

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA**

**DISEÑO Y CONSTRUCCIÓN DE PROTOTIPOS  
ELECTRÓNICOS Y EL SOFTWARE PARA LA  
ENSEÑANZA DE LAS CIENCIAS FÍSICAS.**

Tesis previa a la obtención del Título de ingeniero en  
Electrónica y Telecomunicaciones.

**SOJOS DÁVALOS FRANCISCO JAVIER**

**Quito, Septiembre, 1999**

### CERTIFICACIÓN

Certifico que el presente trabajo ha sido desarrollado en su totalidad por el Sr. Francisco Sojos D.

A handwritten signature in black ink, appearing to read 'Carlos Novillo M.', is written over a horizontal line. The signature is enclosed within a hand-drawn oval.

Ing. Carlos Novillo M.  
DIRECTOR DE TESIS

..... *A mi querida esposa,  
cómplice de todos mis más  
grandes ideales y proyectos,  
y a la reconfortante presencia  
de mis padres.*

*Un agradecimiento a Dios, y a todos esos seres queridos quienes me brindaron su paciencia y apoyo durante todos estos años. A todos ellos, Gracias.*

# CONTENIDO

	Página
<b><u>INTRODUCCION</u></b> .....	1
<b><u>CAPITULO I</u></b>	
<b>MANEJO DEL PUERTO PARALELO DEL PC</b> .....	5
1.1 ACERCA DEL PUERTO PARALELO .....	5
1.2 INTERFAZ CENTRONICS .....	8
1.2.1 BUSES EN UN COMPUTADOR PC O COMPATIBLE .....	12
1.2.2 DIRECCIONES POSIBLES DEL PUERTO PARALELO .....	13
1.3 MANEJO DEL PUERTO PARALELO DEL PC .....	14
1.3.1 Registros de Entrada y Salida .....	14
1.3.2 Líneas de Protocolo .....	17
1.3.3 Líneas de Tierra .....	18
1.3.4 ¿Registro de datos bidireccional?.....	19
1.3.5 ELECTRÓNICA DEL PUERTO PARALELO, CARACTERISTICAS .....	19
1.4 Manejo de los registros del puerto paralelo mediante el uso de programación computacional (Software) .....	25
1.5 Ventajas y desventajas de la comunicación paralela respecto a la comunicación serial .....	30
<b><u>CAPITULO II</u></b>	
<b>TEORIA Y DISEÑO DEL INTERFAZ GENERAL DEL PROYECTO</b> .....	32
2.1 ANTECEDENTES GENERALES .....	32
2.1.1 DIAGRAMA DE BLOQUES DEL CIRCUITO INTERFAZ .....	34
2.2 TEORIA DEL FUNCIONAMIENTO DEL INTERFAZ BASICO .....	38
2.2.1 TEORIA DEL FUNCIONAMIENTO DEL "SCHMITT TRIGGER" .....	45
2.2.2 DISEÑO DEL INTERFAZ GENERAL DEL PROYECTO .....	52
2.2.3 ELECCION DE LOS ELEMENTOS PARA EL DISEÑO .....	59
2.3 DISEÑO DEL INTERFAZ PARA EL PROTOTIPO "PENDULO SIMPLE" .....	63
2.4 DISEÑO DEL INTERFAZ PARA EL PROTOTIPO DE "CAIDA LIBRE" .....	67
2.5 DISEÑO DEL INTERFAZ PARA EL PROTOTIPO DE "PLANO INCLINADO" .....	70

## CAPITULO III

<b>DISEÑO DEL PROGRAMA EN VISUAL BASIC .....</b>	<b>74</b>
3.1 DIAGRAMA DE BLOQUES DEL PROGRAMA COMPUTACIONAL .....	74
3.2 VENTANAS DE FORMULARIOS Y CODIGO DE PROGRAMA .....	82
3.3 FUNCIONES PUBLICAS UTILIZADAS EN EL PROGRAMA .....	123
3.4 UTILIZACION DE LA HERRAMIENTA IOPORT.OCX .....	135

## CAPITULO IV

<b>RESULTADOS .....</b>	<b>138</b>
4.1 CAMBIO DEL SISTEMA DE ILUMINACION DE LUZ VISIBLE A LUZ INFRARROJA .....	138
4.2 PRUEBAS DE ELECTRONICA .....	145
4.2.1 TABLA DE RESULTADOS EXPERIMENTALES DEL INTERFAZ “PENDULO SIMPLE” .....	145
4.2.2 TABLA DE RESULTADOS EXPERIMENTALES DEL INTERFAZ “CAIDA LIBRE” .....	147
4.2.3 TABLA DE RESULTADOS EXPERIMENTALES DEL INTERFAZ “PLANO INCLINADO” .....	150
4.3 PRUEBAS DE LOS INTERFACES ELECTRÓNICOS MEDIANTE EL USO DE RUTINAS DE PROGRAMA (SOFTWARE) .....	152
4.4 CONCLUSIONES Y RECOMENDACIONES .....	156

## CAPITULO V

<b>MANUAL DEL USUARIO .....</b>	<b>163</b>
5.1 OPERACIÓN DE LOS PROTOTIPOS POR EL USUARIO .....	163
5.1.1 PENDULO SIMPLE.....	163
5.1.2 PROCEDIMIENTOS PARA REALIZAR LA PRACTICA DE “PENDULO SIMPLE” .....	167
5.1.3 CAIDA LIBRE .....	173
5.1.4 PROCEDIMIENTOS PARA REALIZAR LA PRACTICA DE “CAIDA LIBRE” .....	176
5.1.5 PLANO INCLINADO .....	182
5.1.6 PROCEDIMIENTOS PARA REALIZAR LA PRACTICA DE “PLANO INCLINADO” .....	186
5.2 CALIBRACION ELECTRONICA DEL EQUIPO .....	191
5.2.1 CALIBRACION DEL INTERFAZ “PENDULO SIMPLE”.....	191
5.2.2 CALIBRACION DEL INTERFAZ “CAIDA LIBRE” Y “PLANO INCLINADO”.....	194

5.3 MANTENIMIENTO DE LOS EQUIPOS .....	196
5.4 MANEJO DEL PAQUETE COMPUTACIONAL DEL PROYECTO .....	198
5.4.1 INTRODUCCION.....	198
5.4.2 MENU DE OPCIONES .....	200
5.4.2.1 VENTANA DE ACCESO DIRECTO .....	200
5.4.2.2 OPCION "EXPERIMENTO" .....	201
5.4.2.2.1 SIMULACIÓN .....	202
5.4.2.2.2 PRÁCTICA .....	208
5.4.2.3 OPCION "UNIDADES" .....	212
5.4.2.3.1 SISTEMA DE UNIDADES MKS (Metro-Kilogramo-Segundo) .....	212
5.4.2.3.2 SISTEMA DE UNIDADES CGS (Centímetro-Gramo-Segundo) .....	212
5.4.2.3.3 SISTEMA DE UNIDADES INGLES (Pie-Libra-Segundo) .....	213
5.4.2.4 OPCION "VENTANA" .....	213
5.4.2.5 OPCION "AYUDA" .....	214

## ANEXOS

<b>A. DIAGRAMAS DE LOS MODULOS FISICOS .....</b>	<b>215</b>
A.1 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO	
"PENDULO SIMPLE" .....	215
A.1.1 Esquemas Gráficos .....	215
A.1.2 Tabla de longitudes físicas en [cm] .....	220
A.2 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO	
"CAIDA LIBRE" .....	220
A.2.1 Esquemas Gráficos .....	220
A.2.2 Tabla de longitudes físicas en [cm] .....	223
A.3 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO	
"PLANO INCLINADO" .....	224
A.3.1 Esquemas Gráficos .....	224
A.3.2 Tabla de longitudes físicas en [cm] .....	228
<b>B. DISEÑO DE LOS CIRCUITOS IMPRESOS .....</b>	<b>229</b>
B.1 DISEÑO DEL CIRCUITO IMPRESO DE "PENDULO SIMPLE" .....	229
B.2 DISEÑO DEL CIRCUITO IMPRESO DE "CAIDA LIBRE" .....	230
B.3 DISEÑO DEL CIRCUITO IMPRESO DE "PLANO INCLINADO" .....	231
<b>C. MATERIALES UTILIZADOS .....</b>	<b>232</b>
C.1 LISTA DE MATERIALES DEL PROTOTIPO "PENDULO SIMPLE" .....	232
C.2 LISTA DE MATERIALES DEL PROTOTIPO "CAIDA LIBRE" .....	233
C.3 LISTA DE MATERIALES DEL PROTOTIPO "PLANO INCLINADO" .....	234

<b>D. DIAGRAMAS ELECTRONICOS FINALES .....</b>	<b>235</b>
D.1 PENDULO SIMPLE .....	236
D.2 CAIDA LIBRE .....	237
D.3 PLANO INCLINADO .....	238
<b>E. SITIOS WEB IMPORTANTES PARA VISITAR .....</b>	<b>239</b>
E.1 SITIOS WEB PARA OBTENER EL "IOPORT.OCX" .....	239
E.2 INFORMACION ADICIONAL PARA EL PUERTO PARALELO .....	241
<b>F. AYUDA DEL PROGRAMA COMPUTACIONAL "EFC" .....</b>	<b>245</b>
<b>G. DATOS TECNICOS DE LOS SEMICONDUCTORES .....</b>	<b>258</b>

## **BIBLIOGRAFÍA**

BIBLIOGRAFÍA .....	274
--------------------	-----

## INTRODUCCIÓN

El uso de computadores en los campos de la investigación y la educación ha crecido durante las décadas de los 70 y 80. Actualmente, en la década de los 90, aumentó drásticamente con la introducción de los microcontroladores en el mercado. Así los grandes proveedores de estos pequeños dispositivos surgieron, como es el caso de Intel, Motorola, Siemens, Allen Bradley, Philips, British Brown-Bovari, etc., y desplazaron a los ya obsoletos microprocesadores Z-80, 6809, 6502, 8080 y 68000; que aunque fueron importantes en su época, hoy en día son la base de lo utilizado para la comunicación con dispositivos en el medio externo.

En 1980 aproximadamente, los fabricantes de circuitos integrados diseñaron un microcomputador integrado en una sola carcasa. Este tomó el nombre de microcontrolador. Definitivamente lo que diferencia al microprocesador del microcontrolador es que, aunque el primero contiene la CPU (Central Processing Unit), para implementar un microcomputador se necesita de elementos adicionales como: Memoria RAM, memoria ROM y circuitos de Entrada/Salida (I/O) para la comunicación externa. Además, para la transferencia y procesamiento de la información se necesitan tres buses: *Bus de datos*, *bus de dirección* y *líneas de control*.

El microcontrolador, sin embargo, reúne todos estos elementos en un solo paquete. Aún así existe una desventaja, éste dispositivo opera a velocidades relativamente muy bajas frente a las velocidades de operación del microprocesador actual.

Un microcontrolador tiene una memoria RAM (de datos) pequeña. Esto

tiene su razón de ser, pues fue diseñado para almacenamiento limitado de información temporal. Igualmente la memoria ROM (de programa), es de baja capacidad de almacenamiento.

Algo importante de mencionar es que la CPU del microprocesador es más compleja que la de un microcontrolador, pues las instrucciones en este último, están más orientadas a la operación de los bits de entrada y salida.

Fuera de todo este análisis, la comunicación con periféricos externos hoy en día se la hace utilizando estos dispositivos, y aunque la elección del elemento adecuado dependerá de las necesidades del usuario, se dispone de una gran gama de circuitos integrados en el mercado para los diferentes propósitos.

Este proyecto de tesis utiliza un microcomputador, y todo el sistema electrónico diseñado y construido, será operado a través de las instrucciones provenientes del microprocesador del equipo. Adicionalmente, es importante mencionar las siguientes ventajas:

- El costo de construcción del interfaz es reducido.
- Un computador de escritorio, trabajando bajo Windows 95 o 98, es muy convencional en nuestro medio.
- Debido al uso del computador y de todos los beneficios que presenta la programación virtual actualmente, se diseñará un programa computacional utilizando un paquete conocido como el Visual Basic.
- No se subutilizarían las opciones de conexión externa que posee el microcontrolador.

Se inicia entonces, a través del *capítulo I*, con el estudio del puerto paralelo de un computador, con el objeto de conocer las direcciones y los registros

que hacen posible que el microprocesador tenga contacto directo con el mundo exterior, además se estudiarán los bits que conforman cada uno de éstos registros y su funcionalidad. En el *capítulo II*, se procederá con el diseño de la tarjeta para servir de interfaz entre el equipo de análisis (computador) y el medio externo.

El *capítulo III*, está destinado a la elaboración del programa computacional (software), el cual funcionará en ambientes bajo windows. Las rutinas de programación serán ejecutadas en **Visual Basic**, y para ello se requerirá la utilización de un computador, capaz de soportar las características de operación de este lenguaje virtual. Posteriormente, en el *capítulo IV*, se mostrarán los resultados obtenidos a lo largo del diseño. Se realizarán las pruebas de electrónica, así como las pruebas de software, señalándose los valores obtenidos en las mediciones de prueba de los puntos importantes del circuito interfaz. Finalmente se establecerán los cambios que debieron realizarse para la culminación exitosa del proyecto.

El *capítulo V*, corresponde al manual del usuario. En éste se describe la adecuada manipulación de los equipos físicos, la calibración del circuito interfaz, el mantenimiento y la utilización del programa computacional, específicamente creado para el proyecto.

Cabe mencionar, finalmente, que este plan de tesis tuvo un periodo de experimentación (*Fase Piloto*), en un proyecto que fue concluido en la “**Universidad Tecnológica Equinoccial**” (UTE), bajo financiamiento de la “**Fundación para la Ciencia y la Tecnología**” (FUNDACYT). Luego del éxito obtenido, los prototipos construidos, se destinaron a los principales colegios de las ciudades de Quito, Guayaquil y Cuenca con el apoyo del **Ministerio de Educación y Cultura (MEC)**.

Dicho proyecto, en sus primeras dos fases, tomó el nombre: **COMPULAB**

I y COMPULAB II respectivamente, además permitió la obtención de las bases teórico-experimentales, necesarias para concluir el presente trabajo. De hecho se ha pensado que ésta tesis puede servir como apoyo didáctico y funcional para el mejor aprendizaje y comprensión de los fenómenos físicos: **“Péndulo Simple, Caída Libre y Plano Inclinado”**, en los niveles medio y superior: Prepolitécnico y primer año de estudios de la **Escuela Politécnica Nacional**, e instituciones afines (o similares).

# CAPITULO I

## MANEJO DEL PUERTO PARALELO DEL PC.

### 1.1 ACERCA DEL PUERTO PARALELO.

Si se revisa la década de los años 80 se verá que, en algunas partes del mundo, varios fabricantes de computadoras comienzan a construir sus propios equipos pequeños, a tal punto, que la competencia a nivel marcas se acentuó fuertemente en dos regiones: EEUU y el Reino Unido. Aparecen por un lado: Apple, IBM, Commodore, ATARI, Radio Shack, NCR, entre las principales y, por el otro: BBC Micro, ORIC, Sinclair, Tandy, Osborne, etc. Luego países como Japón, España, entre otros se adicionaron a la producción de sus propias máquinas.

El enfoque especial de todos estos equipos fue destinado innatamente a la educación; sin embargo, y muy particularmente, Apple, BBC Micro, Commodore y Sinclair, mostraron sus dotes de ser microcomputadoras destinadas a otros usos específicos.

Así se comenzaron a construir equipos externos para apoyo de enfermos, minusválidos y demás personas que necesitaban un mejor desenvolvimiento en la sociedad de su alrededor. Estos fueron “accesorios” destinados a conectarse en sus computadoras con ayuda de programas computacionales adicionales (*software*). La conexión se realizaba a través de interfaces destinados específicamente para el aparato o a través de puertos específicos incluidos en el

microcomputador. La mayoría de estos puertos fueron del tipo paralelo. Es decir, permitían la entrada y salida de un byte de datos.

Alrededor de 1981, IBM introduce su puerto paralelo y éste comienza a evolucionar paulatinamente incorporando nuevas ventajas en su arquitectura y diseño. IBM fue una marca que se dedicó específicamente a los negocios y a la parte administrativa. De hecho, sus conexiones periféricas no iban más allá de conectarse con equipos de salida (*impresoras, plotters, etc.*), y sus únicos dispositivos de entrada eran modems, joysticks, como extraordinarios elementos de conexión. Así su puerto paralelo se enfocó solamente para salida de datos destinado a una impresora y no permitía velocidades más allá de los 150 Kbps.

Al pasar los años, a finales de los 80's y comienzos de los 90's, las marcas débiles comenzaron a desaparecer absorbidas por otras y en especial nuevas inversiones definieron el poder en el mercado de las computadoras. Así IBM con el sistema PS/2 y Apple con el Macintosh monopolizaron los principales campos de ventas mundiales.

IBM ofreció dos puertos paralelos dentro de su sistema. El primero incluido en un adaptador de video monocromático MDA (*Monochrome Display Adapter*) y el segundo fue denominado "Adaptador Paralelo para Impresora" PPA (*Parallel Printer Adapter*).

Este primer puerto ofrecido era directo y no necesitaba de alguna tarjeta de expansión adicional.

En 1987, IBM diseñó el puerto paralelo para ser utilizado como bidireccional gracias al manejo de un bit del registro de control. Aquel bit fue el bit #5 y un 1 lógico en éste, habilitaba al puerto paralelo para recibir datos.

Según otros estudios, se indica que en otras computadoras el bit de control de bidireccionalidad es el bit #7.

En 1991, los principales constructores de impresoras conformaron la **NPA** “*Network Printing Alliance*” que permitió la estandarización del puerto paralelo para usos de impresión y manejo de dispositivos externos.

La **NPA** fue la que finalmente creó la norma **IEEE 1284** que define al puerto paralelo como un puerto compatible totalmente con los antiguos, capaz de manejar velocidades de hasta 1 Mbyte/sec, además de ser bidireccional. El comité encargado, aprobó esta norma en marzo de 1994.

Debido a la época de transición vivida por el puerto paralelo durante las últimas décadas, se ha decidido tratar en éste trabajo considerando al mismo como un puerto paralelo unidireccional, hasta que se tenga la certeza de contar con un puerto de características bidireccionales en todas las máquinas del mercado. Dicha determinación permite que los prototipos que van a construirse en éste trabajo puedan conectarse a cualquier computador IBM o compatible con tecnología INTEL 80486 o superior, trabajando bajo Windows 95 o 98 con un mínimo de 16 Mbytes en RAM y 20 Mbytes libres en disco duro. Se prefiere, si fuese posible, la utilización de un equipo más sofisticado para efectos de un mejor aprovechamiento de los prototipos construidos, en cuanto a velocidad de procesamiento y visualización gráfica (VGA o SVGA con 256 colores).

## 1.2 INTERFAZ CENTRONICS.

En un interfaz se definen 4 tipos de especificaciones importantes:

1. *Especificaciones mecánicas.*
2. *Especificaciones eléctricas.*
3. *Especificaciones funcionales.*
4. *Especificaciones de procedimiento.*

Las cuatro especificaciones anteriores definen completamente al interfaz Centronics y, en resumen, se detallará lo más importante de ellas.

1. **Especificaciones mecánicas:** Centronics utiliza un conector tipo “Amphenol” de 36 pines del lado de la impresora y un conector **DB-25** del lado del computador. El conector DB-25 es del tipo hembra, se encuentra en la parte posterior del computador y tiene la siguiente forma:

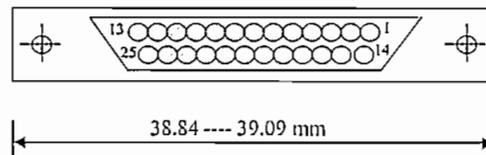


Figura 1.1.- Vista del conector DB-25 H del puerto paralelo.

2. **Especificaciones Eléctricas:** Lógica tipo TTL (*Transistor-Transistor Logic*), es decir:

0L : 0.0V a 0.8 V. (OFF)

1L : 2.4V a 5.0 V. (ON)

*Velocidad de Transmisión máxima:* Bajo la nueva norma implementada IEEE 1284, la velocidad de Transmisión máxima es de 1 Mbyte/sec.

*Distancia máxima de conexión longitudinal de cable:* De 8 a 15 pies (de 2.44 a 4.57 metros), sin embargo se puede incrementar la longitud de alcance físico adicionando extensiones de pórtico.

*Tipo de Transmisión Balanceada.*

*Pines complementados en el Registro de Datos:* Ninguno.

*Pines complementados en el Registro de Status (de estado):* Pin #11 (*Busy*).

*Pines complementados en el Registro de Control:* Pines #1, 14, y 17 (*Strobe, Auto-feed, y Select-Input*).

**3. Especificaciones Funcionales:** En la interfaz centronics se manejan 4 tipos de señales:

- a) *Señales de Datos.*
- b) *Señales de Status (Estado).*
- c) *Señales de Control.*
- d) *Tierras.*

a) **Señales de Datos:** D0 - D7, 8 pines de datos:

DATA	PIN #
D0	2
D1	3
D2	4
D3	5
D4	6
D5	7
D6	8
D7	9

Tabla 1.1.- Disposición física del byte de datos en el DB-25 H.

NOTA: Se consideran los pines como unidireccionales, y sólo para salida de datos.

b) Señales de Status (Estado):

SEÑAL	PIN #
Acknowledge	10
Busy	11
Paper end	12
Select	13
Error	15

Tabla 1.2.- Registro de estado del puerto paralelo.

Donde:

*Acknowledge = Reconocimiento.*

*Busy = Ocupado.*

*Paper end = No hay papel.*

*Select = Selección.*

*Error = Error.*

NOTA 1: Son pines para entrada. “Acknowledge” se utiliza para provocar una interrupción a nivel de la circuitería (hardware). Un flanco negativo en el pin 10, provocará la interrupción. En algunos puertos se requiere de un flanco positivo. Para esto es necesario que el bit 4 del registro de control, se ubique con 11.

NOTA 2: “Busy, Acknowledge”, son líneas de protocolo usadas para la comunicación entre un sistema rápido (*Computador*) y un periférico lento (*Impresora*), ver sección 1.3.2.

c) Señales de Control:

SEÑAL	PIN #
Strobe	1
Auto-Feed	14
Initialize	16
Select Input	17

Tabla 1.3.- Registro de Control del puerto paralelo.

Donde:

*Strobe* = Validación de datos.

*Auto-feed* = Salto de Línea.

*Initialize* = Inicialización.

*Select-Input* = Entrada de Selección.

NOTA : Todos los pines son de entrada y salida. Las salidas son en colector abierto (ver sección 1.3.5.). “Strobe” es parte de las líneas de protocolo antes mencionadas.

d) Tierras: 8 tierras denominadas así:

Tierra	PIN #
GND_a	18
GND_b	19
GND_c	20
GND_d	21
GND_e	22
GND_f	23
GND_g	24
GND_h	25

Tabla 1.4.- Disposición de las tierras en el puerto paralelo.

**NOTA:** Estas líneas permiten el blindaje de las respectivas señales de datos evitándose la interferencia. En el caso de utilizarse cable plano de varias vías, se evita con esta medida el acoplamiento capacitivo entre los conductores paralelos.

