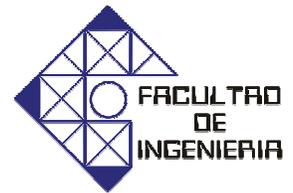




UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA MECÁNICA



## DESARROLLO DEL SISTEMA DE MEDICIÓN Y CONTROL DE UN TRIBÓMETRO TIPO BLOQUE EN ANILLO

Gloriet, Carlos.

Naguanagua, 22 de Junio del 2012



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA MECÁNICA



## DESARROLLO DEL SISTEMA DE MEDICIÓN Y CONTROL DE UN TRIBÓMETRO TIPO BLOQUE EN ANILLO

*Trabajo especial de grado presentado ante la Ilustre Universidad de Carabobo para  
optar al título de Ingeniero Mecánico*

**Tutor académico:**  
Jorge Romero. *MSc, Ing*

**Autor:**  
Gloriet A. Carlos L.

Naguanagua, 22 de Junio del 2012

## **SINOPSIS DEL PROYECTO DE GRADO**

### **TITULO DEL TRABAJO**

Desarrollo del sistema de medición y control de un Tribómetro tipo Bloque en Anillo.

**FECHA DE INICIO DEL TRABAJO:** \_\_\_\_\_

**FECHA DE EX FINAL:** \_\_\_\_\_

**FECHA DE ENTREGA DE EJEMPLARES DEFINITIVOS A DIRECCIÓN DE ESCUELA:** \_\_\_\_\_

### **SINOPSIS:**

Con el sistema automatizado propuesto se pretende incrementar los niveles de eficiencia debido a que no existirían las deficiencias existentes en el banco de ensayo y también se hallarían mejoras significativas en las condiciones de trabajo del estudiante, ya que el mismo no estará sometido a factores que disminuyan su eficiencia y que podrían afectar el aprendizaje del mismo.

El trabajo incluye los planos del circuito electrónico, la descripción del programa, los cálculos y la selección de todos los componentes del sistema.

Como principales conclusiones se tiene que con el sistema automatizado, se implementa tecnología de punta, sustituyendo el sistema manual con alto orden de parada, retardos, inseguridad y retrabajo, por otra parte, se incrementará la confiabilidad y se disminuirán los problemas de errores de apreciación humana en las mediciones.

**CÉDULA DE IDENTIDAD**

**ESTUDIANTES**

**FIRMA**

12.929.938

Gloriet, Carlos

\_\_\_\_\_

**FIRMA**

**PROFESOR GUÍA:** Ing. Jorge, Romero.

\_\_\_\_\_

**JURADO:** \_\_\_\_\_

\_\_\_\_\_

**JURADO:** \_\_\_\_\_

\_\_\_\_\_

# *Agradecimientos*

---

---

A Dios, ante todas las cosas, porque ha sido, es y será mi guía.

A mi Madre, ser único, con todo el amor del mundo, esto es para ti.

A ese ser tan querida (mi esposa), que me otorga su amor y una nueva fase de mi vida.

A mis familiares, por todo el apoyo brindado a lo largo de este trabajo y de mi vida.

A mis amigos(as), que han contribuido con su apoyo de diferentes maneras.

A la Facultad de Ingeniería Mecánica de la Universidad de Carabobo, por darme la oportunidad de realizar mis metas académicas.

A mi tutor académico Prof. Jorge Romero, por sus orientaciones y dedicación para conmigo.

A todos ustedes un eterno agradecimiento.

Carlos L. Gloriet A.

# *Agradecimientos*

---

---

A Dios, ante todas las cosas, porque ha sido, es y será mi guía.

A mi Madre, ser único, con todo el amor del mundo, esto es para ti.

A ese ser tan querida (mi esposa), que me otorga su amor y una nueva fase de mi vida.

A mis familiares, por todo el apoyo brindado a lo largo de este trabajo y de mi vida.

A mis amigos(as), que han contribuido con su apoyo de diferentes maneras.

A la Facultad de Ingeniería Mecánica de la Universidad de Carabobo, por darme la oportunidad de realizar mis metas académicas.

A mi tutor académico Prof. Jorge Romero, por sus orientaciones y dedicación para conmigo.

A todos ustedes un eterno agradecimiento.

Carlos L. Gloriet A.

# *Resumen*

---

---

Con el sistema automatizado propuesto se pretende incrementar los niveles de eficiencia debido a que no existirían las deficiencias existentes en el banco de ensayo y también se hallarían mejoras significativas en las condiciones de trabajo del estudiante, ya que el mismo no estará sometido a factores que disminuyan su eficiencia y que podrían afectar el aprendizaje del mismo.

El trabajo incluye los planos del circuito electrónico, la descripción del programa, los cálculos y la selección de todos los componentes del sistema.

Como principales conclusiones se tiene que con el sistema automatizado, se implementa tecnología de punta, sustituyendo el sistema manual con alto orden de parada, retardos, inseguridad y retrabajo, por otra parte, se incrementará la confiabilidad y se disminuirán los problemas de errores de apreciación humana en las mediciones

# *Índice general*

---

---

<b>ÍNDICE GENERAL</b>	<b>i</b>
<b>ÍNDICE DE FIGURAS</b>	<b>v</b>
<b>ÍNDICE DE TABLAS</b>	<b>vii</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>CAPÍTULO I El Problema</b>	
1.1- PLANTEAMIENTO DEL PROBLEMA	3
1.2- OBJETIVOS	5
1.2.1- OBJETIVO GENERAL	5
1.2.2- OBJETIVOS ESPECÍFICOS	5
1.3- JUSTIFICACIÓN	5
1.4- LIMITACIONES	6
1.5- ALCANCE	7
<b>CAPÍTULO II Marco Teórico</b>	
2.1- ANTECEDENTES	8
2.2- BASE TEÓRICA	9
2.2.1- TRIBOLOGÍA	9
2.2.2- AUTOMATIZACIÓN	10
2.2.2.1- TIPOS DE AUTOMATIZACIÓN	10
2.2.2.2- TÉCNICAS DE CONTROL	11
- TÉCNICAS ANALÓGICAS	11
- TÉCNICAS DIGITALES	11
2.2.3- SISTEMAS DE ADQUISICIÓN DE DATOS (SAD)	12
2.2.3.1- TRANSDUCTORES	13
2.2.3.2- REGISTRADOR DE DATOS	14
2.2.3.3- FUNCIONAMIENTO DE UN REGISTRADOR DE DATOS	15
2.2.3.4- TOPOLOGIA DE UN REGISTRADOR DE DATOS	17
2.2.3.5- TIPOS DE SOFTWARE REGISTRADOR DE DATOS	19

2.2.4- COMUNICACIÓN USB	19
2.2.5- MICROCONTROLADORES PIC	20
2.2.5.1- SISTEMA DE INTERRUPCIONES DEL PIC	22
2.2.6- INTRODUCCIÓN DE SEÑALES	22
2.2.6.1- SENSORES	22
2.2.6.2- CLASIFICACIÓN DE LOS SENSORES SEGÚN EL TIPO DE SEÑAL DE SALIDA	24
2.2.6.3- CLASIFICACIÓN DE LOS SENSORES SEGÚN EL TIPO DE ALIMENTACIÓN	25
2.2.7- LENGUAJE DE PROGRAMACIÓN C#	25
2.2.7.1- CARACTERÍSTICAS DE C#	26
<b>CAPÍTULO III Marco Metodológico</b>	
3.1- NIVEL Y TIPO DE LA INVESTIGACIÓN	29
3.2- DISEÑO DE LA INVESTIGACIÓN	29
3.3-TÉCNICAS DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS	31
<b>CAPÍTULO IV Desarrollo del Trabajo</b>	
4.1- EVALUACIÓN DEL SISTEMA EXISTENTE	33
4.1.1 OBTENCIÓN DE DATOS MEDIANTE ENSAYOS DE FRICCIÓN Y DESGASTE	35
4.1.1.1- CURVAS DE CALIBRACIÓN	36
4.1.1.1.1 CONSTRUCCIÓN DE LA CURVA DE CALIBRACIÓN DE LA VIGA	36
4.1.1.1.2- CONSTRUCCIÓN DE LA CURVA DE PAR TORSOR SIN CARGA	37
4.1.1.1.3- CÁLCULO MANUAL DEL COEFICIENTE DE FRICCIÓN	39
4.1.1.1.4 MEDICIÓN DEL DESGASTE	40
4.2- DETERMINACIÓN DE LA CAUSA RAIZ	41
4.3- DEFINICIÓN DE PROPUESTA Y PLAN DE ACCIÓN	44
4.4- DESCRIPCIÓN DEL SISTEMA DISEÑADO	46
4.4.1- LISTA DE COMPONENTES DEL CIRCUITO ELECTRÓNICO	47
4.4.2- DESCRIPCIÓN DEL CIRCUITO ELECTRÓNICO	49

4.4.3- REQUERIMIENTOS MÍNIMOS DEL COMPUTADOR	50
4.5- MEDICIÓN DEL PAR TORSOR	51
4.5.1- CELDA DE CARGA (J1)	51
4.6- MEDICIÓN DE LA VELOCIDAD ANGULAR DEL SISTEMA	52
4.6.1 SWITCH ÓPTICO (U5)	52
4.7- STBA PROGRAMA PARA LA CAPTURA DE DATOS DEL ENSAYO DEL TRIBÓMETRO BLOQUE EN ANILLO	54
4.7.1- TÉCNICAS DE PROGRAMACIÓN SELECCIONADAS	55
4.7.2- LENGUAJE DE PROGRAMACIÓN SELECCIONADO	55
4.7.3- ALGORITMO DE PROGRAMACIÓN	56
4.7.3.1- Algoritmo principal para la calibración del ensayo	57
4.7.3.2- Algoritmo principal para la captura de datos del ensayo	59
4.7.4- DESCRIPCIÓN DEL ENTORNO DEL PROGRAMA	60
4.7.4.1- DESCRIPCIÓN DEL PANEL DE CONTROL DEL ENSAYO	60
4.7.4.2- DESCRIPCIÓN DEL PANEL DE RESULTADOS Y VARIABLES	62
4.7.4.3- INICIO DEL PROGRAMA	65
4.7.5- DESCRIPCIÓN DEL PANEL VISUALIZACIÓN	66
4.7.5.1- DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (DIAGRAMA)	66
4.7.5.2- DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (GRÁFICA)	67
4.7.5.3- DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (TABLAS)	68
4.7.5.4- DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (CALIBRACIÓN)	69
4.7.6- OTRAS TAREAS REALIZABLES CON EL PROGRAMA	71
4.8 VALIDACIÓN DE LA HERRAMIENTA COMPUTACIONAL	73
CONCLUSIONES	79
RECOMENDACIONES	80
BIBLIOGRAFÍA	81

## ANEXOS

### A- CÓDIGO DE FUNCIONES DEL STBA

A.1- Función del botón captura	84
A.2- Función del botón play	86
A.3- Función del botón pausa	87
A.4- Función del botón stop	88

A.5- Función para dibujar la gráfica del ensayo con carga	89
A.6- Función para dibujar la gráfica de calibración	91
A.7- Función para dibujar diagrama	93
A.8- Función del botón Inicio/Fin Calibración	93
A.9- Función recepción de datos del circuito	94
<b>B- ESQUEMAS, FIGURAS E IMÁGENES</b>	
B.1- Esquema electrónico del Microcontrolador PIC16F690 en el sistema	96
B.2- Esquema electrónico del Microcontrolador PIC18F2550 y Regulador de Voltaje 3V LM317 en el sistema	97
B.3- Reloj comparador Mitutoyo	98
B.4- Ayuda Tribómetro	99
B.5- Resultados de fricción (Medina y Peralta, 2010)	100

# *Índice de Figuras*

---

---

Figura 2.1. El acondicionamiento de señales es un componente importante en un sistema de adquisición de datos.	12
Figura 2.2 Los registradores de datos pueden ser dispositivos basados en PC o autónomos.	15
Figura 2.3 Funcionamiento de un registrador de datos	15
Figura 2.4 Elementos para Aplicaciones de Registro de Datos	16
Figura 2.5 Registradores de Datos Portátiles, Autónomos y de Bajo Costo	18
Figura 2.6 Sistema de Registro de Datos Embebidos Autónomos	18
Figura 2.7 Registradores de Datos Programables Basados en PC	18
Figura 2.8 Microcontrolador.	21
Figura 2.9 Sensor inductivo	23
Figura 2.10 Sensor óptico	24
Figura 3.1 Diagrama de diseño de la investigación	32
Figura 4.1 Tribómetro Bloque en Anillo	34
Figura 4.2 Tribómetro Bloque en Anillo (Vista de Planta)	35
Figura 4.3 Curva de calibración de la viga	37
Figura 4.4 Curva típica de ensayo sin carga.	38
Figura 4.5 Diagrama de fuerzas	39
Figura 4.6 Diagrama causa-efecto	43
Figura 4.7 Circuito Electrónico	46
Figura 4.8 Celda de Carga Instalada en el banco de ensayo	52
Figura 4.9 Interrupción del haz de luz con un disco en el eje del motor	53
Figura 4.10 Switch óptico instalado en el banco de ensayo	54
Figura 4.11 Captura del panel de control del ensayo	60
Figura 4.12 Captura de la interface Variables y Resultados	62
Figura 4.13 Captura de la ventana de inicio del STBA	66
Figura 4.14 Captura de la interface de Pestaña (Diagrama)	67

Figura 4.15 Captura de la interface de Pestaña (Gráficas)	68
Figura 4.16 Captura de la interface de Pestaña (Tablas)	69
Figura 4.17 Botones de calibración	70
Figura 4.18 Captura de la interface de Pestaña (Calibración)	71
Figura 4.19 Despliegue del menú Archivo	72
Figura 4.20 Despliegue del menú Ayuda	73
Figura 4.21. Arreglo ensayo del Tribómetro	74
Figura 4.22. Especímenes de ensayo (Medina y Peralta, 2010)	76
Figura 4.23 Captura pestaña calibración del ensayo realizado	76
Figura 4.24 Captura pestaña tabla del ensayo realizado	77
Figura 4.25 Captura pestaña gráfica del ensayo realizado	78

# *Índice de Tablas*

---

---

Tabla 2.1. Características Eléctricas y Requerimientos Básicos de Acondicionamiento de Señales para los Transductores más Comunes	13
Tabla 4.1 Datos típicos de calibración	36
Tabla 4.2 Datos típicos de ensayo sin carga	38
Tabla 4.3 Componentes del circuito electrónico	47
Tabla 4.4. Sistema Tribológico	75

# *Introducción*

---

---

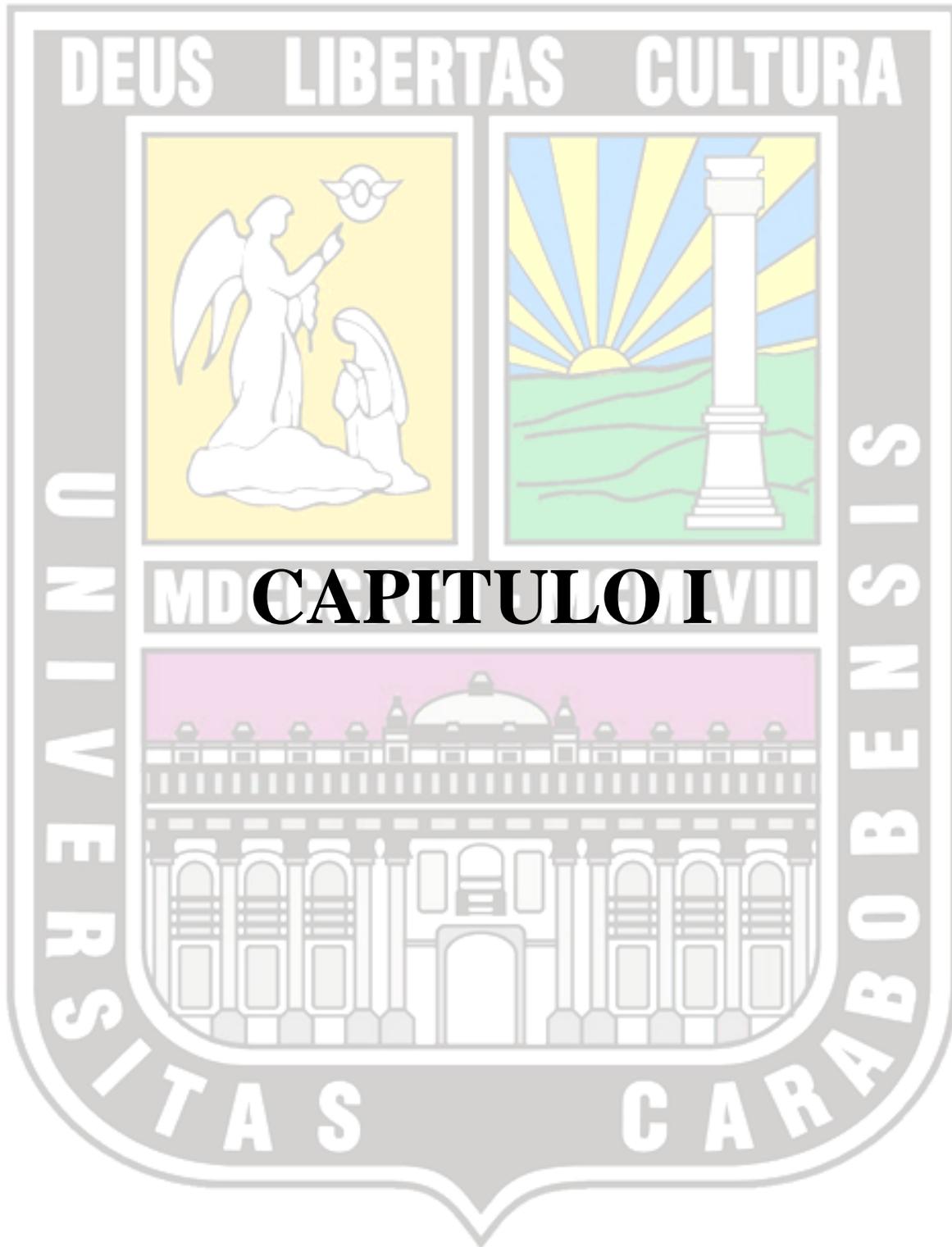
La automatización como una disciplina de la ingeniería es más amplia que un simple sistema de control, abarca la instrumentación industrial, que incluye los sensores y transmisores de campo, los sistemas de control y supervisión, los sistemas de transmisión y recolección de datos y las aplicaciones de software en tiempo real para supervisar y controlar las operaciones de plantas o procesos industriales.

En el Laboratorio de Predicciones de la escuela de Ingeniería Mecánica de la Universidad de Carabobo, se realizan pruebas de simulación en condiciones controladas de fricción y el desgaste de materiales específicos, por medio de un banco de ensayo de tipo bloque en anillo denominado tribómetro. Dichas pruebas se realizan de manera manual, generando pérdida de tiempo para el estudiante, errores de apreciación en la medición y cálculo, entre otros factores.

Dada la situación expuesta anteriormente, resulta conveniente desarrollar una herramienta para el diseño automatizado para la adquisición de datos del tribómetro tipo bloque en anillo; integrada en una herramienta computacional, la cual, permita agilizar de una manera considerable todo el proceso de adquisición de datos del equipo, ofreciendo resultados precisos y confiables en periodos de tiempo mucho más cortos que si se llevara a cabo de manera manual.

El diseño se lleva a cabo en varias etapas que vienen dadas por los requerimientos sistemáticos de la metodología adoptada, y se divide en:

- **Capítulo I:** suministra la información necesaria para conocer el problema planteado, se establecen los objetivos de la investigación, así como las limitaciones y delimitaciones del diseño.
- **Capítulo II:** comprende los lineamientos teóricos y antecedentes que pueden ayudar a resolver el problema planteado y sirven de sustento al diseño del sistema.
- **Capítulo III:** establece el tipo de investigación y los pasos metodológicos seguidos para realizar el proyecto.
- **Capítulo IV:** se desarrolla lo que es el diseño del sistema automatizado, el programa computacional, se seleccionan cada uno de los dispositivos del mismo y se comprueba el funcionamiento del sistema.
- Finalmente se plantean las conclusiones y recomendaciones obtenidas a lo largo del desarrollo de este trabajo.



**CAPITULO I**

# CAPÍTULO 1

## *El Problema*

---

---

### **1.1- PLANTEAMIENTO DEL PROBLEMA**

El Laboratorio de Predicciones se encuentra ubicado en la escuela de Ingeniería Mecánica de la Universidad de Carabobo en Naguanagua Edo. Carabobo, está dirigido a estudiantes de pregrado y postgrado, y su principal objetivo es predecir confiablemente eventos a futuro sobre objetos, personas y el ambiente. Entre las investigaciones más relevantes está el estudio de la fricción y el desgaste mediante ensayos realizados en un tribómetro de tipo Bloque en Anillo.

El tribómetro, es un mecanismo mediante el cual se puede determinar la resistencia al desgaste de un material cuando se pone en contacto con otro y existe movimiento relativo entre ellos, en un determinado medio y así determinar las sollicitaciones más adecuadas para prolongar la vida en servicio del mismo o para establecer su período de utilidad bajo las condiciones preestablecidas. La finalidad de utilizar esta metodología de ensayo, es obtener datos de diferentes materiales y analizar la afinidad de parejas de materiales, que puedan estar sometidos a las condiciones de desgaste adecuadas.

Con los ensayos, se busca analizar el comportamiento ante la fricción y el desgaste, y representar de una manera más precisa y controlada los mecanismos de

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

operación de los componentes mecánicos, midiendo parámetros como temperatura cercana a la zona de contacto, velocidad de deslizamiento, tiempo de ensayo y carga de aplicación entre las superficies en contacto.

Actualmente el proceso de adquisición de datos en los ensayos se realiza de manera manual, tomando de forma directa e indirecta los parámetros ya mencionados. Tal como se puede observar éste proceso manual compromete la confiabilidad de la toma de datos del ensayo debido a que:

1. Para obtener la carga aplicada, se mide la deflexión hecha por una barra metálica adjunto al motor, por lo que existe la posibilidad de errores de apreciación hechas por el encargado de la toma de datos.
2. El tiempo del ensayo se calcula con un cronómetro de mano, por lo que existe también la posibilidad de errores de apreciación hechas por el encargado de la toma de datos.
3. Las revoluciones por minutos realizadas por el motor no son contabilizadas.

Por lo antes expuesto se requiere automatizar la adquisición de datos del equipo en objeto de estudio.

## **1.2 OBJETIVOS**

### **1.2.1 OBJETIVO GENERAL**

Desarrollar sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo.

### **1.2.2 OBJETIVOS ESPECÍFICOS**

1. Comprender el proceso actual del Tribómetro tipo Bloque en Anillo disponible en el Laboratorio de Predicciones a fin de determinar los parámetros a medir.
2. Definir los componentes para la adquisición de datos en el Tribómetro tipo Bloque en Anillo.
3. Instalar los componentes escogidos para el tribómetro en estudio.
4. Diseñar el programa de control de usuario para la adquisición de datos para el Tribómetro tipo Bloque en Anillo disponible en el laboratorio de predicciones.
5. Implementar el sistema diseñado en el Tribómetro tipo Bloque en Anillo disponible en el Laboratorio de Predicciones.

