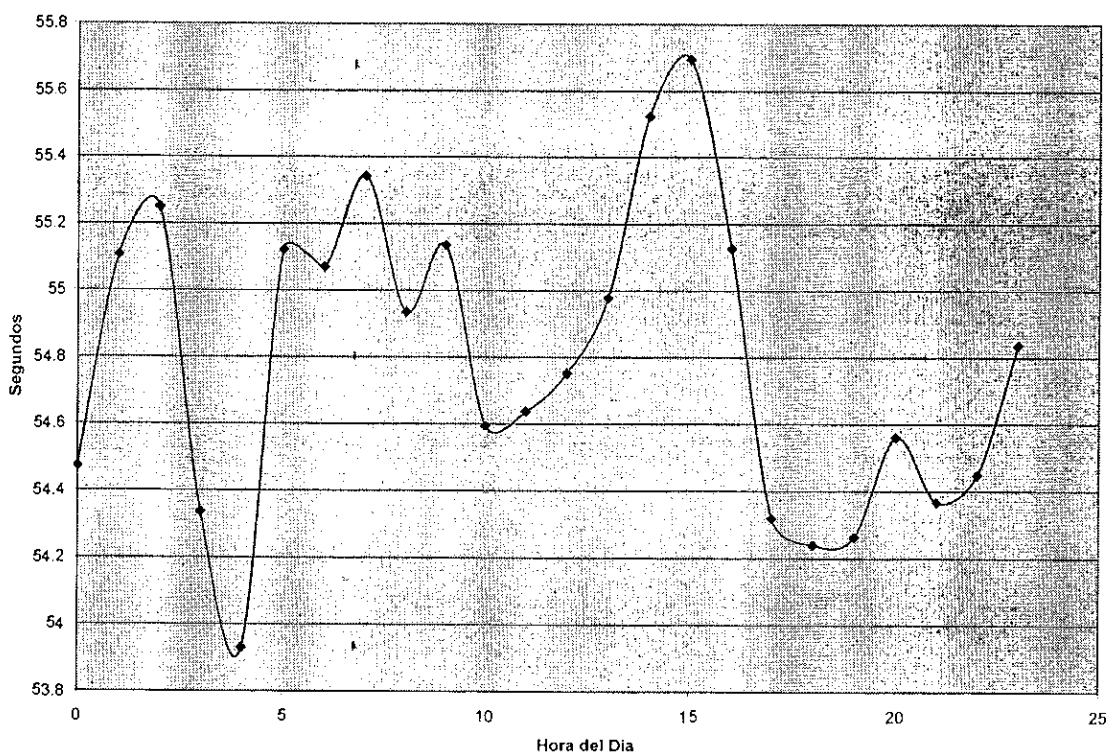
w. Se vuelve al paso a para reiniciar el proceso.

En El apéndice B se presenta el código del programa utilizado para controlar el pesaje y equipo de campo.

## IV. RESULTADOS OBTENIDOS

### A. Resultados y análisis de tiempos

En la gráfica siguiente se observan las variaciones en los tiempos actuales de pesaje, dividiendo el día por horas. Se puede observar que en horas de la madrugada (4am) el tiempo es mínimo, mientras que a las 15 horas se observa el máximo tiempo de espera. Esto nos indica que a esa hora se pueden encontrar el clima con mayor viento en todo el día. Para ello se tomaron un total de 69,000 muestras, tomando el promedio en cada hora. Además, se desecharon los valores mayores a los 200 segundos, debido a que si hay tiempos mayores a esto, se consideran como error o como que un factor externo hizo el proceso más tardado (como error en tarjeta, problemas mecánicos con el camión, etc.)

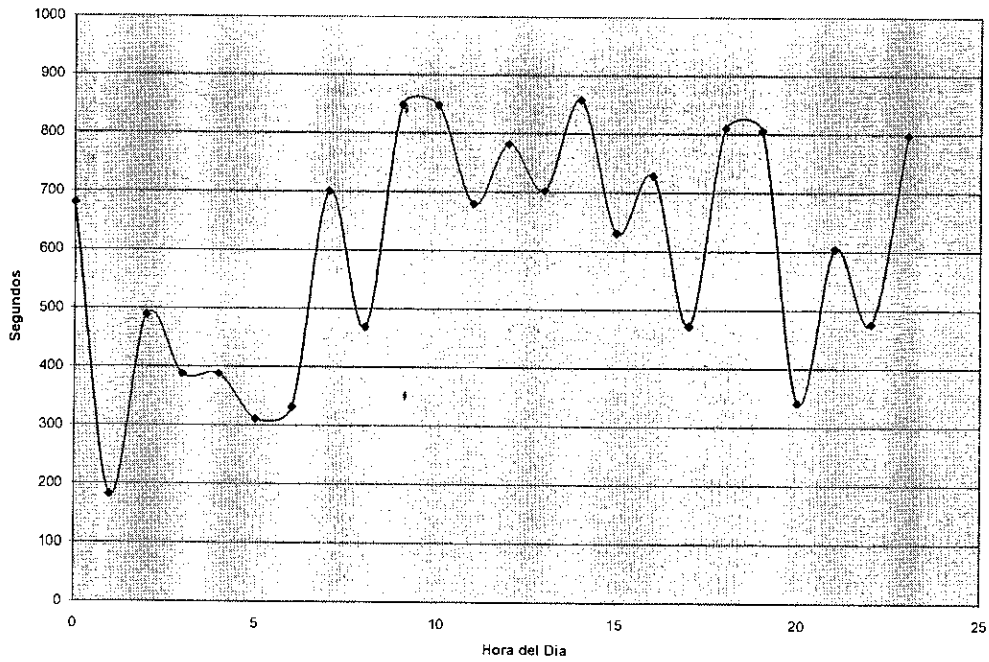


Gráfica A1. Tiempos totales de pesaje por hora

<b>Hora</b>	<b>Tiempo Segundos</b>
0	54.47261815
1	55.10652353
2	55.24885216
3	54.33416771
4	53.92970822
5	55.12094763
6	55.06904577
7	55.34112878
8	54.93412095
9	55.13531287
10	54.59301364
11	54.63604406
12	54.7498704
13	54.97655209
14	55.5227868
15	55.69308176
16	55.12521151
17	54.3177423
18	54.2391514
19	54.26289341
20	54.55859031
21	54.36678404
22	54.44806924

Cuadro A1. Tiempos totales por hora

Además de la gráfica anterior, donde se limitaban los datos a todos aquellos menores de 200 segundos, la siguiente gráfica toma en cuenta los máximos tiempos de espera dividido en las horas del día, donde se puede observar una gráfica más lineal y constante que la de promedios. La baja en las horas de las madrugadas se debe probablemente a que hay un número menor de camiones a esas horas y por consiguiente, tiene menor probabilidad de haber un error de tarjeta o alguna ráfaga de viento al azar.