**4. Especificaciones de Procedimiento:** Aquí se definen los procedimientos para la comunicación por el puerto paralelo, **half duplex** (*transmisión o recepción, una a la vez*) o **full duplex** (*transmisión y recepción, al mismo tiempo*), además de la configuración para construir el cable de comunicaciones; sin embargo, ésta parte no es motivo de estudio en este trabajo, pues se dedicará la atención a la recepción de datos particularmente.

#### **1.2.1 BUSES EN UN COMPUTADOR PC O COMPATIBLE.**

Dentro del computador se manejan dos tipos de buses:

**Bus Interno:** En este bus se encuentran las líneas de datos, de dirección y de control, y algunas líneas de señal de otros circuitos del PC. Aquí se pueden instalar tarjetas de adquisición de datos (*con un mejor tratamiento en cuanto a velocidad de muestreo se refiere*), en terminales o conectores llamados “**Slots**”.

**Bus Externo:** Conformado por dos puertos: El serial y el paralelo. Para efectos de estudio se analiza el puerto paralelo. (*Ver Interfaz Centronics, sección 1.2*). A

través de éstos puertos se conectan los diferentes tipos de periféricos que existen en el mercado y se conectarán también los equipos a construirse en este trabajo.

A diferencia del puerto serial, el puerto paralelo permite sólo comunicaciones de datos establecidas entre equipos situados a corta distancia de conexión. (*Ver Especificaciones Eléctricas, sección 1.2, y sección 1.5*).

### **1.2.2 DIRECCIONES POSIBLES DEL PUERTO PARALELO.**

En toda la bibliografía de los PC's y del DOS (*Disk Operating System*), se definen tres direcciones posibles denominadas LPT1, LPT2 y LPT3. Donde las siglas LPT significan "*Line Printer*". Cada una de éstas tiene una dirección base específica (3BCh, 378h, 278h respectivamente), y en base a éstos, se numerarán el resto de registros que conforman el puerto.

Como cada puerto maneja 12 líneas de datos de salida y 5 líneas de datos de entrada, se necesitarán manejar 3 palabras de 8 bits cada una para poder manipular todas estas líneas de E/S. Entonces se requiere de tres registros consecutivos para este trabajo y se los verá en la siguiente sección.

Como se mencionó, los puertos LPT tendrán una dirección base particular asignada, por ejemplo:

LPT1 puede usar una de las tres direcciones ó puede tener hasta dos direcciones, esto es utilizar la dirección 3BCh o también la 378h ó ambas.

LPT2 sólo puede tener una dirección de dos establecidas: 378h ó 278h.

LPT3 estará sólo con la dirección 278h.

Cuando arranca el computador, es el ROM BIOS el que se encarga de identificar la dirección del puerto paralelo y el que estudia la existencia de éstos. La forma en cómo lo hace es escribiendo un byte en cada uno de ellos y leyendo lo escrito. Si la lectura del puerto es la misma que la que se escribió, entonces el puerto existe.

### 1.3 MANEJO DEL PUERTO PARALELO DEL PC.

#### 1.3.1 Registros de Entrada y Salida.

Se conoce la existencia de tres direcciones básicas en uso dentro del puerto paralelo. Si se las define en una tabla, se las podría clasificar de la siguiente forma:

DIRECCION BASE HEX	DIRECCION BASE DEC	RANGO HEX
3BCh	956	3BCh-3BFh
378h	888	378h-37Fh
278h	632	278h-27Fh
300h	768	300h-340h

Tabla 1.5.- Direcciones básicas asignadas al puerto paralelo.

Seguro parecerá extraña la aparición repentina de la dirección 300h y es que IBM XT, AT, y los modelos de microprocesadores INTEL superiores al 386, llevan consigo una dirección asignada para desarrollo de prototipos experimentales, cuya base es 300h.

La primera columna de la *tabla 1.5*, indica las posibles direcciones que se definirán como *BASE*. Entonces a partir de ésta, las direcciones *BASE+1* y *BASE+2*, serán estudiadas como localidades con un registro de un número de bits determinado, los cuales se utilizarán para entrada o para salida de acuerdo a la dirección asignada. Resumiendo, la *tabla 1.6* puede ser de ayuda:

DIRECCION	NUMERO DE BITS	ENTRADA	SALIDA	LECTURA	ESCRITURA
BASE	8	*NO	SI	SI	SI
BASE+1	5	SI	NO	SI	NO
BASE+2	4	SI	SI	SI	SI

Tabla 1.6.- Características de los registros de datos, estado y control.

**NOTA :** (\*) Considerando que el puerto paralelo es un puerto unidireccional.

Nótese que la segunda columna de la *tabla 1.6* llamada “**NUMERO DE BITS**”, indica solamente los bits del registro disponibles por el usuario. No deberá confundirse lo que es **ENTRADA** con lo que es **LECTURA** y lo que es **SALIDA** con lo que es **ESCRITURA**, pues las columnas 5 y 6 “**LECTURA Y ESCRITURA**” respectivamente, indicarán sólo el último byte que ha sido procesado por el puerto, y no su externabilidad. En la dirección **BASE** por ejemplo, los datos están diseñados sólo para salidas y es que dentro del puerto, existen dispositivos del tipo “**LATCH**” (Retenedores) 74LS374, flip-flop’s que actúan como buffers de entrada, pero “*leer*” el buffer solo dirá el último valor escrito en estas salidas y no permitirá la “*entrada*” de datos externo, a no ser que

por algún artificio físico se pueda poner las salidas como entradas. Se deberá tener cuidado en esto pues, debido a la poca información que se tenga del puerto, se puede llegar a destruir los circuitos internos del mismo. Esto es la razón fundamental por la que en este trabajo se considera al puerto paralelo con características unidireccionales.

Finalmente como direcciones de entrada se definirán las direcciones **BASE+1**, **BASE+2** y como direcciones de salida se tendrán a **BASE**, **BASE+2** de acuerdo a la última tabla. Ahora sí, se puede describir cómo está distribuida cada dirección con su registro de bits interno:

#### DIRECCION BASE

	D7	D6	D5	D4	D3	D2	D1	D0
	DATA							
# PIN	9	8	7	6	5	4	3	2

Tabla 1.7.- Disposición del registro de datos.

#### DIRECCION BASE+1

	D7	D6	D5	D4	D3	D2	D1	D0
	IN	IN	IN	IN	IN			
	*BUSY	ACK	PAPER END	SELECT	ERROR			
# PIN	11	10	12	13	15			

Tabla 1.8.- Disposición del registro de estado.

**NOTA:** (\*) Indica pin complementado.

## DIRECCION BASE+2

	D7	D6	D5	D4	D3	D2	D1	D0
				INTERNAL CONTROL	IN/OUT	IN/OUT	IN/OUT	IN/OUT
				** IRQ ENABLE	*SELECT INPUT	INITIALIZE	*AUTO-FEED	*STROBE
# PIN DB-25					17	16	14	1
# PIN CENT.					36	31	14	1

Tabla 1.9.- Disposición del registro de control.

### NOTAS:

1. (\*) Indican pines complementados.
2. (\*\*) El bit 4 es utilizado para la habilitación del **IRQ7**. Este bit puesto en 1, hace que se active la interrupción.
3. Las salidas son de colector abierto. (*Ver sección 1.3.5*).

### 1.3.2 Líneas de Protocolo.

Estas líneas permiten que la comunicación entre el computador y un periférico sea mucho más rápida. Por ejemplo, en el caso de un periférico como una impresora, existen tres líneas de protocolo que son **Strobe**, **Busy** y **Acknowledge**, traducidos como: “Validación de datos, Ocupado y Reconocimiento”, respectivamente. La combinación en el uso de estas líneas, hace que la comunicación de datos sea comprensible y óptima. Ampliando más, estas líneas trabajan así:

**STROBE (Validación de datos):** Es un pulso con el que el computador le indica a la impresora, que el byte o la información emitida es válida. Este bit es complementado. Para que la información enviada al periférico sea reconocida como válida, la transición negativa deberá ocurrir en un tiempo de 1  $\mu$ S.

**BUSY (Ocupado):** Señal que la impresora envía al Computador. Un nivel alto indica el estado de ocupado por parte del periférico y el computador sabrá que temporalmente no puede enviar más información. El pulso deberá ser mayor a 0.5  $\mu$ S en la transición positiva.

**ACKNOWLEDGE (Reconocimiento):** Es una señal procedente del periférico, que indica que ha recibido los datos. Se activa luego de la desactivación de **STROBE**.

### 1.3.3 Líneas de Tierra.

Realizan básicamente dos tareas: Unir las referencias entre el computador y el periférico utilizado para que sean comunes, y realizar el blindaje de cada una de las líneas de información contra ruidos externos que puedan interferir con la transmisión-recepción de los datos, además, contra interferencias entre las mismas líneas de datos debido a la posibilidad de acoplamiento capacitivo (en el caso de utilizarse cable plano de varias vías).

#### 1.3.4 ¿Registro de datos bidireccional?

Se ha discutido ya acerca de la bidireccionalidad del puerto paralelo (*y su evolución desde que aparecieran las primeras máquinas de IBM*). No todas las computadoras en el mercado poseen un puerto paralelo bidireccional, sin embargo, en las que si lo tienen, existe la posibilidad de manipular la dirección de flujo de los datos, usando un bit del registro de control (bit #5) o en otros equipos el (bit #7). Así se tendrá el puerto configurado para entrada o salida, si el bit se encuentra con valores de 1 ó 0 respectivamente.

Los bits del registro de datos (BASE h), véase tabla 1.7, manejan corrientes bajas que no superan los 20 mA de consumo; entonces no se podrá conectar cargas externas demasiado grandes como motores, relés, lámparas, etc. Para esto se deberá diseñar interfaces específicos o circuitos electrónicos especialmente contruidos para esos propósitos. Con la finalidad de conocer la “capacidad” de manejo de corriente que tiene el puerto paralelo, cada bit del registro de datos opera en modo **source/sink** (*Fuente/sumidero*), esto es 2.6/24 mA. Nótese que se tiene una corriente máxima de 24 mA. Valores más altos de corriente, podrían producir daños al puerto.

#### 1.3.5 ELECTRÓNICA DEL PUERTO PARALELO, CARACTERÍSTICAS.

El *registro de datos* está constituido por una combinación de dispositivos electrónicos digitales tipo “LATCH” en donde se pueden apreciar los circuitos integrados 74LS373, de ocho flip-flop’s tipo D y salidas Q tri-state (de tres

estados), controladas por la entrada **G** (habilitación). El pin **OE** (*Output Enable*) está puesto a tierra, esto hace que el integrado opere permitiendo el flujo de información dirigido hacia los pines del puerto. Nótese que al poner **OE** en  $V_{cc}$ , el puerto se volverá unidireccional en el otro sentido, es decir la información entrará desde el exterior, a través del puerto, al computador. Puede considerarse una forma de hacer que el puerto paralelo sea bidireccional, sin embargo, esto implicaría alterar el circuito de la tarjeta principal del computador, volviéndose este procedimiento inadecuado.

También se dispone de un circuito integrado “buffer”, el 74LS244 que corresponde a un *Controlador/Receptor (line driver/line receiver)* con *disparador de Schmitt (Schmitt trigger)* y salidas del tipo *tri-state (tres estados)*. Este circuito integrado recibe el bus de datos de salida del 74LS373 y de acuerdo al pin **G** inversamente habilitado, permitirá que la información ubicada en el puerto ingrese al computador. Cuando el computador ejecuta un comando de lectura del puerto, el 74LS244 es el que realiza este trabajo. Debe recalarse que no se reciben datos del exterior, sino solamente se da lectura al último byte ubicado en el puerto.

El diagrama electrónico del puerto, que tiene relación con el registro de datos (*BASE h*), véase tablas 1.5 y 1.7, se muestra en la *figura 1.2*. En cambio, el diagrama del puerto que corresponde al *registro de status (registro de estado)*, cuya dirección es (*BASE+1 h*), véase tablas 1.5 y 1.8, es muy similar también al diagrama de la *figura 1.2*, pero con la diferencia de que el pin **OE** del 74LS373, está conectado contra  $V_{cc}$ . Es por esto entonces, que el registro (*BASE+1 h*), es

un registro destinado únicamente para la entrada de datos provenientes del exterior.

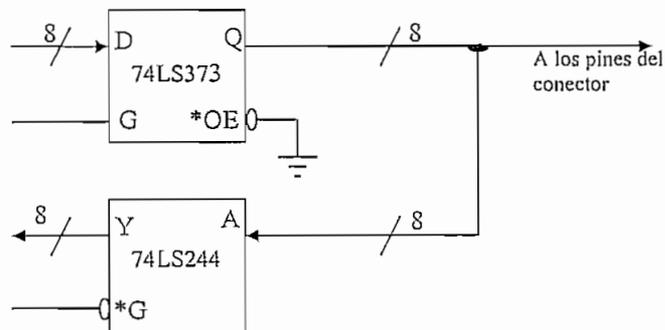


Figura 1.2.- Configuración electrónica del puerto paralelo (Registro de datos).

De la misma forma, se puede realizar una lectura inmediata del puerto cuando se ha ingresado información por el *registro de status* (*registro de estado*), usando una instrucción de entrada propia del lenguaje de programación que disponga el usuario.

Finalmente, dentro del puerto paralelo se tiene el *registro de control*, cuya dirección es ( $BASE+2h$ ), véase tablas 1.5 y 1.9. Los pines del puerto que corresponden a éste registro, tienen la característica de "colector abierto" y deberán manejarse con ayuda de resistencias *pull-up*. Este método permite que los pines trabajen tanto para entrada como para salida de datos.

Si el *registro de control* se utiliza para salida, la instrucción de programa a utilizarse para el envío de un byte, es directa (véase apartado 1.4, Manejo de los registros del puerto paralelo mediante el uso de programación computacional,

*software*). En cambio, si el registro de control se lo utiliza para entrada de datos, previamente se tendrá que programar todos los pines que corresponden a este registro, con 1's. Utilizando QBASIC, se puede ejecutar las siguientes líneas de programa:

```
REM entradas: SELECT-INPUT, INITIALIZE, AUTO-FEED, STROBE del registro de control
DEF SEG = 0
BaseAddress = &H400 + 8
LPT = PEEK(BaseAddress) + (PEEK(BaseAddress + 1) * 256)
OUT (LPT + 2), (INP(LPT + 2) OR 15 : REM decimal 239 en el puerto
OUT (LPT + 2), 4 : REM los cuatro bits están en un nivel alto
```

Las cuatro primeras líneas reconocen el puerto LPT asignado en el computador. La penúltima línea pone el valor decimal 239 en el puerto LPT, es decir, en binario los pines se programan así:

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	0	1	1	1	1

Nótese que solamente nos interesa los 4 bits menos significativos, pues éstos corresponden a los pines del puerto que definen el *registro de control*. Debido a que ciertos bits de éste registro son complementarios, por ejemplo D3, D1 y D0, (véase *tabla 1.9*), al medir con un voltímetro las salidas, se verá que los 4 bits menos significativos tienen de la siguiente forma:

D3	D2	D1	D0
0	1	0	0

Entonces, resta por programar el registro de control usando el valor decimal de 4, para poner externamente todos los pines del puerto en 1L. Para esto se utiliza la última línea del programa anterior. Finalmente, el puerto de control estará en capacidad de recibir datos desde el exterior y los bits de salida quedarán situados de la siguiente forma:

D3	D2	D1	D0
1	1	1	1

**Análisis de una configuración en colector abierto.-** Los diagramas de la *figura 1.3* indican dos estados típicos de una configuración en colector abierto, esto es, el colector del transistor se conecta en paralelo con una resistencia *pull-up*. El valor de esta resistencia se calcula en función de la corriente que circulará por el colector del transistor. En los primeros diseños realizados, las salidas de los puertos de control eran inversores de colector abierto 7405, con resistencias pull-up de 4.7 KΩ.

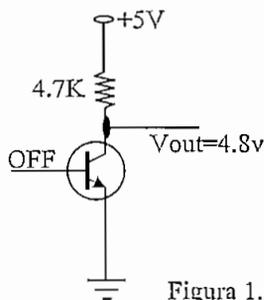


Figura 1.3A

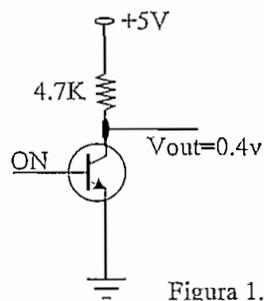


Figura 1.3B

Figura 1.3.- Salidas en colector abierto.

La *figura 1.3A* muestra un transistor que recibe un nivel bajo en la base. Una configuración como ésta, en la que el transistor opera en corte y saturación, predice que al recibir un nivel bajo en la base, el transistor se comportará como un interruptor abierto (*región de corte*), esto es, se tendrá que  $V_{ce} \approx V_{cc}$  e  $I_c \approx 0$ . Por esta razón, el colector del transistor estará en un nivel alto:  $V_{out} = +4.8V$ . Entonces, una vez conectada la resistencia *pull-up*, se van a tener dos voltajes de referencia, uno de +5V (fuente) y el otro de +4.8V (por caída de voltaje en la resistencia), en donde se va a deducir una corriente según la ley de Ohm de:

$$(5-4.8)V / 4.7K\Omega = 42.55 \mu A$$

Como puede verse la corriente es muy baja. Esto significa que si la salida del transistor (+4.8V) se cortocircuita a tierra, la corriente en la resistencia sería de:

$$(5-0)V / 4.7K\Omega = 1.06 \text{ mA}$$

Nótese que al cortocircuitar una salida de colector abierto, que inicialmente estuvo con un nivel alto de 4.8 V, el puerto no corre riesgo de daño electrónico en sus componentes internos, en otras palabras, se puede usar éste transistor para fines de entrada de datos, previa programación del mismo con un nivel alto (1L) a la salida.

En la *figura 1.3B* se tiene en cambio, que el transistor recibe en la base un nivel alto. Este nivel alto será suficiente para que el transistor se sature, esto

significa que  $V_{ce} \approx 0.25 \text{ V}$  y la  $I_c = I_{max}$ , permitiendo que el voltaje de salida en el colector sea:  $V_{out} \approx 0.25\text{V}$ . La resistencia *pull-up* se ubica en paralelo al colector del transistor y se puede calcular la corriente que circula por la misma, de la siguiente forma:

$$(5-0.25)\text{V} / 4.7\text{K}\Omega = 1.011 \text{ mA}$$

Análogamente al caso anterior, existe la posibilidad de conectar la salida **Vout** a un nivel alto, sin embargo, este procedimiento no se realiza comunmente.

Entonces, de acuerdo a éste análisis, lo conveniente es programar las salidas de colector abierto del puerto, con niveles altos, para que puedan funcionar como entradas.

#### 1.4 Manejo de los registros del puerto paralelo mediante el uso de programación computacional (Software).

En éste proyecto se utilizará un programa diseñado en **VISUAL BASIC**, para trabajar con los equipos físicos, "péndulo simple, plano inclinado y caída libre", sin embargo, los comandos a utilizarse para el manejo del puerto paralelo no pueden ser ejecutados por este paquete computacional, debido a que no se dispone de instrucciones de programación directas que trabajen sobre el puerto paralelo del computador. Es por eso, que se crean generalmente rutinas en lenguaje C o ensamblador como apoyo en el desarrollo del programa principal. La desventaja de este procedimiento es que, al hacerse llamadas externas a rutinas

en otro lenguaje, el control del programa inicial se pasa momentáneamente al C, y existe la posibilidad de perderse la información obtenida. Actualmente hay un sitio en el Internet, en donde se puede bajar información referente a un archivo **IOPORT.OCX**, (véase *ANEXOS literal F*) éste archivo sirve de apoyo directo al VISUAL BASIC, y crea un control en el programa principal, el cual, al ser llamado, activa el puerto paralelo y analiza los pines destinados para entrada. Otra opción es la creación de un archivo tipo (.DLL), que lo utilizaría WINDOWS al momento de necesitarse el chequeo del puerto paralelo.

NOTA: El programa **IOPORT.OCX**, es un archivo que puede obtenerse gratuitamente, con una duración de máximo 30 días de uso. La licencia tiene un valor de aproximadamente USD \$15.00 para un número de 1 a 9 usuarios por cada uno.

Las antiguas versiones de Basic, es decir, todas aquellas que antecedieron a los lenguajes gráficos actuales, como por ejemplo **BASIC**, **Microsoft GW-BASIC**, **Microsoft QBASIC**, permitían el manejo de los registros del puerto paralelo directamente con comandos simplificados. En **QBASIC**, por ejemplo, una línea de programa puede seleccionar una dirección para entrada y otra para salida. La forma en que se prepara un puerto para salida es la siguiente:

#### **OUT** *dirección, dato*

Se entiende que el comando **OUT** pone al puerto en condición de salida, y adicionalmente deberemos indicar la dirección a la cuál se necesita enviar datos.

Debe recordarse que *dirección* puede ser (*BASE h*) ó (*BASE+2 h*), y corresponden a las direcciones posibles para salida de datos en el puerto paralelo. La dirección (*BASE+1 h*) sólo se usa para entrada de datos y se la describirá más adelante. El valor hexadecimal o decimal que tenga la dirección utilizada dependerá del puerto habilitado, o que sea de uso del computador: **LPT1**, **LPT2** o **LPT3** (*Ver sección 1.2.2 y tabla 1.5 de la sección 1.3.1*).

El *dato* será un valor en base hexadecimal, binaria, octal, o decimal, que corresponderá a un registro de 8 bits direccionados hacia el puerto, por ejemplo, si se intenta activar los pines: 9, 8, 7, 6, 5, 4, 3, 2, que corresponden a la dirección (*378 h*), en un puerto **LPT1**, se tendrá que enviar un registro completo de ocho 1's a la salida, esto es poner **D7**, **D6**, **D5**, **D4**, **D3**, **D2**, **D1**, **D0** respectivamente en un nivel alto, así:

```
OUT &h378, hFF
```

Nótese que el *dato* puesto, es el valor hexadecimal **FF** que equivale en binario a (**11111111**), y será igual a 255 en decimal. Este dato entonces, podrá ser escrito en cualquier base numérica, siempre y cuando se escriba antes del valor, la letra que identifica a dicha base. Por ejemplo, en la línea anterior, se tiene que a un lado de **FF** hay una letra **h**, ésta indica que el valor que se va a escribir es hexadecimal.

Entonces, en binario quedará:

```
OUT &h378, b11111111
```

o en decimal:

## OUT &h378, d255

En esta última línea, no es necesario escribir la letra d, pues se puede igualmente interpretar dicho valor como decimal. Igual tratamiento puede recibir el número usado para definir la dirección o etiqueta a donde se destinarán los datos de salida.

Para adquirir información del puerto externo se deberá usar otro comando, el mismo que tiene como siglas: INP (*In port*). Como se trata de un dato externo, el computador necesitará almacenarlo en la memoria para procesarlo adecuadamente. Este valor introducido se guarda en una variable cualquiera, y la línea de programa a utilizarse, será la siguiente:

```
variable = INP(dirección)
```

Donde *dirección* será una dirección asignada para usarla como entrada, por ejemplo *BASE+1 h*. Si se sabe que se está usando el puerto LPT1 en el cual, *BASE+1 h* corresponde a (379h), la línea de programa se escribirá así:

```
variable = INP(&h379)
```

Esta dirección también se puede escribir en base decimal (*Véase la tabla 1.5*). En la variable se almacenará el byte obtenido a través del puerto. Este byte estará conformado por los cinco bits del registro de estado, que son bits de

entrada, es decir se tendrá almacenado en una localidad de memoria un valor binario conformado por 5 bits. Cada uno de estos bits proporcionará información del exterior en forma independiente. Aunque el registro es de 8 bits, los 5 bits más significativos están disponibles al usuario (*véase la tabla 1.8*).