## **1.3 JUSTIFICACIÓN**

Actualmente la recopilación de datos del tribómetro en el laboratorio de predicciones que está ubicado en la escuela de Ingeniería Mecánica en la Universidad de Carabobo se realiza de manera manual, debido a esto se requiere de mucho tiempo para ejecutar varios ensayos, tomando en consideración otras variables a medir como son la velocidad del motor, el tiempo que demora la captura de datos, entre otros. Lo que conlleva a una serie de consecuencias que se observan notablemente, como

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

pérdida de tiempo, material utilizado y recursos, así como también errores de apreciación visual, lo cual se traduce en importantes pérdidas económicas y de tiempo.

Con el sistema automatizado propuesto se pretende incrementar los niveles de eficiencia debido a que no existirían las deficiencias anteriormente planteadas y también se hallarían mejoras significativas en las condiciones de trabajo del estudiante, ya que el mismo no estará sometido a factores que disminuyan su eficiencia y que podrían afectar el ensayo, algunas de estas mejoras serían:

- ✓ Visualización en tiempo real del ensayo a ejecutar.
- ✓ Fácil manejo del ensayo por medio de la interface de programación.
- ✓ Fácil adquisición de datos del ensayo a ejecutar con alta precisión y exactitud.

Todo lo antes expuesto muestra las ventajas que traería la automatización del sistema de medición realizada por el tribómetro.

#### **1.4 LIMITACIONES**

- Disponibilidad de Información bibliográfica sobre investigaciones similares.

## **1.5 ALCANCE**

- Este proyecto contemplará el desarrollo e implementación de medición, control y transmisión de datos para el Tribómetro tipo Bloque en Anillo disponible en el laboratorio de predicciones.
- El desarrollo del sistema se basará sólo en el tribómetro tipo Bloque en Anillo disponible en el Laboratorio de Predicciones de la Escuela de Mecánica de la Facultad de Ingeniería.
- El proyecto se desarrollará con recursos, materiales y disponibilidad de planta física que puedan lograrse en el centro de investigación de materiales de la facultad de ingeniería, conjuntamente con el laboratorio de predicciones adscrito al departamento de diseño y automatización de la escuela de ingeniería mecánica con factible financiamiento del CDCH y FORACIT u otros.

# CAPÍTULO 2

## *Marco Teórico*

---

---

### **2.1 ANTECEDENTES**

Calderón y González, (2002), diseñaron un programa de computación para simulación de los procesos de taladrado, fresado, limado y rectificado. Su objetivo principal fue estudiar los procesos metalmecánicos más resaltantes y las variables fundamentales, para luego, desarrollar el software de control. La investigación permitió concluir que por medio de la comparación de los resultados obtenidos se validó el funcionamiento del programa, mostrando, que la desviación de los valores obtenidos se encuentra dentro del rango permitido.

Flores y Gonzales, (2003), automatizaron el proceso de medición y control de diámetros en tubos, para el control estadístico de procesos en tiempo real. Su objetivo fundamental fue estudiar la variabilidad del proceso de medición de los distintos diámetros desarrollados en el proceso que definen la población del sistema diseñado. La investigación permitió concluir que con el sistema automatizado, se implementa tecnología de punta, sustituyendo el sistema manual con alto orden de parada, retardos, inseguridad y retrabajo, por otra parte, se incrementaría la confiabilidad y no se generarían problemas de salud ocupacional.

Hernández y Tovar, (2008), diseñaron un sistema automatizado para un banco de trefilado de barras acero; cuyo diseño se basó en la realización de toda la estructura física (por medio de elementos de máquinas y elementos finitos) y el sistema automatizado (a través de lógica cableada y controlador lógico programable), debido a que su función principal es mantener un ciclo continuo y progresivo tal que al terminar el trefilado de cada barra se pueda iniciar nuevamente el proceso.

Trejo, (2009), diseñó un sistema para la medición y control remoto de variables utilizadas en procesos industriales. El objetivo principal fue establecer las variables principales que intervienen directamente en los procesos industriales para luego diseñar la interface de usuario. Finalmente, este estudio permitió concluir que las variables seleccionadas en los procesos industriales cumplen con la funcionalidad del sistema, realizando una selección de parámetros más resaltantes en los procesos involucrados.

## **2.2 BASE TEÓRICA**

### **2.2.1 TRIBOLOGIA**

La tribología es la ciencia y la tecnología que involucran las superficies interactuantes en movimiento relativo y todos los tópicos y prácticas conexas. Comprende el estudio de la fricción, el desgaste y lubricación con el propósito de entender en detalle la interacción del contacto entre las superficies y prescribir mejoras a una aplicación determinada.

El término tribología fue empleado por primera vez a mediados de 1960's en Inglaterra por Peter Jost quien liderizaba un comité del Ministerio Británico de Educación y Ciencia. Este comité generó en el año 1966 un reporte que destacaba sobre los grandes ahorros que pudieran lograrse por la utilización de mejoras en los procedimientos de diseño y lubricación. (Lubricar, 2007).

## **2.2.2 AUTOMATIZACIÓN**

Según Vallejo y Vallejo (2006), la automatización se ha entendido como una tecnología en la cual se aplican los sistemas mecánicos, electrónicos y computarizados, con el fin de operar y controlar la producción, de bienes físicos de consumo, además, involucra una gran variedad de sistemas y procesos que se ejecutan con mínima o ninguna intervención del ser humano.

### **2.2.2.1 TIPOS DE AUTOMATIZACIÓN**

Según Gonzales, (2004), se pueden encontrar los siguientes tipos:

- **Automatización fija:** se utiliza cuando el volumen de producción es muy alto y, por tanto, se puede justificar económicamente el alto costo del diseño de equipo especializado para procesar el producto, con un rendimiento alto y tasas de producción elevadas. Un ejemplo típico puede ser la fabricación de automóviles. Un inconveniente de la automatización fija es que su ciclo de vida depende de la vigencia del producto en el mercado.
- **Automatización programable:** se emplea cuando el volumen de producción es relativamente bajo y hay una diversidad de productos a obtener. En este caso el equipo de producción es diseñado para adaptarse a las variaciones de configuración del producto y ésta adaptación se realiza por medio de software. Un ejemplo podría ser la fabricación de diferentes tipos de tornillos bajo pedido.
- **Automatización flexible:** es más adecuada para un rango de producción medio. Los sistemas flexibles poseen características de la automatización fija y de la automatización programada. Suelen estar constituidos por una serie de estaciones de

trabajo interconectadas entre sí por sistemas de almacenamiento y manipulación de materiales, controlados en su conjunto por una computadora.

- **Automatización total:** es el escalón final en el que, idealmente, la fabricación se realizaría sin intervención humana.

### **2.2.2.2 TÉCNICAS DE CONTROL**

#### **- TÉCNICAS ANALÓGICAS**

Son el método más antiguo y dio lugar a las técnicas de control clásicas. El proceso analógico de señales puede ser mecánico, neumático, hidráulico, eléctrico, electrónico y óptico. El componente fundamental que permitió el desarrollo del control analógico fue el amplificador electrónico, inventado en la década de los 50. La aplicación principal de las técnicas analógicas es la realización de controladores de procesos continuos industriales: mecanismos y máquinas movidos por motores eléctricos, procesos con fluidos, hornos, entre otros. (Gonzales, 2004)

#### **- TÉCNICAS DIGITALES**

Estas se utilizan para controlar procesos continuos. Los controladores analógicos, y en particular, el Proporcional Integral Derivativo (PID), hoy día se realizan y comercializan en su versión digital. El campo de aplicación es el mismo pero las prestaciones de los digitales son muy superiores a las de los analógicos. Resulta más fácil sintonizarlos y están preparados para poder ser operados a distancia a través de buses de comunicación. (Gonzales, 2004)

### **2.2.3 SISTEMAS DE ADQUISICIÓN DE DATOS (SAD)**

Los sistemas de adquisición de datos (SAD) basados en PC son usados en un amplio rango de aplicaciones en los laboratorios, en el campo y en el piso de una planta de manufactura. Típicamente, los dispositivos SAD son instrumentos de propósito general diseñados para medir señales de voltaje. La mayoría de los sensores y transductores generan señales que debe acondicionar antes de que un dispositivo SAD pueda adquirir con precisión la señal. Este proceso, conocido como acondicionamiento de señal, incluye funciones como amplificación, filtrado, aislamiento eléctrico y multiplexaje. Además, existen otros sensores que requieren de excitación de voltaje o corriente, completar una configuración de puente, linearización o amplificación para que puedan operar de manera correcta. (National Instruments, 2012)

La mayoría de los sistemas SAD basados en PC incluyen algún tipo de acondicionamiento de señal además del dispositivo SAD y la PC, como lo muestra la Figura 2,1.

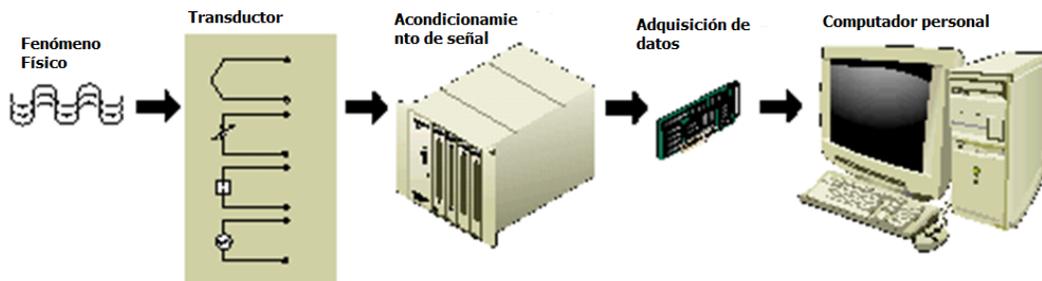


Figura 2.1. El acondicionamiento de señales es un componente importante en un sistema de adquisición de datos. (National Instruments, 2012)

### 2.2.3.1 TRANSDUCTORES

Los transductores son dispositivos que convierten un fenómeno físico, como temperatura, carga, presión o luz a otro. Los transductores más comunes convierten estas características físicas en señales eléctricas como voltaje y resistencia. Las características de los transductores definen muchos de los requerimientos del acondicionamiento de señales necesario en un sistema SAD. La Tabla 2,1 muestra un resumen de las características básicas y el acondicionamiento requerido para los transductores más comunes. (National Instruments, 2012).

Tabla 2.1. Características eléctricas y requerimientos básicos de acondicionamiento de señales para los transductores más comunes (National Instruments, 2012)

<b>Sensor</b>	<b>Características Eléctricas</b>	<b>Requerimientos de Acondicionamiento de Señales</b>
Termopar	Salida de bajo voltaje Baja sensibilidad Salida no lineal	Sensor de temperatura de referencia (para compensación de unión fría) Alta amplificación Linealización
RTD <small>(del inglés: resistance temperature detector)</small> detector de temperatura resistivo	Baja resistencia (típicamente 100 ohms) Baja sensibilidad Salida no lineal	Excitación de corriente Configuración de 3 o 4 cables Linealización
Galga extensiométrica	Baja resistencia Baja sensibilidad Salida no lineal	Excitación de voltaje o corriente Alta amplificación Configuración de puente Linealización Calibración

Tabla 2.1. Características eléctricas y requerimientos básicos de acondicionamiento de señales para los transductores más comunes (Continuación)

(National Instruments, 2012)

Salida de corriente	Salida de 4 a 20 mA	Resistencia de precisión
Termistor	Dispositivo resistivo Alta resistencia y sensibilidad Salida no lineal	Excitación de voltaje o corriente con resistencia de referencia Linealización
Acelerómetros activos	Alto nivel de salida de voltaje o corriente Salida lineal	Alimentación de energía Amplificación moderada

### **2.2.3.2 REGISTRADOR DE DATOS**

Registrar y reproducir datos es una aplicación común de medición. En su forma más básica, el registro de datos es la medida y reproducción de parámetros físicos o eléctricos en un periodo de tiempo, (ver figura 2.2). Los datos pueden ser temperatura, tensión, desplazamiento, flujo, presión, voltaje, corriente, resistencia, potencia y muchos otros parámetros. Se puede clasificar una amplia variedad de productos como registradores de datos, desde dispositivos básicos que realizan una sola medida hasta dispositivos más complejos que ofrecen funciones de análisis y pantallas integradas. Muchas aplicaciones se involucran más que en adquirir o reproducir señales, algunas veces involucran una combinación de análisis, análisis en línea, visualización, generación de reportes y compartir datos. Además, las aplicaciones están comenzando a requerir la adquisición y el almacenamiento de otro tipo de datos, como reproducir sonido y video en conjunto con los otros parámetros medidos durante una prueba de choque de automóvil. (National Instruments, 2012)



Figura 2.2 Los registradores de datos pueden ser dispositivos basados en PC o autónomos.

### 2.2.3.3 FUNCIONAMIENTO DE UN REGISTRADOR DE DATOS

Un registrador de datos funciona con sensores para convertir fenómenos físicos y estímulos en señales electrónicas como voltaje o corriente. Estas señales electrónicas son convertidas o digitalizadas en datos binarios. El dato binario es analizado fácilmente por programas y almacenado en un disco duro de PC o en otro medio de almacenamiento como tarjetas de memoria y CDs. Ver figura 2.3. (National Instruments, 2012).

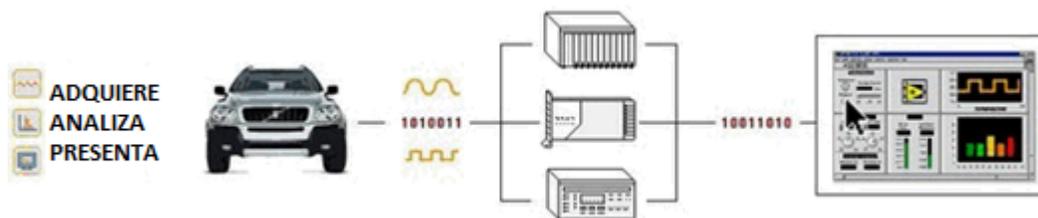


Figura 2.3 Funcionamiento de un registrador de datos. (National Instruments, 2012)

Unos cuantos componentes que cada registrador de datos deben incluir, (National Instruments, 2012):

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

- Hardware para digitalizar lo que se está registrando, incluyendo sensores, acondicionamiento de señales y hardware para conversión analógico a digital.
- Almacenamiento de datos a largo plazo, generalmente memoria interna o una PC.
- Software de registro de datos para adquisición de datos, análisis y presentación.

La habilidad de realizar medidas de sensores y almacenar los datos para usarlos en un futuro es por definición, una característica de un registrador de datos. Sin embargo, una aplicación para registro de datos rara vez requiere adquisición de datos y almacenamiento solamente. Inevitablemente, es necesaria la habilidad de analizar y presentar los datos para determinar los resultados y tomar decisiones basados en los datos registrados. Una aplicación completa de registro de datos generalmente requiere la mayoría de los elementos ilustrados en la figura 2.4.

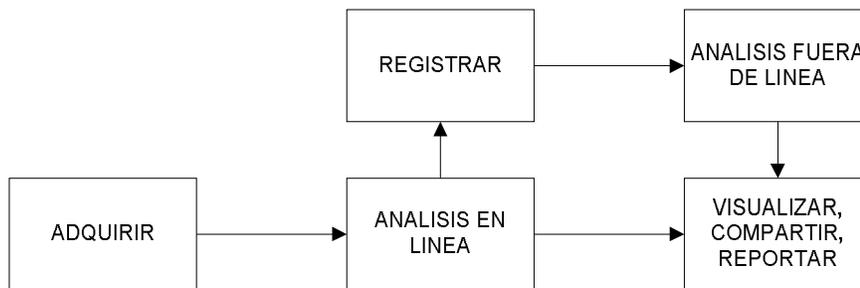


Figura 2.4 Elementos para Aplicaciones de Registro de Datos.

- **Adquirir**– Este paso incluye sensores y hardware de registro de datos, así como la conversión de fenómenos físicos en señales digitales.
- **Análisis en Línea**– Este paso incluye cualquier análisis que se quiera hacer antes de almacenar los datos. Un ejemplo común de esto es convertir las medidas de voltaje en unidades científicas significativas, como grados Celsius. Se pueden realizar estos

cálculos antes de registrarlos. Controlar parte de un sistema basado en medidas de corriente - por ejemplo, un conmutador - es también parte de análisis en línea. Cada aplicación de software de registro de datos debe realizar la conversión de binario a voltaje y la conversión de voltaje a unidades científicas.

- **Registrar**– Este paso se refiere al almacenamiento de datos analizados incluyendo cualquier formato requerido para los archivos de datos.
- **Análisis Fuera de Línea**– Este paso incluye cualquier análisis que se quiera hacer después de almacenar los datos. Un ejemplo común es buscar tendencias en datos históricos o reducción de datos.
- **Visualizar, Compartir, Reportar**– Este paso incluye la creación de cualquier reporte que se necesite realizar para presentar sus datos. Sin embargo, también se puede presentar datos directamente desde el análisis en línea. Esto representa la habilidad para monitorear y ver los datos conforme se adquiera y analiza además de simplemente ver datos históricos.

#### **2.2.3.4 TOPOLOGÍAS DE REGISTRADORES DE DATOS**

- **Autónomo**

El hardware autónomo puede medir y registrar datos continuamente sin conexión a una PC. La lógica de aplicación ocurre en el procesador de registro de hardware. El almacenamiento de datos ocurre en el hardware con la habilidad de transferir datos a una PC. Ver figura 2.5 y 2.6.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*



Figura 2.5 Registradores de Datos Portátiles, Autónomos y de Bajo Costo (National Instruments, 2012)



Figura 2.6 Sistema de Registro de Datos Embebidos Autónomos (National Instruments, 2012)

• **Basado en PC**

La operación requiere conexión de tiempo completo a una PC en cualquier tecnología de bus apropiada. El almacenamiento de datos se lleva a cabo en la memoria de la PC principalmente. Ver figura 2.7.



Figura 2.7 Registradores de Datos Programables Basados en PC (National Instruments, 2012)

### 2.2.3.5 TIPOS DE SOFTWARE REGISTRADOR DE DATOS

- **Listo para Usar:** El software listo para usar es un software de funcionalidad fija sin la habilidad de escalar, configurar o personalizar más allá de su estado original.
- **Configurable:** El software configurable proporciona funcionalidad básica con la habilidad de añadir o modificar funciones, algoritmos y otros pasos definidos por el usuario. No requiere que el usuario modifique el software a través de programación.
- **Programable:** El software programable tiene un entorno abierto donde el usuario puede crear su “Identificación de Usuario” personalizado, lógica de ejecución, procesamiento de señales y análisis, y formatos y registro.

### 2.2.4 COMUNICACIÓN USB (UNIVERSAL SERIAL BUS):

El Universal Serial Bus (bus serie universal USB), es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos.

USB fue diseñado para estandarizar la conexión de periféricos, como mouse, teclados, joysticks, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, sistemas de adquisición de datos, módems, tarjetas de red, tarjetas de sonido y discos duros externos. Su éxito ha sido total, habiendo desplazado a conectores como el puerto serie ,puerto paralelo ,puerto de juegos, Apple Desktop Bus o PS/2 a mercados-nicho o a la consideración de dispositivos obsoletos a eliminar de los modernos ordenadores, pues muchos de ellos pueden sustituirse por dispositivos USB que implementen esos conectores.

Su campo de aplicación se extiende en la actualidad a cualquier dispositivo electrónico o con componentes, desde los automóviles (las radios de automóvil

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

modernas van convirtiéndose en reproductores multimedia con conector USB o iPod) a los reproductores de Blu-ray los modernos juguetes. Se han implementado variaciones para su uso industrial e incluso militar. Pero en donde más se nota su influencia es en los teléfonos inteligentes (smartphones), tabletas y videoconsolas, donde ha reemplazado a conectores propietarios casi por completo.

Desde 2008, aproximadamente 6 millardos de dispositivos se encuentran actualmente en el mercado global, y alrededor de 2 millardos se venden cada año.

Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra. Para ello existen concentradores (llamados USB hubs) que incluyen fuentes de alimentación para aportar energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites). (Bateson, 2001).

### **2.2.5 MICROCONTROLADORES PIC (PERIPHERAL INTERFACE CONTROLLER)**

El nombre actual no es un acrónimo. En realidad, el nombre completo es PIC micro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico).

Los PIC son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.

El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de entrada y salida, y

el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba microcódigo simple almacenado en ROM para realizar estas tareas; y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador. Ver figura 2.8. (Cuenca, Angulo y Angulo, 1998).

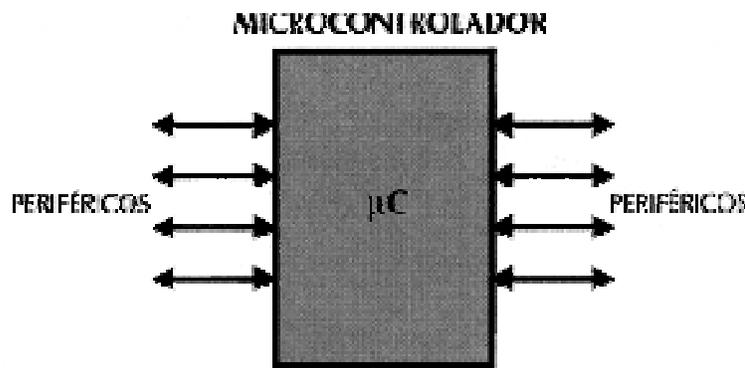


Figura 2.8 Microcontrolador (National Instruments, 2012)

En 1985 la división de microelectrónica de General Instrument se separa como compañía independiente que es incorporada como filial (el 14 de diciembre de 1987 cambia el nombre a Microchip Technology y en 1989 es adquirida por un grupo de inversores) y el nuevo propietario canceló casi todos los desarrollos, que para esas fechas la mayoría estaban obsoletos. El PIC, sin embargo, se mejoró con EPROM (siglas en inglés de Erasable Programmable Read-Only Memory, ROM programable borrable), para conseguir un controlador de canal programable. Hoy en día multitud de PICs vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores, etc.) y con memoria de programa desde 512 a 32.000 palabras (una palabra corresponde a una instrucción en lenguaje ensamblador, y puede ser de 12, 14, 16 ó 32 bits, dependiendo de la familia específica de PIC micro). (Cuenca, Angulo y Angulo, 1998).

### **2.2.5.1 Sistema de Interrupciones del PIC:**

El manejo de interrupciones es una técnica de programación, basada en un mecanismo automático en el hardware del microcontrolador, que permite dar atención a algún periférico interno o externo, únicamente en el momento en que ésta se requiera. Una interrupción es en realidad una llamada a una subrutina, pero iniciada por el hardware del propio periférico y no por la instrucción "CALL" (llamada). La interrupción puede ocurrir en cualquier instante.