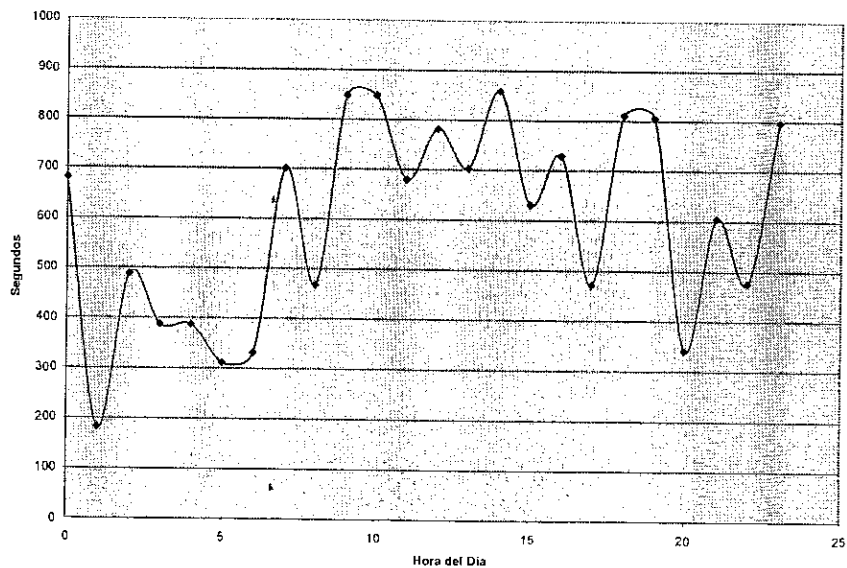


Gráfica A2. Máximos tiempos de pesaje por hora

Hora	Tiempo Máximo Segundos
0	680
1	182
2	488
3	387
4	387
5	311
6	331
7	700
8	468
9	848
10	848
11	679
12	781
13	702
14	858
15	630
16	728
17	472
18	810
19	805
20	341
21	604
22	476
23	797

Cuadro A2. Tiempos máximos de pesaje por hora

En la gráfica A3, se toma el número de ocurrencias de tiempos mayores a los 200 segundos con la misma cantidad de muestras, en la cual se puede observar que ente las 9 horas y las 12 horas es cuando se ha encontrado una mayor incidencia de estos casos.



Gráfica A3. Cantidad de camiones por hora con tiempos mayores a 200 seg

Hora	Cantidad de Camiones
0	9
2	5
3	5
4	2
5	3
6	5
7	23
8	13
9	28
10	30
11	36
12	31
13	14
14	13
15	20
16	16
17	17
18	13
19	11
20	10
21	6
22	12
23	12

Cuadro A3. Cuenta de tiempos de pesaje por hora

## B. Análisis antes y después de la automatización

Para comparar los beneficios en tiempo que se ahorra el proceso de pesaje mediante la automatización de las basculas, se presenta a continuación diferentes histogramas de frecuencias para desplegar de forma gráfica la variabilidad dentro del proceso con base al conjunto de datos (tiempos) recolectados.

1. Datos antes de la automatización. Se realizó una estimación de la medida para la recolección de los datos antes de que se implementara la automatización, se recolectaron 1000 datos los cuales son una medida representativa del universo de datos según el siguiente teorema<sup>1</sup>:

$$n = \left( \frac{z_{\alpha/2} \sigma}{e} \right)^2$$

donde  $n$  es igual al número mínimo que se necesita de muestras para obtener el grado deseado de exactitud.

$Z$  área bajo la curva normal según la tabla estadística para un grado de confianza de 95%

$\sigma$  Desviación estándar de las muestras

$e$  error estimado en la toma de muestras

<i>Datos</i>	
Desviación Estándar ( $\sigma$ )	35.67426
$Z_{95\%}$	1.65
Error estimado ( $e$ )	2 Segundos

Cuadro B1. Datos para calcular representatividad de la muestra

Este cálculo da como resultado que se deben tener al menos 866 datos para que el análisis de la muestra sea representativo.

Según los 1000 datos obtenidos antes de la automatización se obtiene lo siguiente:

<sup>1</sup> Teorema 7.2 Probabilidad y Estadística, Walpole y Myers

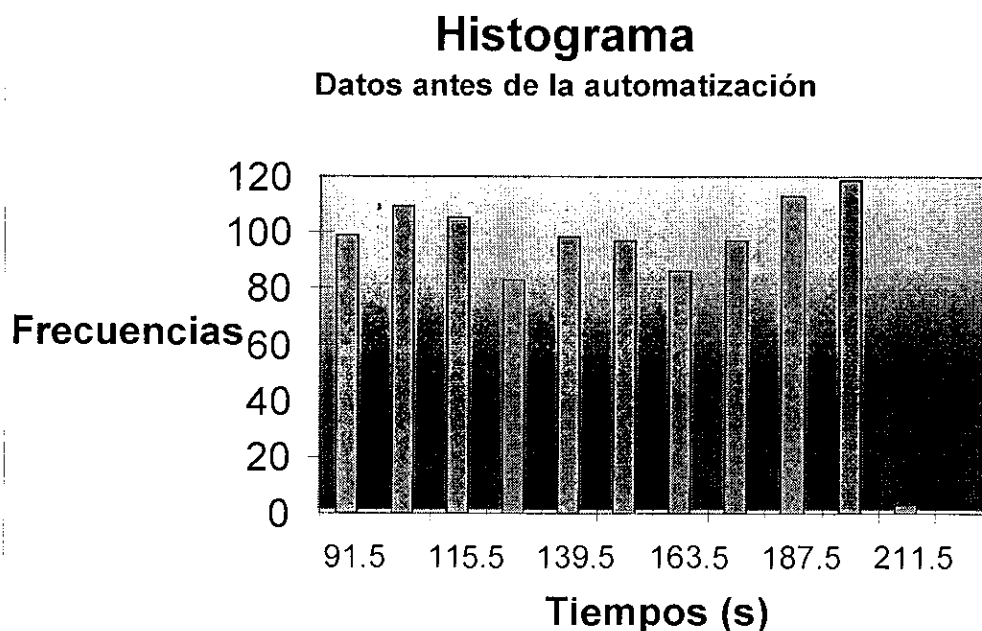


Gráfico B1. Histograma antes de automatización

Se puede inferir del gráfico anterior que el proceso no está controlado ya que los tiempos de pesaje no son uniformes y varían entre 80 y 200 segundos, esto se debe a que el proceso solía hacerse manualmente, el tiempo en que el piloto se baja del camión y entrega la papelería, el operador de datos ingresa los pesos es variable, debido a que el piloto desciende del camión, hay variaciones adicionales en la báscula.

2. Datos después de la automatización. Los datos recolectados después de la automatización representan el 100% del universo de muestras por lo tanto no es necesario aplicar la fórmula anterior para determinar el número de muestras necesarias para realizar el análisis.