Para analizar la información que lleva cada uno de los 5 bits de entrada del registro de estado, se tendrá que usar una máscara binaria. Esto permitirá el análisis individual de cada uno de los bits que conforman el byte. Por ejemplo, para saber el estado lógico del pin **BUSY** (#11) del conector DB-25H, se deberá ejecutar la siguiente línea de programa:

```
variable = INP(&h379) AND 128
```

El valor decimal de 128 corresponde en binario a (10000000), entonces al realizar la operación lógica, el resultado contendrá solamente al bit más significativo, en este caso el bit D7, que corresponde al pin **BUSY**, según la *tabla 1.8*:

D7	D6	D5	D4	D3	D2	D1	D0
AND							
1	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0

El resultado decimal, leído por el computador, será un valor de 128 o 0, dependiendo si el valor ingresado fue un nivel alto o un nivel bajo, respectivamente.

### **1.5 Ventajas y desventajas de la comunicación paralela respecto a la comunicación serial.**

En la comunicación paralela se pueden llegar a tener más desventajas frente a las ventajas que tiene la comunicación serial. Sin embargo, es necesario nombrar las siguientes como las más importantes:

#### **Ventajas:**

- La comunicación paralela es más efectiva, por la capacidad de disponer más canales de acceso (hasta 8 simultáneamente).
- Es más rápida que la comunicación serial, pues puede alcanzar velocidades de transmisión de hasta 1 Mbyte/sec.
- No utiliza sincronismo.
- Exige comandos simples del BASIC para su manipulación. En la comunicación serial, la programación tiene otro tipo de comandos especiales.

#### **Desventajas:**

- No sirve para comunicaciones a larga distancia, en ese caso lo óptimo es la comunicación serial.
- Es muy costosa respecto a la comunicación serial.
- Es más complicada de realizar.

- Si se usa cable plano de varias vías para la comunicación, puede existir acoplamientos capacitivos entre conductores y producirse errores a nivel de datos.

## CAPITULO II

### TEORIA Y DISEÑO DEL INTERFAZ GENERAL DEL PROYECTO.

#### 2.1 ANTECEDENTES GENERALES.

El interfaz que habrá de construirse, deberá traducir las señales externas a una lógica entendible por el computador, para su análisis y aplicación en el fenómeno físico que se experimente.

Se ha pensado en construir tres equipos para simular el efecto físico producido por un “Péndulo”, un “Plano Inclinado” y un sistema de “Caída Libre”. Cada uno de éstos equipos tiene un sistema de estudio diferente; sin embargo, la manera de como se va a sensar los fenómenos en particular es la misma. Se utiliza básicamente un sistema transductor de luz para poder interpretar el fenómeno en estudio. Además, por motivo de costos, contaremos con elementos que produzcan y reciban luz visible que es adicionalmente fácil de encontrarlos en el mercado convencional.

Se ha pensado la forma de relacionar el fenómeno físico en proceso con el computador, esto es, provocando un haz luminoso e interrumpiéndolo con algún evento en particular, como puede ser un cuerpo físico en pleno movimiento. La computadora, que será el equipo que realice el estudio de lo ocurrido en los prototipos, traducirá los datos recibidos a través del puerto paralelo utilizando un paquete computacional, para el adecuado manejo e interpretación de los datos externos. En definitiva, se hará que tanto los prototipos como el programa diseñados, se interrelacionen para cumplir con el objetivo en estudio.

Si se analizan las circunstancias en las que se va a desenvolver el fenómeno, se podría decir que no hay nada de diferente en su realización física. La forma en la que se ejecute será la tradicional y su estudio matemático igual. Dentro del prototipo, se ubicarían los testigos electrónicos que estudiarán el fenómeno en particular y, en base a éstos, se podrá finalmente analizar en detalle de lo que esté sucediendo.

Debe indicarse que éste trabajo no tiene como objetivo hacer las cosas fáciles al estudiante, sino más bien prestar una ayuda práctica para que el estudio y análisis del fenómeno físico sean mucho más asimilables por la persona inmersa en este campo. Entonces es importante que, paralelamente a la práctica, exista otra persona que realice los mismos cálculos en la forma tradicional, utilizando las ecuaciones del movimiento en estudio. Solo así se estará seguro de que el aprendizaje sea mucho más eficiente, debido a que se generarían errores que motiven la investigación de su procedencia.

El computador será una herramienta que, dentro de la interrelación hombre-máquina, proveerá de un ambiente amigable al usuario y permitirá su uso como un verdadero cronómetro, que se usará en la medición del tiempo en que se realiza una actividad en especial. Con esto se indica prácticamente que, lo que se medirá en todo momento, es el tiempo en que el móvil analizado realiza una actividad específica, según el tema que se esté estudiando "*Péndulo, Plano Inclinado y Caída Libre*". Con dicha variable de tiempo, determinada por el computador durante la práctica, se calcularán las variables del movimiento que son de interés dentro del estudio físico particular elegido.

Para poder estar más cerca de la realidad de lo que es la física, se puede plantear algunos puntos de observación para hacer un seguimiento preciso del movimiento del cuerpo en estudio, esto es, volver al equipo un sistema mucho más inteligente para el análisis, sin embargo el objetivo de este proyecto será de plantear el uso de un número básico de dos sensores por módulo, con el fin de tener una apreciación de ciertas partes importantes en el análisis del movimiento y que son suficientes para generar una respuesta a los requerimientos planteados, por ejemplo: En el prototipo “Caída Libre”, dos sensores situados en un panel vertical darían información de un cuerpo desplazándose hacia abajo; en donde uno de ellos situado como referencia superior, determinará una cierta velocidad inicial  $V_0$  generada por el móvil al paso por ese punto; y el otro situado en la parte inferior, indicará lo referente a la velocidad final  $V_f$  en este último punto. Con toda esta información y las ecuaciones del movimiento, no se requiere más que manipular las variables físicas y obtener los resultados.

#### **2.1.1 DIAGRAMA DE BLOQUES DEL CIRCUITO INTERFAZ.**

En esta sección se verá la forma de representar la idea electrónica que se construirá mediante bloques generales, en donde cada uno de estos tendrá su importancia en la consecución de los objetivos para el funcionamiento del interfaz deseado.

Es de suma importancia que se detalle la operación de cada uno de estos bloques y la influencia que presentan en el circuito total; sin embargo, se prefiere

hacerlo en la siguiente sección y por ahora dar prioridad a ligeros comentarios de lo que se hará en cada parte del diagrama.

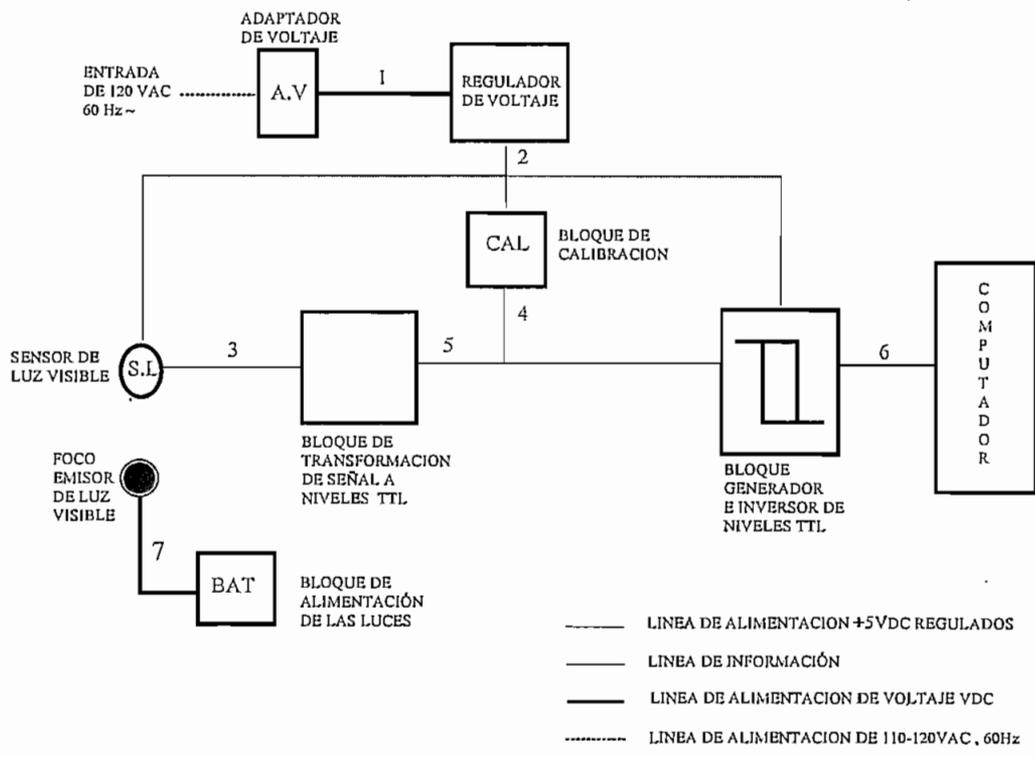


Figura 2.1.- Diagrama de bloques del interfaz electrónico.

**DETALLES :**

1. Línea proveniente del adaptador o fuente de alimentación principal. Sale del adaptador de voltaje. Este adaptador rectifica el voltaje alterno entrante (110-120 Vac, 60 Hz) de la red y lo transforma en un voltaje continuo de salida en rangos de 3, 4.5, 6, 7.5, 9, y 12 Vdc, que pueden ser elegidos por el usuario, con una carga máxima de hasta 500mA. La línea 1 se direcciona hacia la entrada del bloque regulador de voltaje.

2. *Línea de alimentación proveniente del bloque regulador de voltaje.* En esencia, esta línea es aquella que provee de alimentación regulada hacia el resto de dispositivos del circuito interfaz. El bloque de donde ésta proviene es el “*Bloque Regulador de Voltaje*”. Este bloque contiene principalmente un dispositivo regulador de voltaje que permite en la entrada voltajes comprendidos entre 7-30 Vdc, y que son tomados de la línea 1, que corresponde a la salida rectificadora del adaptador y cuyas características son del tipo universal. A la salida del dispositivo regulador se tiene un voltaje de alimentación estable en +5.02 voltios (con capacidad de corriente a plena carga de hasta 1 A), que se destina a la polarización de los dispositivos TTL. Debido a que el adaptador universal tiene un voltaje de salida muy inestable, ya sea en cualquier posición seleccionada, la adición del bloque regulador de voltaje es muy importante en el circuito y evitará la degradación o destrucción de los elementos electrónicos del interfaz. En realidad, la capacidad de carga que soporta la línea de salida del bloque regulador, estará limitada por la capacidad de carga de la fuente de alimentación que la precede. Los puntos de alimentación regulados comprenden: El sensor, el bloque de calibración, el bloque generador e inversor de niveles TTL.

3. *Línea de información generada por el sensor.* Conecta el dispositivo sensor con el bloque de transformación de señal. Lleva la corriente entregada por el sensor que responde al grado de incidencia luminosa que éste reciba en la juntura base-colector. Esta corriente deberá ingresar al bloque de

transformación en donde se dispone de un dispositivo BJT trabajando en corte y saturación, para producir niveles TTL de salida.

4. *Línea proveniente del bloque de calibración.* Controla la sensibilidad del sensor de luz y manipula el nivel de voltaje de entrada al bloque generador-inversor de niveles TTL.
5. *Línea de salida del bloque de transformación.* Toma los niveles lógicos obtenidos por un dispositivo BJT que opera en corte y saturación, de acuerdo a la actividad del sensor de luz y los lleva al bloque generador e inversor de niveles TTL para ser corregidos.
6. *Línea generadora de los niveles lógicos resultantes.* Los niveles de salida son enviados al computador para su tratamiento bajo un programa específicamente creado para el efecto. Estos datos de salida son el resultado de la operación de un dispositivo “Schmitt trigger” ubicado dentro del bloque generador e inversor de niveles TTL, que toma la forma de onda de salida del BJT del bloque de transformación de señal, y la mejora produciendo una forma de onda más cuadrada, apta para ser introducida al computador. Debido a que el lazo de histéresis del dispositivo “Schmitt trigger” es inverso, entonces los datos de salida hacia el computador también sufrirán el mismo efecto.
7. *Línea de energización de las luces.* A través de ésta línea se comandará la energización de una lámpara por medio del bloque de alimentación de luces.

## 2.2 TEORIA DEL FUNCIONAMIENTO DEL INTERFAZ BASICO.

El interfaz que se diseñará consiste en un sistema que permite que un sensor de luz (Fototransistor tipo NPN), ubicado externamente, sea el dispositivo que produzca una corriente tal que su magnitud dependerá del grado de luz que reciba éste del exterior. La luz recibida es canalizada, a través de una lente que permite la concentración y la convergencia de ésta, hacia la juntura fotosensible Base-Colector del transistor, que deberá estar inversamente polarizada para el correcto funcionamiento del dispositivo. Según las especificaciones técnicas de este tipo de elementos electrónicos, se deben considerar dos tipos de corrientes que se clasifican en una corriente en obscuridad ( $I_D$ ) y otra a plena luz ( $I_L$ ). La primera por lo general esta determinada en valores de decenas de  $nA$  y será generada en momentos en que una luz tenue o baja, escasamente incide sobre el fototransistor. La segunda se encuentra por lo general en el orden de las centenas de  $\mu A$ , como valores típicos, e indica la magnitud de corriente generada en circunstancias en las que el fototransistor se encuentra iluminado por un haz de luz incidente en la juntura fotosensible de éste.

Es momento de crear un primer bosquejo del circuito que se necesita para conseguir unificar el funcionamiento de un fototransistor, con un sistema electrónico capaz de interpretar la información proporcionada por éste dispositivo y adaptarla para que sea fácilmente utilizada por el computador para su procesamiento. Así entonces, el primer paso es plantear cómo se conectará el fototransistor y cómo se realizará la conexión a un transistor BJT tipo NPN, para polarizarlo adecuadamente y que pueda trabajar en la región de corte y saturación,

y así finalmente obtener niveles altos y bajos en el colector del BJT como salida.

La *figura 2.2* muestra la idea planteada.

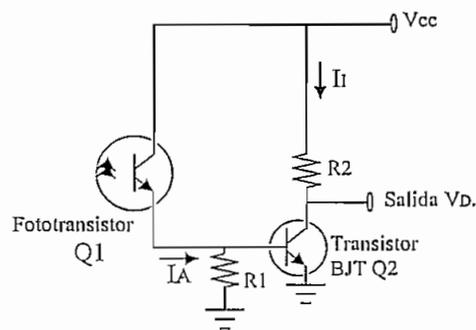


Figura 2.2.- Conexión del sensor a un BJT que trabaja en la región no lineal.

Como se ha visto, se hará incidir un haz luminoso sobre el dispositivo optoelectrónico, que una vez montado en el prototipo final, el móvil en estudio lo traspasará cortándolo por un cierto tiempo y produciendo un cambio en las condiciones de iluminación que el transistor detectará y responderá con valores de corriente diferentes. Por ejemplo, si el fototransistor no está iluminado, la corriente producida es muy baja, casi imperceptible, y no será capaz de activar la base del BJT de la *figura 2.2*. Pero si el fototransistor se encuentra iluminado, se tendrá un valor de corriente significativo en el circuito y que puede ser utilizado por la base del BJT para su funcionamiento.

Una vez que se ha definido los niveles de corriente del fototransistor en condiciones de luz u oscuridad, la cual tomará el nombre de ( $I_A$ ), ésta se conducirá a través de una conexión directa desde el emisor del fototransistor, a la base del BJT tipo NPN, que trabaja en corte y saturación, como se ve en la *figura 2.2*. Una resistencia situada entre la base del BJT ( $R1$ ) y tierra, de un valor

adecuado, permitirá conseguir un voltaje de polarización en la base del transistor de bajo valor. Analizando la operación del BJT, se deberá clasificarla en dos partes:

1. Al tener un valor de corriente bajo en el fototransistor ( $I_A$ ), en condiciones de oscuridad, corriente casi imperceptible que ingresa a la base del transistor BJT, la corriente de colector ( $I_I$ ) es también muy baja. Esta situación fuerza al transistor a entrar en la región de corte y por lo tanto su voltaje  $V_{ce}$  tenderá a ser alto y como el emisor está conectado a tierra, este voltaje  $V_{ce}$  deberá ser algo menor que  $V_{cc}$ . Nótese que en el colector se tendrá un voltaje de salida alto. Para mayor información, se puede observar los resultados experimentales obtenidos (*ver  $V_D$  en el capítulo IV, sección 4.2.1, tabla 4.1*).
2. Si ahora, el fototransistor está siendo iluminado, el valor de corriente ( $I_A$ ) que ingresa a la base del BJT es máximo, la corriente de colector ( $I_I$ ) también será alta y por lo tanto el transistor operará obligatoriamente en la región de saturación donde el  $V_{ce}$  tiende a ser cero. Esto explica que a la salida del transistor ( $V_D$ ), se tendrá un voltaje de aproximadamente 0 voltios, ya que el emisor del BJT está conectado directamente a tierra (*ver  $V_D$  en el capítulo IV, sección 4.2.1, tabla 4.1*).

Es importante notar que hasta ahora todo el proceso realizado muestra la apariencia del diseño de un convertidor análogo a digital (A/D), sin embargo, los

pocos elementos utilizados dan muestra de otra realidad, que permite obtener niveles TTL de un sistema analógico como es la luz a partir de electrónica diferente a la convencional.

Continuando con el bosquejo del circuito a implementarse, surge la necesidad de introducir una resistencia variable que permita al circuito buscar las condiciones óptimas de trabajo, y que además, el sensor pueda afinarse para un mejor desempeño dentro del circuito. Según la *figura 2.2*, la resistencia insertada  $R2$  sirve para ese efecto y probablemente sea preferible ubicar en esa posición un potenciómetro para los fines ya mencionados. En definitiva, se requiere que  $R2$  pase a ser un potenciómetro y que éste dispositivo permita la calibración del sistema electrónico general. Con el objeto de evitar que la fuente de voltaje  $V_{cc}$  entre en cortocircuito cuando el potenciómetro adquiriera una de las posiciones extremas, esto es  $0\ \text{ohmios}$ , se montará adicionalmente una resistencia  $R3$  que evitará que este problema suceda. El esquema de la *figura 2.3*, indica los nuevos cambios y adiciones.

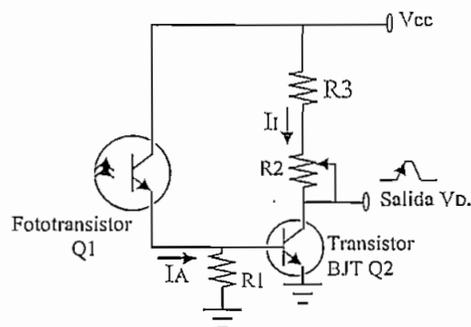
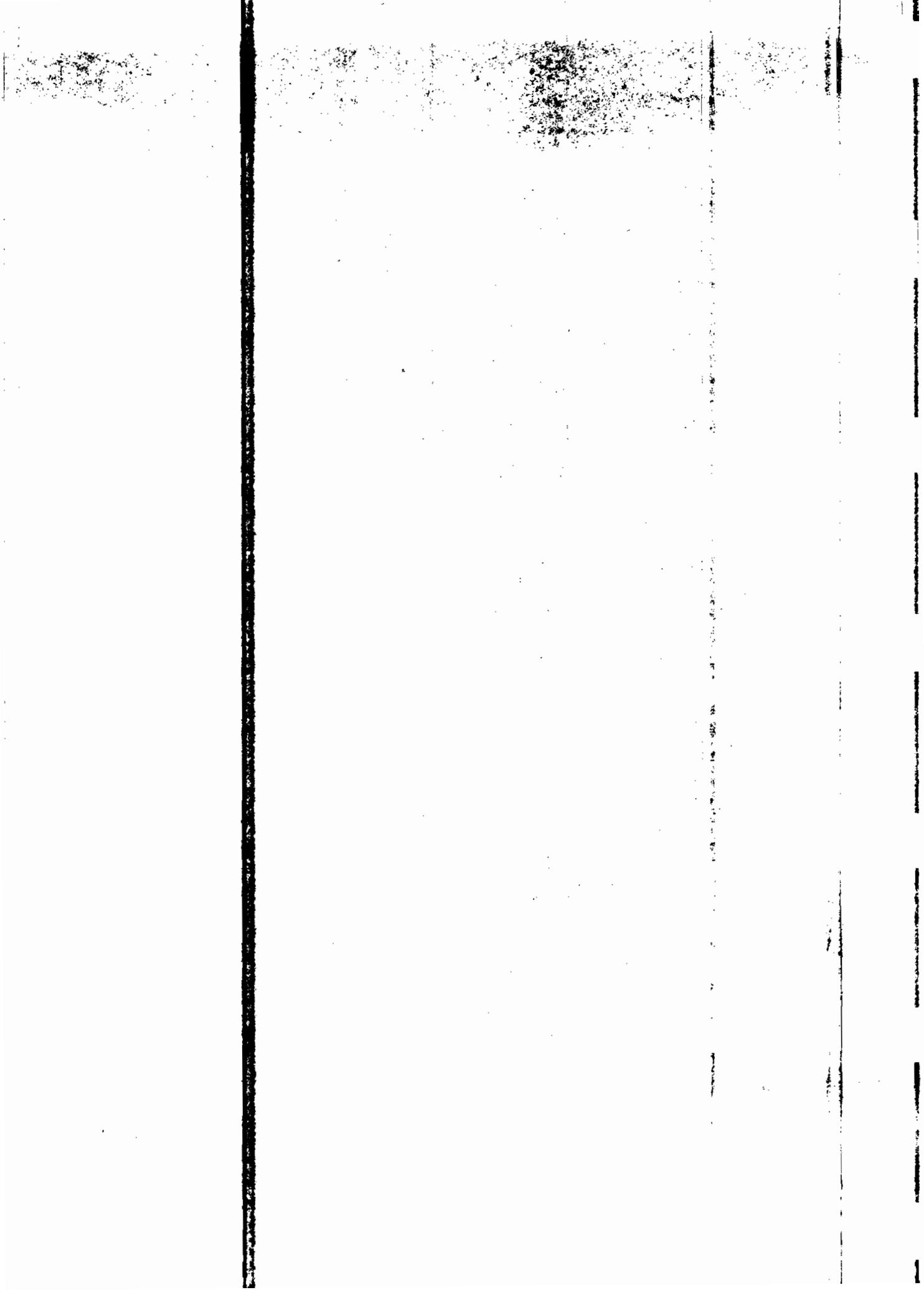


Figura 2.3.- Diagrama electrónico con control de sensibilidad y protección de la fuente.

De acuerdo a la *figura 2.3*, se ha insertado un potenciómetro ( $R2$ ) entre  $V_{cc}$  y el colector del transistor BJT ( $Q2$ ), y que cumplirá con dos funciones: Primero, la de permitir mayor circulación de corriente al fototransistor ( $Q1$ ) a través del colector, esto le dará la característica de regulador de sensibilidad del dispositivo optoelectrónico; y, segundo, para evitar que  $V_{cc}$  entre en cortocircuito cuando el transistor BJT ( $Q2$ ) opere en la región de saturación; sin embargo, el voltaje de colector del BJT ( $Q2$ ), se verá disminuido por la presencia de este elemento resistivo en el momento en que el transistor opere en la región de corte. De hecho, será algo menor que el voltaje de alimentación  $V_{cc}$ , porque en el arreglo en serie de las resistencias ( $R2+R3$ ) del colector del BJT ( $Q2$ ), se tiene una cierta caída de potencial.