Al generarse la señal de interrupción, el microcontrolador termina primero la ejecución de la instrucción que se esté procesando en ese momento, guarda en el STACK ó PILA la dirección de memoria de código de la siguiente instrucción y posteriormente hace un llamado a la "subrutina de atención de interrupciones", cuyo inicio siempre se encuentra en la dirección 04H de la memoria de código, en el caso de la mayoría de los microcontroladores 16F. (Cuenca, Angulo y Angulo, 1998).

## **2.2.6 INTRODUCCIÓN DE SEÑALES**

### **2.2.6.1 SENSORES**

Según Serrano (1997), son dispositivos capaces de convertir las variaciones de una magnitud física en variaciones de magnitud eléctrica o magnética denominadas habitualmente señales.

#### **- Sensores inductivos**

Están formados por un devanado enrollado en un núcleo (ver figura 2.9). Al aproximar el extremo del devanado a un objeto metálico, cambia la inductancia del primero, éste cambio puede monitorearse por el efecto que produce en un circuito resonante y sirve para activar un interruptor. Sólo se puede usar para detectar objetos

metálicos y funciona mejor con metales ferrosos en un rango que va desde 1 mm a unos 30 mm de distancia, con una posible resolución del orden de decimas de milímetro.

Son frecuentemente usados como interruptores de final de carrera ya que poseen algunas ventajas con respecto a los electromecánicos, tales como: ausencia de contacto con el objeto a detectar, robustez mecánica, resistencia a ambientes agresivos, altas temperaturas y bajo precio.



Figura 2.9 Sensor inductivo.

#### - Sensores ópticos

Emplean fotocélulas como elementos de detección, algunos tipos disponen de un cabezal que incorpora un emisor de luz y la fotocélula de detección, actuando por reflexión y detección del haz de luz reflejado sobre el objeto que se pretende detectar. Otros tipos trabajan a modo de barrera y están previstos para detección a mayores distancias con fuentes luminosas independientes del cabezal detector. Ambos tipos suelen trabajar con frecuencias luminosas en la gama de los infrarrojos. Ver figura 2.10.

Otra variante de los detectores ópticos son los de fibra óptica, que tienen los puntos de emisión-detección de luz separados de la unidad generadora, y unidos a ella mediante la fibra. De ésta forma, la detección puede llevarse a puntos inaccesibles para las fotocélulas de barrera o reflexión, aprovechando la flexibilidad de la fibra. Estos,

disponibles en versiones de reflexión y de barrera, tienen distancias de detección desde 3 mm a 10 m, pudiendo detectar objetos muy pequeños sobre piezas que se mueven a gran velocidad u objetos estáticos.

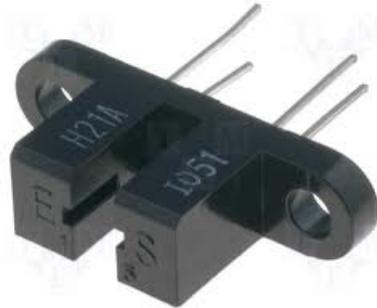


Figura 2.10 Sensor óptico.

#### **2.2.6.2 CLASIFICACIÓN DE LOS SENSORES SEGÚN EL TIPO DE SEÑAL DE SALIDA**

Según Serrano (1997), éstos pueden clasificarse de la siguiente manera.

- **Sensores analógicos:** son aquellos que proporcionan como salida un valor de tensión o corriente variable en forma continúa dentro del campo de medida. Es frecuente que para éste tipo sensores se incluya una etapa de salida para suministrar señales normalizadas de 0 V a 10 V ó de 4 mA a 20 mA.
- **Sensores digitales:** son aquellos que proporcionan como salida una señal codificada en forma de pulsos o en forma de una palabra digital codificada en binario o cualquier otro sistema.
- **Sensores todo o nada:** indican únicamente cuando la variable detectada rebasa un cierto umbral o límite. Pueden considerarse como un caso límite de los sensores digitales en el que se codifican sólo dos estados.

### 2.2.6.3 CLASIFICACIÓN DE LOS SENSORES SEGÚN EL TIPO DE ALIMENTACIÓN

- **Sensores pasivos:** son aquellos con alimentación externa para su funcionamiento, por lo general se basan en la modificación de la impedancia eléctrica o magnética de un material bajo determinadas condiciones físicas o químicas (resistencia, capacidad, inductancia, reluctancia, entre otras). Este tipo de sensores, debidamente alimentados, provocan cambios de tensión o de corriente en un circuito, los cuales son recogidos por el circuito de interfaz.
- **Sensores activos:** son aquellos que no poseen alimentación externa, en realidad son generadores eléctricos, generalmente de pequeña señal, por ello no necesitan alimentación exterior para funcionar, aunque si suelen necesitarla para amplificar la débil señal del captador.

### 2.2.7 LENGUAJE DE PROGRAMACIÓN C#

**C#** (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi. (O’Reilly, 2001).

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el **lenguaje nativo de .NET** La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención

de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic. (O'Reilly, 2001).

### **2.2.7.1 CARACTERÍSTICAS DE C#**

Con la idea de que los programadores más experimentados puedan obtener una visión general del lenguaje, a continuación se recoge de manera resumida las principales características de C#. Algunas de las características aquí señaladas no son exactamente propias del lenguaje sino de la plataforma .NET en general. Sin embargo, se comentan aquí también en tanto que tienen repercusión directa en el lenguaje, aunque se indicará explícitamente cuáles son este tipo de características cada vez que se toquen:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:

El código escrito en C# es **autocontenido**, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL o el tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.

- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción **foreach** que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión

de un tipo básico **string** para representar cadenas o la distinción de un tipo **bool** específico para representar valores lógicos.

- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: **encapsulación, herencia y polimorfismo**.

Por otro lado y a diferencia de Java, en C# se ha optado por hacer que todos los métodos sean por defecto sellados y que los redefinibles hayan de marcarse con el modificador **virtual** (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales. Además, un efecto secundario de esto es que las llamadas a los métodos serán más eficientes por defecto al no tenerse que buscar en la tabla de funciones virtuales la implementación de los mismos a la que se ha de llamar. Otro efecto secundario es que permite que las llamadas a los métodos. (O'Reilly, 2001).

- **Instrucciones seguras:** Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un **switch** ha de terminar en un **break** o **go to** que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.

- **Sistema de tipos unificado:** A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada **System. Object**, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”) A diferencia de Java, en C# esto también es aplicable a los tipos de datos básicos. Además, para conseguir que ello no tenga una repercusión negativa en su nivel de rendimiento, se ha incluido un mecanismo transparente de **boxing** y **unboxing** con el que se consigue que sólo sean tratados como objetos cuando la situación lo requiera, y mientras tanto puede aplicárseles optimizaciones específicas.

El hecho de que todos los tipos del lenguaje deriven de una clase común facilita enormemente el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo. (O’Reilly, 2001).

- **Extensibilidad de tipos básicos:** C# permite definir, a través de **estructuras**, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro **ref**. (O’Reilly, 2001).

- **Extensibilidad de operadores:** Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores incluidos los de conversión, tanto para conversiones implícitas como explícitas-cuando se apliquen a diferentes tipos de objetos.

# CAPÍTULO 3

## *Marco Metodológico*

---

---

### MARCO METODOLÓGICO

En este capítulo se presenta el tipo de metodología a utilizar para realizar la investigación y el conjunto de pasos que la describe, los cuales permiten la organización necesaria para obtener de manera satisfactoria la solución al problema planteado.

#### **3.1- NIVEL DE LA INVESTIGACIÓN**

El presente trabajo de grado es de tipo experimental ya que se debe indagar acerca del fenómeno tribológico, es decir, estudiar a fondo el proceso de fricción y de desgaste. Así mismo se debe conocer acerca de los ensayos a realizar que involucra el procedimiento para determinar el coeficiente de fricción y las ecuaciones y así poder diseñare el sistema de adquisición de datos y todos los componentes que lo conforman.

#### **3.2- DISEÑO DE LA INVESTIGACIÓN**

El diseño de la investigación se refiere a estrategias concebidas por los investigadores para responder el problema planteado. De acuerdo a los objetivos específicos, el proyecto se desarrolla conforme a la siguiente estructura (Ver figura 3.1):

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

Primera Fase, Comprender el proceso actual del tribómetro:

Se recopilará información tanto teórica como de campo para la realización del desarrollo de la formulación del problema con el fin de obtener claramente los objetivos, limitaciones y alcances presentes en la investigación, con la finalidad de generar soluciones que se ajusten realmente a los problemas actuales.

Segunda Fase, Definir los componentes de adquisición de datos:

Se evaluarán los datos de campo y otras fuentes con la finalidad de obtener a profundidad los problemas a resolver.

Tercera Fase, generación de soluciones:

Se buscará exponer la solución más adecuada y efectiva para el proceso de adquisición de datos de un tribómetro Bloque en Anillo, para ello se contará con la ayuda del tutor académico.

Cuarta Fase, diseño cuantitativo del sistema:

Es en donde se diseñará el sistema el cual ya incluye dimensionamiento, cálculos y selección de los componentes que intervienen en él, asegurando así la eficiencia del mismo.

Quinta fase, validación de sistema diseñado:

Una vez reunida la información necesaria, se procede a establecer las consideraciones de diseño y a elaborar la metodología para el diseño de programa. Se

realizan en esta etapa pruebas de ensayos ejecutados manualmente, acompañados de los refinamientos que sean necesarios.

Sexta fase, generación de conclusiones y recomendaciones del proyecto:

Con el conocimiento previo del sistema diseñado se generarán las acciones pertinentes tanto para la instalación como operación del mismo, así como también la generación de conclusiones referentes a los objetivos planteados anteriormente.

### **3.3 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS**

La información y los datos para desarrollar este proyecto se obtienen a través de revisiones bibliográficas y la resolución escrita de problemas para la comprensión teórica. De la misma manera se obtiene la documentación necesaria para la determinación de las técnicas de programación adecuadas y adiestramiento en el manejo del lenguaje de programación que finalmente se utilice.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

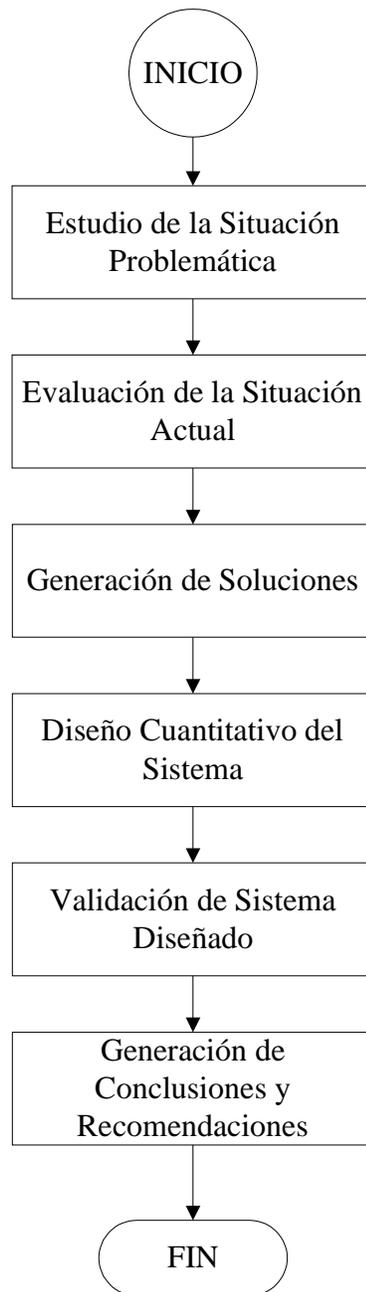


Figura 3.1 Diagrama de diseño de la investigación

# CAPÍTULO 4

## *Desarrollo de la Solución Propuesta*

---

---

En este capítulo se presenta el diseño, describiendo el sistema de medición para el Tribómetro Bloque en Anillo y sus partes. Luego se presentarán los componentes y los diagramas del sistema electrónico que lo conforma, concluyendo con la presentación del programa computacional de adquisición de datos y sus funciones.

### **4.1 EVALUACIÓN DEL SISTEMA EXISTENTE**

El tribómetro bloque en anillo que se encuentra en el Laboratorio de Predicciones de la escuela de Ingeniería Mecánica de la Universidad de Carabobo y fue construido en la década de los 80, (Turkington, 1980).

El objetivo principal del equipo es el ensayo por fricción para la obtención de datos precisos relativos según la rata de desgaste en distintos tipos de materiales y su relación con los parámetros más destacados que afectan el proceso. Es de considerar de alto interés medir el coeficiente de fricción entre los materiales expuestos al ensayo. Ver figura 4.1.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

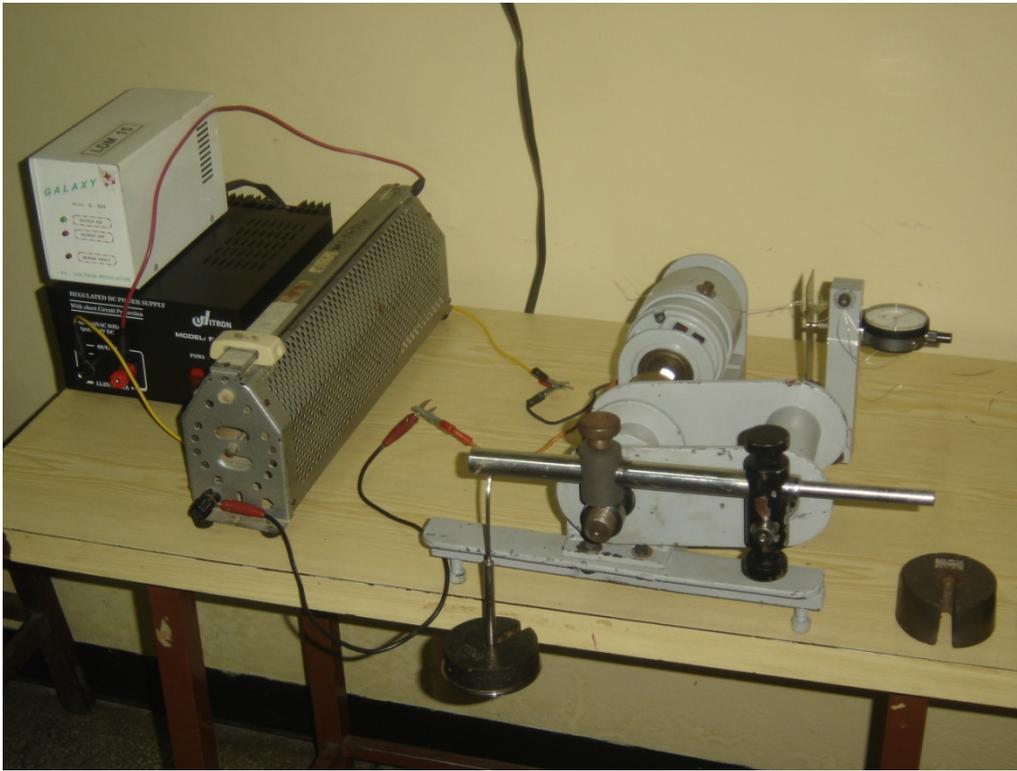


Figura 4.1 Tribómetro Bloque en Anillo. (Turkington, 1980)

El sistema cuenta con una viga calibrada, que debe presentar una relación lineal entre la fuerza aplicada y la deflexión de la misma. Esto se cumple para los metales en general dentro de su rango elástico de deformación (Ver figura 4.2). Para medir la deformación de la viga, se cuenta con un reloj comparador de marca Mitutoyo (Ver Anexo B.3), y se cuenta con una base lo suficientemente rígida en comparación con la viga, permitiendo variar fácilmente la posición del cuadrante para su fácil lectura. Está constituido también por un motor de corriente continua de tipo campo imanes permanentes. Las características principales del motor es la facilidad de variar la velocidad de giro con solo variar la tensión de alimentación.

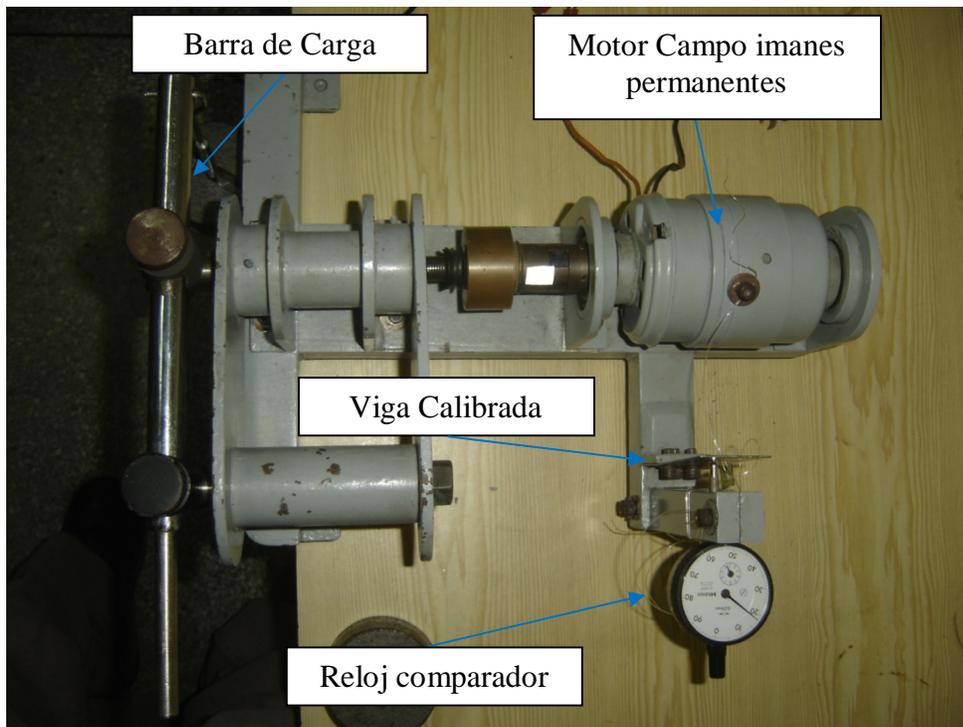


Figura 4.2 Tribómetro Bloque en Anillo (Vista de Planta). (Turkington, 1980)

#### 4.1.1 OBTENCIÓN DE DATOS MEDIANTE ENSAYOS DE FRICCIÓN Y DESGASTE

Para la adquisición de datos se realizaba la calibración del equipo mediante la realización de curvas experimentales las cuales deben construirse a partir de datos obtenidos de ensayos preliminares. La deflexión de esta curva calibrada es medida con el comparador tomando como medida indirecta del par torsor requerido para vencer la fuerza de fricción entre las dos especímenes en contacto, luego se realiza el ensayo de los materiales seleccionados. En la sección siguiente se describe la metodología de calibración manual que se utilizaba.

#### **4.1.1.1 CURVAS DE CALIBRACIÓN**

##### **4.1.1.1.1 CONSTRUCCIÓN DE LA CURVA DE CALIBRACIÓN DE LA VIGA**

La curva de calibración de la viga representa la relación entre la carga aplicada y la deflexión de la misma, indicada en el comparador. Para la obtención de la curva, se colocan pesas conocidas. A continuación se describen los pasos que se tomaron para la construcción de la curva:

- a) Se coloca una cuerda de Nylon y el colgador de pesas sobre la polea conectado a la viga empotrada (Ver Fig. 4.2).
- b) Anotar la lectura del comparador, la cual se toma como posición inicial de la viga.
- c) Se coloca una pesa de 1,97 N (200 gr) en el colgador y se anota la lectura del comparador.
- d) Repetir el paso c), añadiendo pesas hasta llegar a un peso total de 5,9 N (600 gr).

Como modo de ejemplo los valores de peso y deflexión se ordenan en la tabla 4.1, a partir del cual se construye la figura 4.3.

Tabla 4.1 Datos típicos de calibración. (Turkington, 1980)

Peso (Kg)	Deflexión (mm)
0,0	0
1,0	0,79
2,0	1,37
3,0	2,37
4,0	3,15

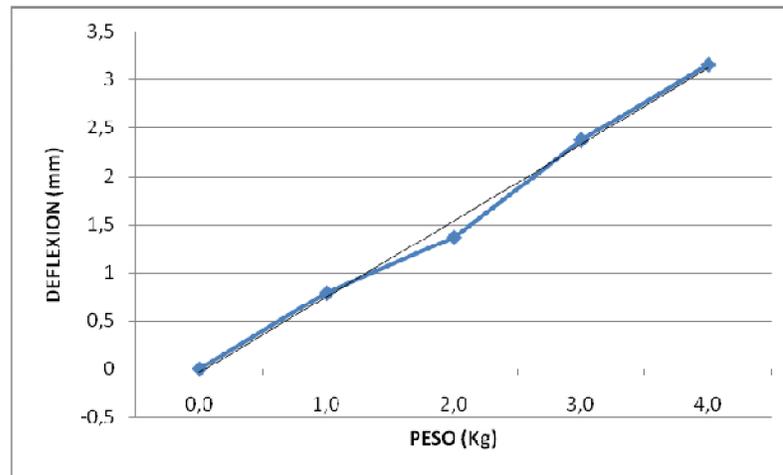


Figura 4.3 Curva de calibración de la viga, (Turkington, 1980)

#### 4.1.1.1.2 CONSTRUCCIÓN DE LA CURVA DE PAR TORSOR SIN CARGA

El par torsor de reacción indicado por la deflexión de la viga calibrada, es igual a la suma del par torsor requerido para vencer la fricción entre los dos especímenes y el par torsor gastado en fricción de cojinetes y debido al aire. Por lo tanto, es necesario conocer el valor del componente torque final " $T_f$ " el cual varía con la velocidad, por lo tanto se construye una curva de ensayo de la forma siguiente:

- a) Separar el espécimen fijo del espécimen rotativo.
- b) Anotar la lectura inicial del comparador.
- c) Usando un reóstato, ajustar la velocidad del motor a un valor inicial (ej. 500 RPM) y anotar la lectura del comparador.
- d) Repetir el paso c), haciendo variar la velocidad hasta 3500 RPM.