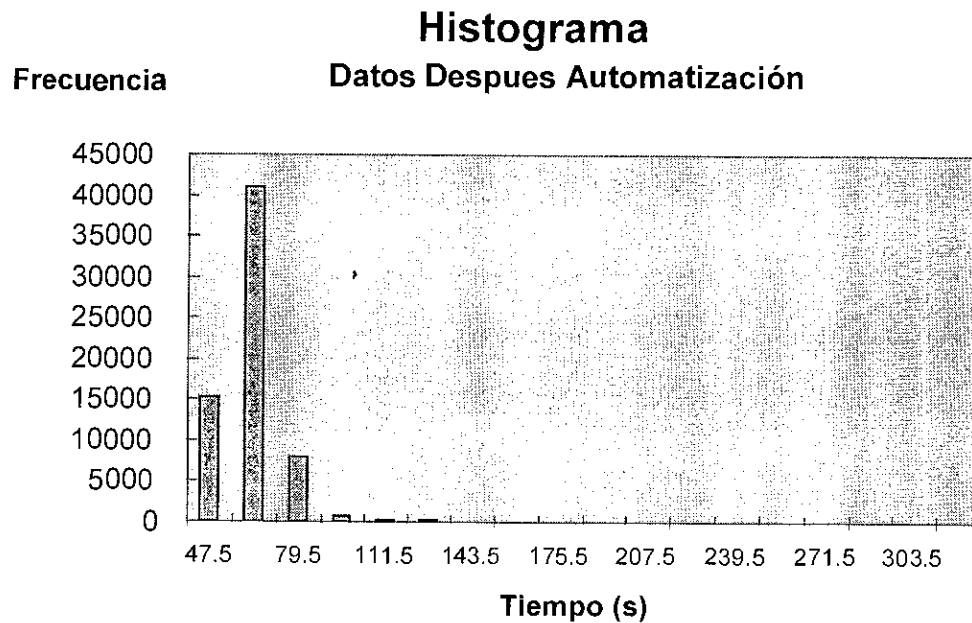


Gráfico B2. Histograma después de automatización

Se extrae del gráfico anterior la nueva tendencia en cuanto a los tiempos y se puede concluir que el proceso de pesaje se encuentra controlado y no tiene variaciones significativas, algunos de los tiempos se encuentran aislados hacia la derecha, éstos pueden ser interpretados como momentos cuando el viento es mayor y esto causa una inestabilidad en la báscula por unos instantes. Esto obliga al programa a tomar otras muestras y tardar un poco más de tiempo antes de guardar los datos.

En el Apéndice C se muestra la justificación financiera del proyecto de automatización como complemento de este estudio y para presentar el beneficio económico que esto conlleva.

## V. CONCLUSIONES Y RECOMENDACIONES

### A. Conclusiones

Luego de haber desarrollado el trabajo de investigación y la puesta en marcha del sistema, se obtuvieron los siguientes resultados:

1. Se eliminó totalmente la necesidad de personal de báscula, y con ello se aumentó la velocidad, se eliminaron los errores tipográficos y se redujo el costo del pesaje.
2. Se logró utilizar una misma infraestructura de báscula para poder pesar tanto camiones cargados (bruto) como camiones vacíos (tara), obteniendo así una mayor flexibilidad y economía.
3. También fue efectivo el método para conversión de protocolo Wiegand a protocolo serial para así lograr una comunicación eficaz entre la lectora y la computadora.
4. Por medio de las tarjetas de proximidad y una llave primaria en la base de datos, se pudo obtener una identificación única dentro de la planta, lo cual facilita todos los procesos de almacenamiento y reconocimiento.
5. Las barreras fueron un control muy efectivo para el control de flujo de los camiones.
6. Se logró, por medio del método de bajar la barrera después de recibir la señal de la tarjeta de proximidad, encerrar al camión durante todo el proceso de pesaje, y en caso de algún error fatal, este no tiene opción de poder salir hasta que no se resuelva el error.
7. Por medio de la bitácora local y de base de datos es posible consultar en todo momento el estado de la báscula, así como poder rastrear una operación.
8. Se observó que a ciertas horas del día en ciertas básculas, el viento es un factor crucial en la toma de pesos, y que se deben tomar muchas muestras para lograr un resultado aceptable (menos de 30kg de diferencia).

## B. Recomendaciones

1. Es recomendable implementar el algoritmo de esperar diez segundos entre las mediciones del cero (justo después que el camión sale de la báscula), ya que de lo contrario es muy probable que la barrera impacte al camión.
2. No utilizar detector de campo magnético para bajar las barreras, debido a que los camiones no son detectados con fidelidad por estos dispositivos, por el contrario, la solución de esperar que la báscula estabilice a cero es una mejor opción.
3. Colocar las barreras sobre bases de concreto altas para que la barrera quede a un nivel apropiado para un camión. De lo contrario, éstas podrían ser quebradas por los camiones si no se tiene una buena visibilidad.
4. Activar el modo de rebote de las barreras, para que sean capaces de bajar automáticamente 10 segundos después y no crear confusión en el flujo de los camiones.
5. Implementar algún tipo de protección para el viento cerca de las celdas de la báscula para evitar variaciones muy grandes a ciertas horas del día.

## V. BIBLIOGRAFÍA

- Jonson, Jonathan. CIM <http://www.tech.purdue.edu>
- Oregon State University. *Computer Integrated Manufacturing* 04/22/03 artículo  
<http://ie.oregonstate.edu/facilities/cim/>
- ATMI. *ATMI Computer Integrated Manufacturing (CIM) Standard for Spinning*  
<http://www.techexchange.com/thelibrary/ATMI.html>
- SEMATECH. *Computer Integrated Manufacturing (CIM) Framework, Architecture, Principles and guidelines, 7 version.* scott Hawker, 1996,  
[www.sematech.org/public/docubase/document/3214aeng.pdf](http://www.sematech.org/public/docubase/document/3214aeng.pdf)
- Amano Cincinnati. *ATP-1700 Parking Gate*  
<http://www.adautogate.com/amano.html>
- Persona Printers. *How Proximity Cards Work*  
<http://www.id-enhancements.com/proximity-cards.html>
- Wich Technology. *How Reader Technology Works*  
[http://www.kantech.com/tech\\_support/reference\\_material/body\\_wichTechnology.htm](http://www.kantech.com/tech_support/reference_material/body_wichTechnology.htm)
- California Polytechnic and State University. *CIM in Manufacturing*  
*Henry C. Co*  
[www.csupomona.edu/~hco/TOM420/](http://www.csupomona.edu/~hco/TOM420/)
- Walpole, Ronald ; Myers Raymond. 1992. *Probabilidad y Estadística.* 4<sup>ta</sup> edición,  
McGraw-Hill, México, D.F. 797 Págs.

## VI. APÉNDICE

### A. Código del PIC para conversión de protocolo Wiegand a RS-232

```
LIST          P=16F877
#include      <P16F877.INC>
#define      XTAL 2000000          ; Frecuencia del xtal usado
__CONFIG    b'11111100000101'
;__CONFIG    b'00011100000101'    ;; CP OFF - PWRT ON
                                       ;; - WDT OFF - XT_OSC

CBLOCK 0x20
    YAMANDO
    YAMANDOTWICE
    TEMPRX
    d1
    d2
    d3
    a1
    a2
ENDC

=====
; INICIO DEL PROGRAMA
=====
ORG          0x000          : Vector de reset
CALL INICIALIZAR
GOTO MAIN

=====
; INTERRUPTS
=====
ORG          0x04          : Vector de interrupt
BTFSC PIR1, RCIF          : chequea si fue el receive
GOTO INTSRX
GOTO REGRESA

INTSRX
CLRWDT
BCF INTCON, GIE
BANKSEL RCREG
BCF RCSTA, CREN          ; Si hay error de overflow, se limpia
MOVWF RCREG, W          ; Se mueve el registro de recepcion a W
BANKSEL TEMPRX          ; Se almacena lo recibido en la var TEMPRX
MOVWF TEMPRX
BCF STATUS, Z
MOVLW A'a'
SUBWF TEMPRX, W
```

```

BTFSK STATUS, Z
GOTO ABRIRA
MOVLW A'b'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRB
    CLRWDT
MOVLW A'c'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRC
MOVLW A'd'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRD
MOVLW A'A'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRAA
MOVLW A'B'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRBB
CLRWDT
MOVLW A'C'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRCC
    MOVLW A'D'
SUBWF TEMPRX, W
BTFSK STATUS, Z
GOTO ABRIRDD
CLRWDT
    GOTO REGRESA