Debido a esto, se propone la utilización de un dispositivo “*Schmitt trigger*” que tome las señales de salida del colector del transistor ( $Q2$ ) y las mejore para ser utilizadas por el puerto paralelo del computador.

**NOTA:** Cabe mencionar que la salida digital en el colector del BJT ( $Q2$ ) es adecuada y, además, comparable con las características de la lógica TTL para su tratamiento mediante el uso de un computador. Cuando ( $Q2$ ) se encuentra en corte, el voltaje de salida en el colector debería ser aproximadamente  $V_{cc}$ . En realidad resulta ser más bajo que  $V_{cc}$ , y mejoraría si, al bloquear la luz incidente sobre el sensor, éste es ubicado en un lugar oscuro, lejos de la incidencia de cualquier iluminación externa que no sea la misma fuente de luz propia del sensor; por ejemplo, podría ser una cavidad oscura diseñada específicamente para



ser capaz de mantener al sensor aislado de la luz exterior. Sin embargo, en este proyecto es preferible asegurar que el equipo pueda operar en cualquier medio, sea éste claro u oscuro. Para esto será necesario verificar que el nivel alto de salida en el colector del BJT ( $Q_2$ ), es adecuado y que estando el sensor en condiciones normales, sean éstas en luz u oscuridad, los niveles altos y bajos deberán ser dirigidos hacia un dispositivo “*Schmitt trigger*” para mediante éste, conseguir el mejoramiento en la señal de salida que se destinará al computador. La *figura 2.4*, muestra la conexión del colector del transistor ( $Q_2$ ) hacia un dispositivo “*Schmitt trigger*”, el cual trabaja generalmente como un dispositivo “*conformador de onda cuadrada*” y que dentro de este proyecto será de suma importancia en la comunicación hacia el computador.

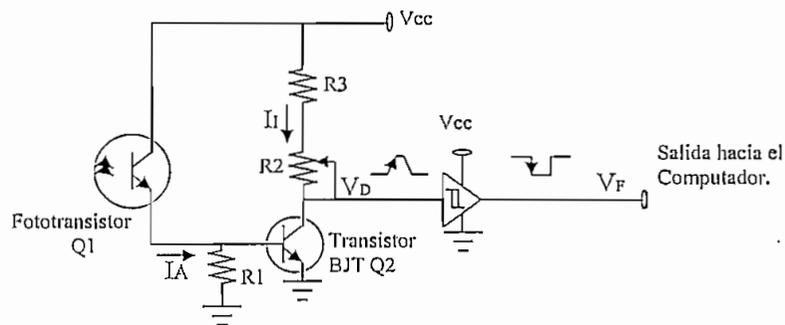
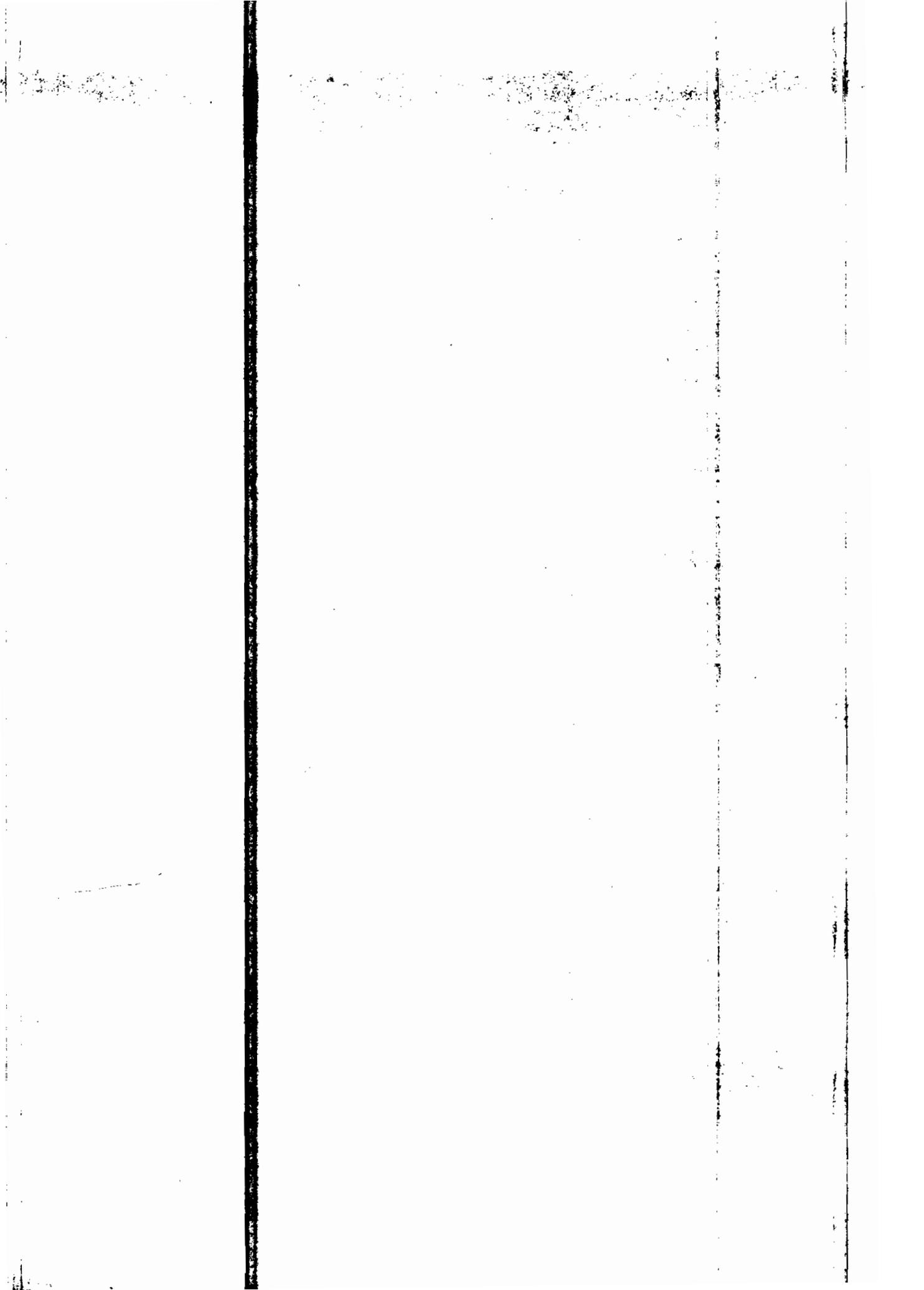


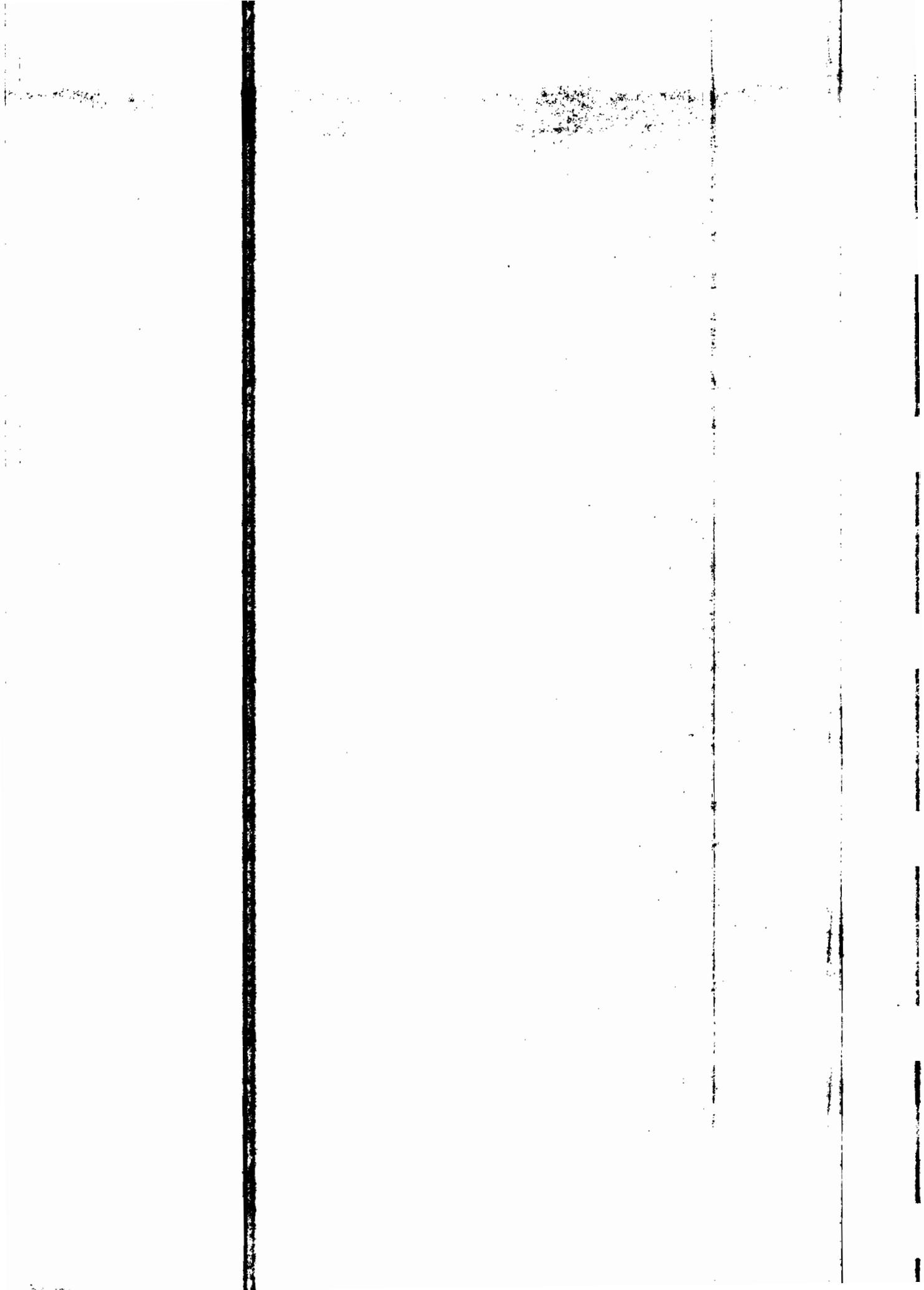
Figura 2.4.-Diagrama electrónico del circuito básico de interfaz hacia el computador.

La *figura 2.4* muestra cómo se puede conectar un dispositivo “*Schmitt trigger*” al circuito electrónico presentado en la *figura 2.3*, para tomar los niveles lógicos otorgados por el colector del BJT ( $Q_2$ ) y poderlos transformar en niveles aceptables, capaces de ser manipulados por un sistema externo (*computador*).



Este dispositivo, que será detallado ampliamente en la sección 2.2.1, toma la salida digital del colector del BJT y lo interpreta así:

1. Si se tiene un nivel bajo en el colector del transistor ( $Q2$ ), con aproximadamente 0.2 voltios, es decir cuando el fototransistor ha sido previamente iluminado, entonces se deduce que el nivel de voltaje de entrada al *Schmitt trigger* ( $V_D$ ) es aún menor que el voltaje umbral  $V_{T-}$  característico del lazo de histéresis del disparador, y que típicamente está en 0.9 voltios (véase lazo de histéresis y especificaciones técnicas del 74LS14 en ANEXOS). Entonces, el voltaje de salida ( $V_F$ ), del dispositivo disparador, se mantendrá en un nivel alto de aproximadamente 4.46 voltios, según los datos proporcionados por el fabricante. Conforme el cuerpo en estudio cruce el haz luminoso y evite que el sensor siga siendo incidido por éste, se producirá que la corriente entregada por el fototransistor decrecerá en magnitud, por lo tanto el BJT se abrirá (entrará en corte), pasando el colector inicialmente desde un nivel bajo a un nivel alto. Esta transición positiva, finalmente disparará al *Schmitt trigger* por su flanco negativo luego de haberse superado el voltaje umbral  $V_{T+}$  típicamente en 1.7 voltios, para tener una salida ( $V_F$ ) ahora en 0 voltios.
2. Si ahora se tiene un nivel alto en el colector del transistor ( $Q2$ ), con la premisa de que el fototransistor dejó de estar iluminado, entonces el voltaje ( $V_D$ ) tomado por la entrada del *Schmitt trigger* será superior al voltaje de umbral  $V_{T+}$  establecido en el lazo de histéresis del dispositivo, esto hace que el

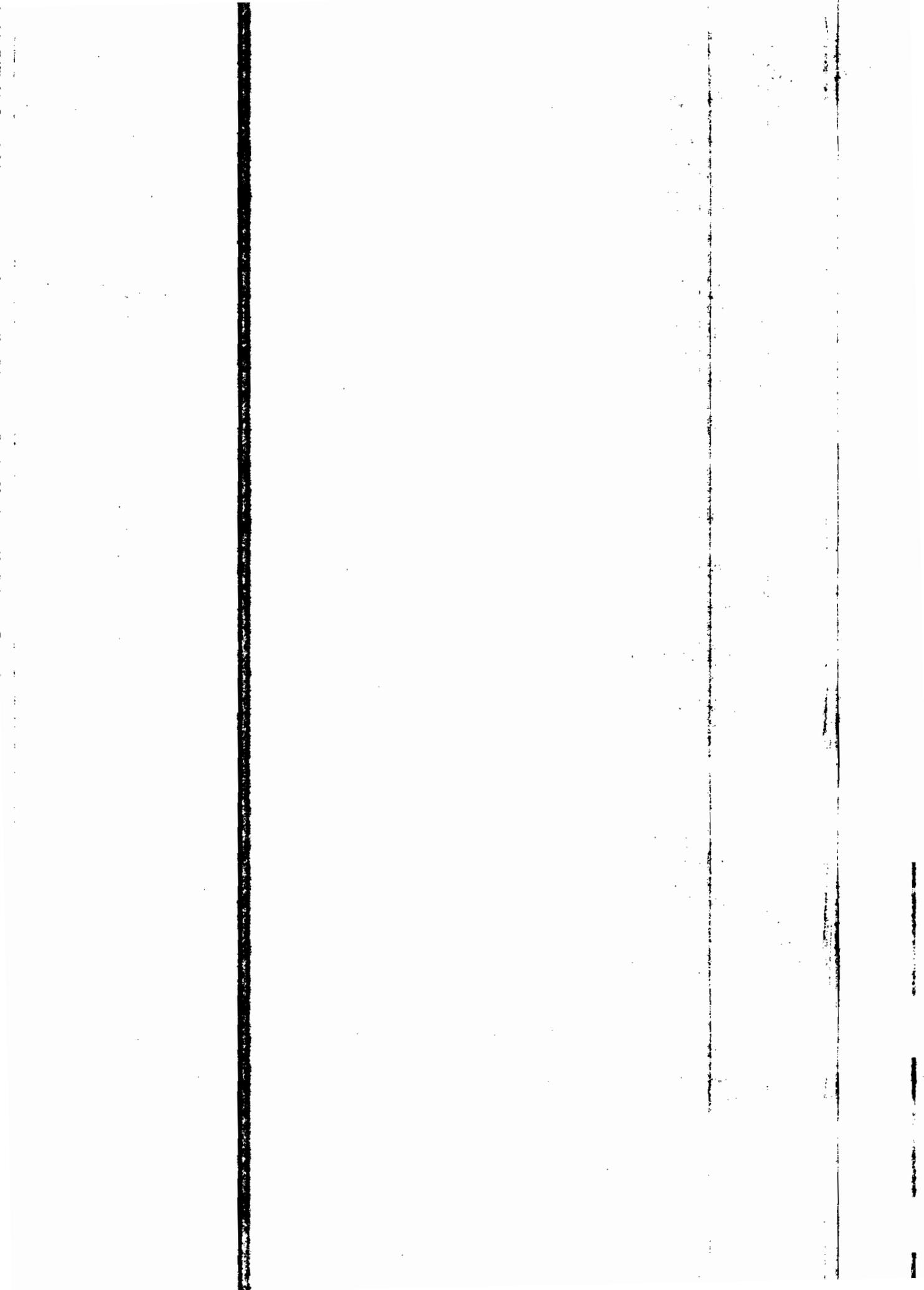


resultado ( $V_F$ ) sea un nivel bajo de aproximadamente 0.2 voltios típicamente, como valor de salida hacia el computador, (*ver lazo de histéresis y especificaciones técnicas del 74LS14 en ANEXOS*). Pero en el momento en que el sensor vuelve a estar iluminado, es decir que el móvil traspasó el haz de luz incidente, se produce un cambio en la magnitud de la corriente entregada por el fototransistor, es decir, la corriente ( $I_A$ ) se incrementa. Este cambio obliga al colector del BJT a pasar de un nivel alto a un nivel bajo en ( $V_D$ ). La entrada del *Schmitt trigger* permanecerá a la espera de que el voltaje ( $V_D$ ) disminuya, sobrepasando el voltaje umbral  $V_T$  establecido por el lazo de histéresis y se disparará por flanco positivo, para finalmente tener en la salida ( $V_F$ ) un nivel alto que será tomado por el computador.

**NOTA:** Para que se ejecute todo este proceso, el interfaz deberá ser calibrado correctamente (*véase Calibración electrónica del Interfaz, capítulo V, sección 5.2*).

### 2.2.1 TEORIA DEL FUNCIONAMIENTO DEL “SCHMITT TRIGGER”.

El “*Schmitt trigger*” es un dispositivo muy utilizado hoy en día por sus aplicaciones diversas en el campo de la electrónica digital. Por ejemplo, si una señal de entrada ruidosa produce cambios de estado no deseados, el *Schmitt trigger* reduce este problema mediante el uso de dos voltajes de referencia, conocidos como voltajes umbrales. Mientras el valor pico a pico del ruido no

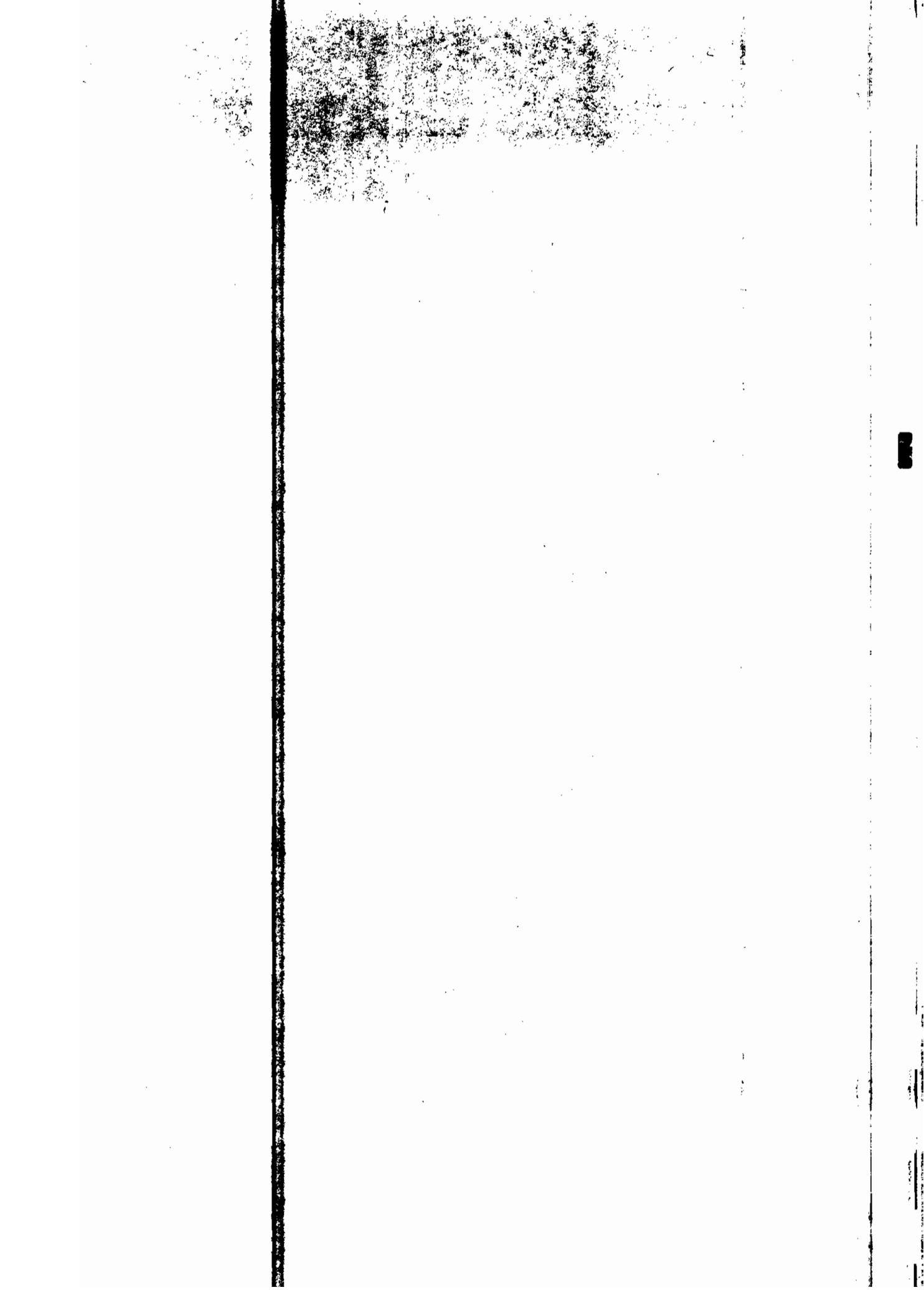


supere al voltaje de umbral positivo del disparador, no se registrará un cambio de estado en el mismo.

Otra aplicación de este tipo de dispositivos es aquella en donde se requiere de transiciones rápidas de nivel, pues algunos dispositivos TTL convencionales producen en sus salidas respuestas lentas, incluso otros las deforman. Al usar un Schmitt trigger en reemplazo del inversor, este tipo de inconvenientes es superado y la transición obtenida es segura y rápida (por esta razón se llama a estos dispositivos de disparo “Circuitos conformadores de onda cuadrada”).

Dos tipos de funcionamiento le caracterizan a este dispositivo: Uno operando en modo *directo* y otro en modo de *inversor*, este último es característico dentro de los elementos electrónicos digitales comerciales. Cada modo de operación se basa en un diagrama de estado que se conoce como “*Lazo de Histéresis*”, en el cual se determinan dos puntos críticos considerados como “*puntos de disparo o flancos*” y que están ubicados en los extremos del lazo.

Si se considera el modo de operación *directo*, se obtendrá un disparo por flanco positivo cuando el voltaje de entrada ( $V_i$ ) aumente hasta llegar al máximo establecido por el umbral ( $V_{H+}$ ), caso contrario la salida ( $V_o$ ) se mantendrá en un nivel bajo. Si en cambio, la salida ( $V_o$ ) se encuentra en un nivel alto y el voltaje de entrada ( $V_i$ ) por algún motivo decrece, el disparo por flanco negativo sucederá cuando se exceda el umbral ( $V_{H-}$ ) para finalmente tener una salida ( $V_o$ ) en nivel bajo. Cabe indicar, que mientras no se supere el valor umbral negativo, la salida ( $V_o$ ) seguirá situada en un nivel alto. La *figura 2.5* muestra una típica curva de histéresis con operación en modo *directo*.