Como modo de ejemplo los valores de velocidad y deflexión se ordenan en la tabla 4.2 a partir del cual se construye la figura 4.4.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

Tabla 4.2 Datos típicos de ensayo sin carga. (Turkington, 1980)

RPM	Lectura Inicial Comparador (mm)	Deflexión (mm)
0	0,9	0
500	1,25	0,35
1000	1,37	0,47
1500	1,43	0,53
2000	1,48	0,58
2500	1,52	0,62
3000	1,58	0,68

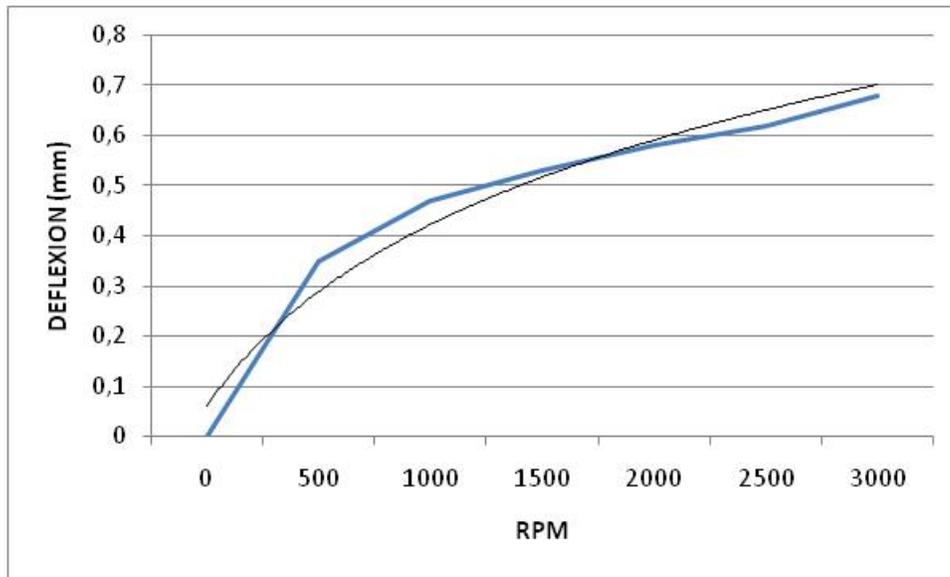


Figura 4.4 Curva típica de ensayo sin carga. (Turkington, 1980)

Esta curva se usa para calcular el coeficiente de roce o fricción en un ensayo específico según se explica a continuación en la sección 4.1.1.1.3

#### 4.1.1.1.3 CÁLCULO MANUAL DEL COEFICIENTE DE FRICCIÓN

Una vez realizadas la curva de calibración y la curva de par torsor sin carga, según las secciones 4.1.1.1.1 y 4.1.1.1.2, se procede a calcular el coeficiente de fricción resultante de las fuerzas entre los dos especímenes en contacto deslizante. Ver figura 4.5.

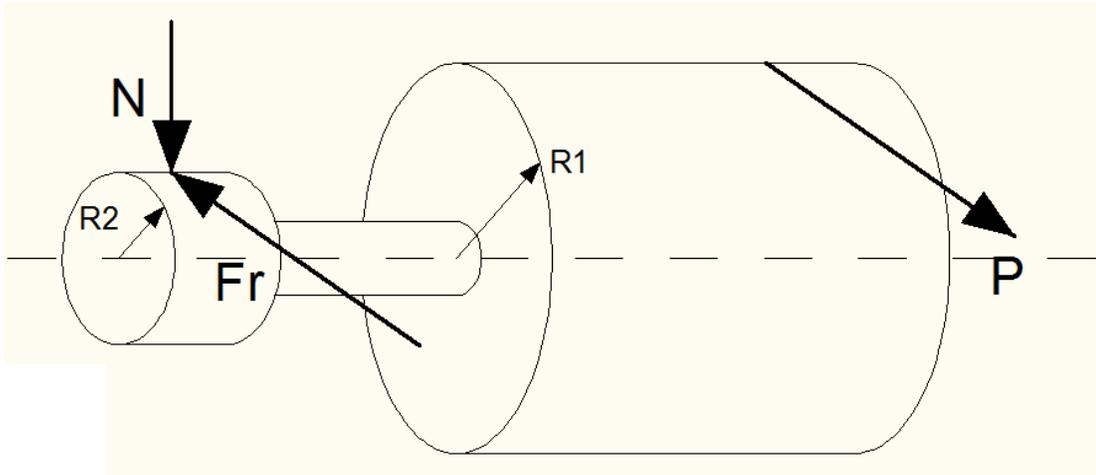


Figura 4.5 Diagrama de fuerzas.

El procedimiento se describe a continuación:

- Anotar la lectura del comparador y la velocidad angular de ensayo del espécimen rotativo.
- A partir de la curva de la figura 4.3, se lee el peso  $P'$  correspondiente a esa deflexión.
- A partir de la curva de la figura 4.4, se obtiene el valor de deflexión, correspondiente a la velocidad del ensayo.
- Volviendo a la figura 4.3, se lee el peso  $P_F$  correspondiente a esta deflexión.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

- e) Se lleva  $P'$  y  $P_F$  a pares torsores multiplicando por el radio ( $R_1$ ) de la polea.
- f) Se calcula el par torsor debido a la fricción entre especímenes,  $T$ .
- g) Se calcula la fuerza de fricción  $Fr$ , dividiendo  $T$  entre el radio ( $R_2$ ) del espécimen rotativo.
- h) El coeficiente de fricción es el cociente entre la fuerza de fricción ( $Fr$ ) y la carga normal ( $N$ ).

La carga normal es la suma de las pesas colocadas en el colgador, multiplicada por la ventaja mecánica del brazo de cargas.

#### **4.1.1.1.4 MEDICIÓN DEL DESGASTE**

La forma más sencilla de detectar y medir el desgaste, es mediante el pesaje del espécimen, ya que así la cantidad total de desgaste es expresada como un sólo número. Por supuesto, esto no indica la forma en que se distribuye el desgaste sobre la superficie. El procedimiento se resume en los siguientes pasos:

- a) Pesar inicialmente el espécimen fijo o el rotativo, después de una limpieza cuidadosa con acetona o tetracloruro de carbono. Se monta el espécimen e iniciar el ensayo.
- b) Cuando se desea medir el desgaste, interrumpir el ensayo, e desmonta el espécimen, se limpia cuidadosamente y se pesa de nuevo.
- c) La diferencia entre los pesos, es el desgaste total durante el periodo del ensayo.

## **4.2 DETERMINACIÓN DE LA CAUSA RAIZ**

En los ensayos que se realizan con el tribómetro bloque en anillo, se busca analizar el comportamiento ante la fricción y el desgaste, y representar de una manera más precisa y en ambiente controlado los mecanismos de operación de los componentes mecánicos de deslizamiento, midiendo parámetros como temperatura cercana a la zona de contacto, velocidad de deslizamiento, tiempo de ensayo y carga de aplicación entre las superficies en contacto.

El proceso de adquisición de datos en los ensayos se realiza de manera manual, tomando de forma directa e indirecta los parámetros ya mencionados. Los cuales se describen a continuación:

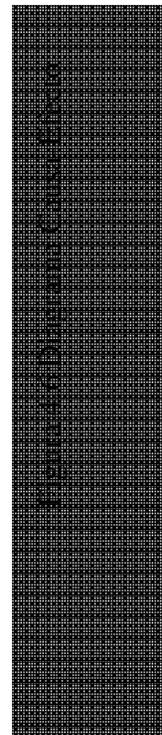
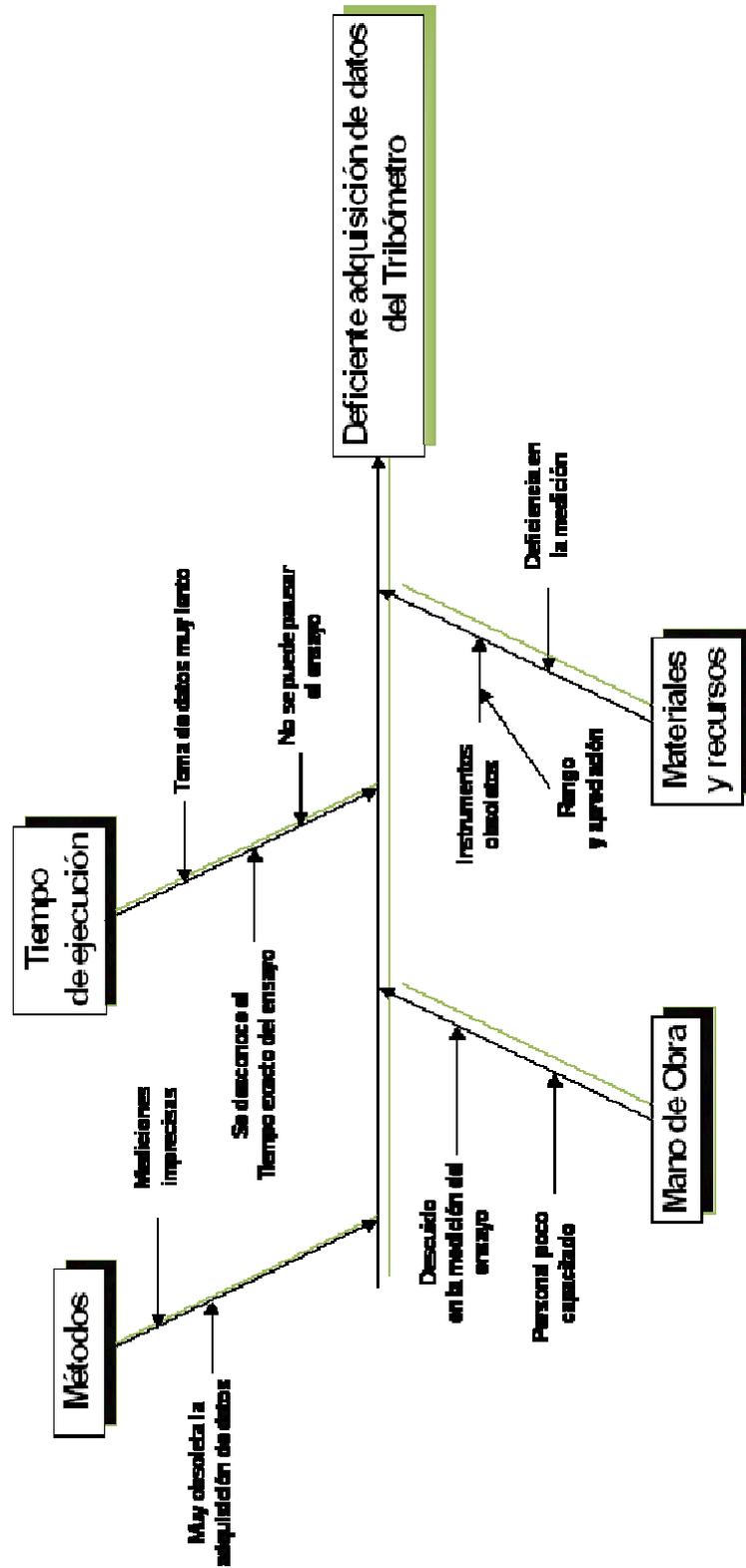
1. Para obtener la carga aplicada por el motor, se mide la deflexión hecha por una barra metálica adjunto al motor por medio de un comparador analógico, por lo que existe la posibilidad de errores de apreciación hechas por el encargado de la toma de datos.
2. El tiempo de ejecución se toma con un cronómetro de mano, por lo que existe la posibilidad de error de apreciación hechas por el alumno o persona que realiza el ensayo.
3. Las revoluciones por minutos realizadas por el motor no son contabilizadas de manera precisa o existe posibilidad de error de apreciación a la hora de la medición del mismo.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

1. El personal encargado de realizar las pruebas, toma de manera manual los datos obtenidos y calcula los parámetros directos e indirectos que rigen el ensayo.

Para la optimización del equipo existente, se realiza un análisis del proceso para aislar las causas que originan el mal resultado. El análisis se basó en las experiencias, en los hechos y datos que describen la situación existente.

Los resultados del análisis del problema se presentan de manera simplificada en la figura 4.6:



### **4.3 DEFINICIÓN DE PROPUESTAS Y PLAN DE ACCIÓN**

Una vez identificadas las causas, se toma la decisión acerca de las acciones para disminuirlas o eliminarlas. Para ello se definen las propuestas de acción que se presentan a continuación.

Para la selección de las mejores alternativas de acción se consideran los siguientes factores:

- **Efectividad:**

El sistema automatizado que se desea implantar resolverá en gran parte el problema de adquisición de datos que se presenta. Ya que el dispositivo diseñado para el banco es capaz de obtener los parámetros en tiempo real y calcular las variables que rigen el ensayo.

Entre los parámetros que se obtienen son:

- ✓ Revoluciones por minuto del motor.
- ✓ Par torsor del ensayo.
- ✓ Tiempo de ensayo.
- ✓ Fuerza de fricción.
- ✓ Coeficiente de fricción.
- ✓ Gráfica del ensayo.
- ✓ Tabla de datos.

Otros parámetros que se visualizan en el ensayo son:

- ✓ Radio de la polea.
- ✓ Radio del elemento rotativo.
- ✓ Temperatura promedio del elemento a ensayar.
- ✓ Fuerza normal en el elemento rotativo.

- **Factibilidad:**

El sistema se puede implantar en el laboratorio de predicciones de la facultad de ingeniería de la Universidad de Carabobo en la escuela de Ingeniería Mecánica, ya que es un sistema de automatización muy pequeño y de fácil instalación para el banco de ensayo.

- **Factor tiempo:**

El sistema que se desea implantar es una solución a corto plazo, ya que los resultados de su implantación son inmediatos. Y el usuario puede aplicar las herramientas que el sistema le ofrece.

- **Orientación al usuario:**

Para satisfacer los requisitos del usuario, se diseña un programa que es de fácil acceso y manipulación, mejorando interacción entre el ensayo y la persona encargada de ejecutar las pruebas.

#### **4.4 DESCRIPCIÓN DEL SISTEMA DISEÑADO**

El sistema se encarga que adquirir y calcular datos del ensayo realizado por el tribómetro bloque en anillo, mostrando sus resultados mediante una interface de usuario por medio de un programa computacional, permitiendo de este modo la visualización en tiempo real del ensayo en ejecución. También se persigue conseguir la fácil ejecución del ensayo por medio de la interface computacional.

Este dispositivo, es un circuito electrónico construido para adquirir y registrar los datos obtenidos en el ensayo del tribómetro tipo bloque en anillo (Ver figura 4.7). Dicho sistema, lo forman ciertos componentes que se muestran en la tabla 4.3 a continuación se realiza una breve descripción. Los esquemas electrónicos del circuito se muestran en los anexos 1 y 2.

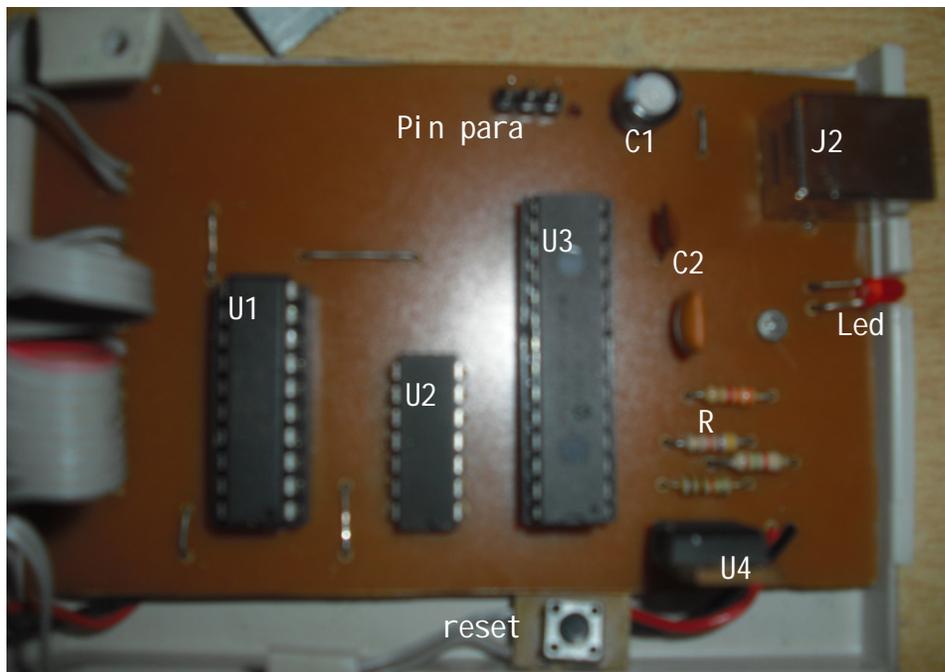
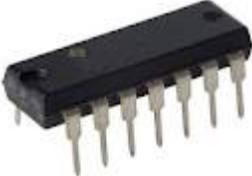


Figura 4.7 Circuito Electrónico

#### 4.4.1 LISTA DE COMPONENTES DEL CIRCUITO ELECTRÓNICO

Tabla 4.3 Componentes del circuito electrónico

Cant.	Descripción	Nomenclatura	Imagen
1	Microcontrolador <b>PIC16F690</b>	U1	
1	Microcontrolador <b>PIC18F2550</b>	U3	
1	Switch analógico <b>CD4066</b>	U2	
1	Regulador de Voltaje 3V <b>LM317</b>	U4	
1	Switch óptico (tipo herradura) <b>H21A1</b>	U5	

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

Tabla 4.3 Componentes del circuito electrónico (Continuación)

1	Conector <b>USB tipo B</b>	J2	
1	Condensador Electrolítico 10uF/16V	C1	
1	Condensador cerámico 0,1uF	C2	
2	Resistencias 1/4W 330Ohm	R1, R2	
4	Resistencias 1/4W 5,1KOhm	R3, R4	
1	Celda de carga extraída de una balanza Modelo SF-400. Rango 7 kg. Apreciación 10g	J1	
1	Placa base baquelita	--	

#### 4.4.2 DESCRIPCIÓN DEL CIRCUITO ELECTRÓNICO

A continuación se describen cada uno de los componentes instalados en el circuito electrónico:

- **Microcontrolador PIC16F690, (U1):**

El PIC16F690 establece la conexión entre el display digital y la celda de carga, también activa el apagado y encendido automático del circuito cuando está inactivo. El esquema electrónico del circuito se muestra en el anexo B.1.

- **Microcontrolador PIC18F2550, (U3):**

Es el maestro o “cerebro” de todo el circuito. Éste PIC es el que establece la conexión USB e intercambia datos con el software (STBA) que se describirá en la sección siguiente. El esquema electrónico del circuito se muestra en el anexo B.2

- **Switch analógico CD4066, (U2):**

Es el switch controlado por el PIC16F690 (U1) para encender y apagar automáticamente la celda de carga.

- **Regulador de Voltaje 3V LM317, (U4):**

Este dispositivo es el regulador de voltaje necesario para convertir los 5V del puerto de comunicaciones USB a los 3V que necesita la celda de carga instalada para su funcionamiento. El esquema electrónico del circuito se muestra en el anexo B.1.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

- **Conector USB tipo B, (J2):**

Este es un conector hembra de tipo “B”, igual al que poseen las impresoras modernas de comunicación por USB. Debe conectarse un cable común para impresoras USB, macho tipo “B” a macho tipo “A” y el extremo tipo “A” a cualquiera de los puertos del computador.

- **Condensador Electrolítico 10uF/16V, (C1):**

El condensador C1 permite obtener la capacidad eléctrica elevada en el circuito, es el componente necesario para establecer la comunicación USB, sin él, el PIC no puede comunicarse con el software.

- **Condensador cerámico 0,1uF, (C2):**

El C2 es el condensador necesario para estabilizar el voltaje de alimentación y disminuir las pérdidas en altas frecuencias.

- **Resistencias 1/4W 330Ohm, 1/4W 5,1KOhm (R1, R2, R3 y R4):**

Las resistencias instaladas en el circuito cumplen la función de estabilización del circuito, regulando la intensidad de corriente que la atraviesa.

#### **4.4.3 REQUERIMIENTOS MÍNIMOS DEL COMPUTADOR:**

- ✓ Procesador Pentium 266 MHz o superior con Windows 98/NT/2000/Me/XP/Vista/7.
- ✓ Unidad de CD ROM y/o DVD.
- ✓ Puerto de comunicaciones USB.

Además es necesario Un (1) Cable USB macho tipo “A” a macho tipo “B” (Como el que usan las impresoras).

## **4.5 MEDICIÓN DEL PAR TORSOR:**

### **4.5.1 CELDA DE CARGA (J1):**

Como se describió en la sección 4.1, es necesario construir la curva de calibración y la curva de par torsor sin carga para luego de forma experimental y con ayuda de las gráficas poder obtener el par torsor resultante en el ensayo, luego de la obtención de este valor, se procede a calcular la fuerza de fricción resultante y luego el coeficiente de fricción del ensayo realizado en ese instante. Para esto es necesario realizar largos procedimientos de adquisición manual de datos, llevando así errores de apreciación en la lectura o simplemente elevar el rango de errores en la medición del mismo.

Para evadir la construcción de dichas gráficas de forma manual, se instaló una celda de carga extraída de una balanza electrónica (Ver Tabla 4.3), con la finalidad de adquirir directamente la fuerza ejercida por el motor al momento de iniciar el ensayo y obtener los datos de manera instantánea.

La celda de carga instalada (J1), es un transductor que es utilizado para convertir una fuerza aplicada por el motor en una señal eléctrica (Ver figura. 4.8). Esta conversión es indirecta y se realiza en dos etapas. Mediante un dispositivo mecánico, la fuerza que se desea medir deforma una serie de galgas extensiométricas. La galga extensiométrica convierte el desplazamiento o deformación en señales eléctricas. Esta celda de carga se compone de cuatro galgas extensiométricas conectadas en una configuración tipo puente de Wheatstone.

La señal eléctrica de salida es típicamente del orden de unos pocos milivóltios y debe ser amplificada mediante un amplificador de instrumentación antes de que pueda ser utilizada. La salida del transductor se conecta en un algoritmo para calcular la fuerza aplicada al transductor.

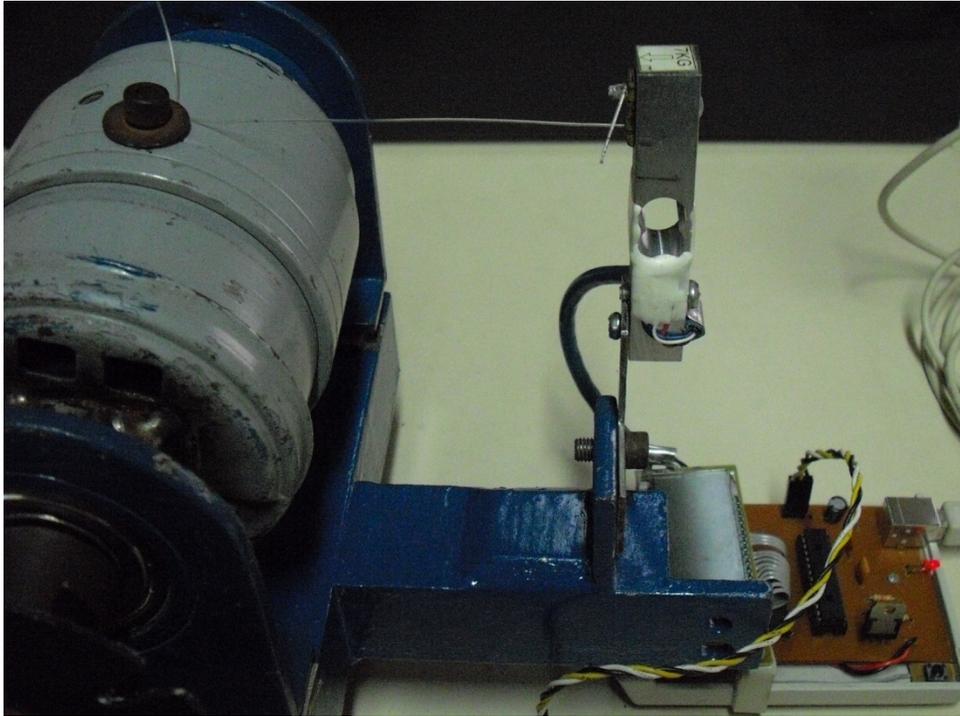


Figura 4.8 Celda de Carga Instalada en el banco de ensayo

## **4.6 MEDICIÓN DE LA VELOCIDAD ANGULAR DEL SISTEMA**

### **4.6.1 SWITCH ÓPTICO (U5)**

Para medir la velocidad angular del motor de manera sencilla y rápida, sólo se necesita contar el número de vueltas en un intervalo de tiempo determinado. Para ello se instaló un switch óptico al circuito electrónico. De igual manera es necesario amplificar la señal eléctrica de salida del dispositivo mediante un amplificador de instrumentación antes de que pueda ser utilizada.