```

```

REGRESA
CLRWDT

```

```

    BCF INTCON, RBIF                ; Clear flag de interrupt
    BSF INTCON, GIE                 ; Re enable global ints
    BANKSEL RCSTA
    BSF RCSTA, CREN                 ;Se rehabilita la recepcion
    RETFIE                          ; return from interrupt
    GOTO MAIN

```

```

ABRIRAA
    BSF PORTA, 0
    GOTO REGRESA

```

```

ABRIRBB
    BSF PORTA, 1
    GOTO REGRESA

```

```

ABRIRCC

```

```

BSF PORTA, 2
GOTO REGRESA

ABRIRDD
BSF PORTA, 3
GOTO REGRESA

ABRIRA
BSF PORTA, 0
CALL DELAY
CALL DELAY
BCF PORTA, 0
GOTO REGRESA
;
ABRIRB
BSF PORTA, 1
CALL DELAY
CALL DELAY
BCF PORTA, 1
GOTO REGRESA

ABRIRC
BSF PORTA, 2
CALL DELAY
CALL DELAY
BCF PORTA, 2
GOTO REGRESA
;
ABRIRD
BSF PORTA, 3
CALL DELAY
CALL DELAY
BCF PORTA, 3
GOTO REGRESA

=====
; INICIALIZACION !
=====
INICIALIZAR
BANKSEL TRISA ; TODO EL PUERTO A,D DE SALIDA
CLRF TRISA
CLRF TRISD

MOVLW .255 ; PUERTO B DE ENTRADA
MOVWF TRISB

MOVLWB'01111111' ; TX output. RX input
MOVWF TRISC
MOVLW b'00000110' ; A/D APAGADO EN TODO EL PUERTO A
MOVWF ADCON1

BANKSEL PORTA
MOVLW .255
MOVWF PORTA
MOVWF PORTD

```

```

CLRF PORTA           ; CLEAR EL PUERTO A Y B
CLRF PORTB
CLRF PORTD

BANKSEL RCSTA
MOVLWB'10010000'    ; Serial port y continuous receive
MOVWF RCSTA

BANKSEL SPBRG
MOVLW .64
MOVWF SPBRG         ; 19200 bps

BANKSEL TXSTA
MOVLWB'00100100'    ; Serial port and continuous transmit
MOVWF TXSTA
BANKSEL PIE1
MOVLWB'00100000'    ; Enable Receive Interrupt RCIE
MOVWF PIE1

BANKSEL OPTION_REG ; Pull ups en puerto B
BCF OPTION_REG, 7

BANKSEL RCSTA
MOVLW B'10010000'   ; Enable serial port
MOVWF RCSTA

MOVLW b'11000000'
MOVWF INTCON        ; Serial port interrupt
MOVLW .79
CALL TRANSMIT
MOVLW .75
CALL TRANSMIT
CLRWDI
RETURN

MAIN:
        NOP

INICIO7:
        CLRWDI
btfss  PORTB,7       ;0x06,7 ;0022 1f86
goto   MANDAR7       ;0023 283b
bef    YAMANDO,7     ;0024 13a0
        bcf        YAMANDOTWICE, 7

INICIO6:
        CLRWDI
btfss  PORTB,6       ;0x06,6 ;0025 1f06
goto   MANDAR6       ;0026 2843

```

```

    bcf YAMANDO,6           ;0027 1320
      bcf YAMANDOTWICE, 6

INICIO5:
      CLRWDT
    btfss PORTB,5           ;0x06.5 ;0028 1e86
    goto MANDAR5           ;0029 284b
    bcf YAMANDO,5         ;002a 12a0
      bcf YAMANDOTWICE, 5

INICIO4:
      CLRWDT
    btfss PORTB,4           ;0x06.4 ;002b 1e06
    goto MANDAR4           ;002c 2853
    bcf YAMANDO,4         ;002d 1220
      bcf YAMANDOTWICE, 4

INICIO3:
      CLRWDT
    btfss PORTB,3           ;0x06.3 ;002e 1d86
    goto MANDAR3           ;002f 285b
    bcf YAMANDO,3         ;0030 11a0
      bcf YAMANDOTWICE, 3
      CLRWDT

INICIO2:
      CLRWDT
    btfss PORTB,2           ;0x06.2 ;0031 1d06
    goto MANDAR2           ;0032 2863
    bcf YAMANDO,2         ;0033 1120
      bcf YAMANDOTWICE, 2

INICIO1:
      CLRWDT
    btfss PORTB,1           ;0x06.1 ;0034 1e86
    goto MANDAR1           ;0035 286b
    bcf YAMANDO,1         ;0036 10a0
      bcf YAMANDOTWICE, 1

INICIO0:
      CLRWDT
    btfss PORTB,0           ;0x06.0 ;0037 1e06
    goto MANDAR0           ;0038 2873
    bcf YAMANDO,0         ;0039 1020
      bcf YAMANDOTWICE, 0
    goto MAIN              ;003a 2822

MANDAR7:
      btfss YAMANDOTWICE, 7
      goto SETTWISE7
    btfsc YAMANDO,7         ;003b 1ba0
    goto INICIO6           ;003c 2825
    movlw A'A'             ;003d 3061
    call TRANSMIT          ;003e 207b
    bsf YAMANDO,7         ;0041 17a0
    goto INICIO6           ;0042 2825

```

```

SETTWICE7:
    bsf YAMANDOTWICE, 7
    call delay2
    goto INICIO6

MANDAR6:
    btfss YAMANDOTWICE, 6
    goto SETTWICE6
    btfsc YAMANDO,6 ;0043 1b20
    goto INICIO5 ;0044 2828
    movlw A'a' ;0045 3061
    call TRANSMIT ;0046 207b
    bsf YAMANDO,6 ;0049 1720
    goto INICIO5 ;004a 2828
SETTWICE6:
    bsf YAMANDOTWICE, 6
    call delay2
    goto INICIO5

MANDAR5:
    btfss YAMANDOTWICE, 5
    goto SETTWICE5
    btfsc YAMANDO,5 ;004b 1aa0
    goto INICIO4 ;004c 282b
    movlw A'B' ;004d 3062
    call TRANSMIT ;004e 207b
    bsf YAMANDO,5 ;0051 16a0
    goto INICIO4 ;0052 282b
SETTWICE5:
    bsf YAMANDOTWICE, 5
    call delay2
    goto INICIO4

MANDAR4:
    btfss YAMANDOTWICE, 4
    goto SETTWICE4
    btfsc YAMANDO,4 ;0053 1a20
    goto INICIO3 ;0054 282e
    movlw A'b' ;0055 3062
    call TRANSMIT ;0056 207b
    bsf YAMANDO,4 ;0059 1620
    goto INICIO3 ;005a 282e
SETTWICE4:
    bsf YAMANDOTWICE, 4
    call delay2
    goto INICIO3

MANDAR3:
    btfss YAMANDOTWICE, 3
    goto SETTWICE3
    btfsc YAMANDO,3 ;005b 19a0

```

```

goto INICIO2          :005c 2831
movlw A'C'           :005d 3063
call TRANSMIT        :005e 207b
bsf YAMANDO,3       :0061 15a0
goto INICIO2         :0062 2831