Considerando ahora el modo de operación *inverso*, es importante mencionar que el disparo será por flanco negativo a un nivel bajo en la salida ( $V_o$ ), cuando el voltaje de entrada ( $V_i$ ) al disparador aumente hasta llegar a sobrepasar el voltaje umbral positivo ( $V_{H+}$ ). El disparo será por flanco positivo a un nivel alto en la salida ( $V_o$ ), cuando el voltaje de entrada al dispositivo disminuya hasta el punto de ser igual al voltaje umbral negativo ( $V_{H-}$ ).

La *figura 2.6* muestra la curva característica de este modo de operación. En ella se puede apreciar los flancos positivo y negativo, por los cuales el disparador cambia de un nivel bajo a alto y de un nivel alto a bajo, respectivamente en su salida.

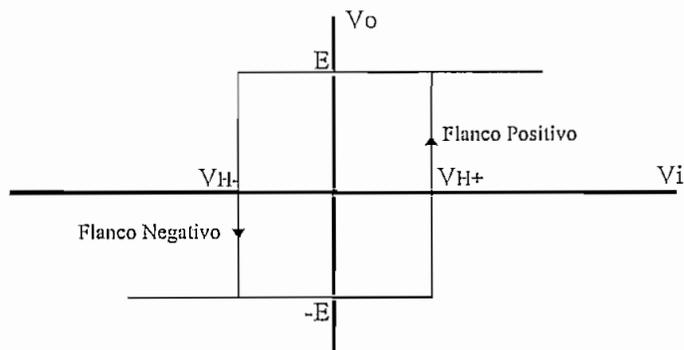


Figura 2.5.- Lazo de histéresis de un Schmitt trigger en modo directo.

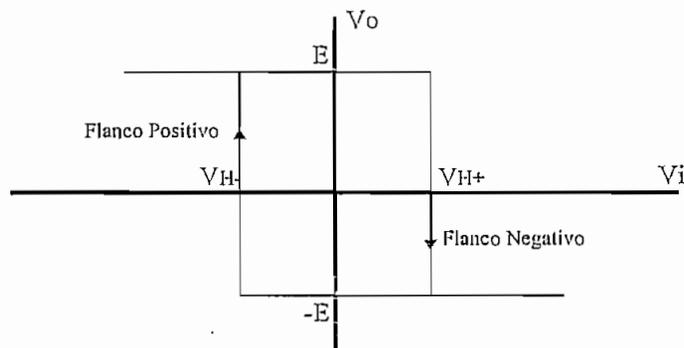
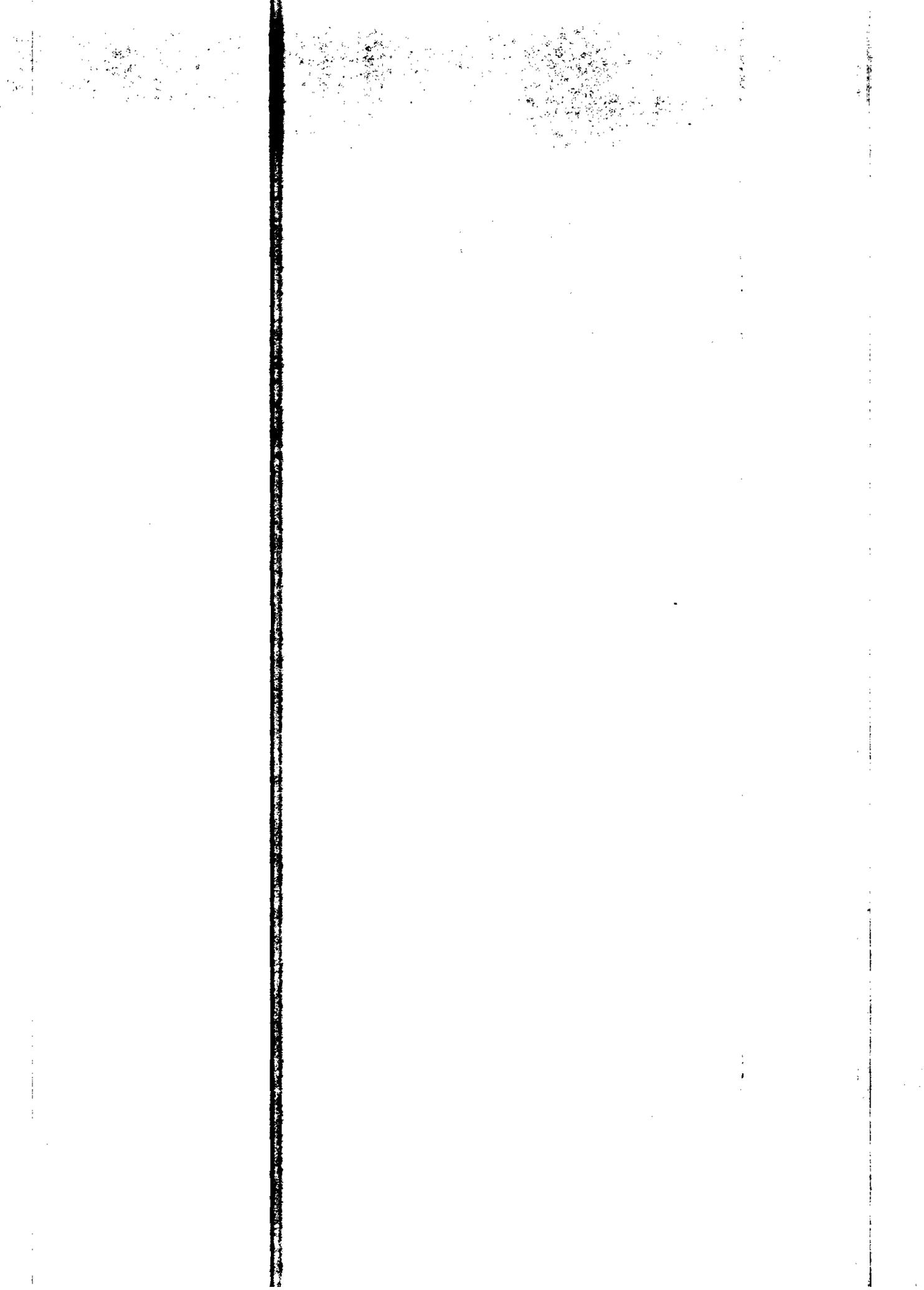


Figura 2.6.- Lazo de histéresis de un Schmitt trigger en modo inverso.



Un ejemplo gráfico en donde se puede describir una de las aplicaciones del “*Schmitt trigger*” como conformador de ondas cuadradas es aquel de la *figura 2.7*, en donde se tiene una señal variable como entrada y de frecuencia conocida. A ésta señal le corresponde la función del disparador en modo directo. Nótese que mientras crece el voltaje de la entrada ( $V_{in}$ ) desde el punto 0 hasta ( $V_{H+}$ ) voltios, la salida ( $V_o$ ) se mantiene en un nivel bajo. Al sobrepasar el valor umbral ( $V_{H+}$ ), se ve el disparo por el flanco positivo a un nivel alto en la salida. La señal entrante llega a su pico máximo y comienza a decrecer cruzando por cero hasta llegar al punto de ( $V_{H-}$ ), donde aún se mantiene la salida del disparador en un nivel alto. Luego de cruzar el punto umbral negativo, el disparo ocurre por el flanco negativo para finalmente tener la salida ( $V_o$ ) en un nivel bajo.

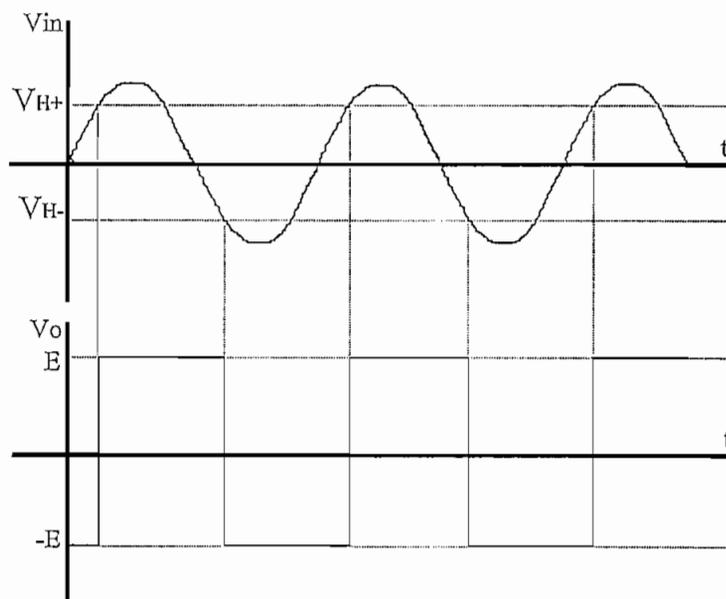


Figura 2.7.- Aplicación del Schmitt trigger como formador de onda cuadrada.

Los disparadores o *Schmitt triggers* pueden ser representados dentro de la electrónica analógica por arreglos de elementos activos y pasivos. Entre los elementos activos pueden mencionarse el uso de amplificadores operacionales que permiten que la construcción de éstos sea simplificada. Si se suma a esto unas pocas resistencias como elementos pasivos, se puede conformar finalmente un *Schmitt trigger* que puede ser representado en el diagrama de la figura 2.8:

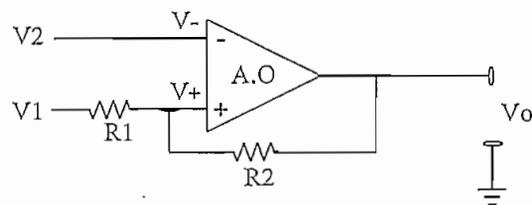


Figura 2.8. Diagrama electrónico básico del Schmitt trigger con A.O.

Como puede verse, el *Schmitt trigger* consiste en un amplificador operacional que se encuentra realimentado positivamente, operando así dentro de la región no lineal. Un amplificador operacional que trabaja en la región no lineal, es aquel en donde el voltaje de salida corresponde a los voltajes en saturación positiva o negativa.

Un amplificador operacional ideal tiene las siguientes características:

1. Ganancia en lazo abierto infinita  $G \rightarrow \infty$
2. Impedancia de entrada infinita  $Z_{in} \rightarrow \infty$
3. Impedancia de salida  $Z_{out} \rightarrow 0$
4. Ancho de Banda  $AB \rightarrow \infty$

Entonces si la ganancia en lazo abierto es infinita, y como la relación de ganancia en un amplificador operacional es de:

$$(V_+) - (V_-) = V_o / G$$

Se tiene que:

$$(V_+) - (V_-) = 0$$

Entonces para poder analizar el funcionamiento de un operacional se necesitará de las siguientes herramientas:

1. Conocer que  $(V_+) - (V_-) = 0$ , esto significa que  $(V_+) = (V_-)$
2. La corriente en  $V_+$  y en  $V_-$  es también de 0

Finalmente, estamos en condiciones de obtener las ecuaciones para el diseño de un *Schmitt trigger*. Según la *figura 2.8*, se puede verificar que existe una malla, en donde por el teorema de *Kirchoff* se tiene que:

$$(V_1 - V_+) / R_1 = (V_o - V_+) / R_2 \quad (1)$$

Desarrollando:

$$(V_1 - V_+) * R_2 = (V_o - V_+) * R_1$$

$$V_1 * R_2 - V_+ * R_2 = V_o * R_1 - V_+ * R_1$$

Pero se sabe que:  $(V_+ - V_-) = 0 \therefore V_+ = V_-$

y también que:  $V_2 = (V_-)$ , entonces reemplazando los términos:

$$V_1 * R_2 - V_2 * R_2 + V_2 * R_1 = V_o * R_1$$

$$V_1(R_1/R_2) + V_2(R_1 - R_2)/R_1 = V_o \quad (2)$$

Con la ecuación (2) se puede obtener el primer caso particular del *Schmitt trigger*, que es aquel que corresponde a la curva de histéresis de la *figura 2.5*. Para ello hay que definir que el voltaje de entrada  $V_2$  esté conectado a tierra, esto es  $V_2 = 0$  voltios, entonces se tendrá:

$V_1(R_2/R_1) = V_o$ , de donde:

$$V_1 = (R_1/R_2)V_o \quad (3)$$

La ecuación (3) corresponde a la ecuación del voltaje umbral positivo ( $V_{H+}$ ) del disparador y además indica el voltaje de entrada al dispositivo capaz de dispararlo por el flanco positivo. Con el mismo valor pero negativo, se tiene el voltaje umbral ( $V_{H-}$ ), y también definirá el punto donde se disparará el dispositivo por flanco negativo.

Sin embargo, se trata de diseñar un *Schmitt trigger inversor* que corresponda a la curva de histéresis de la *figura 2.6*, para ello se tendrá que partir definiendo que el voltaje ( $V_1$ ) de la *figura 2.8* está conectado a tierra, esto es  $V_1 = 0$ . Las ecuaciones de acuerdo al teorema de *kirchoff*, deberán redefinirse así:

$$(V_+ - V_1) / R_1 = (V_o - V_+) / R \quad (4)$$

Como se sabía que:  $(V_+ - V_-) = 0 \therefore V_+ = V_-$

y además:  $V_2 = V_-$ ,

Entonces reemplazando los términos:

$$V_2 * R_2 - V_1 * R_2 + V_2 * R_1 = V_o * R_1$$

Dividiendo ambos miembros para R1:

$$V_2(R_1 + R_2)/R_1 - V_1(R_2/R_1) = V_o$$

Finalmente se concluye que con  $V_1 = 0$ , la ecuación (4) tiene la siguiente forma:

$$V_2 = R_1 * V_o / (R_1 + R_2) \quad (5)$$

El voltaje  $V_2$  de la ecuación (5) indica el voltaje umbral ( $V_{H+}$ ) por el cual el dispositivo se disparará por el flanco negativo a un nivel bajo de salida. Para el voltaje ( $V_{H-}$ ), se tiene la misma ecuación (5) pero invertida en el signo, y en ese punto se definirá el disparo por flanco positivo hacia un nivel alto en la salida.

### 2.2.2 DISEÑO DEL INTERFAZ GENERAL DEL PROYECTO.

En este punto, se procederá con el diseño del interfaz básico, pues en él se definirán los elementos electrónicos necesarios que permitan la conexión de un sensor de luz, con un sistema capaz de interpretar la respuesta del mismo y transformar ésta respuesta en una señal digital de salida, para enviarla finalmente hacia un equipo de análisis externo (computador). Este modelo básico, que servirá de base para los diseños particulares de los tres prototipos a construirse, partirá del gráfico proporcionado en la *figura 2.4*, el cual muestra claramente el circuito que se adapta a los propósitos mencionados. Adicionalmente se completará la idea del diseño mediante la conexión de todo el interfaz hacia el pin #11 (*Busy*) del puerto paralelo de un computador PC o compatible, pin que servirá como entrada de la información digital al computador. Las ocho tierras del

conector paralelo, se deberán conectar a la referencia general. La *figura 2.9* muestra el circuito electrónico básico con sus detalles.

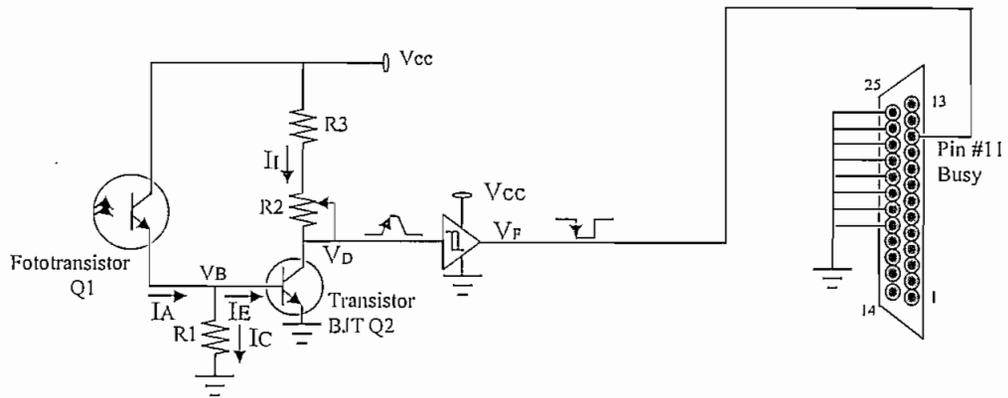


Figura 2.9.- Diagrama electrónico del interfaz básico.

Comenzando el proceso de diseño, veamos que sobre Q1 se tiene la siguiente ecuación, donde  $V_B$  es el voltaje en la resistencia R1 (voltaje de Base):

$$V_{CC} = V_{CEQ1} + V_B \quad (1)$$

Como  $V_B = V_{BE} = 0.6V$ , entonces:

$$V_{CEQ1} = 5V - 0.6V = 4.4V$$

El fototransistor *NTE 3034A*, según los manuales, aproximadamente entrega  $500 \mu A$  cuando se encuentra iluminado. Esta corriente  $I_L$  deberá ingresar a la base del BJT para saturarlo y para eso se deberá cumplir que:  $I_C \ll I_E$  y también  $I_E \approx I_L$ , donde  $I_C$  es la corriente en la resistencia R1,  $I_E$  es la corriente que ingresa a la base del BJT e  $I_L$  es la corriente dada por el fototransistor cuando recibe luz.

Entonces, sea  $I_L = 100 I_C$

$$I_C = I_L / 100 = 500\mu A / 100 = 5\mu A$$

Y como el voltaje  $V_B = I_C * R_1 = 0.6V$  (2)

se tiene que:

$$R_1 = 0.6V / 5\mu A = 120 K\Omega.$$

Si  $R_1$  se toma como  $R_1 = 220 K\Omega, \pm 5\%$ , la operación del BJT en saturación es más segura pues mayor cantidad de corriente ingresa a la base de éste, mientras que en menor grado fluye por  $R_1$ .

Volviendo a calcular  $I_C$ , se ve que:

$$I_C = V_B / R_1 = 0.6V / 220 K\Omega = 2.72 \mu A$$

Para el cálculo del valor del potenciómetro ( $R_2$ ) del circuito, se deberá considerar que, cuando el BJT está en saturación, la mayor cantidad de corriente ( $I_I$ ) fluye a través del potenciómetro ( $R_2$ ). Esta corriente en las mejores condiciones, esto es, cuando el sensor está siendo iluminado, se puede plantearla como:

Sea  $I_I \approx 10 \text{ mA}$ , entonces:

$$V_{CC} = I_I * R_2 + V_{CE_{SAT}(Q_2)} \quad (3)$$

Donde:

$V_{CC}$  = Voltaje de una fuente regulada de alimentación = +5 V<sub>DC</sub>.

$R_2$  = Valor en ohmios del potenciómetro total.

$V_{CE_{SAT}(Q_2)}$  = Voltaje típico del BJT entre el colector y el emisor cuando se está en saturación e igual a 0.25V máximo. Con éstos datos se puede calcular:

$$R_2 = (V_{CC} - V_{CE_{SAT}(Q_2)}) / I_I = (5V - 0.25V) / 10\text{mA} = 475 \Omega.$$

Se propone un valor de  $1\text{ K}\Omega$  en el potenciómetro ( $R2$ ), para efectos de apertura en la calibración del circuito. Este valor de resistencia calculada puede dividirse en una resistencia fija ( $R3$ ) de  $56\ \Omega$ , en serie con un potenciómetro ( $R2$ ) de  $1\text{ K}\Omega$ . Con esto se puede proteger la fuente de alimentación  $V_{cc}$  en el caso de que ( $R2$ ) llegue a su extremo de cortocircuito, es decir  $R2 = 0\ \Omega$ .

Como la salida del BJT ( $V_D$ ) tiene características digitales (ver figura 2.9), se puede conectar en ese punto, la entrada de un “Schmitt trigger”, que tome la forma de onda de  $V_D$ , y la mejore produciendo una señal cuadrada invertida apta para ser llevada finalmente como salida al puerto paralelo del computador, en especial al *pin #11 (Busy)*.

Por último, resta por definir la fuente de alimentación a utilizarse. En el mercado existen muchas alternativas respecto a sistemas de alimentación. Para este proyecto en especial, se ve la necesidad de utilizar un adaptador de voltaje universal para satisfacer la demanda de carga del circuito interfaz diseñado. Sin embargo, habrá que idealizar la forma de conectar dicho adaptador a nuestro sistema, asegurando que el voltaje entregado por éste, no sea nocivo para la circuitería del interfaz construido. Dicho esto, y debido a que los adaptadores universales por lo general no son dispositivos de alimentación regulados, se procederá a utilizar un sistema de regulación de voltaje muy convencional, que utiliza un circuito integrado y algunos elementos electrónicos adicionales de bajo costo. Este regulador deberá tener como características principales una salida de 5 voltios estables y, además, permitir en la entrada un rango de voltaje de 7 a 30 voltios. La corriente necesaria para cubrir la demanda de carga, será de 1 A.

Algo que es importante recalcar es que el adaptador de voltaje también deberá permitir una carga de 1 A para abastecer de energía a todo el sistema, pues si la fuente de alimentación es insuficiente para soportar la carga externa, ésta puede sufrir daños en su interior y posteriormente un mal funcionamiento.

La *figura 2.10*, muestra una fuente de alimentación regulada muy utilizada en el mercado, que utiliza un circuito integrado fácil de conseguir en los almacenes de electrónica.

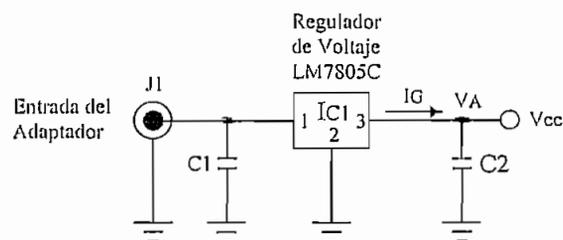


Figura 2.10.- Fuente de alimentación regulada para el circuito interfaz.

En el diagrama anterior se muestra una configuración típica de un dispositivo regulador de voltaje. En éste aparecen dos capacitores C1 y C2, los cuales trabajan como filtros de rizado. Aunque la alimentación entregada por el adaptador a través del conector *J1* es una señal rectificada, el fabricante recomienda la utilización de los dos capacitores para efectos de que el integrado opere adecuadamente. C1 reducirá cualquier rizado pequeño que se presente a la entrada del regulador de voltaje, mientras que C2 asegura un nivel de voltaje DC más plano en el tiempo a la salida del regulador.

Finalmente, se debe diseñar un sistema de iluminación simple que sirva como fuente de luz para hacerlo incidir sobre el sensor (*Q1*) del diseño. En vista

que trabajaremos con luz visible, requeriremos de un arreglo simple que consiste en una batería de 1.5 voltios y un foco que produzca luz convergente a un solo punto. Esta característica lo tienen algunos focos en el mercado que producen iluminación concentrada en un solo punto, como es el caso del número de parte 272-1174 de Radio Shack (#112), que corresponde a un foco de 1.2 voltios, 250 mA. Existe también otra posibilidad de utilizar el número de parte 272-1124 (#243) que corresponde a un foco de 2.33 voltios, 0.27 A. Se necesitará adicionalmente de un interruptor convencional (*SI*), el cual comandará la alimentación de la batería hacia el foco (*LPI*). Aunque el sensor (*QI*) puede trabajar en condiciones de luz del ambiente, es preferible utilizar este sistema de iluminación propia para lograr un mejor desempeño del fototransistor en condiciones operativas. La *figura 2.11*, muestra un diagrama eléctrico elemental del circuito que se requiere para la alimentación del foco *LPI*. Este está asistido por una batería de 1.5 voltios, la cual se irá gradualmente desgastando según el grado de consumo de corriente del foco utilizado. El interruptor *SI*, permitirá el paso de corriente hacia la lámpara.