Para contabilizar la cantidad de giros del sistema, se procede a interrumpir un haz de luz infrarrojo una vez por cada revolución. Para ello, se acopló al eje del motor una especie de disco con una ranura en él (ver figura 4.9), En las figura siguiente se hace evidente que el comportamiento del disco es el opuesto al de la pestaña, pero para el propósito deseado no afecta la medición.

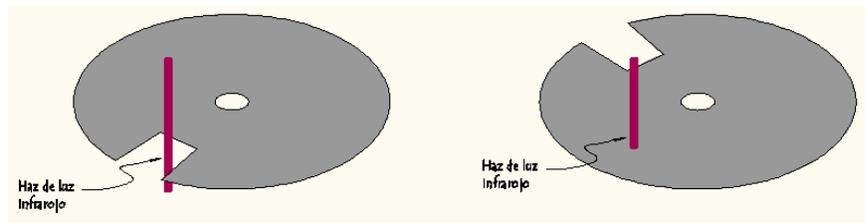


Figura 4.9. Interrupción del haz de luz con un disco en el eje del motor

El haz de luz infrarrojo proviene del switch óptico (U5), el cual integra en un mismo encapsulado, un LED emisor infrarrojo y un fototransistor que recibe el haz de luz. Mientras el haz no sea interrumpido (Ver Figura 4.10), el fototransistor envía 0V, y al momento de la interrupción, éste cambia la salida a 5V, generando así un tren de pulsos de la misma frecuencia a la que gira el motor.

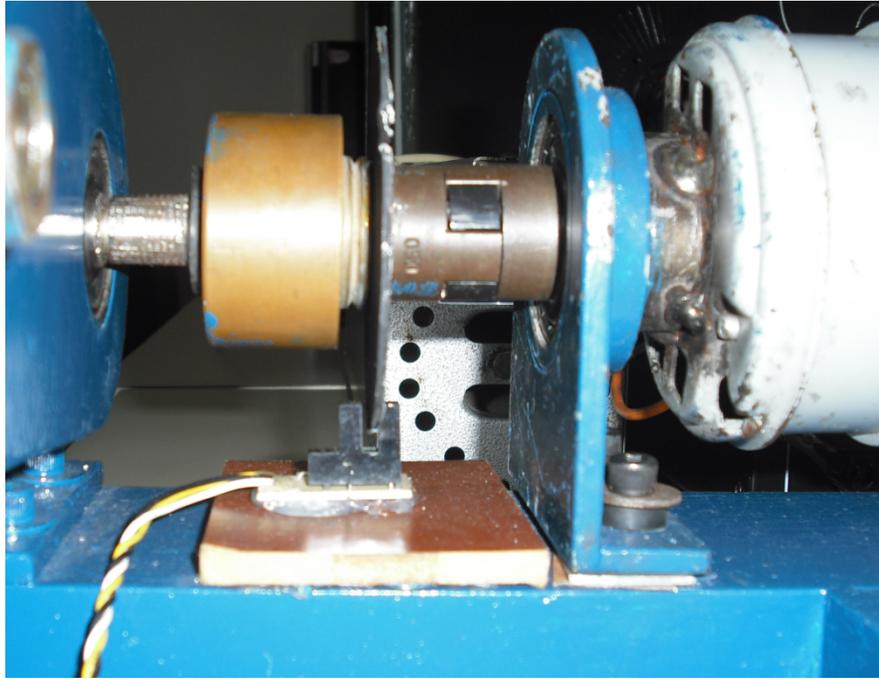


Figura 4.10 Switch óptico instalado en el banco de ensayo

#### **4.7 STBA: PROGRAMA PARA LA CAPTURA DE DATOS DEL ENSAYO DEL TRIBOMETRO BLOQUE EN ANILLO**

STBA (Software Tribómetro Bloque en Anillo) es el nombre de la herramienta computacional creada en este trabajo para la adquisición de datos del ensayo. Desarrollada en Microsoft Visual Basic 2008, el cual es compatible con todas las versiones del sistema operativo Microsoft Windows. Permite aproximar los valores del ensayo en ejecución de manera casi instantánea. Basta con especificar unos pocos datos acerca de las condiciones de trabajo del ensayo y en instantes visualizar los resultados.

En el Anexo “C” se incluye la herramienta computacional en CD para instalación.

#### **4.7.1 TÉCNICAS DE PROGRAMACIÓN SELECCIONADAS**

Debido a que el algoritmo de diseño incorpora estructuras secuenciales, de selección e iteración, es necesario recurrir a la programación estructurada. La programación modular ayuda a dividir un programa complejo en módulos ó subprogramas de menor complejidad, con el fin de hacerlo más legible y manejable.

La programación basada en objetos permite elaborar un entorno de aplicación dibujando objetos provenientes de una galería de objetos prediseñados (botones, cajas de texto, etiquetas, entre otros). Esto reduce el tiempo de desarrollo en gran medida, ya que el programador no tiene que detenerse a crear dichos objetos desde cero. Adicionalmente se requieren técnicas de programación para acceso a bases de datos.

#### **4.7.2 LENGUAJE DE PROGRAMACIÓN SELECCIONADO**

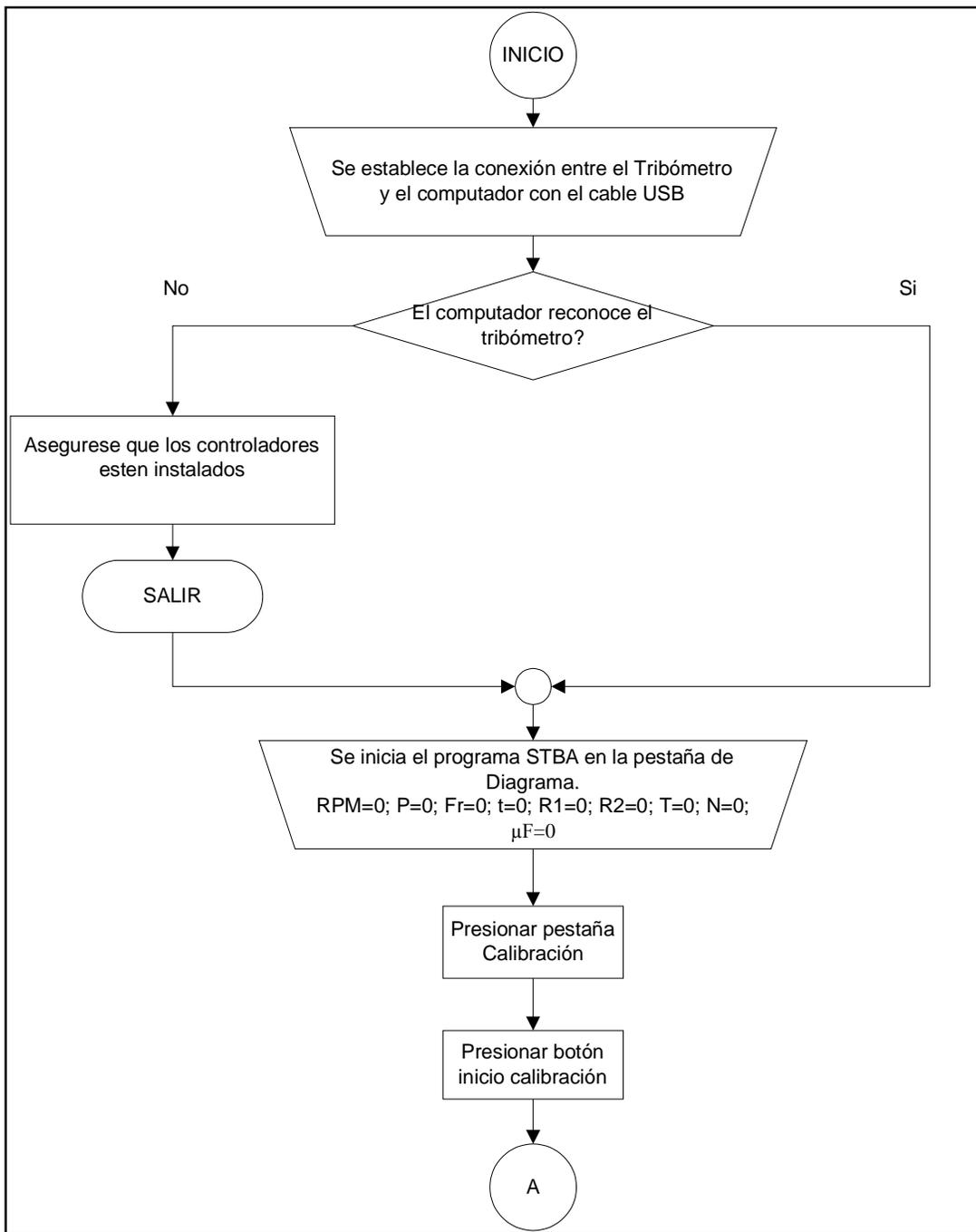
Microsoft Visual Basic 2008 es un lenguaje de programación basado en el entorno Microsoft Windows. Este lenguaje permite aplicar las técnicas de programación requeridas. Por otra parte, el equipo realizador ya cuenta con experiencia previa en el uso del lenguaje y algunos códigos que realizan determinadas tareas básicas, que combinadas permiten crear aplicaciones robustas en corto tiempo.

#### **4.7.3 ALGORITMO DE PROGRAMACIÓN**

- **Algoritmo principal para la calibración del ensayo:** En este algoritmo realiza la ejecución del programa para la obtención de datos que se ejecutará en el laboratorio para la calibración del banco de ensayo en vacío, muestra paso a paso las funciones del programa utilizado para la obtención del mismo.
  
- **Algoritmo principal para la captura de datos del ensayo:** En este algoritmo realiza la ejecución del programa para la obtención de datos que se ejecutará en el laboratorio para la obtención de datos en tiempo real del banco de ensayo con carga aplicada, muestra paso a paso las funciones del programa utilizado para la obtención del mismo.

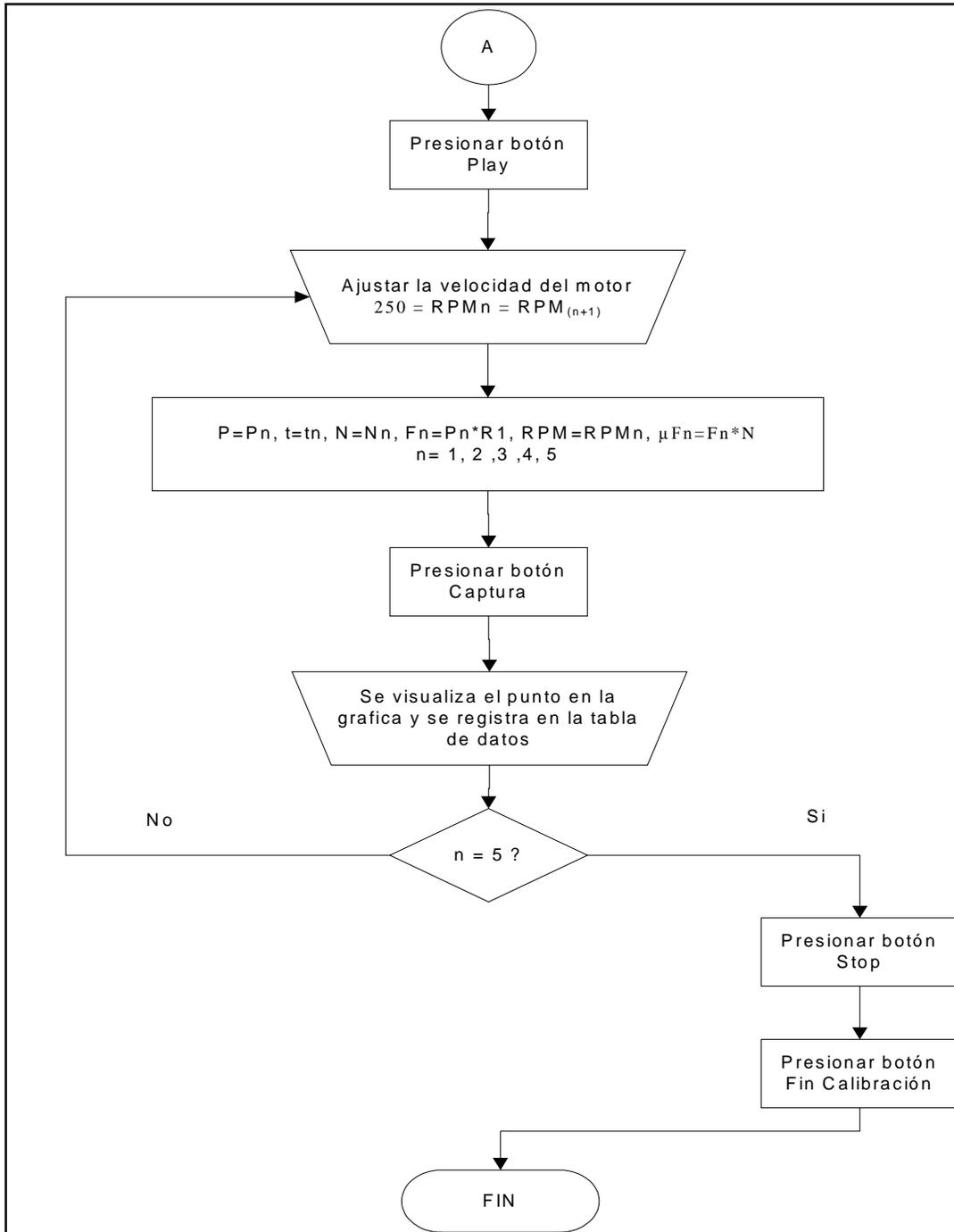
Los algoritmos descritos se presentan a continuación.

### 4.7.3.1 Algoritmo principal para la calibración del ensayo



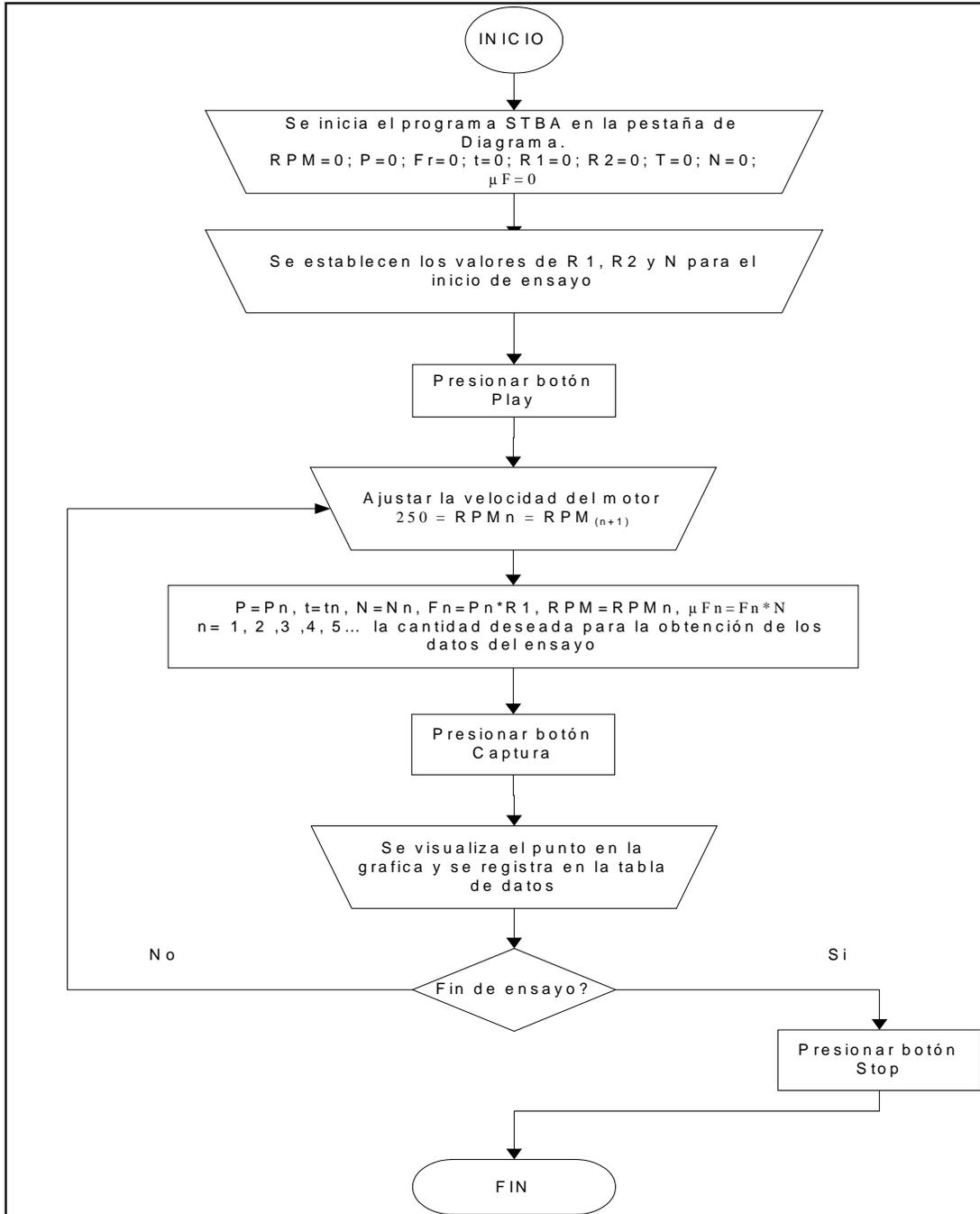
Algoritmo principal para la calibración del ensayo (página 1 de 2).

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*



**Algoritmo principal para la calibración del ensayo (página 2 de 2).**

4.7.3.2 Algoritmo principal para la captura de datos del ensayo



#### 4.7.4 DESCRIPCIÓN DEL ENTORNO DE PROGRAMA

El entorno de cada una de las pestañas del STBA consta de dos paneles: uno superior (panel *Visualización*) y uno inferior (panel *Resultados y Variables*). La tarea de cada panel viene dada por sus nombres.

El panel *Visualización* permite especificar toda la información acerca de las condiciones de trabajo del ensayo (Diagrama, Gráfica, Tabla de datos y tabla de calibración; curva de calibración).

El panel *Resultados y Variables* es un espacio dispuesto para presentar de forma textual los resultados en tiempo real y las correspondientes al ensayo en ejecución y está siempre visible asimismo cambie de pestaña. En la parte superior derecha del panel *Visualización* se encuentran los botones de control del ensayo y también está siempre visible asimismo se cambie de pestañas, (Ver figura 4.11), el cual se describe a continuación.

##### 4.7.4.1 DESCRIPCIÓN PANEL DE CONTROL DEL ENSAYO

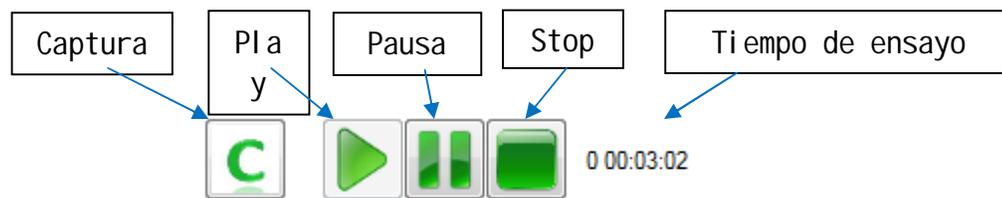


Figura 4.11 Captura del panel de control del ensayo

- **Play:**

El botón play está configurado para iniciar el ensayo, dando comienzo a la visualización en tiempo real de los datos adquiridos por el circuito electrónico ya descrito en la sección anterior. La descripción del código se muestra en el anexo A.2.

- **Pausa:**

Este botón detiene temporalmente el ensayo, pausando el cronómetro visible en el panel de control de ensayo. La descripción del código se muestra en el anexo A.3.

- **Stop:**

Detiene completamente el ensayo en ejecución, reiniciando el cronómetro y borrando todos los datos registrados por el programa. Es necesario realizar nuevamente la calibración del ensayo. La descripción del código se muestra en el anexo A.4.

- **Captura:**

Este botón permite obtener una captura de datos de manera instantánea, registrando los valores del ensayo en ejecución. La descripción del código se muestra en el anexo A.1.

- **Tiempo de ensayo:**

Muestra el tiempo transcurrido luego de iniciado el ensayo, el formato del cronómetro es; Días: Horas: Minutos: Segundos.

#### 4.7.4.2 DESCRIPCIÓN PANEL RESULTADOS Y VARIABLES

En este panel se muestran los valores y resultados del ensayo en tiempo real, tanto como las variables ajustables las cuales rigen el banco de ensayo. Ver figura 4.12.



Figura 4.12 Captura de la interface Variables y Resultados

A continuación se describen los valores de derecha a izquierda (Ver Figura 4.12), que se muestran en el panel *Resultados y Variables*:

- **Velocidad (rpm):**  
Muestra la velocidad angular del ensayo censada por el switch óptico (U5) en RPM.

- **Peso (P):**

El peso P es la magnitud escalar de la fuerza ejercida por el motor, medida por la celda de carga, esta fuerza es multiplicada por un factor de conversión para obtener el valor de la fuerza en Kgf, para luego posteriormente utilizar este valor en el cálculo del par torsor.

- **Fuerza de Fricción (Fr):**

Es la fuerza resultante del cálculo obtenido entre el par torsor del motor durante la ejecución del ensayo y la calibración del banco del ensayo previamente realizada.

El valor de la fuerza de fricción mostrada se calcula de la siguiente manera:

- a) Se obtienen los valores de Fuerzas P y P' medidas por la celda de carga, donde P es la fuerza ejercida por el motor en plena carga y P' es la fuerza ejercida por el motor sin carga que se obtuvo previamente en la pestaña de Calibración el cual se describe en la sección 4.7.5.4.
- b) Se llevan los valores P y P' a pares torsores multiplicando por el radio de la polea, es decir:

$$T = P \cdot R1 \qquad \text{Ecuación 4.1}$$

$$T' = P' \cdot R1 \qquad \text{Ecuación 4.2}$$

T' representa el par torsor al eje, T es el par torsor usado para vencer la fricción en el cojinete y debido al aire.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

c) Se calcula el par torsor debido a la fricción entre especímenes,  $T_f$ .

$$T_f = T - T' \quad \text{Ecuación 4.3}$$

d) Se calcula la fuerza de fricción,  $Fr$  dividiendo  $T_f$  entre el radio del espécimen rotativo (R2)

$$Fr = T_f/R2 \quad \text{Ecuación 4.4}$$

- **Tiempo de Ensayo (t):**

Muestra el tiempo del ensayo en ejecución. El formato del cronómetro es (Días: Horas: Minutos: Segundos).