```

SETTWICE3:

```

    bsf YAMANDOTWICE, 3
    call delay2
    goto INICIO2

```

MANDAR2:

```

                btfss YAMANDOTWICE, 2
                goto   SETTWICE2
btfsc YAMANDO,2 :0063 1920
goto INICIO1    :0064 2834
movlw A'c'      :0065 3063
call TRANSMIT   :0066 207b
bsf YAMANDO,2   :0069 1520
                goto INICIO1          :006a 2834

```

SETTWICE2:

```

    bsf YAMANDOTWICE, 2
    call delay2
    goto INICIO1

```

MANDAR1:

```

                btfss YAMANDOTWICE, 1
                goto   SETTWICE1
btfsc YAMANDO,1 :006b 18a0
goto INICIO0    :006c 2837
movlw A'D'      :006d 3064
call TRANSMIT   :006e 207b
bsf YAMANDO,1   :0071 14a0
goto INICIO0    :0072 2837

```

SETTWICE1:

```

    bsf YAMANDOTWICE, 1
    call delay2
    goto INICIO0

```

MANDAR0:

```

                btfss YAMANDOTWICE, 0
                goto   SETTWICE0
btfsc YAMANDO,0 :0073 1820
goto INICIO7    :0074 2822
movlw A'd'      :0075 3064
call TRANSMIT   :0076 207b
bsf YAMANDO,0   :0079 1420
goto INICIO7    :007a 2822

```

SETTWICE0:

```

    bsf YAMANDOTWICE, 0
    call delay2

```

goto INICIO7

```
=====!  
; RUTINA DE TRANSMISION !  
=====!
```

TRANSMIT

```
CLRWDT  
BTFSS PIR1,TXIF ;ESPERA QUE TERMINE DE TRANSMITIR  
GOTO TRANSMIT  
MOVWF TXREG ;MANDA W A TX  
RETURN
```

```
=====!  
; RUTINA DE DELAY !  
=====!
```

DELAY:

```
clrwdt ;999990 cycles  
movlw 0x07  
movwf d1  
movlw 0x2F  
movwf d2  
movlw 0x03  
movwf d3  
DELAY_0:  
decfsz d1, f  
goto $+2  
decfsz d2, f  
goto $+2  
decfsz d3, f  
goto DELAY_0  
;  
goto $+1 ;6 cycles  
goto $+1  
goto $+1  
;  
return ;4 cycles (including call)
```

delay2:

```
movlw .1  
movwf a1  
decfsz a1, f  
goto delay2  
nop  
return
```

end

## B. Código del Programa en Visual Basic

```
Public Class frmMain
```

```
    Inherits System.Windows.Forms.Form
```

```
    Dim Mistring As String
```

```
    Dim Cuenta As Integer
```

```
    'Para azucar
```

```
    Dim CodigoArea, CodigoIngenio, CodigoProducto As Integer
```

```
    Dim CodigoTipoProducto, CodigoBodega, CodigoEmpaque, BodegaAuxiliar  
As Integer
```

```
    Dim PesoBrutoTemp As Double
```

```
    Dim PesoBrutoIngenio, PesoTaraIngenio As Double
```

```
    Dim TipodeProducto As Integer
```

```
    Const Azucar = 1
```

```
    Const Otros = 0
```

```
    Private Sub CheckStatus()
```

```
        PanelStatus.Text = Status
```

```

Select Case Status
    Case "Esperando 0"
        Esperar0()
    Case "Esperando Camion"
        AbrirEntrada()
    Case "Cerrar Entrada"
        CerrarEntrada()
    Case "Tomar Pesos"
        TomarPesos()
    Case "Guardar Datos"
        GuardarDatos()
    Case "Saliendo"
        Saliendo()
    Case "Error Tarjeta"
        ErrorTarjeta()
    Case "Error Peso"
        ErrorPeso()
    Case "Error Bandera"
        ErrorBandera()
    Case "Error Desconocido"
        ErrorDesconocido()
End Select

```

```
End Sub
```

```

Private Sub Esperar0()
    'Inicializar timer para 0
    Timer0.Enabled = True
End Sub

```

```

Private Sub TomarPesos()
    TimerInterPesos.Enabled = True
End Sub

```

```

Private Sub Saliendo()
    TalanqueraS = True

    Status = "Esperando 0"
    ActualizarTalanqueras()
End Sub

```

'Se le manda el numero de tarjeta como parametro y la busca en la base de datos, si no existe lo encierra  
' y tambien si la bandera no es valida.

```

Private Sub ParsearTarjeta(ByVal NumeroTarjeta As String)
    Dim Recset As New ADODB.Recordset()

    Dim Query As String

    ComTarjeta.PortOpen = False

```

```

        If Status = "Leer Tarjeta" Then
            Query = "SELECT numerotarjeta, bandera, numeroingreso,
codigoarea, ingenioenvia, codigoproducto, pesobrutoexpo, "
            Query = Query & " codigoproducto, codigotipoproducto,
codigobodega, codigoempaque, bodega_auxiliar, pesobrupoingenio,
pesotaraingenio"
            Query = Query & " FROM ingreso_azucar WHERE numerotarjeta =
'" & NumeroTarjeta & "'"

            Recset.Open(Query, Con1, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
            If Recset.EOF And Recset.BOF Then
                Recset.Close()
                Recset = New ADODB.Recordset()
                Query = "SELECT * FROM OP_DETALLE WHERE NUMEROTARJETA =
'" & NumeroTarjeta & "'"
                Recset.Open(Query, Con1,
ADODB.CursorTypeEnum.adOpenStatic, ADODB.LockTypeEnum.adLockReadOnly)
                If Not Recset.EOF Then
                    TipodeProducto = Otros
                    Bandera = Recset.Fields("bandera").Value
                    Select Case Bandera
                        Case 1, 3
                            txTarjeta.Text = NumeroTarjeta
                            txNumeroIngreso.Text =
Recset.Fields("DetalleID").Value
                            TiempoInicial = Now 'Para calcular cuanto
tarda el proceso de pesaje
                            AgregarLog(NumeroTarjeta & " - " &
txNumeroIngreso.Text & "(otros productos)")
                            Status = "Cerrar Entrada"
                            Recset.Close()
                        Case Else
                            Status = "Error Bandera"
                            Recset.Close()
                    End Select
                Else
                    AgregarLog("Tarjeta no registrada, no tiene
asociado numero de ingreso")
                    Status = "Error Tarjeta"
                    Recset.Close()
                End If

                Exit Sub
            Else
                TipodeProducto = Azucar
                Bandera = Recset.Fields("bandera").Value
                Select Case Bandera
                    Case 1, 3