En la *figura 2.12*, se muestra una idea de como se intenta relacionar el medio físico con el computador. Cuando se logra apuntar un haz de luz hacia la lente de un sensor, ese haz luminoso provoca que el sensor conduzca corriente. Si de alguna forma un cuerpo cualquiera cruza el haz, interrumpiéndolo, entonces el sensor prácticamente no conducirá. Esta situación del sensor es llevada a un sistema electrónico para su análisis y procesamiento. Esta tesis utiliza este método para estudiar los tres fenómenos físicos propuestos mediante la asistencia

de un programa diseñado en un computador. Si el sistema *foco-sensor* se monta estratégicamente en un módulo físico, teniéndose la lámpara frente al sensor, entonces un cuerpo físico cualquiera se podrá mover entre los dos sistemas y producir cambios en las condiciones eléctricas del sensor.

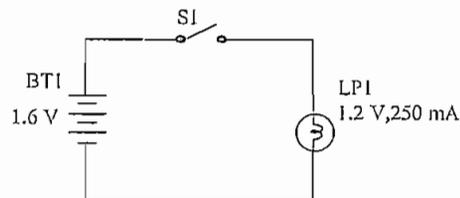


Figura 2.11.- Diagrama del sistema básico de iluminación con luz visible.

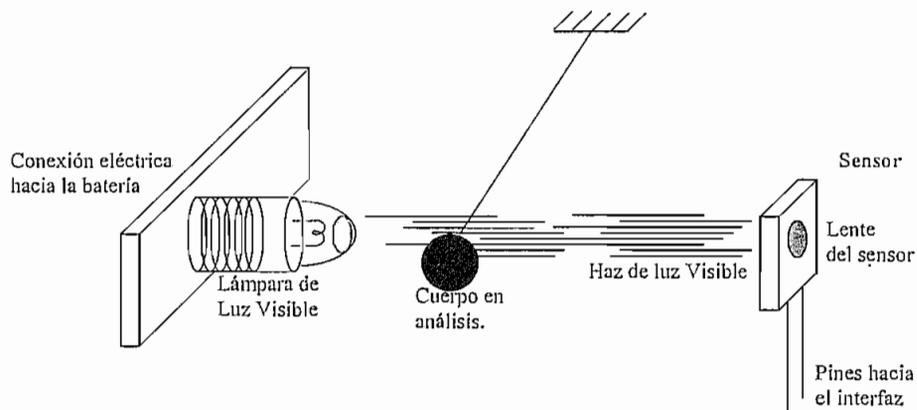


Figura 2.12.- Sistema Foco-Sensor como elementos de análisis del fenómeno físico.

Partiendo de la *figura 2.12*, en la cual se refleja la forma en que una masa pendular es detectada al cruzar un haz generado de luz, se idealizará también, bajo el mismo método, los sistemas adecuados para estudiar el movimiento en un

plano inclinado y en un sistema de caída libre. Los tres equipos a construirse se basan en el mismo funcionamiento: “Un móvil que en su trayectoria de movimiento cruza en algún punto un haz de luz, provocando variaciones de las condiciones eléctricas en un sensor ubicado estratégicamente para el análisis del fenómeno físico”.

### 2.2.3 ELECCION DE LOS ELEMENTOS PARA EL DISEÑO.

La elección de los elementos electrónicos que se utilizarán en el interfaz será de mucha importancia en este proceso. En un diseño por lo general se trata de utilizar el menor número de elementos, que éstos se obtengan fácilmente en el mercado y que además se adapten lo mejor posible para lograr los objetivos planteados. Se comenzará entonces, eligiendo los dispositivos idóneos para montar el circuito de la *figura 2.9*, detallando el por qué de su uso en el diseño.

**IC1: 7805C (NTE 960):** De acuerdo al diagrama de bloques del interfaz mostrado en la *figura 2.1*, la alimentación para todo el circuito consistirá en un sistema integrado básicamente por un regulador de voltaje. Este dispositivo permitirá una entrada no regulada proveniente del adaptador o fuente de alimentación externa, y proporcionar un voltaje de salida de +4.96 voltios DC estables suficientes para alimentar el circuito interfaz. No se puede prescindir de este regulador debido a que los adaptadores o fuentes de voltaje de bajo costo ofrecidas en el mercado, son muy vulnerables a la carga externa conectada, y el interruptor que poseen para selección de los diferentes voltajes de salida, goza de

poca confiabilidad. Es así, que a la entrada de éste dispositivo, se puede ingresar un voltaje continuo de 7 a 20 voltios según datos proporcionados por el fabricante. A la salida se tendrá un voltaje estable en +4.96 voltios DC, que se suministrará a la carga (*circuito final*). Según las notas técnicas del regulador de voltaje (*véase en ANEXOS, NTE960*), se debe evitar no sobrepasar el máximo voltaje de entrada permitido (+35 Voltios DC), y así evitar daños severos al integrado. Obsérvese así mismo que la corriente máxima de salida a plena carga del regulador de voltaje puede ser hasta 1 A, más que suficiente para todo el diseño. Si de alguna forma se alcanza este valor de corriente crítica, será conveniente el uso de un disipador de calor sujeto a la estructura del integrado. Se debe recordar que la máxima corriente que soporta el regulador a plena carga es de 1 A, sin embargo, el hecho de que pueda ofrecer ese máximo valor de corriente dependerá de la fuente de alimentación que antecede al dispositivo. Por ejemplo, si se trata de una fuente de alimentación que soporta hasta 500 mA, el regulador proporcionará máximo 500 mA de corriente a la carga.

**Q1: NTE 3034A:** Este dispositivo sensor ofrece una capacidad de corriente de 500  $\mu$ A típico cuando incide luz sobre él, valor adecuado para ser dirigido hacia la base de un transistor de propósito general (*Q2*), para operar dentro de la región no lineal. Así mismo éste fototransistor cuando no le incide luz, presenta niveles de corriente de oscuridad bajos casi imperceptibles (*en el orden de las centenas de nA*), que son suficientes para dejar de excitar la base del BJT (*Q2*).

**Q2: TRANSISTOR 2N3904 (NTE 123AP):** Este es un transistor para propósitos generales, usado convencionalmente y que soporta hasta 100 mA de corriente continua en el colector. Para estos niveles de corriente no necesita de ningún disipador adicional, pues no maneja corrientes relativamente altas. A través de su estructura disipa hasta 500 mW. Dentro de sus características presenta un voltaje  $V_{CE_{SAT}}$  de 0.25 voltios máximo y no necesita de corrientes elevadas en base para ser conducido a saturación.

**R1: Resistencia de  $120\text{ K}\Omega$ ,  $\pm 5\%$ ,  $\frac{1}{4}\text{ W}$ :** Resistencia que permite la polarización de la base del 2N3904 (*Q2*). Variando esta resistencia se puede controlar la operación del BJT en saturación, pues cambia el valor del voltaje colector-emisor ( $V_{ce}$ ), obteniéndose un valor lo más cercano a 0.25 voltios.

**R2: Potenciómetro de  $1\text{ K}\Omega$  (Para instalación en carcasa, con perilla):** De acuerdo al diseño, este elemento se lo elige de  $1\text{ K}\Omega$ , para efectos de brindar al sistema una amplia apertura en la calibración para el funcionamiento del fototransistor. Deberá ser mecánicamente resistente para soportar el manejo del usuario.

**R3: Resistencia de  $56\ \Omega$ ,  $\pm 5\%$ ,  $\frac{1}{4}\text{ W}$ :** Resistencia utilizada para protección de cortocircuito de la fuente cuando el BJT está en saturación, y cuando el potenciómetro de  $1\text{ K}\Omega$  (*R2*) se encuentre en la posición de  $0\ \Omega$ . Este valor de

resistencia es adecuado, pues valores más altos, quitan rango de sensibilidad al potenciómetro ( $R2$ ) colocado en serie.

**C1 Y C2: Capacitores supresores de rizado:** Estos dispositivos se han elegido de acuerdo a las especificaciones del fabricante del integrado 7805C, en donde se sugiere el uso de capacitores C1 y C2 para aplanar la forma de onda de salida del regulador, reduciendo cualquier rizado que pueda presentarse. De acuerdo a los datos técnicos proporcionados por el fabricante del integrado (*véase en ANEXOS, NTE960*), se obtienen las siguientes características eléctricas de funcionamiento:

$$V_{OUT} = 5V, V_{IN} = 10V, I_O = 500mA, C1 = 2.2 \mu F \text{ y } C2 = 1.0 \mu F, P_D = 1.5 W.$$

Para este diseño se ha decidido utilizar capacitores de cerámica cuyos valores son:  $C1 = C2 = 0.01 \mu F$ , debido a que el voltaje a la entrada del regulador tiene un nivel de rizado insignificante y no se necesita de filtros mayores.

**IC2: Circuito Integrado 74LS14N (NTE74LS14):** Contiene 6 inversores con entradas del tipo “*Schmitt trigger*”, que se usará en el diseño para producir salidas lógicas rápidas y sobre todo invertir y mejorar la forma de onda que entrega el colector del BJT ( $Q2$ ), al operar en corte y saturación.

Como el integrado pertenece al grupo de los dispositivos TTL, puede ser alimentado por la fuente regulada de +5 VDC.

Este integrado vuelve congruente la información dada por el sensor; “cuando hay luz incidente sobre éste, la salida del *Schmitt trigger* entrega un 1L, cuando no existe luz sobre el sensor, la salida del *Schmitt trigger* es ahora de 0L”.

### 2.3 DISEÑO DEL INTERFAZ PARA EL PROTOTIPO “PENDULO SIMPLE”.

Para iniciar el diseño de este prototipo, se utilizará como base el diseño realizado en el apartado 2.2.2. El prototipo “*Péndulo Simple*”, consiste en un sistema conformado por un panel vertical de madera, en donde se montarán dos sensores de luz, similares al sensor (*Q1*) de la *figura 2.9*. Estos sensores se llamarán (*Q3*) y (*Q1*), los cuales serán montados en la parte superior e inferior respectivamente del panel de madera, como se muestra en la *figura 2.13*. Mayores detalles del diseño de la estructura física, véase en *ANEXOS, diagramas de los módulos físicos, diagrama del prototipo péndulo simple*.

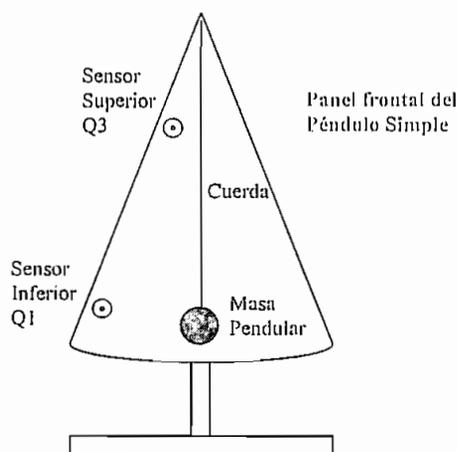


Figura 2.13.- Disposición de los dos sensores de luz en el panel vertical del “péndulo simple”.

Entonces, la existencia de dos sensores, obligará al diseño electrónico a cambiar su apariencia original. El motivo de la introducción de los dos dispositivos, se debe a la necesidad de realizar la práctica de un péndulo, utilizando dos tipos de longitudes. Recordemos que el objetivo de este proyecto, es demostrar que la frecuencia pendular no depende de la masa del péndulo, pero si depende inversamente de la longitud del hilo de donde se sujeta la masa.

Por lo tanto, al montar los sensores en dos posiciones, una posición superior y otra inferior, se está haciendo que el péndulo tenga la posibilidad de trabajar con dos longitudes de cuerda, una longitud pequeña ( $l$ ), y una longitud grande ( $L$ ) respectivamente. La variación de la masa en cada una de las longitudes de cuerda, no provocará cambios en la frecuencia de oscilación pendular. Por lo tanto, en el prototipo se podrá experimentar con varios tipos de masas, masa menor ( $m$ ) y masa mayor ( $M$ ).

Debido a la naturaleza del movimiento físico a analizarse y como cada masa pendular tiene su propio sensor para su estudio, es necesario la activación particular de cada uno de éstos dispositivos para analizar el movimiento pendular. Es así que, para el diseño de "*Péndulo Simple*", se utilizarán dos fototransistores y se los conectarán para ser activados individualmente por medio de un interruptor  $SI$ , instalado en el circuito interfaz. Dicho interruptor comandará la energización de cada uno de los colectores de los dos dispositivos, y así, se podrá seleccionar la operación particular de los dos sensores. Si  $SI$  es un interruptor de tres posiciones, por medio de éste, se conectarán los colectores de los fototransistores contra  $V_{cc}$ , de la siguiente manera:

- La posición *izquierda* conectará el colector del sensor *superior* a  $V_{cc}$ ,
- La posición *derecha* conectará el colector del sensor *inferior* a  $V_{cc}$ .
- La posición *central* se reserva para el “*stand-by*” de los sensores.

Finalmente, los emisores de los dos fototransistores ( $Q1$  y  $Q3$ ), se unirán para luego conectarse a la base del BJT ( $Q2$ ).

Adicionalmente se ha insertado un segundo interruptor en el circuito, se trata de  $S2$ . Este dispositivo es un interruptor simple *ON/OFF* y permitirá el paso de la corriente proveniente de la fuente de alimentación, hacia el regulador de voltaje  $7805C$ . tal como se muestra en la *figura 2.14*.

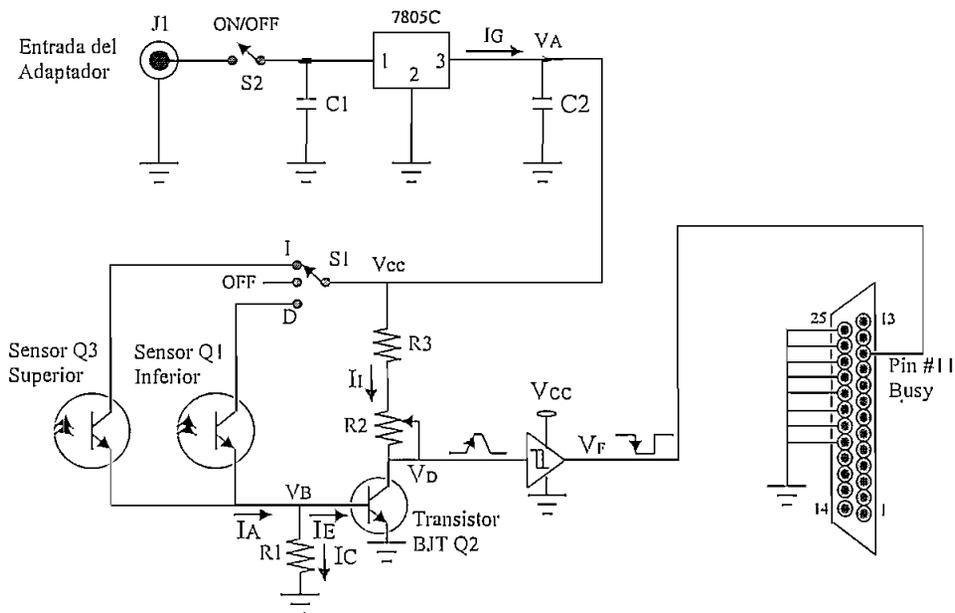


Figura 2.14.- Esquema electrónico del interfaz de “Péndulo Simple”.

El conmutador ( $S1$ ), que ha sido utilizado para seleccionar el sensor, sea éste superior ( $I$ ), o inferior ( $D$ ), tiene dentro de su estructura otro conmutador

montado paralelamente y aislado eléctricamente. Con esta última parte del dispositivo y de la misma forma: sensor **superior** (*posición izquierda*) y sensor **inferior** (*posición derecha*), se accionará la alimentación de la luz **superior** y luz **inferior**, respectivamente.

El diseño utilizará un sistema de iluminación montado en postes de acrílico específicamente contruidos para esto (*véase en ANEXOS, diagramas de los módulos físicos, diagrama del prototipo “péndulo simple”*). Este sistema utiliza dos focos pequeños cuyo voltaje de alimentación es de máximo 2.2 voltios. Debido a su delicada construcción, se los alimentará con una batería de 1.6 voltios individualmente. Ya que las lámparas de este tipo consumen mucha energía, las baterías deberán ser renovarse cada vez que se requiera hacerlo. Es preferible contar con una fuente de alimentación regulada que asista las lámparas utilizadas para iluminar los sensores, pues el uso de baterías no es adecuado por el costo y limitada vida útil.

En el caso del uso prolongado de las baterías, el usuario del equipo notará la debilidad en la intensidad de la luz aplicada sobre los sensores y deberá proceder a cambiar las mismas. Para ello dentro del interfaz se ha montado un portabaterías para hacer fácil el reemplazo de éstas. La *figura 2.15*, muestra el diagrama eléctrico del sistema de iluminación para “péndulo simple”.

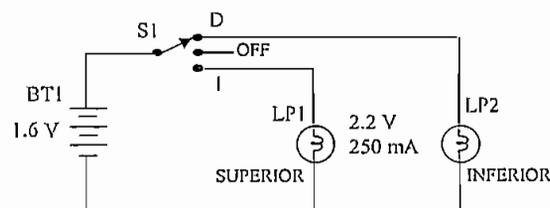


Figura 2.15.- Esquema eléctrico del sistema de iluminación de “péndulo simple”.

Para la manipulación del potenciómetro R2 en el interfaz, será necesario ubicar éste dispositivo en una caja de acrílico diseñada específicamente para proteger el circuito interfaz del prototipo. R2 estará sujeto en la parte frontal de la caja, (véase en ANEXOS, diagramas de los módulos físicos), y será manipulado mediante una perilla para la calibración del circuito electrónico.

#### 2.4 DISEÑO DEL INTERFAZ PARA EL PROTOTIPO DE “CAIDA LIBRE”.

El prototipo de “caída libre” consistirá en un módulo vertical en forma de U, que permita el montaje de dos sensores  $Q1$  y  $Q3$ , distribuidos sobre una de las paredes laterales, con el fin de ubicar los puntos de monitoreo, de la caída de un cuerpo en estudio. Un sensor se ubicará en la parte superior y el otro en la parte inferior de la pared lateral del módulo. Para ser excitados por luz proveniente de una fuente luminosa, será adicionalmente necesario montar los respectivos focos, exactamente al frente de cada uno de los sensores, es decir distribuidos sobre la otra pared lateral del equipo.

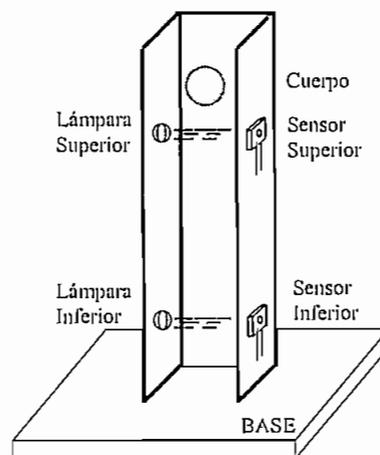


Figura 2.16.- Posicionamiento de luces y sensores para “Caída Libre”.

La *figura 2.16*, muestra el modelo del equipo físico e indica la ubicación de los sensores y los focos a utilizarse para el estudio de la caída libre de un cuerpo. Para mayor información en cuanto a diseño y dimensiones del prototipo, véase *ANEXOS, diagramas de los módulos físicos, diagrama del prototipo "caída libre"*.

En cuanto al diseño electrónico que corresponde a éste prototipo, el nuevo interfaz deberá cambiar en varios aspectos importantes. Primero, el equipo utilizará dos sensores funcionando a la vez, a diferencia del proyecto de péndulo simple, en el cual, el funcionamiento de los dos sensores era coordinado por un conmutador (S1). Para conseguir el funcionamiento de los dos dispositivos, se diseñará dos sistemas similares al de la *figura 2.9*, y se los conectará en paralelo para estudiar, al mismo tiempo, el movimiento del cuerpo en caída libre. Esto implicará el uso de dos entradas al puerto del computador, recibiendo información simultánea y paralela, además de dos sistemas electrónicos similares al diseño básico de la *figura 2.9* y alimentados simultáneamente por *Vcc*. Dicho esto, lo conveniente es montar los circuitos de acuerdo a la *figura 2.17*, donde se muestra el esquema electrónico del nuevo interfaz.

Para "caída libre", adicionalmente se reemplaza el interruptor doble S1, el cual en el proyecto de "péndulo simple" era utilizado para el accionamiento del sistema de iluminación y los sensores, para ahora ser un interruptor destinado solamente a energizar las dos lámparas, para la iluminación de los dos sensores en forma simultánea.

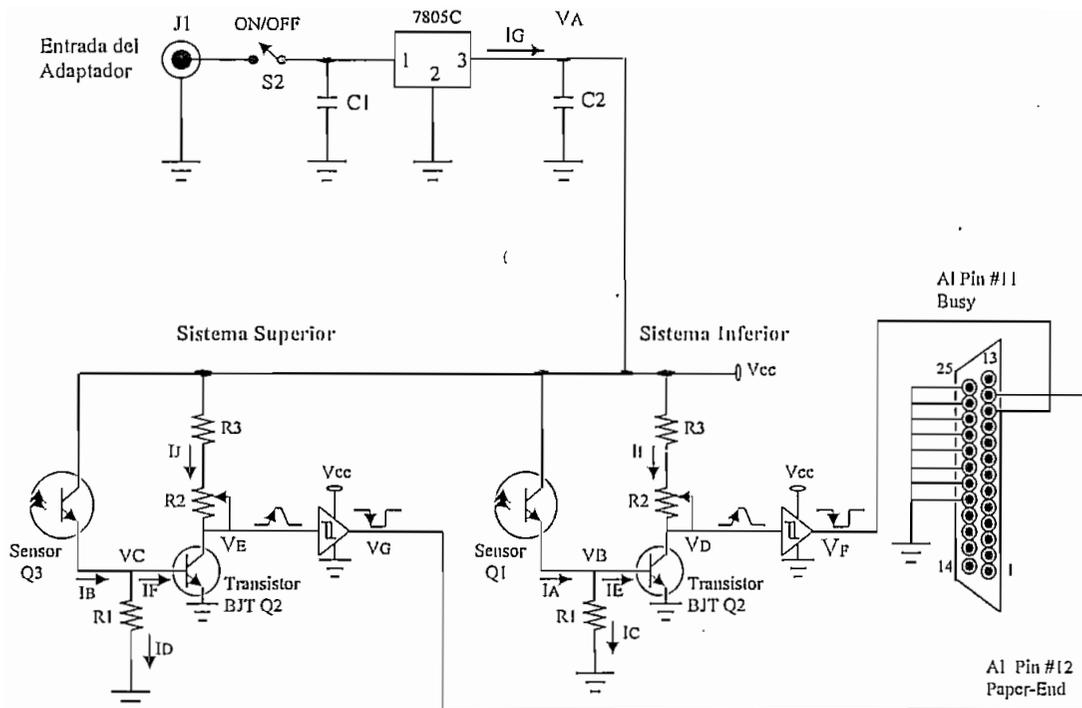


Figura 2.17.- Diagrama electrónico del interfaz para el prototipo de "Caída Libre".