- **Radio 1 (R1):**

Esta variable puede ajustarse antes y/o durante el ensayo. Es el radio del motor, que es necesario para obtener el par torsor ejercido por el mismo.

- **Radio 2 (R2):**

Esta variable puede ajustarse antes y/o durante el ensayo. Es el radio del espécimen rotativo a ensayar, que es necesario para obtener el par torsor ejercido por el mismo.

- **Temperatura (T):**

Esta variable puede ajustarse antes y/o durante el ensayo. Es la temperatura estimada que alcanza la zona de ensayo.

- **Peso (N):**

Esta variable puede ajustarse antes y/o durante el ensayo. Es la suma de las pesas colocadas en el colgador, multiplicada por la ventaja del brazo de cargas.

- **Coefficiente de Roce ( $\mu F$ ):**

Es el valor resultante entre el cociente de la fuerza de rozamiento calculada anteriormente y el peso (N) ajustado previamente, es decir;

$$\mu F = Fr/N$$

Ecuación 4.5

La carga normal (N) es la suma de las pesas colocadas en el colgador.

#### 4.7.4.3 INICIO DEL PROGRAMA

Para iniciar el software es necesario ejecutar el archivo “STBA.exe” que se encuentra en el CD de datos suministrado. Se abrirá una ventana de diálogo el cual muestra, en la parte derecha los puertos de comunicaciones disponibles en el computador, a la izquierda el botón “Conectar” donde se dará inicio a la ventana principal del Programa. (Ver figura 4.13). También cuenta con los botones; minimizar y salir, que cumplen con las mismas funciones de una ventana normal de Windows. La descripción del código para la recepción de datos del circuito se muestra en el anexo A.9

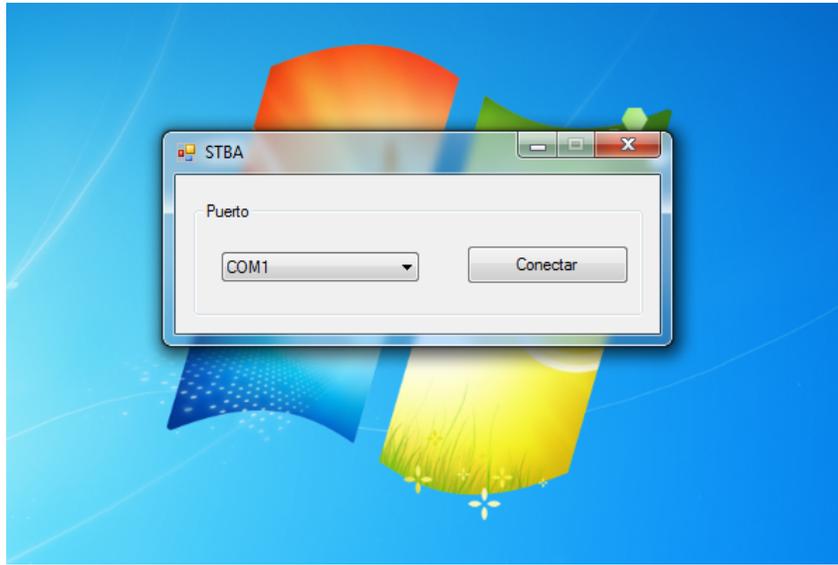


Figura 4.13 Captura de la ventana de inicio del STBA

#### **4.7.5 DESCRIPCIÓN PANEL VISUALIZACIÓN**

Como se explicó anteriormente en el panel *Visualización* permite especificar toda la información visual acerca de las condiciones del ensayo en ejecución ya sea Diagrama, Gráfica, Tabla de datos y tabla de calibración; el procedimiento de calibración se describirá en la sección 4.7.5.4.

##### **4.7.5.1 DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (DIAGRAMA)**

En el entorno de Diagrama se visualiza una imagen característica propia del tribómetro tipo bloque en anillo, el cual muestra las fuerzas resultantes y los radios característicos del mismo. Estas variables están identificadas con nomenclaturas que se describen en el panel *Resultados y Variables*. (Ver figura 4.14). La descripción del código para colocar el diagrama se muestra en el anexo A.7

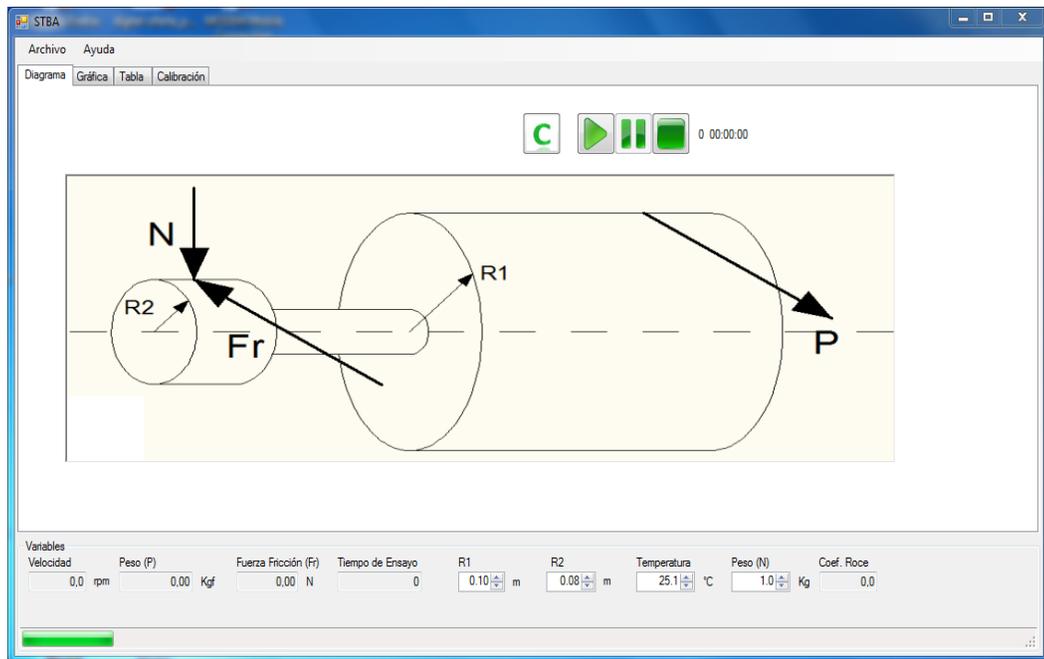


Figura 4.14 Captura de la interface de Pestaña (Diagrama)

#### 4.7.5.2 DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (GRÁFICA)

En este entorno, el software permite construir un gráfico en tiempo real de los valores obtenidos por el sistema de medición, sólo con presionar el botón Captura “C” luego de estabilizar el ensayo, el valor que se registra en la interface forma un punto de referencia en el gráfico mencionado. (Ver Figura 4.15). La Abcisa  $X$  registra los valores de RPM y la abscisa  $Y$  registra los valores de la Fuerza de roce (Fr). La descripción del código para realizar la gráfica del ensayo con carga se muestra en el anexo A.5

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

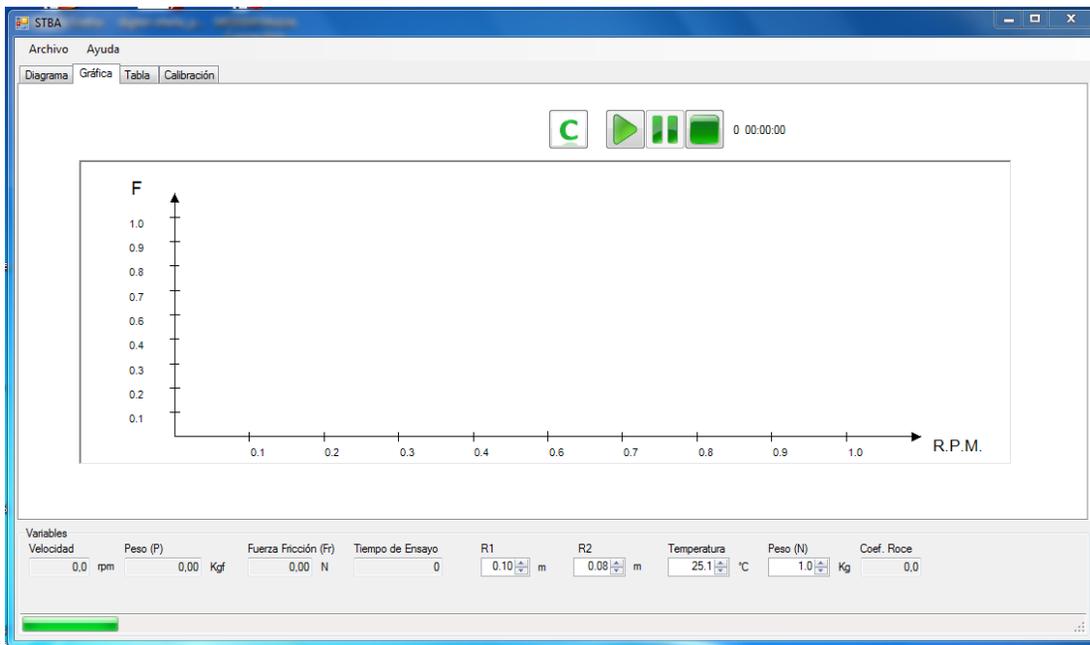


Figura 4.15 Captura de la interface de Pestaña (Gráficas)

#### **4.7.5.3 DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (TABLAS)**

El entorno de Tablas organiza los valores obtenidos por el sistema durante el ensayo, es necesario presionar el botón Captura “C” luego de estabilizar el ensayo y posteriormente se registran los datos construyendo una tabla, el cual se puede exportar para su estudio. (Ver Figura 4.16). En la tabla se registra el número de la captura, el tiempo de ensayo, la velocidad angular del motor, la Fuerza de fricción, la temperatura, el peso (N) y el coeficiente de fricción.

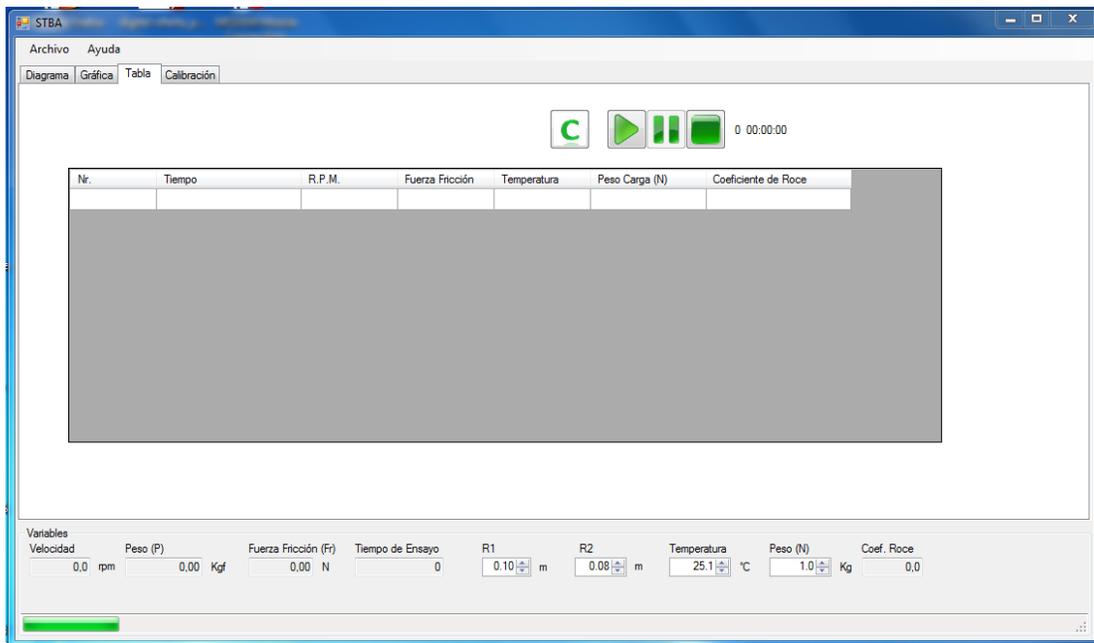


Figura 4.16 Captura de la interface de Pestaña (Tablas)

#### 4.7.5.4 DESCRIPCIÓN DEL ENTORNO DE PESTAÑA (CALIBRACIÓN)

Este entorno es el más importante del software, ya que si no se realiza la calibración del banco de ensayo sin carga de forma correcta, la captura de datos puede ser errónea y obtener valores pocos significativos.

A continuación se describen los pasos para realizar la calibración:

- a) Presionar sobre la pestaña *Calibración*.
- b) Pulsar el botón (Inicio Calibración), se iniciará el cronómetro de calibración y se empezarán a recibir las lecturas provenientes del circuito.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

En el entorno se muestra un botón de (Inicio Calibración) el cual cambiará a (Fin Calibración) al hacer click con el botón derecho del mouse (Ver Figura 4.17). La descripción del código para la función inicio/fin Calibración, se muestra en el anexo A.8.



Figura 4.17 Botones de Calibración

- a) Ajustar la velocidad del motor hasta obtener el valor mínimo deseado, presionar el botón *Capturar* para guardar el valor en la tabla de calibración y en la gráfica, Ver figura 4.18.
- b) Una vez obtenida la cantidad de puntos deseados (recomendable 5 como mínimo), pulsar el botón *Fin de Calibración* para terminar el proceso de Calibración.

En este entorno, el software permite construir un gráfico de calibración y una tabla de calibración en tiempo real de los valores obtenidos por el sistema de medición, sólo con presionar el botón Captura “C” luego de estabilizar el ensayo, el valor que se registra en la interface forma un punto de referencia en el gráfico mencionado y ordena los valores del ensayo de calibración tal como se muestra en la figura 4.18. La Abcisa *X* registra los valores de RPM y la abscisa *Y* registra los valores del peso (P) censada por la celda de carga. La descripción del código para realizar la grafica de calibración se muestra en el anexo A.6

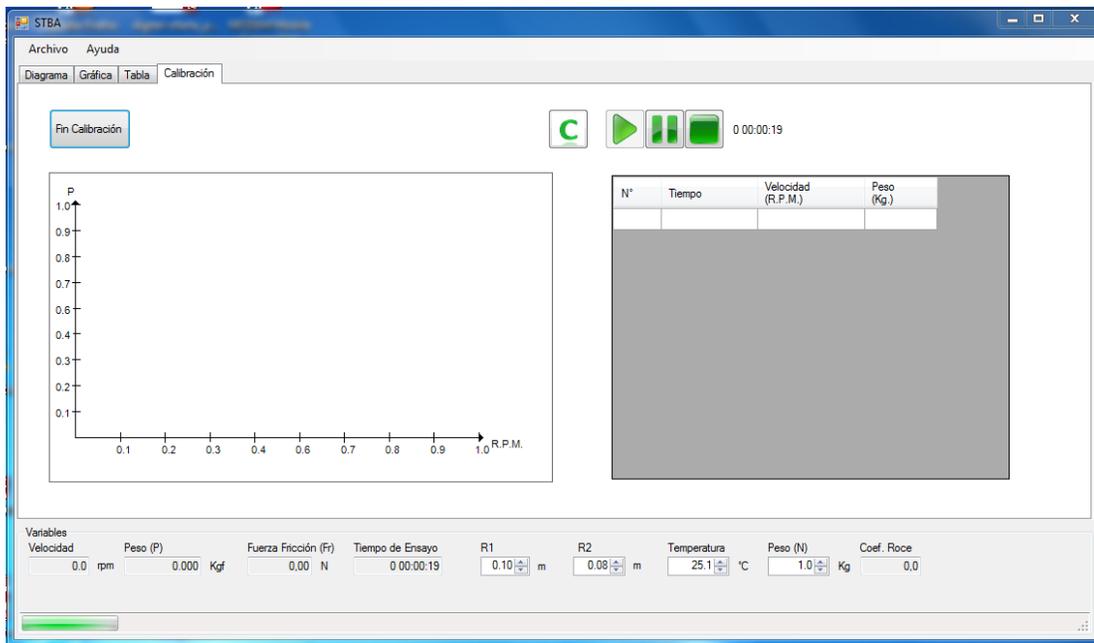


Figura 4.18 Captura de la interface de Pestaña (Calibración)

#### 4.7.6 OTRAS TAREAS REALIZABLES CON EL PROGRAMA

- **Exportar a Excel:**

El software también permite exportar los resultados obtenidos en las tablas, exportándolo a Microsoft Excel instalado en su Computador. Permitiendo guardar los resultados para su posterior utilización. (Ver Figura 4.19).

- **Salir:**

Esta opción permite cerrar el programa directamente desde el menú archivo.

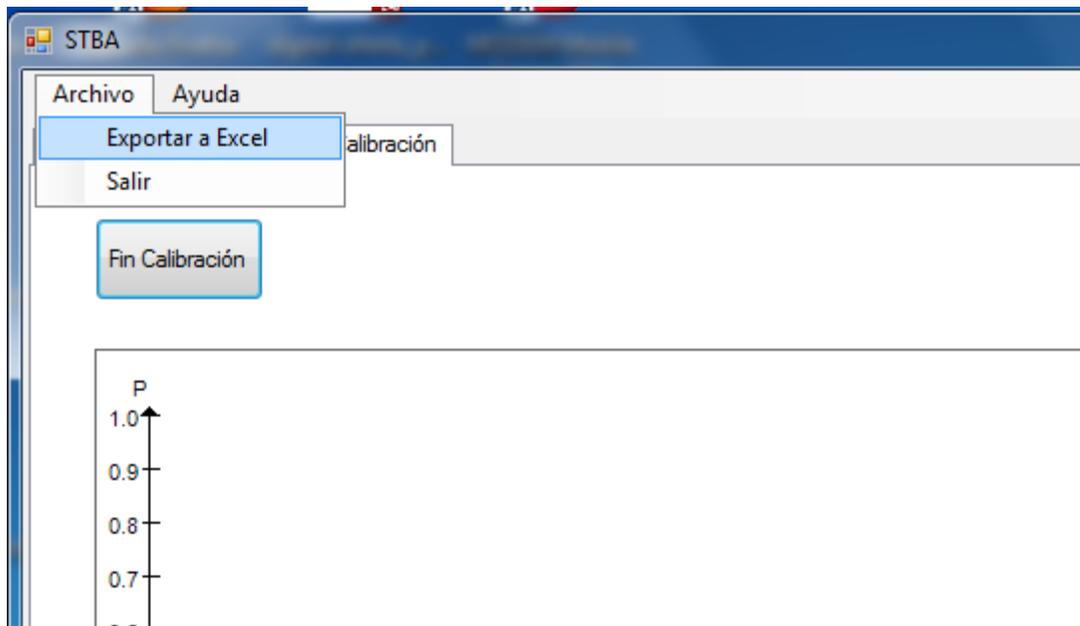


Figura 4.19 Despliegue del menú Archivo.

- **Ayuda:**

En el menú ayuda se ejecuta un archivo tipo PDF que describe brevemente el modo de calibración a realizarse antes de comenzar a iniciar las pruebas de medición en el banco de ensayo del tribómetro tipo bloque en anillo. (Ver figura 4.20). El contenido del archivo PDF se muestra en el anexo B.4.

- **Acerca de:**

En el menú acerca de muestra una breve descripción del banco de ensayo, su fecha de realización y datos de interés.

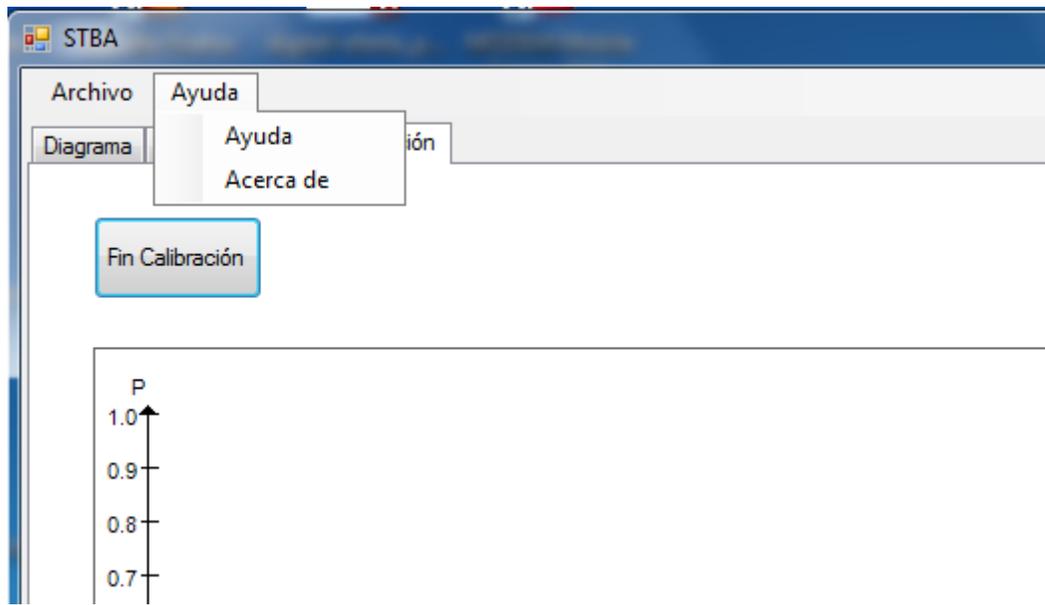


Figura 4.20 Despliegue del menú Ayuda.

#### 4.8 VALIDACIÓN DE LA HERRAMIENTA COMPUTACIONAL

En esta sección se validan los resultados de la herramienta computacional STBA. El arreglo del ensayo consistió en colocar el motor de la máquina en serie con un reóstato y una fuente de poder con ayuda de un computador de escritorio.