```

```

        txTarjeta.Text = NumeroTarjeta
        txNumeroIngreso.Text =
Recset.Fields("NumeroIngreso").Value
        TiempoInicial = TimeOfDay 'Para calcular
cuanto tarda el proceso de pesaje
        AgregarLog(NumeroTarjeta & " - " &
txNumeroIngreso.Text)
        CodigoArea = Recset.Fields("CodigoArea").Value
        CodigoIngenio =
Recset.Fields("IngenioEnvia").Value
        CodigoProducto =
Recset.Fields("CodigoProducto").Value
        'PesoBrutoTemp =
Recset.Fields("pesobrutoexpo").Value

        CodigoTipoProducto =
Recset.Fields("CodigoTipoProducto").Value
        CodigoBodega =
Recset.Fields("CodigoBodega").Value
        CodigoEmpaque =
Recset.Fields("CodigoEmpaque").Value
        BodegaAuxiliar =
Recset.Fields("Bodega_Auxiliar").Value

        PesoBrutoIngenio =
Recset.Fields("PesoBrupoIngenio").Value
        PesoTaraIngenio =
Recset.Fields("PesoTaraIngenio").Value
        PesoBrutoTemp =
Recset.Fields("pesobrutoexpo").Value
        Status = "Cerrar Entrada"
        Recset.Close()
        Case Else
            Status = "Error Bandera"
            Recset.Close()
        End Select
    End If
End Sub

End If
End Sub

'Agregar a log ok, append a archivo y a log list, recibe parametro
el texto a escribir
Private Sub AgregarLog(ByVal texto As String)
    Dim temp As String
    If Log.Items.Count > 300 Then
        Log.Items.Clear()
    Else
        temp = Format(Now(), "HH:mm") & " " & texto
        Log.Items.Add(temp)
    End If
End Sub

```

```

        Log.SelectedIndex = Log.Items.Count - 1
    End If
    FileOpen(1, Application.StartupPath & "\" & Format(Now(),
"MMddyyyy") & ".txt", OpenMode.Append, OpenAccess.Write,
OpenShare.LockReadWrite, 50)
    Print(1, temp & Chr(13) & Chr(10))
    FileClose(1)
End Sub

Private Sub btnReiniciar_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs)

End Sub

'-----Errores -----
-----
Private Sub ErrorTarjeta()
    ComBascula.PortOpen = False
    AgregarLog("Error con tarjeta. Reinicie bascula para
continuar. Peso NO ALMACENADO");
    MsgBox("Error con tarjeta. Reinicie bascula para continuar.
Peso NO ALMACENADO")

End Sub

Private Sub ErrorPeso()
    AgregarLog("Error en peso bruto o tara, la diferencia es mayor
de 750kg")
    ComBascula.PortOpen = False
    MsgBox("Error en peso bruto o tara, la diferencia es menor de
750kg. Reinicie bascula para continuar. Peso NO ALMACENADO")
End Sub

Private Sub ErrorBandera()
    AgregarLog("Camion tiene registros de bruto y tara, error de
bandera")
    ComBascula.PortOpen = False
    MsgBox("Camion tiene registros de bruto y tara, error de
bandera. Reinicie bascula para continuar. Peso NO ALMACENADO")
End Sub

Private Sub ErrorDesconocido()
    ComBascula.PortOpen = False
    MsgBox("Problema de logistica. Bascula sin camion. Reinicie
bascula y vuelva a pesar. Peso NO ALMACENADO")
End Sub

'-----Talanqueras -----
-----

```

```

Private Sub ActualizarTalanqueras()
    Dim valor As Double
    valor = 0
    If TalanqueraE Then
        valor = valor + 1
        lblEntrada.Text = "Abierta"
        lblEntrada.ForeColor = lblEntrada.ForeColor.Green
    Else
        lblEntrada.ForeColor = lblEntrada.ForeColor.Red
        lblEntrada.Text = "Cerrada"
    End If

    If TalanqueraS Then
        valor = valor + 2
        lblSalida.Text = "Abierta"
        lblSalida.ForeColor = lblSalida.ForeColor.Green
    Else
        lblSalida.Text = "Cerrada"
        lblSalida.ForeColor = lblSalida.ForeColor.Red
    End If
    valor = Val(valor)
    D1PortWritePortUchar(&H378, valor)
End Sub

```

```

Private Sub AbrirEntrada()
    TalanqueraE = True
    TalanqueraS = False
    ActualizarTalanqueras()
    ComTarjeta.PortOpen = True
    Status = "Leer Tarjeta"
    AgregarLog("Listo para leer tarjeta")

```

End Sub

```

Private Sub CerrarEntrada()
    TalanqueraS = False
    TalanqueraE = False
    ActualizarTalanqueras()
    Status = "Tomar Pesos"
End Sub

```

'-----Inicializar y load -----  
-----

```

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    On Error GoTo 1
    Inicializar()
    Status = "Esperando 0"
    Con1 = New ADODB.Connection()

```

```

        Con1.Open("Expogranel", "system", "administrator")
        On Error GoTo 2
        ComBascula.PortOpen = True
        Exit Sub
1:
        MsgBox("Error de Inicializacion de base de datos, revise que la
conexion ODBC Expogranel exista")
        Exit Sub
2:
        MsgBox("Error de Comunicacion con la bascula")
        Exit Sub
End Sub

'Inicializar variables globales y estados
Private Sub Inicializar()
    NumeroTarjeta = ""
    txTarjeta.Text = ""
    txNumeroIngreso.Text = ""
    txPesoBruto.Text = ""
    Bandera = ""
    NombreBascula = System.Environment.MachineName
    Me.Text = NombreBascula & " ver. 5.0 (Otros Productos)"
    TalanqueraE = False
    TalanqueraS = False
    ActualizarTalanqueras()
    Indice = 0
    TresCeros = 0
    ListPesos.Items.Clear()
    TimerInterPesos.Enabled = False
    Reiniciando = False
End Sub

Private Sub frmMain_Close(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Closed
    TalanqueraE = False
    TalanqueraS = False
    ActualizarTalanqueras()
End Sub

Private Sub btnReiniciar_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnReiniciar.Click
    Reiniciar()
End Sub

Private Sub Reiniciar()
    AgregarLog("Bascula reiniciada a mano")
    If ComBascula.PortOpen = False Then ComBascula.PortOpen = True
    Inicializar()
    TalanqueraS = True
    TalanqueraE = True

```

```

        ActualizarTalanqueras()
        Status = "Esperando 0"
    End Sub
'-----Timers-----
-----
    Private Sub TimerInterPesos_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TimerInterPesos.Tick
        Dim i As Integer
        Dim Menor, Mayor As Double
        TimerInterPesos.Enabled = False
        Indice = Indice + 1
        If Indice = 13 Then
            'Debug
            Menor = Pesos(1)
            Mayor = Pesos(1)
            For i = 2 To 12
                If Val(Pesos(i)) < Menor Then Menor = Val(Pesos(i))
                If Val(Pesos(i)) > Mayor Then Mayor = Val(Pesos(i))
            Next
            'Debug
            'Menor = 10

            If Math.Abs(Mayor - Menor) < 30 Then
                If Menor <> 0 Then
                    AgregarLog("Pesos en rango adecuado")
                    Status = "Guardar Datos"
                Else
                    AgregarLog("Error de Logistica, bascula sin
camion")
                    Status = "Error Desconocido"
                End If
            Else
                Indice = 0
                AgregarLog("Pesos con mas de 30 kilos de diferencia")
                TimerInterPesos.Enabled = True
                ListPesos.Items.Clear()
            End If
            AgregarLog(Pesos(1) & "-" & Pesos(2) & "-" & Pesos(3) & "-"
& Pesos(4) & "-" & Pesos(5) & "-" & Pesos(6) & "-" & Pesos(7) & "-" &
Pesos(8) & "-" & Pesos(9) & "-" & Pesos(10) & "-" & Pesos(11) & "-" &
Pesos(12))
            Else
                Pesos(Indice) = txPeso.Text
                ListPesos.Items.Add(Indice & " - " & txPeso.Text)
                TimerInterPesos.Enabled = True

            End If

        End Sub

    End Sub