De igual forma que en el proyecto anterior, el interruptor S2 cumplirá con la función de energizar todo el circuito interfaz, y será instalado entre la salida de la fuente de alimentación y la entrada al regulador 7805C.

Otro punto de importancia en los cambios efectuados en el diseño, será la disposición de los elementos de manipulación del usuario, como son las perillas. Estas deberán montarse en la caja de acrílico, la cual albergará todo el circuito interfaz, y deberán estar suficientemente separadas para facilidad de operación y comodidad en el proceso de calibración del sistema, (*véase en ANEXOS, diagramas de los módulos físicos*).

Finalmente ubicaremos las dos salidas lógicas obtenidas, en los pines #11 (Busy) y #12 (Paper End) del conector DB-25 del puerto paralelo. El pin #11



primero como base y el segundo móvil respecto al primer plano, sujetos a través de una bisagra ubicada en un lado común de las dos piezas, para efectos de que el plano móvil disponga de libertad de giro, de acuerdo a la *figura 2.19a* y *2.19b*. La disposición de los sensores a lo largo del plano, para el análisis del movimiento de un cuerpo, se realiza ubicando un sensor en la parte superior del plano móvil y otro en la parte inferior del mismo. Adicionalmente, se montarán dos postes de iluminación, en los cuales se montarán las lámparas que proveerán de luz visible a cada sensor instalado.

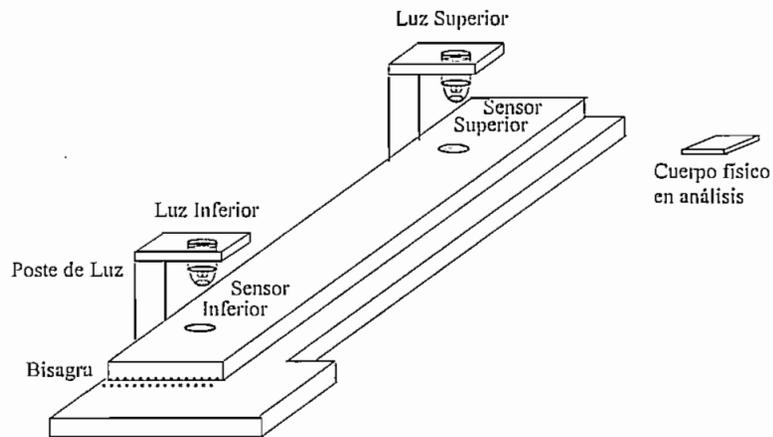


Figura 2.19a.- Modelo físico del prototipo de "Plano Inclinado".

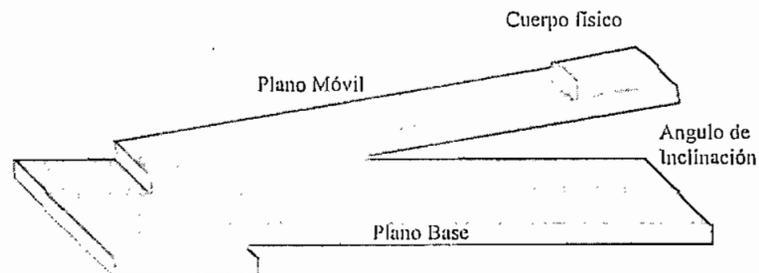


Figura 2.19b.- Vista 3D "Plano Inclinado".

De igual forma, el computador recibirá la información de cada uno de éstos sensores a través de los pines #11 (Busy) y #12 (Paper End) que corresponderán al sensor superior e inferior respectivamente. En lo que respecta a disposición de perillas para calibración del sistema, modelo de la caja de acrílico del circuito interfaz, y diseño de la estructura mecánica puede revisarse en *ANEXOS, diagramas de los módulos físicos*.

Los interruptores *S1* y *S2*, son dispositivos que tienen igual funcionamiento que dentro del interfaz diseñado para "Caída Libre". En la *figura 2.20*, se muestra el diagrama electrónico para el interfaz del prototipo "Plano Inclinado", y la *figura 2.21* indica el circuito de iluminación de los sensores ubicados en el plano móvil.

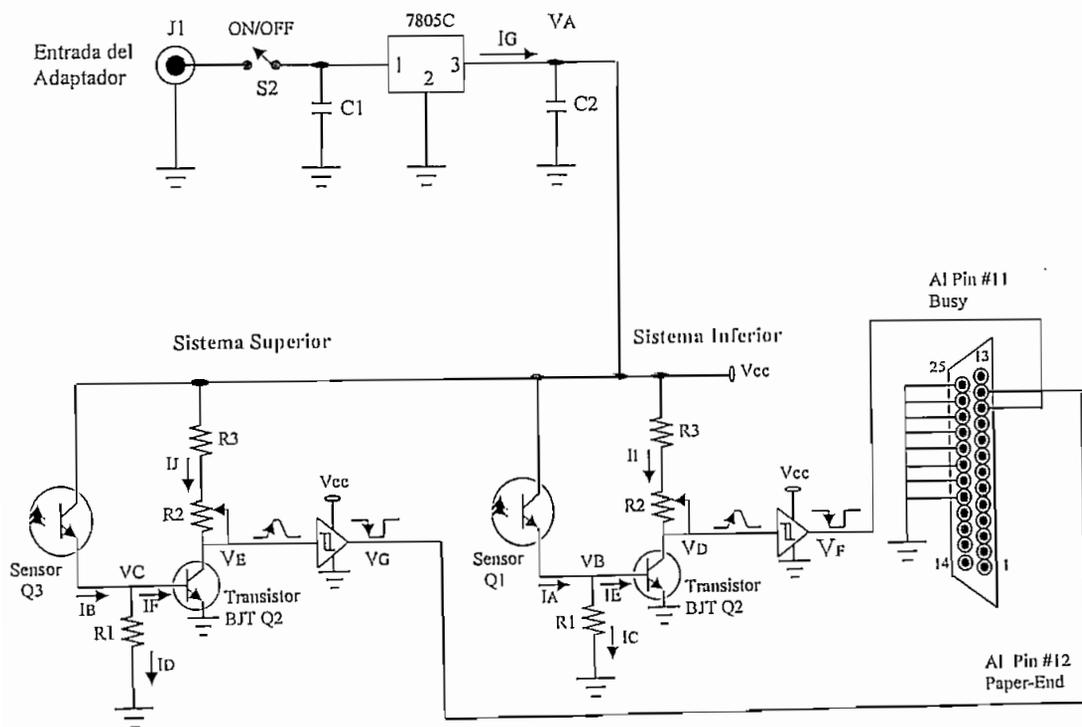


Figura 2.20.- Diagrama electrónico del interfaz para el prototipo de "Plano Inclinado".

## CAPITULO III

### DISEÑO DEL PROGRAMA EN VISUAL BASIC.

#### 3.1 DIAGRAMA DE BLOQUES DEL PROGRAMA COMPUTACIONAL.

El diseño del programa en éste proyecto está basado en una filosofía de programación orientada a objetos **OOP** (*Object Oriented Programing*), distinta a la programación estructurada convencional.

En programación estructurada, las acciones a realizar se caracterizan por algoritmos los mismos que se implementan con procedimientos y funciones. A éstos algoritmos se aplican las estructuras de datos adecuadas y funcionan. En cambio, la programación orientada a objetos, como su nombre lo indica, se basa en objetos. Cada objeto tiene sus características, y lo más importante es que se pueden tener características comunes entre objetos, esto es lo que le hace diferente a este tipo de programación. Un objeto es como algo de la vida real, que contiene un conjunto de atributos y características que describen su naturaleza y funcionalidad.

En el diseño de un programa basado en una filosofía orientada a objetos, se tiene la siguiente secuencia de acciones:

- **Definir el Problema.**- ¿Qué es lo que se quiere hacer?, ¿Cuáles son los objetivos del programa?.
- **Identificar los objetos y sus clases.**- Plantear las herramientas u objetos con los cuales se trabajará e interactuará en el programa.

- **Determinar los métodos requeridos para cada clase.**- Establecer cómo actuará un determinado objeto en forma particular.
- **Escribir el programa principal.**- Este consiste en el código general: Variables, dimensionamiento, declaraciones , funciones, etc.
- **Determinar los elementos que constituyen cada clase.**- Para un objeto o clase de objeto, se determinarán los elementos que permitan su funcionamiento e interrelación con otros objetos.
- **Realizar los métodos de cada clase.**- Ejecutar los métodos establecidos para cada clase.

Para diseñar el programa en Visual Basic (Lenguaje orientado a Objetos), se define el problema de la siguiente forma:

*Se requiere crear un programa que permita al usuario estudiar tres fenómenos físicos característicos: "Caída Libre", "Plano Inclinado" y "Péndulo Simple". Para esto se contará con el hardware de los tres equipos, construidos específicamente para reproducir dichos fenómenos.*

El programa mostrará un menú para la realización de dos actividades básicas:

1. **Simulación.** En la cual se realizará la simulación del movimiento de tres fenómenos en estudio: "Caída Libre", "Plano Inclinado" y "Péndulo Simple".
2. **Práctica.** Dentro de la cual, el usuario interactúa con los equipos construidos en el proyecto.

La *figura 3.1, pag. 76*, indica el diagrama de bloques de la parte principal del programa. Según la elección dada, se opta por uno de los tres casos disponibles para estudio: **CL = Caída Libre, PI = Plano Inclinado, PS = Péndulo Simple.**

## PROGRAMA PRINCIPAL

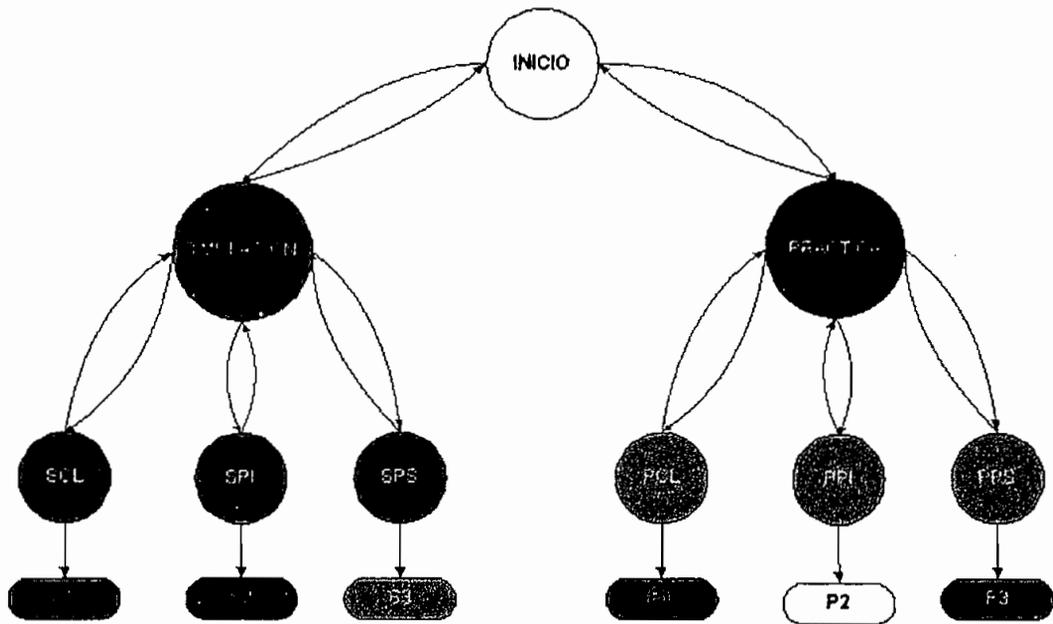


Figura 3.1.- Diagrama de bloques del programa principal.

Nótese que cada experiencia a elegirse, ya sea dentro de la simulación o dentro de la práctica, tiene relación directa con un bloque de ejecución particular:

### **SIMULACION:**

- **SCL** = Simulación de Caída Libre, sección **S1**.
- **SPI** = Simulación de Plano Inclinado, sección **S2**.
- **SPS** = Simulación de Péndulo Simple, sección **S3**.

### **PRACTICA:**

- **PCL** = Simulación de Caída Libre, sección **P1**.
- **PPI** = Simulación de Plano Inclinado, sección **P2**.
- **PPS** = Simulación de Péndulo Simple, sección **P3**.

## BLOQUES DE SIMULACION DE LOS TRES EXPERIMENTOS

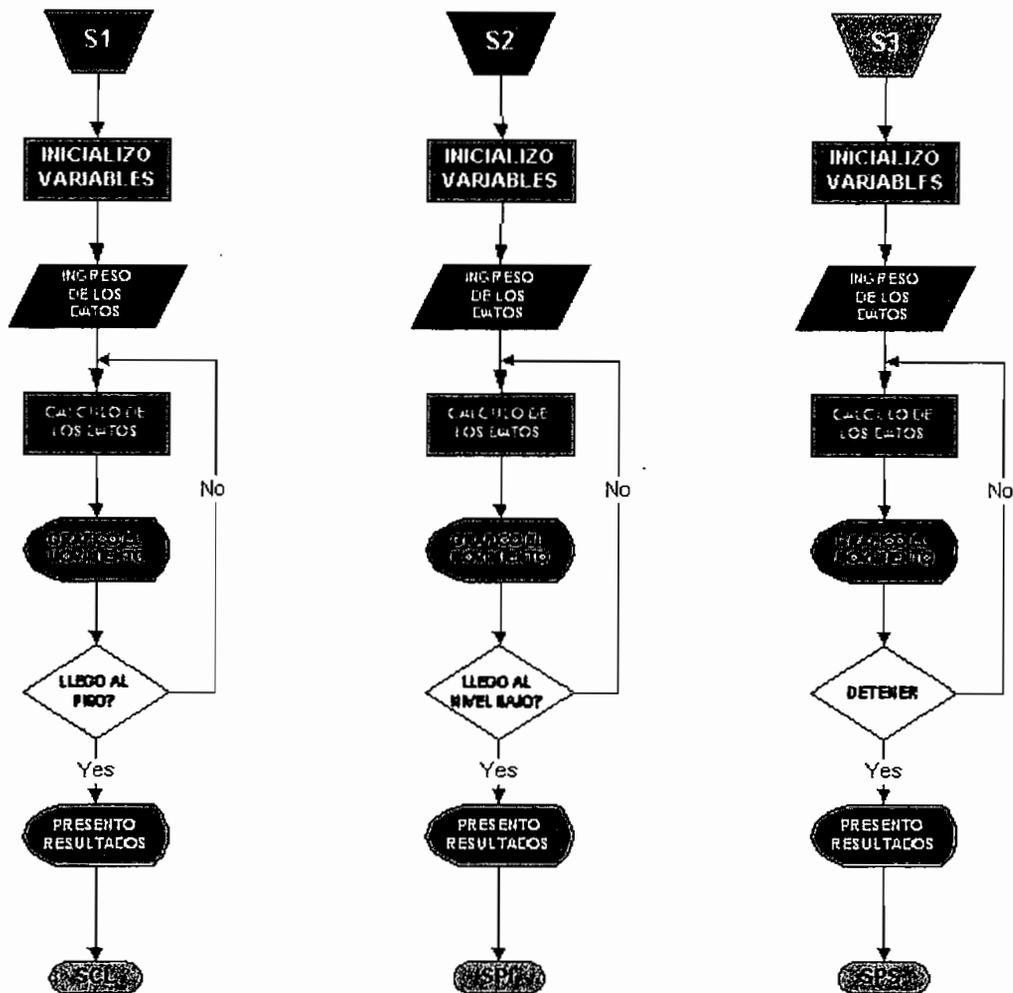


Figura 3.2.- Diagrama de bloques para la parte de simulación de los tres experimentos.

La *figura 3.2*, indica los diagramas de bloques para cada caso particular en el proceso de simulación. En cualquiera de los tres casos, el proceso es similar: Se inicializan las variables, reseteándose los contadores y variables almacenadas. Se solicita el ingreso de información válida para la simulación, luego se usan rutinas

de cálculo para resolver las ecuaciones del movimiento y así graficar la simulación del cuerpo.

Para "Caída Libre" y "Plano Inclinado", se indaga la finalización del movimiento. En el caso de ser verdadero, se procede a mostrar los resultados finales, de lo contrario, continúa el proceso de cálculo del movimiento. Similar situación sucede con "Péndulo Simple", con la diferencia de que el movimiento es perpetuo y el usuario decide cuándo termina la simulación.

BLOQUES CORRESPONDIENTES A LAS PRÁCTICAS EN LOS TRES EXPERIMENTOS

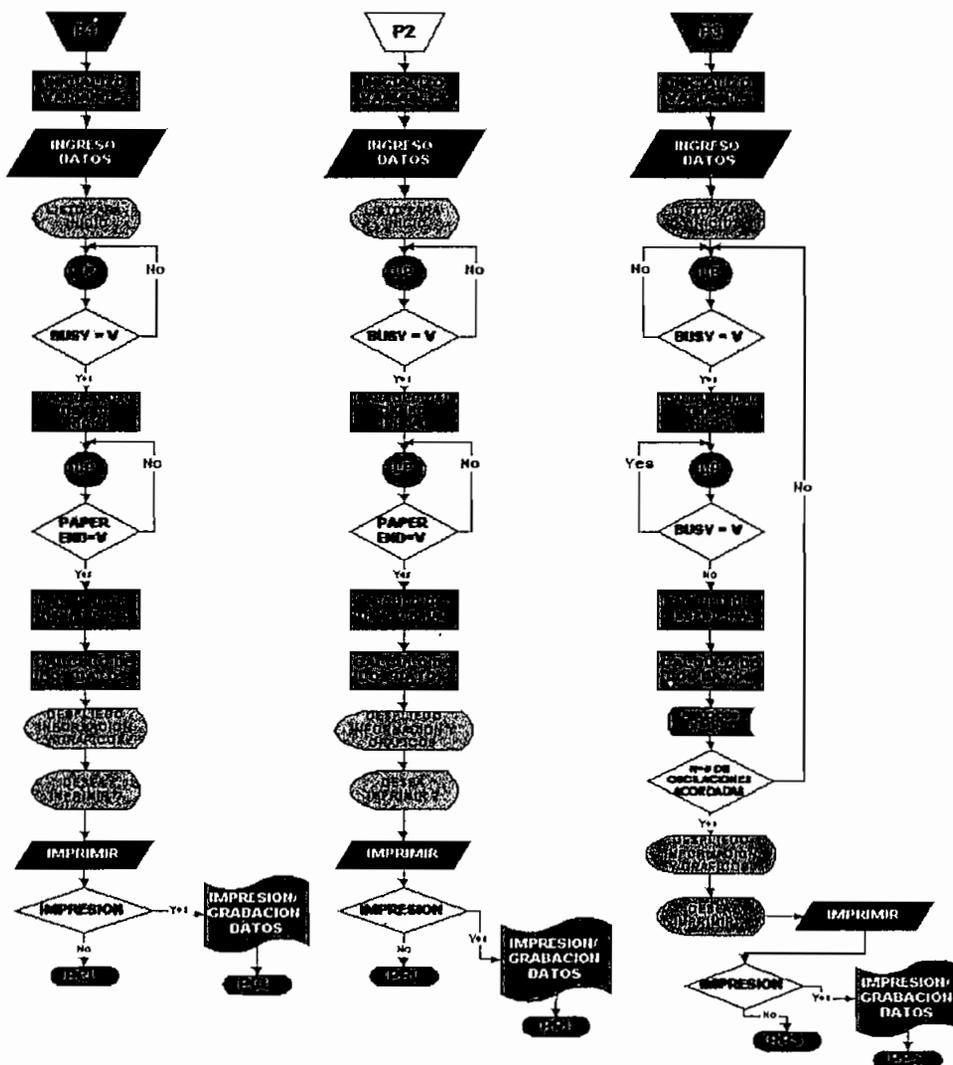


Figura 3.3.- Diagrama de bloques de la parte práctica en los experimentos físicos.

La figura 3.3, pag. 78, muestra los diagramas de bloques correspondientes a las partes prácticas de los tres experimentos físicos. Para "Caída Libre" y "Plano Inclinado" se empieza con la inicialización de las variables, luego se da entrada a los datos de información respectivos de la práctica. El programa genera un mensaje de aviso para indicar al usuario la preparación y el comienzo del experimento.

El programa deberá conocer el estado del puerto, para esto se realiza la lectura del mismo (**LP**), tal como se describe en el diagrama de bloques de la figura 3.4, pag. 81. Es ese momento se procede con la verificación del pin #11 del puerto (**Busy**), el mismo que muestra el estado del sensor superior en los prototipos "Caída Libre" y "Plano Inclinado". Si se detecta un estado de falso, el programa reconoce que aún no empieza el movimiento y continua chequeando el puerto, caso contrario se toma una muestra del tiempo en ese instante. Inmediatamente a este proceso se vuelve a leer el puerto, esta vez verificándose la actividad del pin #12 del puerto (**Paper-End**).

Si el pin #12 presenta una respuesta falsa, es decir, si el cuerpo aún no es detectado por el sensor inferior, entonces continuará calculándose el tiempo de caída entre los dos sensores, de lo contrario se realizan los cálculos respectivos utilizándose las ecuaciones del movimiento, para luego desplegar la información resultante obtenida.

Los resultados obtenidos pueden ser enviados directamente a una impresora o almacenados en un archivo de texto, de otra forma se puede optar por reiniciar la práctica (**PCL**) ó (**PPI**).

En "Péndulo Simple", luego del ingreso de los datos y el aviso de iniciación de la práctica, se da lectura al pin #11 del puerto paralelo (LP). Si existe una respuesta negativa, el puerto seguirá a la espera de detectar un estado diferente. De otro modo, si la respuesta es positiva, se mide el tiempo actual, dándose enseguida lectura al mismo puerto (LP). En este punto, si se recibe una afirmación, se procede a leer otra vez el puerto hasta que se reciba un estado de falso, el cual indica que el puerto detectó la primera oscilación cumplida por el péndulo.

Llegado a este estado, se realiza el cálculo de los datos, se aumenta el contador de períodos en una unidad y se compara el número de períodos sucedidos con el número de períodos solicitados por el usuario. Si la comparación es falsa, se vuelve a repetir el proceso de lectura del puerto (LP), de otra forma el programa despliega la información de los resultados, con la opción de que éstos puedan ser impresos o enviados a un archivo de texto.

El bloque de lectura (LP) es utilizado para acceso al puerto paralelo del computador. En este se reciben las llamadas para la verificación del estado actual de los pines de entrada #11 (*Busy*) y #12 (*Paper-End*), utilizados en el proyecto. Luego de realizarse la lectura del puerto se retorna al lugar de donde se hizo la llamada. La *figura 3.4, pag.81*, muestra su configuración.



### BLOQUE DE LECTURA DEL PUERTO PARALELO

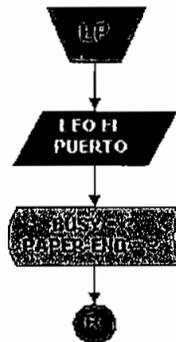


Figura 3.4.- Diagrama de bloques del acceso al puerto paralelo.

Tanto para "Caída Libre", "Plano Inclinado" o "Péndulo Simple", deberá existir la opción de un bloque de calibración del equipo (calibración por Software). Este bloque de código permitirá la puesta a punto o afinamiento de los prototipos para obtener una adecuada respuesta de los mismos. El bloque de calibración es una herramienta del usuario al elegir la opción de **Práctica** con los tres prototipos y se describe en la *figura 3.5*.