El motor de la máquina es un motor de corriente continua con un voltaje de 12 voltios y una velocidad máxima de 3000 RPM. La fuente de poder utilizada es una fuente con una entrada de 110 VDC y una salida de 13.8 VDC, 17 Ampere, marca Nitron, modelo PS105. En vista de que el voltaje de salida en la fuente es 1,8 Voltios por encima del que consume el motor, fue necesario utilizar un reóstato para controlar el voltaje suministrado a la máquina. Mediante el mismo se intentó controlar la velocidad de la máquina a 2000 RPM el cual se validó la velocidad de giro con un tacómetro digital marca Entek (Entach), serial nr. 2020261, la cual es una velocidad

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

adecuada para así comparar los resultados obtenidos con los antecedentes que dispone el equipo de investigación. En la figura 4.21 se muestra el arreglo final para el ensayo:



Figura 4.21 Arreglo ensayo del Tribómetro

Las condiciones del ensayo tribológico así como las propiedades de los materiales involucrados en el ensayo se encuentra en la tabla 4.5 y figura 4.22 respectivamente:

Tabla 4.4. Sistema Tribológico

<b>SISTEMA TRIBOLÓGICO</b>				
<b>TIPO DE ENSAYO: ENSAYO ESFERA SOBRE ANILLO</b>				
<b>VARIABLES OPERATIVAS</b>				
CARGA (N)		VELOCIDAD (m/s)	DISTANCIA (m)	
12,201		3,1416	2000	
<i>Tipo de movimiento:</i> Deslizamiento continuo			<i>Duración del ensayo:</i> 10 minutos aproximadamente	
<b>ESTRUCTURA DE LOS TRIBO-SISTEMAS</b>				
CARACTERÍSTICAS		PIN	RING	OBSERVACIÓN
<i>Tipo de material/elemento</i>		ACERO AISI Cr 52 100	ACERO INOXIDABLE SAF 2507	AIRE
<i>Geometría y dimensiones</i>		ESFÉRICO D= 6 MM	ANILLOS D <sub>EXT</sub> =30 mm espesor= 4mm	-
<i>Composición (% peso)</i>		1,35 a 1,65 Cr 0,90 a 1,10 C 0,15 a 0,35 Si 0,25 a 0,45 Mn	25 Cr ; 7 Ni 4 Mo ; 0,27 N	-
<i>Dureza promedio</i>		838 HV	240 HB	La dureza del anillo pudo variar por los ensayos anteriores
<i>Rugosidad</i>		0,100 µm	Desconocida	No se midió la rugosidad inicial del anillo

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

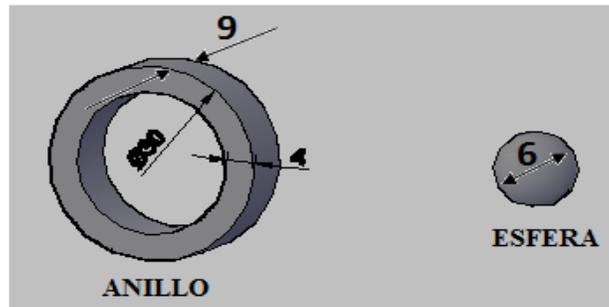


Figura 4.22. Especímenes de ensayo (Medina y Peralta, 2010)

A continuación se presentan los datos obtenidos en el ensayo realizado, la fuerza de fricción y los valores de coeficiente de fricción para la carga trabajada:

En la figura 4.23 se muestra la curva de calibración construida para el ensayo a realizar, tomando 6 valores para una mejor apreciación en los resultados.

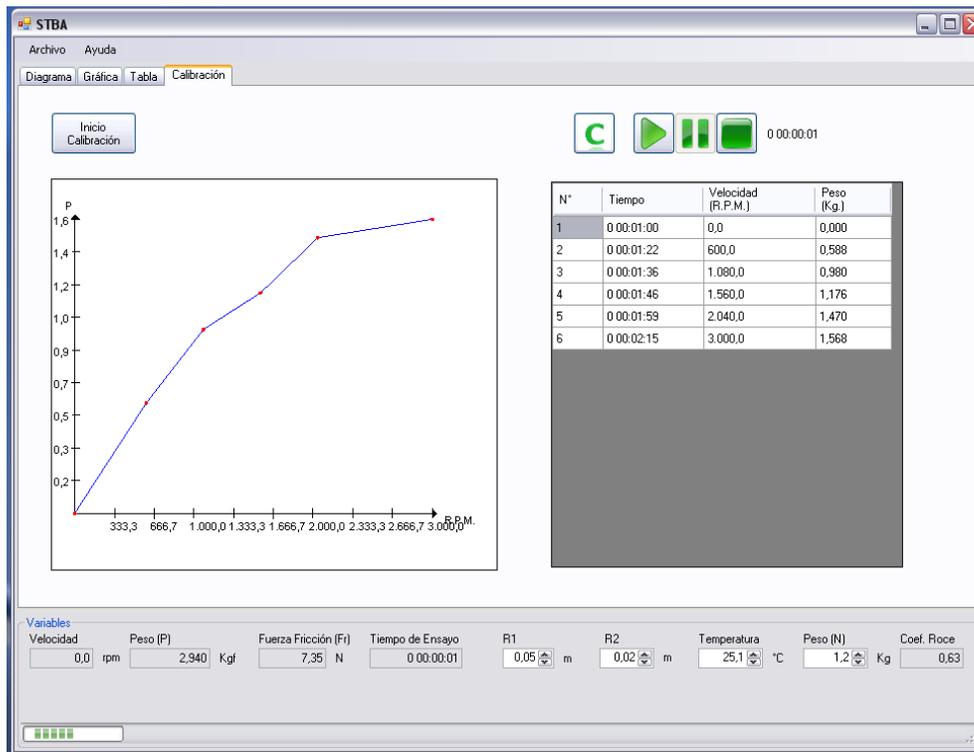
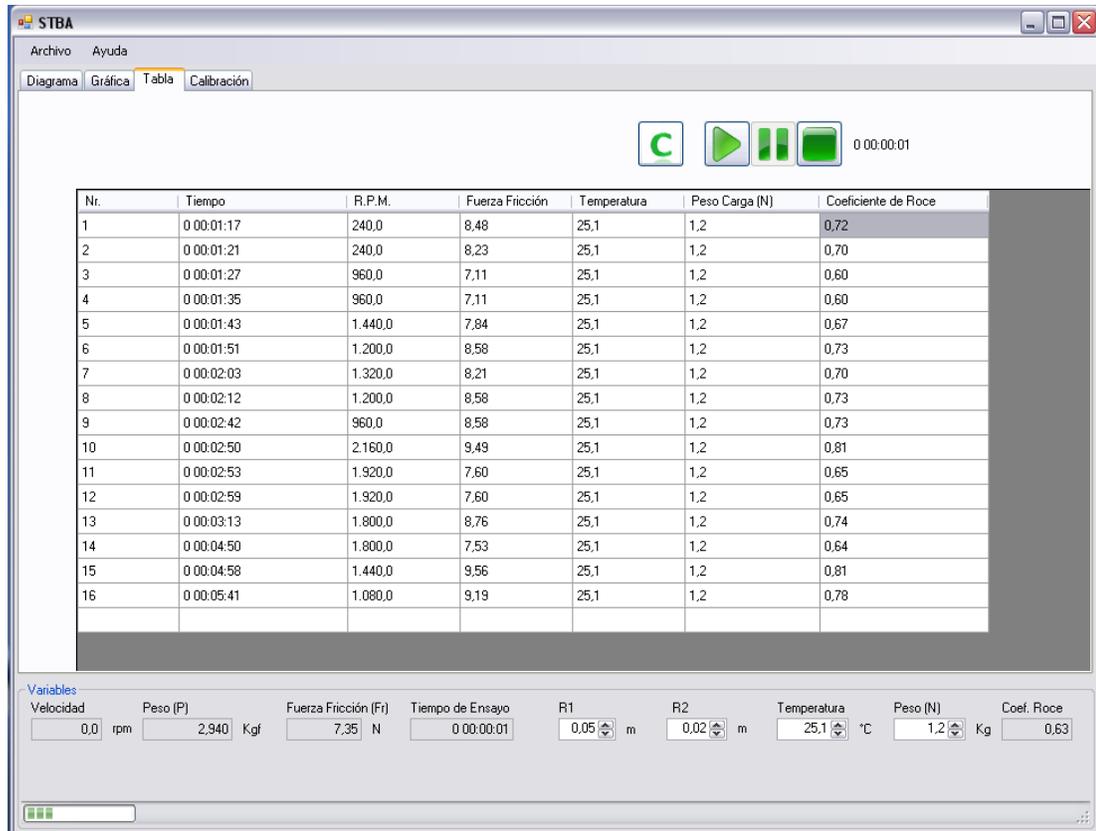


Figura 4.23 Captura pestaña calibración del ensayo realizado

#### Capítulo 4. Desarrollo de la solución

En la figura 4.24 se muestra la tabla construida durante el ensayo realizado, registrando los valores de fuerza de fricción, RPM, tiempo, coeficiente de fricción, carga aplicada y temperatura (este valor de temperatura es sólo referencial como se explicó en la sección 4.7.4.2).



The screenshot shows the STBA software interface. At the top, there are menu options: Archivo, Ayuda, Diagrama, Gráfica, Tabla (selected), and Calibración. Below the menu is a control panel with buttons for 'C' (Clear), Play, Stop, and a timer showing 00:00:01. The main area contains a data table with 7 columns: Nr., Tiempo, R.P.M., Fuerza Fricción, Temperatura, Peso Carga (N), and Coeficiente de Roce. Below the table is a 'Variables' section with input fields for various parameters.

Nr.	Tiempo	R.P.M.	Fuerza Fricción	Temperatura	Peso Carga (N)	Coeficiente de Roce
1	00:01:17	240,0	8,48	25,1	1,2	0,72
2	00:01:21	240,0	8,23	25,1	1,2	0,70
3	00:01:27	960,0	7,11	25,1	1,2	0,60
4	00:01:35	960,0	7,11	25,1	1,2	0,60
5	00:01:43	1.440,0	7,84	25,1	1,2	0,67
6	00:01:51	1.200,0	8,58	25,1	1,2	0,73
7	00:02:03	1.320,0	8,21	25,1	1,2	0,70
8	00:02:12	1.200,0	8,58	25,1	1,2	0,73
9	00:02:42	960,0	8,58	25,1	1,2	0,73
10	00:02:50	2.160,0	9,49	25,1	1,2	0,81
11	00:02:53	1.920,0	7,60	25,1	1,2	0,65
12	00:02:59	1.920,0	7,60	25,1	1,2	0,65
13	00:03:13	1.800,0	8,76	25,1	1,2	0,74
14	00:04:50	1.800,0	7,53	25,1	1,2	0,64
15	00:04:58	1.440,0	9,56	25,1	1,2	0,81
16	00:05:41	1.080,0	9,19	25,1	1,2	0,78

**Variables**

Velocidad: 0,0 rpm    Peso (P): 2,940 Kgf    Fuerza Fricción (Fr): 7,35 N    Tiempo de Ensayo: 00:00:01

R1: 0,05 m    R2: 0,02 m    Temperatura: 25,1 °C    Peso (N): 1,2 Kg    Coef. Roce: 0,63

Figura 4.24 Captura pestaña tabla del ensayo realizado.

*Desarrollo del sistema de medición y control para la adquisición de datos de un tribómetro tipo Bloque en Anillo*

En la figura 4.25 se muestra la gráfica construida durante el ensayo, obtenida con los valores de fuerza de fricción y RPM.

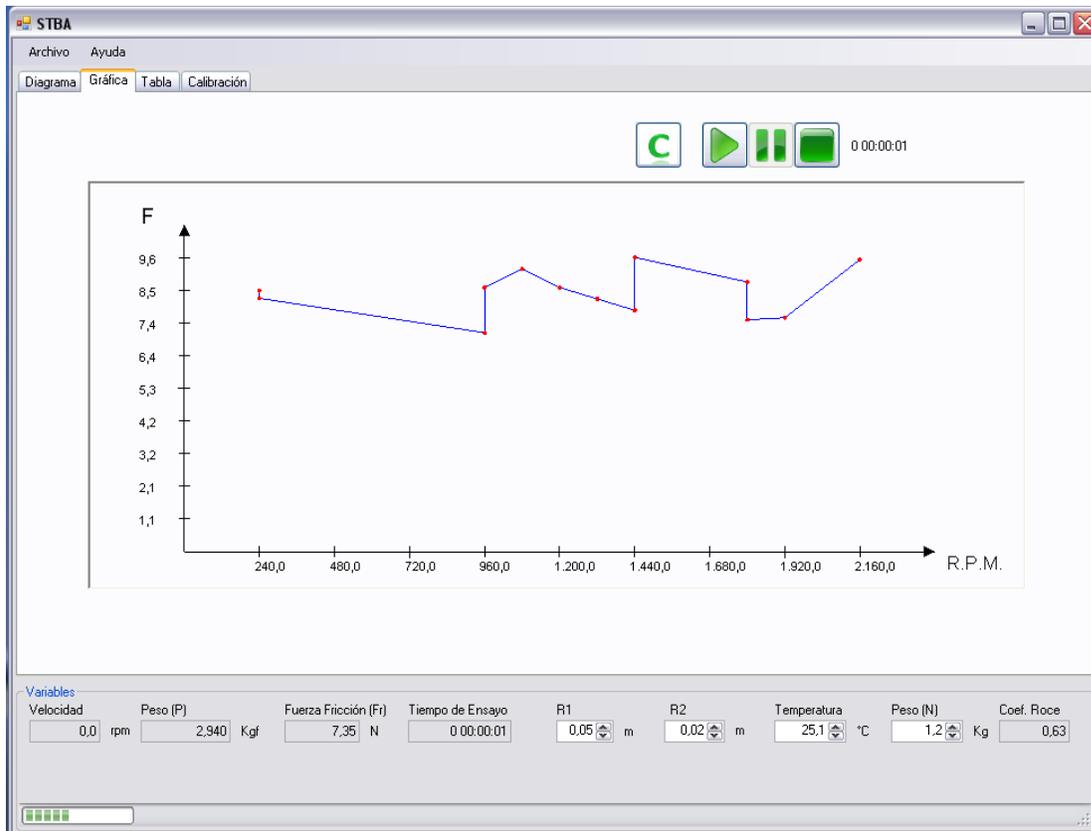


Figura 4.25 Captura pestaña gráfica del ensayo realizado

En la curva comparativa (figura 4.25), se puede apreciar que los valores más altos del coeficiente de fricción se obtienen para la fuerza de fricción de 9,49 N, con un valor de 0,81. El valor del coeficiente de fricción para una velocidad de giro de 1920 RPM es de 0,65 como se muestra en las capturas 11 y 12 de la figura 4.24, se puede apreciar que los resultados son similares a los obtenidos por los ensayos realizados en el trabajo de investigación realizado por Medina y Peralta en el 2010 (Ver anexo B.5). Aunque debido a la falta de estabilidad presentada por el reóstato utilizado, la velocidad del motor era muy variable y no se alcanzó una continuidad en las RPM del motor, obteniendo valores distintos en el ensayo.

# *Conclusiones*

---

---

Con el desarrollo de la investigación se concluye el logro satisfactorio de los objetivos planteados así como cabe destacar los siguientes comentarios finales:

- Para la adquisición de fuerza de fricción se instaló una celda de carga extraída de una balanza con la finalidad de remplazar el sistema de medición anterior el cual era muy obsoleto.
- Con el sistema automatizado, se implementa tecnología de punta, sustituyendo el sistema manual con alto orden de parada, retardos, inseguridad y retrabajo, por otra parte, se incrementará la confiabilidad y no se generarán problemas de errores de apreciación humana en las mediciones.
- La herramienta computacional desarrollada es sencilla, de poco peso y funcional. Requiere de pocos datos para actuar, agiliza la ejecución de los cálculos del ensayo. También es capaz de analizar dichos resultados y de exportar observaciones al usuario en caso de que alguno(s) de los valores se encuentre(n) fuera de ciertos límites de operación, contribuyendo a un mejor entendimiento del comportamiento de tales dispositivos.
- El sistema de medición automatizado de la fuerza de fricción funciona satisfactoriamente en el rango de velocidades por encima de 250 RPM. Para velocidades menores a 3500 RPM.
- Finalmente, el desarrollo de este trabajo especial de grado demuestra la versatilidad de las computadoras, la utilidad que tienen y lo necesarias que se han vuelto para el desarrollo de mejores proyectos de ingeniería.

## *Recomendaciones*

---

---

A continuación se plantean una serie de sugerencias para la realización de futuras mejoras y el mejor aprovechamiento de la metodología de ensayo y la herramienta computacional STBA.

- Utilizar la metodología de adquisición de datos y la herramienta computacional STBA en el proceso de aprendizaje-enseñanza de asignaturas relacionadas al área de procesos tribológicos y en el desarrollo de proyectos de ingeniería.
- Se recomienda crear un manual de operación, para estandarizar el procedimiento de funcionamiento del sistema, mejorando la seguridad y evitando pérdidas de tiempo.
- Se recomienda reemplazar el reóstato por un sistema de control de velocidades más preciso, para conseguir variar las velocidades del motor utilizado y así aumentar aún más la veracidad de los datos obtenidos por el programa computacional.
- Se recomienda implementar el sistema de medición y la herramienta computacional en otros bancos de ensayos existentes o en investigaciones futuras.
- Por último se recomienda reemplazar el motor existente en el banco de ensayos por un motor nuevo o en mejores condiciones, ya que el existente no opera en su valor nominal y afecta las condiciones ideales del ensayo.

## *Referencias Bibliográficas*

---

Acosta Gerardo, (2010). Automatización industrial. Apuntes capítulo VI. Argentina.

Recuperado Marzo 2012 de <http://www.fio.unicen.edu.ar/>

Christian Nagel, Morgan Skinner, Jay Glynn, Karli Watson, Ollie Cornes, Jerod Moemeka y publicado por Wrox Press en 2001.

Enciclopedia Wikipedia. Microcontroladores.

Recuperado Diciembre 10, 2011 de <http://es.wikipedia.org>

González, J. (2004) Automatización de procesos industriales. Universidad del País Vasco. Apuntes de clase de Automatización de Procesos Industriales.

Recuperado Julio 01, 2012 de <http://www.vc.ehu.es/>

Hernández y Tovar, (2008) Diseño de un sistema automatizado para un banco de trefilado de barras de acero. Valencia: Universidad de Carabobo.

*HITECH* Software LLC (2010). Produce el compilador PICC Lite para C. Este puede obtenerse en forma gratuita de su sitio web. La versión completa de este producto es uno de los compiladores más poderosos existentes para microcontroladores PIC.

Recuperado febrero 2012 de <http://www.htsoft.com/>

Introducción al sistema de control Technology, 7ma Edición, (2001) Robert N. Bateson, 706 Pp.

Turkington, K (1980) Diseño y fabricación de una máquina para ensayo de roce.

Valencia: Universidad de Carabobo

Lubricar. (2008), Teoría y Práctica de la Lubricación

Recuperado Enero 2012 de (<http://www.lubricar.net/teoria.htm>).

Medina y peralta, (2010) Evaluación de las Propiedades Tribológicas en el Acero Dúplex. Valencia: Universidad de Carabobo.

Microcontroladores PIC, (1998). La solución en un chip. Martin Cuenca, E. Angulo JM Y Angulo, segunda edición.

Microchip Technology (s.f.), Inc. Sitio del fabricante de los microcontroladores PIC. Aquí encontrará hojas de datos y notas de aplicaciones.

Recuperado Enero 2012 de <http://www.microchip.com/>

National Instruments, (s.f.). Registradores de datos

Recuperado Enero 2012 de (<http://zone.ni.com/devzone/cda/tut/p/id/10643>).

Programming C#, (2001), escrito por Jesse Liberty y publicado por O'Reilly en 2001

Anexo A:  
Código de Funciones del STBA

## Anexo A.1: Función del botón Captura.

```
private void capturar_Click(object sender, EventArgs e)
{
    if (calibrando)
    {
        if (puntosCalib == null)
        {
            puntosCalib = new PointF[1];
        }
        else
            Array.Resize<PointF>(ref puntosCalib,
                puntosCalib.Length + 1);

        puntosCalib[puntosCalib.Length - 1] =
            new
            PointF(float.Parse(velocidad_textBox.Text.ToString()),
                float.Parse(peso_textBox.Text.ToString()));
        ordenarPuntosCalib();

        object[] fila = { puntosCalib.Length,
            labelTiempo1.Text.ToString(),
            velocidad_textBox.Text, peso_textBox.Text };

        if (tabla2.Rows[0].Cells[0].Value == null)
        {
            tabla2.Rows[0].SetValues(fila);
        }
        else
        {
            tabla2.Rows.Add();
            object[] fila0 = new object[6];
            for (int k = 0; k < 4; k++)
                fila0[k] = tabla2.Rows[tabla2.RowCount -
1].Cells[k].Value;
            tabla2.Rows[tabla2.RowCount - 2].SetValues(fila0);
            tabla2.Rows[tabla2.RowCount - 1].SetValues(fila);
        }

        grafCalibracion.Refresh();
    }
    else if (!play1.Enabled)
    {
        if (puntosGrafica == null)
        {
            puntosGrafica = new PointF[1];
        }
        else
            Array.Resize<PointF>(ref puntosGrafica,
                puntosGrafica.Length + 1);

        puntosGrafica[puntosGrafica.Length - 1] =
            new PointF(float.Parse(velocidad_textBox.Text),
```

```

        float.Parse(friccion_textBox.Text));
ordenarPuntos();

        object[] fila = { puntosGrafica.Length,
labelTiempo1.Text.ToString(), velocidad_textBox.Text,
friccion_textBox.Text, temperatura_textBox.Value.ToString(),
pesoCarga.Value, coef_roce_textBox.Text };

        if (tabla1.Rows[0].Cells[0].Value == null)
        {
            tabla1.Rows[0].SetValues(fila);
        }
        else
        {
            tabla1.Rows.Add();
            object[] fila0 = new object[6];
            for (int k = 0; k < 7; k++)
                fila0[k] = tabla1.Rows[tabla1.RowCount -
1].Cells[k].Value;
            tabla1.Rows[tabla1.RowCount - 2].SetValues(fila0);
            tabla1.Rows[tabla1.RowCount - 1].SetValues(fila);
        }
        graficaPanel.Refresh();
    }
}

```

## Anexo A.2: Función del botón Play.

```
private void play_Click(object sender, EventArgs e)
{
    if (!serialPort1.IsOpen)
    {
        ProgressBar1.Value = 0;
        leyendoDisp = false;
    }
    else
    {
        try
        {
            serialPort1.Close();
            serialPort1.Open();
            ProgressBar1.Value = ProgressBar1.Maximum;
        }
        catch (Exception error)
        {
            ProgressBar1.Value = 0;
        }
    }
    try
    {
        byte[] envio = { Iniciar };
        serialPort1.Write(envio, 0, envio.Length);
        timer1.Enabled = true;

        play1.Enabled = false;
        play2.Enabled = false;
        play3.Enabled = false;
        play4.Enabled = false;

        pausa1.Enabled = true;
        pausa2.Enabled = true;
        pausa3.Enabled = true;
        pausa4.Enabled = true;
        ProgressBar1.Style = ProgressBarStyle.Marquee;
    }
    catch (Exception error)
    {
        MessageBox.Show("Dispositivo Desconectado.", "Error en el
Puerto", MessageBoxButtons.OK);
    }
}
```

### Anexo A.3: Función del botón Pausa.

```
private void pausa_Click(object sender, EventArgs e)
{
    try
    {
        timer1.Enabled = false;

        pausa1.Enabled = false;
        pausa2.Enabled = false;
        pausa3.Enabled = false;
        pausa4.Enabled = false;

        play1.Enabled = true;
        play2.Enabled = true;
        play3.Enabled = true;
        play4.Enabled = true;
        if (serialPort1.IsOpen)
        {
            byte[] envio = { Detener };
            serialPort1.Write(envio, 0, envio.Length);
        }
    }
    catch (Exception error)
    {
        MessageBox.Show("No se pudo enviar el comando.", "Error
en el Puerto", MessageBoxButtons.OK);
    }
}
```