```

```
Private Sub Timer0_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer0.Tick
```

```
    If IsNumeric(txPeso.Text) Then
        If Val(txPeso.Text) = 0 Then
            TresCeros = TresCeros + 1
            If TresCeros >= 3 Then
                Timer0.Enabled = False
                If ReIniciando Then EscribirLog(Tipo)
                Status = "Esperando Camion"
                Inicializar()
                AgregarLog(Status)
                TresCeros = 0
            End If
        End If
    End If
End If
```

```
End Sub
```

```
'-----Com ports -----
```

```
'Lee el puerto de la tarjeta y parsea el numero, si encuentra un enter, la parsea
```

```
'Comunicacion de la bascula a la compu
```

```
Private Sub ComBascula_OnComm(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComBascula.OnComm
```

```
    Dim TempPeso As String
    Cuenta = Cuenta + 1
```

```
Mistring = Mistring & ComBascula.Input
```

```
'Text1.Text = Com.Input
```

```
If InStr(1, Mistring, Chr(13)) > 0 Then
```

```
    'Para tomar muestras dispensas
```

```
    If Cuenta > 25 Then
```

```
        Cuenta = 0
```

```
        'el peso
```

```
        TempPeso = Val(Mid(Mistring, 5, 6))
```

```
        txPeso.Text = TempPeso
```

```
        'Verifica estabilidad del peso
```

```
    End If
```

```
Mistring = ""
```

```
End If
```

```

    CheckStatus()
End Sub

```

```

'-----Base de datos-----

```

```

Private Sub GuardarDatos()
    'Insert a la base de datos
    'Chequeo de banderas
    ' insert
    Dim Query As String
    Dim crc As String
    Dim PesoBrutoTemp As Double

    'análisis de laboratorio
    If IsNumeric(PesoBrutoTemp) And TipodeProducto = Azucar Then
        AgregarAnálisis(PesoBrutoTemp - Val(txPeso.Text))
    End If
    'On Error Resume Next

    'peso bruto
    If Bandera = 1 Then
        Tipo = "BRUTO"

        AgregarLog("Guardando peso bruto: " & txPeso.Text)

        If TipodeProducto = Azucar Then
            Query = "UPDATE ingreso_azucar set pesobrutoexpo = '" &
txPeso.Text & "', fechabascuabruto = to_date('" & Format(Now, "dd-MM-
yyyy HH:mm") & "', 'DD-MM-YYYY HH24:Mi'), bandera = '3',
usuariobascuabruto = '" & NombreBascu & "' WHERE numerotarjeta = '"
& NumeroTarjeta & "'"
            Conl.Execute(Query)
        Else
            'Cuando es bruto de azucar, es tara de otros productos.
            Query = "UPDATE op_detalle set pesotara ='" &
txPeso.Text & "', bandera = '3' WHERE NUMEROTARJETA = '" &
NumeroTarjeta & "'"
            Conl.Execute(Query)
        End If

        AgregarLog("Peso almacenado, abriendo salida")
        ReIniciando = True
        Status = "Saliendo"

    ElseIf Bandera = 3 Then 'peso tara
        Tipo = "TARA"

        PesoBrutoTemp = Val(txPesoBruto.Text)
    End If
End Sub

```

```

        AgregarLog("Guardando peso tara: " & txPeso.Text)
        If TipodeProducto = Azucar Then
            Query = "UPDATE ingreso_azucar set pesotaraexpo = '" &
txPeso.Text & "', fechabasculatara = to_date('" & Format(Now, "dd-MM-
yyyy HH:mm") & "', 'dd-MM-yyyy HH24:Mi'), bandera = '10',
usuariobasculatara = '" & NombreBascula & "' WHERE numerotarjeta = '" &
NumeroTarjeta & "'"
            Else
                'Cuando es tara de azucar, es bruto de otros productos.
                Query = "UPDATE op_detalle set pesobruto ='" &
txPeso.Text & "', bandera = '10' WHERE NUMEROTARJETA = '" &
NumeroTarjeta & "'"
            End If
            Con1.Execute(Query)
            Con1.Execute("commit")
            AgregarLog("Peso almacenado, abriendo salida")
            ReIniciando = True
            Status = "Saliendo"
        Else
            Status = "Error Bandera"
        End If
    End Sub

```

End Sub

```

Private Sub EscribirLog(ByVal Tipo As String)
    Dim TiempoTotal As Long
    Dim Query As String
    Dim crc As String
    On Error GoTo 1

    TiempoTotal = DateDiff(DateInterval.Second, TiempoInicial, Now)
    AgregarLog("Proceso tomo un total de: " & TiempoTotal & "
segs")
    'El log con los valores que fueron validados como estables
    crc = Pesos(1) + 2 * Pesos(2) + Pesos(3) + 2 * Pesos(4) +
Pesos(5) + 2 * Pesos(6)
    Query = "INSERT INTO REC_LOGS_BASCULAS (codigoarea, bascula,
numeroingreso, p1, p2, p3"
    Query = Query & " , p4, p5, p6, p7, p8, p9, p10, p11, p12,
crc, fecha, tipo, tiempototal) "
    Query = Query & "VALUES ('1' , " & "'" & NombreBascula & "', '"
& txNumeroIngreso.Text
    Query = Query & "', '" & Pesos(1) & "', '" & Pesos(2) & "', '"
& Pesos(3) & "', '" & Pesos(4) & "', '" & Pesos(5) & "', '" & Pesos(6)
& "', '" & Pesos(7) & "', '" & Pesos(8) & "', '" & Pesos(9) & "', '"
& Pesos(10) & "', '" & Pesos(11) & "', '" & Pesos(12)
    Query = Query & "', '" & crc & "', to_date('" & Format(Now,
"dd-MM-yyyy HH:mm") & "', 'DD-MM-YYYY HH24:Mi'), '" & Tipo & "', '" &
TiempoTotal & "'"
    Con1.Execute(Query)

```

```

        Exit Sub
1:    AgregarLog("Error almacenando log de pesos y tiempo total")
        AgregarLog(Query)
        AgregarLog(Err.Description)
    End Sub

    Private Sub ComTarjeta_OnComm(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ComTarjeta.OnComm
        TalanqueraE = False
        ActualizarTalanqueras()
        NumeroTarjeta = NumeroTarjeta & ComTarjeta.Input
        If InStr(NumeroTarjeta, Chr(13)) <> 0 Then
            NumeroTarjeta = Microsoft.VisualBasic.Left(NumeroTarjeta,
Len(NumeroTarjeta) - 2)
            ParsearTarjeta(NumeroTarjeta)

        End If
    End Sub

    Private Sub MenuItem4_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MenuItem4.Click
        TalanqueraE = False
        TalanqueraS = False
        ActualizarTalanqueras()
        ComBascula.PortOpen = False
        AgregarLog("Bascula Cerrada a mano")
        MsgBox("Bascula Cerrada, debe reiniciar bascula para
continuar")
    End Sub

    Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem5.Click
        Reiniciar()
    End Sub

    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem2.Click
        Me.Close()
    End Sub

    'Para agregar analisis de laboratorio

    Private Sub AgregarAnalisis(ByVal PesoNeto As String)
        Dim rsAnalisis As New ADODB.Recordset()
        Dim rsCantidad As New ADODB.Recordset()
        Dim Query As String