#### Anexo A.4: Función del botón Stop.

```
private void stop_Click(object sender, EventArgs e)
{
    try
    {
        timer1.Enabled = false;
        duracionDias = 0;
        duracionHor = 0;
        duracionMin = 0;
        duracionSeg = 0;
        string temp = duracionDias + " " +
duracionHor.ToString("D2") + ":" + duracionMin.ToString("D2") + ":" +
duracionSeg.ToString("D2");
        labelTiempo1.Text = temp;
        labelTiempo2.Text = temp;
        labelTiempo3.Text = temp;
        labelTiempo4.Text = temp;
        textBox6.Text = temp;

        pausa1.Enabled = false;
        pausa2.Enabled = false;
        pausa3.Enabled = false;
        pausa4.Enabled = false;

        play1.Enabled = true;
        play2.Enabled = true;
        play3.Enabled = true;
        play4.Enabled = true;
        ProgressBar1.Value = 0;
        ProgressBar1.Style = ProgressBarStyle.Continuous;
        if (calibrando)
        {
            calibrando = false;
            Calib.Text = "Inicio Calibración";
        }
        if (serialPort1.IsOpen)
        {
            byte[] envio = { Detener };
            serialPort1.Write(envio, 0, envio.Length);
        }
    }
    catch (Exception error)
    {
        MessageBox.Show("No se pudo enviar el comando.", "Error
en el Puerto", MessageBoxButtons.OK);
    }
}
```

## Anexo A.5: Función para dibujar la gráfica del ensayo con carga.

```
private void graficarFriccion(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    //Eje Y
    g.DrawLine(Pens.Black, graficaPanel.Width * 0.1F,
graficaPanel.Height * 0.1F, graficaPanel.Width * 0.1F,
graficaPanel.Height * 0.9F);

    //Flecha Vertical
    PointF[] triangulo1 = {new PointF(graficaPanel.Width*0.1F-
5,graficaPanel.Height*0.1F+10),
new PointF(graficaPanel.Width*0.1F,graficaPanel.Height*0.1F),
new PointF(graficaPanel.Width*0.1F+5,graficaPanel.Height*0.1F+10)};
    g.FillPolygon(Brushes.Black, triangulo1);

    //Label Fuerza de Fricción
    g.DrawString("F", new Font("Arial", 14), Brushes.Black,
graficaPanel.Width * 0.05F, graficaPanel.Height * 0.05F);

    //Eje X
    g.DrawLine(Pens.Black, graficaPanel.Width * 0.1F,
graficaPanel.Height * 0.9F,
graficaPanel.Width * 0.9F,
graficaPanel.Height * 0.9F);

    //Flecha Horizontal
    PointF[] triangulo2 = {new PointF(graficaPanel.Width*0.9F-
10,graficaPanel.Height*0.9F + 5),
new PointF(graficaPanel.Width*0.9F,graficaPanel.Height*0.9F),
new PointF(graficaPanel.Width*0.9F-10,graficaPanel.Height*0.9F-5)};
    g.FillPolygon(Brushes.Black, triangulo2);

    //Label RPM
    g.DrawString("R.P.M.", new Font("Arial", 13), Brushes.Black,
graficaPanel.Width * 0.91F, graficaPanel.Height * 0.9F);

    if (puntosGrafica != null && puntosGrafica.Length > 2)
    {
        float maximaF = 0;
        float maximaRPM = 0;
        for (int i = 0; i < puntosGrafica.Length; i++)
        {
            if (puntosGrafica[i].X > maximaRPM)
                maximaRPM = puntosGrafica[i].X;
            if (puntosGrafica[i].Y > maximaF)
                maximaF = puntosGrafica[i].Y;
        }
        for (int i = 1; i <= 9; i++)
        {
            g.DrawLine(Pens.Black, graficaPanel.Width * 0.1F -
5, graficaPanel.Height * 0.9F - graficaPanel.Height * 0.08F * i,
```

```

graficaPanel.Width * 0.1F + 5, graficaPanel.Height * 0.9F -
graficaPanel.Height * 0.08F * i);
    g.DrawString((maximaF / 9F * i).ToString("N1"), new
Font("Arial", 8), Brushes.Black, graficaPanel.Width * 0.05F,
graficaPanel.Height * 0.9F - graficaPanel.Height * 0.08F * i - 5);
    g.DrawLine(Pens.Black, graficaPanel.Width * 0.1F +
graficaPanel.Width * 0.08F * i, graficaPanel.Height * 0.9F - 5,
graficaPanel.Width * 0.1F + graficaPanel.Width * 0.08F * i,
graficaPanel.Height * 0.9F + 5);
    g.DrawString((maximaRPM / 9F * i).ToString("N1"),
new Font("Arial", 8), Brushes.Black, graficaPanel.Width * 0.1F +
graficaPanel.Width * 0.08F * i - 5, graficaPanel.Height * 0.9F + 6);
}
for (int x = 1; x < puntosGrafica.Length; x++)
    g.DrawLine(Pens.Blue,
graficaPanel.Width * 0.1F + puntosGrafica[x - 1].X / maximaRPM * 9 *
graficaPanel.Width * 0.08F,
graficaPanel.Height * 0.9F - puntosGrafica[x - 1].Y / maximaF * 9 *
graficaPanel.Height * 0.08F,
graficaPanel.Width * 0.1F + puntosGrafica[x].X / maximaRPM * 9 *
graficaPanel.Width * 0.08F,
graficaPanel.Height * 0.9F - puntosGrafica[x].Y / maximaF * 9 *
graficaPanel.Height * 0.08F);

for (int i = 0; i < puntosGrafica.Length; i++)
{
    g.FillEllipse(Brushes.Red,
graficaPanel.Width * 0.1F + puntosGrafica[i].X / maximaRPM * 9 *
graficaPanel.Width * 0.08F - 2,
graficaPanel.Height * 0.9F - puntosGrafica[i].Y / maximaF * 9 *
graficaPanel.Height * 0.08F - 2,
4, 4);
}
else
{
for (int i = 1; i <= 9; i++)
{
    g.DrawLine(Pens.Black, graficaPanel.Width * 0.1F -
5, graficaPanel.Height * 0.9F - graficaPanel.Height * 0.08F * i,
graficaPanel.Width * 0.1F + 5, graficaPanel.Height * 0.9F -
graficaPanel.Height * 0.08F * i);
    g.DrawString((1 / 9F * i).ToString("N1"), new
Font("Arial", 8), Brushes.Black, graficaPanel.Width * 0.05F,
graficaPanel.Height * 0.9F - graficaPanel.Height * 0.08F * i);
    g.DrawLine(Pens.Black, graficaPanel.Width * 0.1F +
graficaPanel.Width * 0.08F * i, graficaPanel.Height * 0.9F - 5,
graficaPanel.Width * 0.1F + graficaPanel.Width * 0.08F * i,
graficaPanel.Height * 0.9F + 5);
    g.DrawString((1 / 9F * i).ToString("N1"), new
Font("Arial", 8), Brushes.Black, graficaPanel.Width * 0.1F +
graficaPanel.Width * 0.08F * i, graficaPanel.Height * 0.9F + 10);
}
}
}

```

## Anexo A.6: Función para dibujar Gráfica de Calibración.

```
private void grafCalibracion_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.Clear(Color.White);

    float x1 = grafCalibracion.Width * 0.05F;
    float x2 = grafCalibracion.Width * 0.85F;
    float y1 = grafCalibracion.Height * 0.1F;
    float y2 = grafCalibracion.Height * 0.85F;

    //Eje vertical
    g.DrawLine(Pens.Black, x1, y1, x1, y2);
    PointF[] triangulo = {new PointF(x1-5,y1),
                          new PointF(x1,y1-5),
                          new PointF(x1+5,y1)};
    g.FillPolygon(Brushes.Black, triangulo);
    g.DrawString("P", new Font("Arial", 8), Brushes.Black, x1-10,
y1-20);

    //Eje horizontal
    g.DrawLine(Pens.Black, x1, y2, x2, y2);
    PointF[] triangulo2 = {new PointF(x2,y2-5),
                          new PointF(x2+5,y2),
                          new PointF(x2,y2+5)};
    g.FillPolygon(Brushes.Black, triangulo2);
    g.DrawString("R.P.M.", new Font("Arial", 8), Brushes.Black,
x2+12, y2);

    if (puntosCalib != null && puntosCalib.Length > 2)
    {
        float maximaF = 0;
        float maximaRPM = 0;
        for (int i = 0; i < puntosCalib.Length; i++)
        {
            if (puntosCalib[i].X > maximaRPM)
                maximaRPM = puntosCalib[i].X;
            if (puntosCalib[i].Y > maximaF)
                maximaF = puntosCalib[i].Y;
        }

        for (int i = 1; i <= 9; i++)
        {
            g.DrawLine(Pens.Black, x1 - 3, y2 - (y2 - y1) / 9
* i, x1 + 5, y2 - (y2 - y1) / 9 * i);
            g.DrawString((maximaF / 9F * i).ToString("N1"),
new Font("Arial", 8), Brushes.Black, x1 - 22, y2 - (y2 - y1) / 9 * i
- 5);
            g.DrawLine(Pens.Black, x1 + (x2 - x1) / 9 * i, y2
- 5, x1 + (x2 - x1) / 9 * i, y2 + 5);
            g.DrawString((maximaRPM / 9F * i).ToString("N1"),
new Font("Arial", 8), Brushes.Black, x1 + (x2 - x1) / 9 * i - 5, y2 +
6);
        }
    }
}
```

```

    }
    if (maximaF != 0 && maximaRPM != 0)
        for (int x = 1; x < puntosCalib.Length; x++)
            g.DrawLine(Pens.Blue, x1 + puntosCalib[x - 1].X /
maximaRPM * (x2 - x1), y2 - puntosCalib[x - 1].Y / maximaF * (y2 -
y1), x1 + puntosCalib[x].X / maximaRPM * (x2 - x1), y2 -
puntosCalib[x].Y / maximaF * (y2 - y1));

        for (int i = 0; i < puntosCalib.Length; i++)
            g.FillEllipse(Brushes.Red, x1 + puntosCalib[i].X /
maximaRPM * (x2 - x1) - 2, y2 - puntosCalib[i].Y / maximaF
* (y2 - y1) - 2, 4, 4);

    }
    else
    {
        for (int i = 1; i <= 9; i++)
        {
            g.DrawLine(Pens.Black, x1 - 3, y2 - (y2-y1)/9 * i,
x1 + 5, y2 - (y2-y1)/9 * i);
            g.DrawString((1 / 9F * i).ToString("N1"), new
Font("Arial", 8), Brushes.Black, x1 - 22, y2 - (y2-y1)/9 * i - 5);
            g.DrawLine(Pens.Black, x1 + (x2-x1)/9 * i, y2 - 5,
x1 + (x2-x1)/9 * i, y2 + 5);
            g.DrawString((1 / 9F * i).ToString("N1"), new
Font("Arial", 8), Brushes.Black, x1 + (x2-x1)/9 * i - 5, y2 + 6);
        }
    }
}

```

### Anexo A.7: Función para dibujar Diagrama.

```
private void dibujarDiagrama(object sender, PaintEventArgs e)
{
    Graphics g = panell1.CreateGraphics();
    g.DrawImage(Image.FromFile("Diagrama.png"), 0, 0, panell1.Width,
panell1.Height);
}
```

### Anexo A.8: Función botón Inicio/Fin Calibración.

```
private void Calib_Click(object sender, EventArgs e)
{
    try
    {
        if (Calib.Text.CompareTo("Inicio Calibración") == 0)
        {
            calibrando = true;
            play_Click(sender, e);
            if (!play4.Enabled)
                Calib.Text = "Fin Calibración";
        }
        else
        {
            calibrando = false;
            stop_Click(sender, e);
            Calib.Text = "Inicio Calibración";
        }
    }
    catch (Exception error)
    { }
}
```

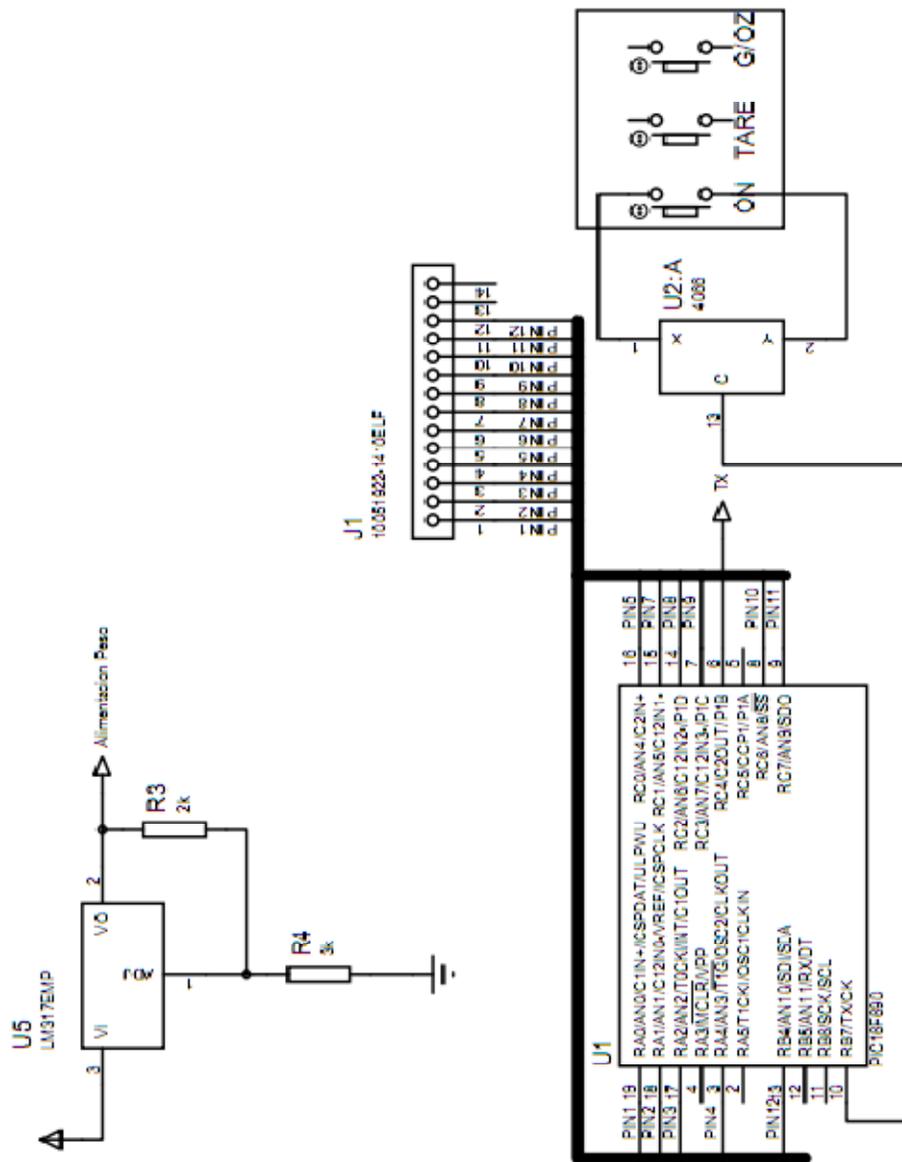
## Anexo A.9: Función Recepción datos del Circuito.

```
private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    try
    {
        if (!leyendoDisp)
        {
            string dato = serialPort1.ReadLine();
            string[] Datos = dato.Split(("
\r\n").ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
            Datos[0] = Datos[0].Substring(2);
            Datos[1] = Datos[1].Substring(2);

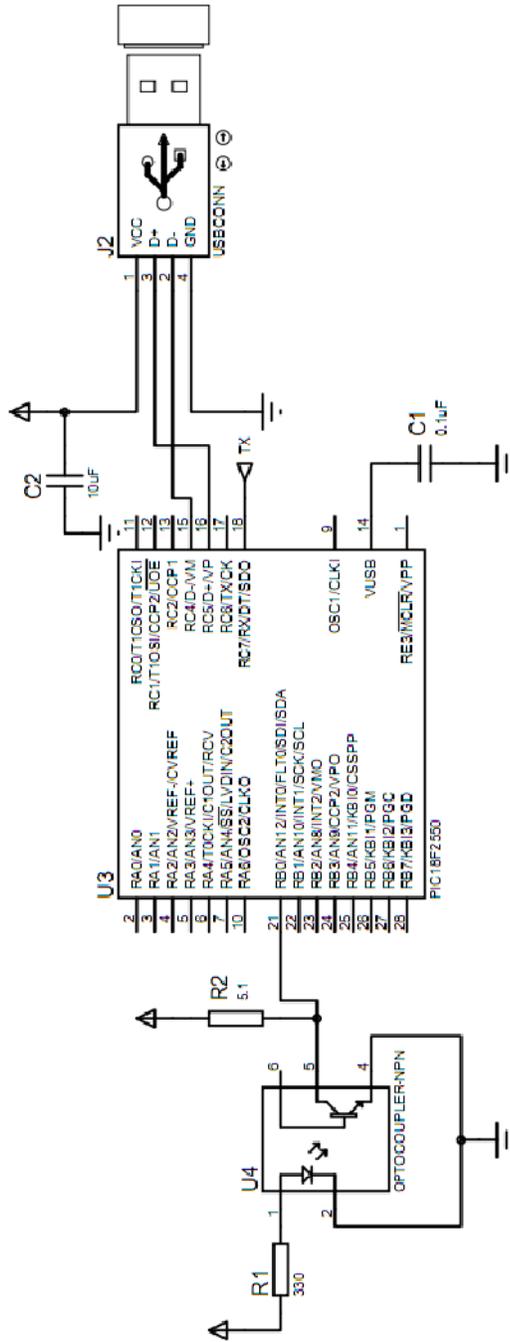
            int vel = int.Parse(Datos[1].ToString());
            vel = vel * 2 * 60;
            velocidad_textBox.Text = (vel).ToString("N1") + "";

            float peso = int.Parse(Datos[0].ToString());
            peso = peso / 1000 * 9.8F;
            peso_textBox.Text = (peso).ToString("N3") + "";
            if (!calibrando && puntosCalib != null &&
puntosCalib.Length > 1)
            {
                float friccion = (peso * R1 -
interpolarCalibracion(vel) * R1) / R2;
                friccion_textBox.Text =
(friccion).ToString("N2");
                coef_roce_textBox.Text = (friccion /
((float)pesoCarga.Value * 9.8F)).ToString("N2");
            }
        }
    }
    catch (Exception error) { }
}
```

Anexo B:  
Esquemas, figuras y tablas



Anexo B.1. Esquema electrónico del Microcontrolador PIC16F690 en el sistema.



Anexo B.2. Esquema electrónico del Microcontrolador PIC18F2550 y Regulador de Voltaje 3V LM317 en el sistema.



Apreciación: 0,01 mm  
Rango: 0 – 9 mm

### Anexo B.3. Reloj Comparador Mitutoyo

**Primeros pasos:**

1. Iniciar el proceso de Calibración para obtener la gráfica de Fuerza vs. R.P.M. con el motor en vacío.
2. Detener el proceso de Calibración. **Nota:** Es necesario detener el proceso de Calibración en el botón *Fin de Calibración* antes de realizar cualquier otra acción.
3. Una vez obtenidos los datos de la calibración se puede proceder a obtener la gráfica y llenar la tabla en las respectivas pestañas. Para ello se debe presionar el botón *Play* y luego de estabilizarlo en cada valor deseado, presionar el botón *Capturar*.

**Calibración**

1. Presionar sobre la pestaña *Calibración*.
2. Hacer Click sobre el botón *Inicio Calibración*. En éste momento se empezarán a recibir las lecturas provenientes del circuito.
3. Al obtener un valor deseado, presionar el botón *Capturar* para guardar el valor en la tabla de calibración y en la gráfica.
4. Una vez obtenida la cantidad de puntos deseados (recomendable 5 como mínimo), pulsar el botón *Fin de Calibración* para terminar el proceso de Calibración.

**Menú Archivo:**

*Exportar a Excel:* abre un nuevo documento de Excel con los datos de la calibración contenidos en la hoja *Calibración* y los datos de la adquisición, en la hoja *Datos* para permitir al usuario su posterior estudio y análisis.

RESULTADOS FRICCIÓN														
EVENTOS	NÚMERO DE MUESTRA	CARGA (N)	DISTANCIA (m)	VELOCIDAD (m/s)	TIEMPO (min)	VALOR INICIAL (mm)	VALOR FINAL (mm)	DEFLEXIÓN (mm)	CARGA P (N)	T' (Nm)	Tf (Nm)	T (Nm)	f (N)	μk
1	12	17.10	500	3.14	2.65	5.97	6.45	0.48	3.21	0.15	0.03	0.13	8.43	0.49
2	10	17.10	2,000	3.14	10.61	5.95	6.25	0.30	2.00	0.10	0.03	0.07	4.60	0.27
3	14	17.10	5,000	3.14	26.53	5.98	6.30	0.32	2.13	0.10	0.03	0.08	5.02	0.29
4	4	12.20	500	3.14	2.65	5.27	5.70	0.43	2.87	0.14	0.03	0.11	7.36	0.60
5	3	12.20	2,000	3.14	10.61	4.22	4.60	0.38	2.94	0.14	0.03	0.11	7.56	0.62
6	1	12.20	5,000	3.14	26.53	4.36	4.70	0.34	2.27	0.11	0.03	0.08	5.44	0.45
7	9	7.30	500	3.14	2.65	4.72	5.05	0.33	2.20	0.10	0.03	0.08	5.23	0.72
8	7	7.30	2,000	3.14	10.61	4.23	4.50	0.27	1.80	0.09	0.03	0.06	3.96	0.54
9	6	7.30	5,000	3.14	26.53	4.23	4.46	0.23	1.53	0.07	0.03	0.05	3.12	0.43
10	13	17.10	500	3.14	2.65	5.95	6.40	0.45	3.01	0.14	0.03	0.12	7.79	0.46
11	2	12.20	5,000	3.14	26.53	4.21	4.60	0.39	2.60	0.12	0.03	0.10	6.51	0.53
12	11	17.10	2,000	3.14	10.61	5.90	6.25	0.35	2.33	0.11	0.03	0.08	5.65	0.33
13	5	12.20	500	3.14	2.65	4.23	4.60	0.37	2.47	0.12	0.03	0.09	6.07	0.50
14	8	7.30	2,000	3.14	10.61	4.82	5.10	0.28	1.87	0.09	0.03	0.06	4.17	0.57
				Total min	172.42									
				Horas	2 h : 52 m									

RPM	2,000
RADIO PARA T (m)	0.015
RADIO PARA Tf (m)	0.0135
RADIO PARA T' (m)	0.0475

Promedio Rég. Transitorio	0.55
Desv. Estándar	0.1064
Promedio Rég. Estable	0.45
Desv. Estándar	0.1280