```

```

Dim Cantidad, Acumulado As Double

Dim AnalisisID As Integer
On Error Resume Next
Query = "SELECT ACUMULADO, CONTADOR "
Query = Query & "FROM LAB_ACUMULADOS WHERE CODIGOAREA = " &
CodigoArea
Query = Query & " AND CODIGOINGENIO = " & CodigoIngenio
Query = Query & " AND CODIGOPRODUCTO = " & CodigoProducto
Query = Query & " AND CODIGOTIPOPRODUCTO = " &
CodigoTipoProducto
Query = Query & " AND CODIGOBODEGA = " & CodigoBodega
Query = Query & " AND BODEGA_AUXILIAR = " & BodegaAuxiliar
Query = Query & " AND CODIGOAREA = " & CodigoArea
If CodigoBodega > 2 Then
    Query = Query & " AND CODIGOEMPAQUE = " & CodigoEmpaque
End If

rsAnalisis.Open(Query, Con1,
ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)

If Not rsAnalisis.EOF Then
    rsCantidad = New ADODB.Recordset()
    Query = "SELECT * FROM LAB_CANTIDADES WHERE CODIGOINGENIO =
" & CodigoIngenio
    Query = Query & " AND CODIGOPRODUCTO = " & CodigoProducto &
" AND CODIGOTIPOPRODUCTO = " & CodigoTipoProducto
    rsCantidad.Open(Query, Con1,
ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
    If Not rsCantidad.EOF Then
        Cantidad = rsCantidad.Fields("CANTIDAD").Value
    Else
        Exit Sub
    End If
    Acumulado = rsAnalisis.Fields("ACUMULADO").Value

    If Cantidad <= Acumulado Then
        rsAnalisis.Fields("ACUMULADO").Value = PesoNeto
        CounterOverflow(rsAnalisis)
    Else
        rsAnalisis.Fields("ACUMULADO").Value = Acumulado +
PesoNeto
        rsAnalisis.Update()
    End If

Else

```

```

        Query = "INSERT INTO LAB_ACUMULADOS (CODIGOAREA,
CODIGOINGENIO, CODIGOPRODUCTO, "
        Query = Query & "CODIGOTIPOPRODUCTO, ACUMULADO,
CODIGOBODEGA, CODIGOEMPAQUE, BODEGA_AUXILIAR, CONTADOR) "
        Query = Query & "VALUES (" & CodigoArea & ", " &
CodigoIngenio & ", " & CodigoProducto & ", " &
        Query = Query & CodigoTipoProducto & ", " & PesoNeto & ", " &
CodigoBodega & ", " & CodigoEmpaque & ", " & BodegaAuxiliar & ", 1 )"
        Conl.Execute(Query)
    End If
End Sub

Private Sub CounterOverflow(ByVal rsAnalisis As ADODB.Recordset)
    Dim Fecha As String
    Dim Query As String
    Dim rsBodega As New ADODB.Recordset()
    Dim rsId As New ADODB.Recordset()
    Dim AnalisisId As Integer
    Dim TipoAnalisis As Integer
    Dim CodigoBodega As Integer
    On Error Resume Next
    'Fecha = Format(Now(), "yyyy-MM-dd H:mm") 'Mysql
    Fecha = " To_Date(' " & Format(Now(), "dd/MM/yyyy H:mm") & "' ,
'dd/mm/yyyy hh24:mi') " 'Oracle

    'Si va a un barco, no a recepcion
    If CodigoBodega > 5 Then
        TipoAnalisis = 3    'Tipo barco
    Else
        TipoAnalisis = 1    'Tipo recepcion
    End If

    Query = "SELECT BODEGAID FROM LAB_BODEGAS WHERE CODIGOINGENIO
=" & CodigoIngenio
    rsBodega.Open(Query, Conl, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
    If Not rsBodega.EOF Then
        CodigoBodega = rsBodega.Fields("BODEGAID").Value
    Else
        CodigoBodega = 0
    End If

    If rsAnalisis.Fields("CONTADOR").Value < 10 Then

        Query = "INSERT INTO LAB_ANALISIS (CODIGOINGENIO,
CODIGOPRODUCTO, "
        Query = Query & "FECHA, CODIGOTIPOPRODUCTO, TIPOANALISIS) "
        Query = Query & "VALUES (" & CodigoBodega & " , " &
CodigoProducto & " , " & Fecha & " , "

```

```

        Query = Query & CodigoTipoProducto & ", " & TipoAnalisis &
    )"
        Con1.Execute(Query)
        rsAnalisis.Fields("CONTADOR").Value =
rsAnalisis.Fields("CONTADOR").Value + 1
        rsAnalisis.Update()
        AgregarLog("Analisis Agregado para ingenio" &
CodigoIngenio)
        Else
            rsAnalisis.Fields("CONTADOR").Value = 1
            rsAnalisis.Update()
            Query = "INSERT INTO LAB_ANALISIS (CODIGOINGENIO,
CODIGOPRODUCTO, "
                Query = Query & "FECHA, CODIGOTIPOPRODUCTO, ANALISISDOBLE,
TIPOANALISIS ) "
                Query = Query & "VALUES (" & CodigoBodega & " , " &
CodigoProducto & " , " & Fecha & " , "
                Query = Query & CodigoTipoProducto & ", 1, " & TipoAnalisis
& " )"
            Con1.Execute(Query)
            rsId = New ADODB.Recordset()
            rsId.Open("SELECT MAX(ANALISISID) AS MAXIMO FROM
LAB_ANALISIS", Con1, ADODB.CursorTypeEnum.adOpenForwardOnly,
ADODB.LockTypeEnum.adLockReadOnly)
            If Not rsId.EOF Then
                AnalisisId = rsId.Fields("MAXIMO").Value
            End If
            rsId.Close()
            Query = "INSERT INTO LAB_ANALISIS_DOBLES (CODIGOINGENIO,
CODIGOPRODUCTO, "
                Query = Query & "FECHA, CODIGOTIPOPRODUCTO,
ANALISISASOCIADO, TIPOANALISIS ) "
                Query = Query & "VALUES (" & CodigoBodega & " , " &
CodigoProducto & " , " & Fecha & " , "
                Query = Query & CodigoTipoProducto & " , " & AnalisisId & " ,
" & TipoAnalisis & " )"
            Con1.Execute(Query)
            Con1.Execute(Query)
            AgregarLog("Analisis Dobles Agregados para ingenio" &
CodigoIngenio)
        End If

    End Sub
End Class

```

### C. Justificación Financiera

Como sección complementaria, se incluye el análisis de retorno de inversión del proyecto, como una ventaja adicional de la automatización. Para el cálculo de flujos de efectivo se tomaron como base los siguientes rubros:

Ahorro Mano de obra (con un aumento del 10% anual)
Desarrollo de Software
Equipos (se incluyen 4 basculas y obra civil)
Mantenimiento de Equipos

Se realizo el análisis para 5 anos, obteniendo los siguientes resultados.

TIR	68%
Periodo de repago	1 año y 3 meses

Con una Tasa Interna de Retorno del 68%, el proyecto se convierte muy atractivo para el inversionista además que el proyecto se paga en un corto plazo. Respecto al mantenimiento de los equipos se puede concluir que este costo es quince veces menor que el pago de salarios por turnos, esto lo hace un proyecto rentable además de todos los beneficios que conlleva.