### BLOQUE DE CALIBRACION (CAL)

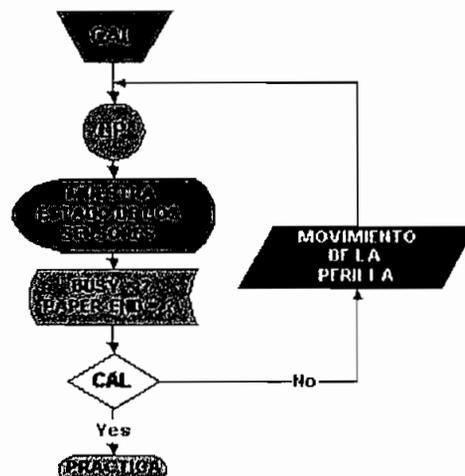


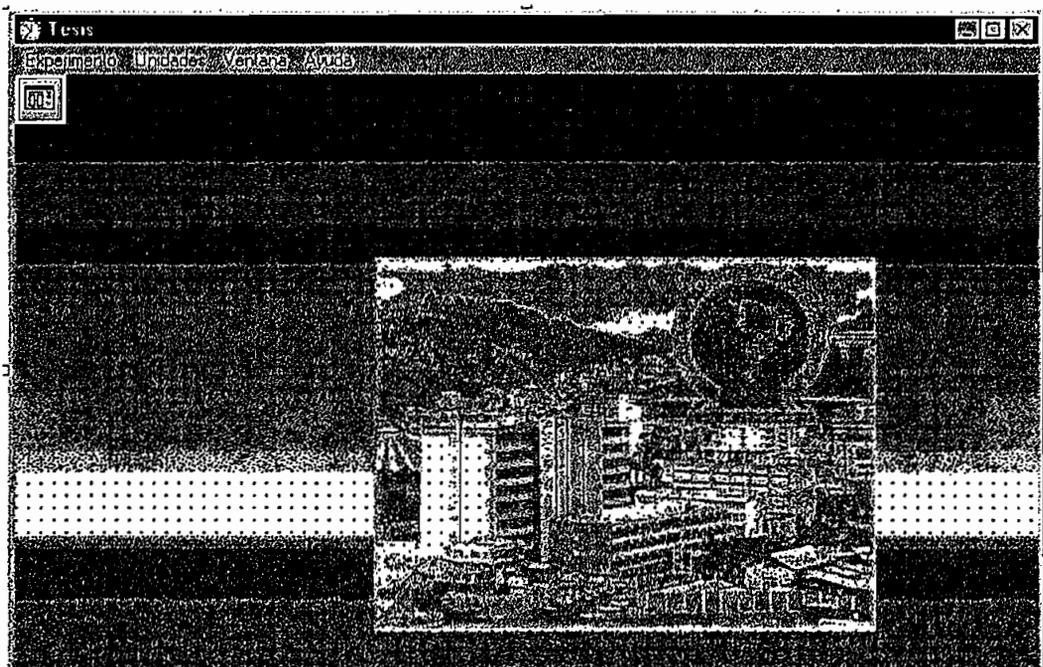
Figura 3.5.- Diagrama de bloques del módulo de Calibración (CAL).



### 3.2 VENTANAS DE FORMULARIOS Y CODIGO DE PROGRAMA.

EFC (Experimentos de Física por Computador), utiliza la filosofía de programación orientada a objetos. Este programa está desglosado en ventanas de trabajo o formularios. Cada formulario contiene los objetos que interactuarán con el usuario y está descrito por código de programa, dentro del cual se escriben los comandos de BASIC que permitirán el funcionamiento de los objetos. En ciertos casos se tienen líneas de código de programa que interrelacionan objetos de otros formularios o hacen llamadas a funciones públicas establecidas. A continuación se muestran las ventanas utilizadas en EFC y el código de programa utilizado:

#### Principal.frm



#### Fprincipal (Code)

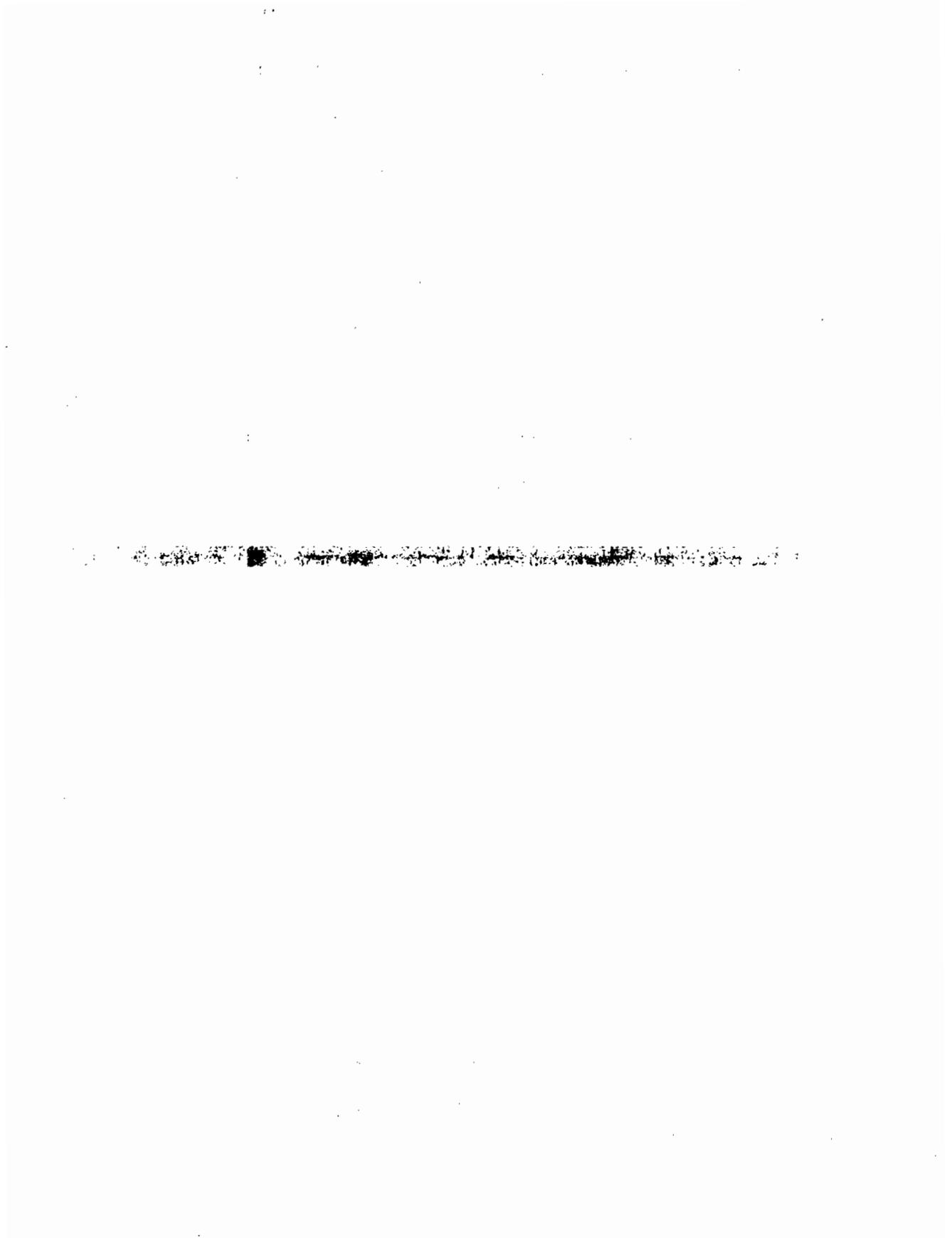
'Acciones a tomarse el momento que el usuario elije  
'una de las opciones del menú "Experimento"

```
Private Sub Experimento_Click(Index As Integer)
```

```
    Select Case Index
```

```
        Case 11:
```

```
            FSimulacion.Show
```



```

    Case 12:
        FPractica.Show
    Case 14:
        End
    End Select
End Sub

Private Sub MDIForm_Load()
    Caratula.Top = 100
    Caratula.Left = 3300
    Caratula.Width = 5175
    Caratula.Height = 6180
    Caratula.Show
End Sub

Private Sub Principal_Click(Index As Integer)
    If Index = 40 Then
        Help_control.HelpFile = ".\Tesis.hlp"
        Help_control.HelpCommand = cmdHelpForceFile
        Help_control.ShowHelp
    End If
End Sub

Private Sub Unidades_Click(Index As Integer)
    Dim i As Integer
    Select Case Index
        Case 21:
            Unidades(21).Checked = True
            Unidades(22).Checked = False
            Unidades(23).Checked = False
        Case 22:
            Unidades(21).Checked = False
            Unidades(22).Checked = True
            Unidades(23).Checked = False
        Case 23:
            Unidades(21).Checked = False
            Unidades(22).Checked = False
            Unidades(23).Checked = True
    End Select
    On Error Resume Next
    ActiveForm.Establecer_Sistema
End Sub

'Acciones a tomarse el momento que el usuario elije
'una de las opciones del menú "Ventana"
Private Sub Ventana_Click(Index As Integer)
    Select Case Index

```

```

Case 31:
  FPrincipal.Arrange vbCascade
Case 32:
  FPrincipal.Arrange vbTileVertical
End Select
End Sub

```

### Carátula.frm



### Carátula (Code)

```

' Al presionar el botón Inicio01, la
' ventana de la carátula desaparece
Private Sub Inicio01_Click()
  Caratula.Hide
  Barra.Show
End Sub

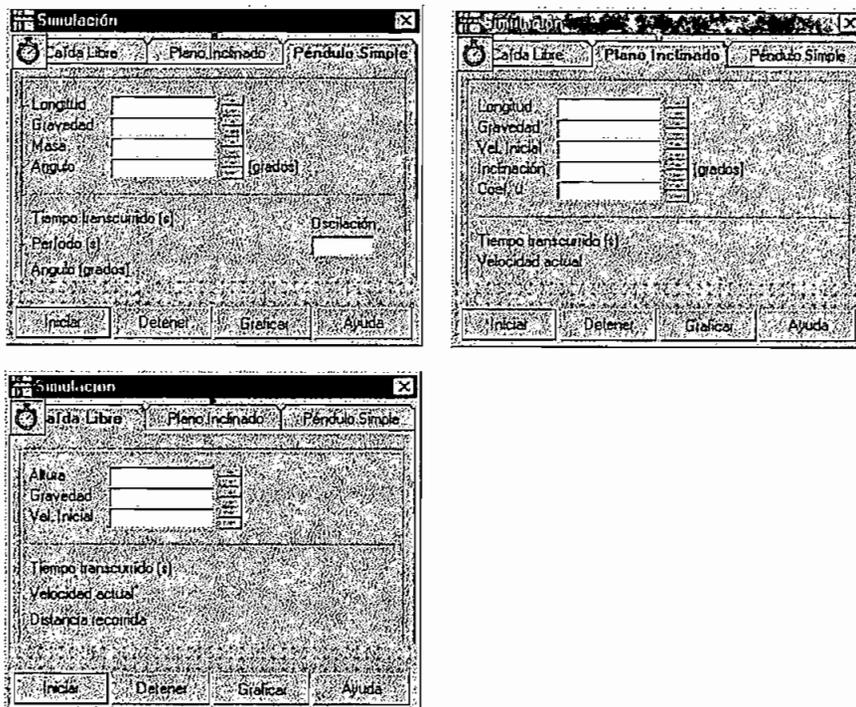
```

```

If Index = 1 Then
    FPractica.SSTab2.Tab = 1
    FPractica.Show
End If
End Sub
Private Sub Form_Load()
    Barra.Top = 0
    Barra.Left = 0
    Barra.Width = 5055
    Barra.Height = 2715
End Sub

```

## Simulación.frm



## Simulación (Code)

```

Private mvarVo As Double 'Variable de velocidad inicial
Private mvarG As Double 'Variable de gravedad
Private mvarS As Double 'Variable de altura
Private mvarI As Double 'Variable de inclinación
Private mvarU As Double 'Variable de rozamiento
Private mvarTo As Double 'Variable de tiempo de inicio
Private mvarAo As Double 'Variable de ángulo inicial
Private mvarA As Double 'Variable de ángulo
Private mvarL As Double 'Variable de Longitud
Private p, q As Integer

```

.....

.....

```

Private r As Double
Private Sub Ayuda_Click()
    If SSTab1.Tab = 0 Then
        h_id = "caida_libre"
    End If
    If SSTab1.Tab = 1 Then
        h_id = "plano_inclinado"
    End If
    If SSTab1.Tab = 2 Then
        h_id = "pendulo_simple"
    End If
    FPrincipal.Help_control.HelpFile = ".\tesis.hlp"
    FPrincipal.Help_control.HelpKey = h_id
    FPrincipal.Help_control.HelpCommand = cdlHelpForceFile
    FPrincipal.Help_control.ShowHelp
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la altura
Private Sub CL_Altura_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(CL_Altura.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Gravedad
Private Sub CL_Gravedad_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(CL_Gravedad.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Velocidad Inicial
Private Sub CL_Velocidad_Inicial_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(CL_Velocidad_Inicial.Text, Chr(KeyAscii)) = 0 Then Exit
Sub
    End Select
    KeyAscii = 0

```

```

End Sub

'Llama al procedimiento que fija el sistema de unidades con
'el que se está trabajando
Private Sub Form_Activate()
    Establecer_Sistema
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Longitud
Private Sub PI_Longitud_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PI_Longitud.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Gravedad
Private Sub PI_Gravedad_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PI_Gravedad.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Inclinación
Private Sub PI_Inclinacion_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PI_Inclinacion.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el número ingresado no sea mayor a 90
Private Sub PI_Inclinacion_Change()
    If Val(PI_Inclinacion.Text) > 90 Then PI_Inclinacion.Text = "90"
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Rozamiento
Private Sub PI_Rozamiento_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii

```

```

    Case 8, 48 To 57: Exit Sub
    Case 46: If InStr(PI_Rozamiento.Text, Chr(KeyAscii)) = 0 Then Exit Sub
End Select
KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Velocidad Inicial
Private Sub PI_Velocidad_Inicial_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PI_Velocidad_Inicial.Text, Chr(KeyAscii)) = 0 Then Exit
Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Longitud
Private Sub PS_Longitud_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PS_Longitud.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para la Masa
Private Sub PS_Masa_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PS_Masa.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

'Verifica que el usuario no ingrese valores inválidos
'(no numéricos o menores a cero) en el campo reservado
'para el Angulo Inicial
Private Sub PS_Angulo_Inicial_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 45: PS_Angulo_Inicial.Text = -1 * Val(PS_Angulo_Inicial.Text)

```

```

    Case 46: If InStr(PS_Angulo_Inicial.Text, Chr(KeyAscii))=0 Then Exit Sub
End Select
KeyAscii = 0
End Sub

```

**'Verifica que el número ingresado no sea mayor a 45  
'ni menor a -45**

```

Private Sub PS_Angulo_Inicial_Change()
    Select Case Val(PS_Angulo_Inicial.Text)
        Case Is > 45: PS_Angulo_Inicial.Text = "45"
        Case Is < -45: PS_Angulo_Inicial.Text = "-45"
    End Select
End Sub

```

**'Verifica que el usuario no ingrese valores inválidos  
'(no numéricos o menores a cero) en el campo reservado  
'para la Gravedad**

```

Private Sub PS_Gravedad_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 8, 48 To 57: Exit Sub
        Case 46: If InStr(PS_Gravedad.Text, Chr(KeyAscii)) = 0 Then Exit Sub
    End Select
    KeyAscii = 0
End Sub

```

**'Detiene los cronómetros que controlan la simulación  
'cuando se presiona el botón de Detener, y restauran  
'los controles a su estado original**

```

Private Sub Detener_Click()
    Select Case SSTab1.Tab
        Case 0:
            Timer0.Enabled = False
            'Calculo de datos para el gráfico
            'y escribe en las tablas correspondientes
            If q = 1 Then
                FGráfico_Funcion.Listax.AddItem 0
                FGráfico_Funcion.Listay1.AddItem Val(CL_Velocidad_Inicial)
                FGráfico_Funcion.Listay2.AddItem 0
                mvarT = Val(CL_Tiempo)
                For p = 1 To 17
                    FGráfico_Funcion.Listax.AddItem Val((mvarT / 17) * p)
                    FGráfico_Funcion.Listay1.AddItem Val(CL_Vf(mvarG, (mvarT / 17) * p, mvarVo))
                    FGráfico_Funcion.Listay2.AddItem Val(CL_s(mvarG, (mvarT / 17) * p, mvarVo))
                Next p
            End If
    End Select

```

```

Case 1:
    Timer1.Enabled = False
    If q = 1 Then
        FGráfico_Funcion.Listax.AddItem 0
        FGráfico_Funcion.Listay1.AddItem Val(PI_Velocidad_Inicial)
        FGráfico_Funcion.Listay2.AddItem 0
        FGráfico_Funcion.Listax2.AddItem Val(PI_At(mvarG, mvarU, mvarI))
        mvarT = Val(PI_Tiempo)
        For p = 1 To 17
            FGráfico_Funcion.Listax.AddItem Val((mvarT / 17) * p)
            FGráfico_Funcion.Listax2.AddItem Val(PI_At(mvarG, mvarU,
                mvarI))
            FGráfico_Funcion.Listay1.AddItem Val(PI_Vf(mvarG, (mvarT / 17)
                * p, mvarI, mvarU, mvarVo))
            FGráfico_Funcion.Listay2.AddItem Val(PI_s(mvarG, (mvarT / 17) *
                p, mvarI, mvarU, mvarVo))
        Next p
    End If
Case 2:
    Timer2.Enabled = False
    If q = 1 Then
        'Calculo de datos para el gráfico
        'y escribe en las tablas correspondientes
        FGráfico_Funcion.Listax.AddItem 0
        FGráfico_Funcion.Listay1.AddItem Val(PS_Angulo_Inicial)
        FGráfico_Funcion.Listay2.AddItem Val(PS_Angulo_Inicial)
        mvarT = Val(PS_Periodo)
        For p = 1 To 17
            FGráfico_Funcion.Listax.AddItem Val((mvarT / 17) * p)
            FGráfico_Funcion.Listay1.AddItem Val(PS_A(mvarAo, mvarL,
                mvarG, (mvarT / 17) * p))
            FGráfico_Funcion.Listay2.AddItem Val(PS_A(mvarAo, mvarL,
                mvarG, (mvarT / 17) * p))
        Next p
    End If
End Select
FPrincipal.Principal(20).Enabled = True
Frame1.Enabled = True
Iniciar.Enabled = True
Detener.Enabled = False
End Sub
'Cierra las ventanas con gráficos antes de cerrar
'la ventana principal de simulación
Private Sub Form_Unload(Cancel As Integer)
    ToggleButton1.Value = False
End Sub

```

'Bloquea la ventana para que no se modifiquen los  
'parámetros bajo los cuales se realiza la simulación,  
'inicializa las variables, activa los cronómetros que  
'controlan cada simulación e inicializa los gráficos

```
Private Sub Iniciar_Click()  
    Frame1.Enabled = False  
    Iniciar.Enabled = False  
    Detener.Enabled = True  
    FPrincipal.Principal(20).Enabled = False  
    mvarTo = Timer  
    q = 0  
    Select Case SSTab1.Tab  
        Case 0:  
            mvarG = Val(CL_Gravedad.Text)  
            mvarS = Val(CL_Altura.Text)  
            mvarVo = Val(CL_Velocidad_Inicial)  
            Call FCaidaLibre.Inicializar(mvarS)  
            p = 1  
            va = 0  
            'r = Val(CL_Altura.Text) / (17 * 3)  
            r = Val(CL_Altura.Text) / 17  
            Timer0.Enabled = True  
  
        Case 1:  
            mvarG = Val(PI_Gravedad.Text)  
            mvarS = Val(PI_Longitud.Text)  
            mvarI = Val(PI_Inclinacion.Text)  
            mvarU = Val(PI_Rozamiento.Text)  
            mvarVo = Val(PI_Velocidad_Inicial.Text)  
            Call FPianoInclinado.Inicializar(mvarS, mvarI)  
            Timer1.Enabled = True  
  
        Case 2:  
            mvarAo = Val(PS_Angulo_Inicial.Text)  
            mvarG = Val(PS_Gravedad.Text)  
            mvarL = Val(PS_Longitud.Text)  
            PS_Periodo = PS_T(mvarG, mvarL)  
            PS_Frecuencia = PS_F(mvarG, mvarL)  
            Call FPenduloSimple.Inicializar(mvarAo)  
            If mvarAo <> 0 Then  
                q = 1  
            End If  
            Timer2.Enabled = True  
    End Select  
End Sub
```

```
'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SB_PLongitud_SpinDown()
    If Val(PI_Longitud.Text) - 0.01 < 0 Then Exit Sub
    PI_Longitud.Text = Str(Val(PI_Longitud.Text) - 0.01)
End Sub
```

```
'Scroll de incremento de cantidad en 0.01
Private Sub SB_PLongitud_SpinUp()
    PI_Longitud.Text = Str(Val(PI_Longitud.Text) + 0.01)
End Sub
```

```
'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SB_SAltura_SpinDown()
    If Val(CL_Altura.Text) - 0.01 < 0 Then Exit Sub
    CL_Altura.Text = Str(Val(CL_Altura.Text) - 0.01)
End Sub
```

```
'Scroll de incremento de cantidad en 0.01
Private Sub SB_SAltura_SpinUp()
    CL_Altura.Text = Str(Val(CL_Altura.Text) + 0.01)
End Sub
```

```
'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SB_SGravedad_SpinDown()
    If Val(CL_Gravedad.Text) - 0.01 < 0 Then Exit Sub
    CL_Gravedad.Text = Str(Val(CL_Gravedad.Text) - 0.01)
End Sub
```

```
'Scroll de incremento de cantidad en 0.01
Private Sub SB_SGravedad_SpinUp()
    CL_Gravedad.Text = Str(Val(CL_Gravedad.Text) + 0.01)
End Sub
```

```
'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SB_SVInicial_SpinDown()
    If Val(CL_Velocidad_Inicial.Text) - 0.01 < 0 Then Exit Sub
    CL_Velocidad_Inicial.Text = Str(Val(CL_Velocidad_Inicial.Text) - 0.01)
End Sub
```

```
'Scroll de incremento de cantidad en 0.01
Private Sub SB_SVInicial_SpinUp()
    CL_Velocidad_Inicial.Text = Str(Val(CL_Velocidad_Inicial.Text) + 0.01)
End Sub
```



```

Private Sub SB_SLAltura_SpinUp()
    CL_Altura.Text = Str(Val(CL_Altura.Text) + 0.01)
End Sub
Private Sub SB_SLAltura_SpinDown()
    If Val(CL_Altura.Text) - 0.01 < 0 Then Exit Sub
    CL_Altura.Text = Str(Val(CL_Altura.Text) - 0.01)
End Sub

Private Sub SB_SLGravedad_SpinDown()
    If Val(CL_Gravedad.Text) - 0.01 < 0 Then Exit Sub
    CL_Gravedad.Text = Str(Val(CL_Gravedad.Text) - 0.01)
End Sub

Private Sub SB_SLGravedad_SpinUp()
    CL_Gravedad.Text = Str(Val(CL_Gravedad.Text) + 0.01)
End Sub

Private Sub SB_SLVinicial_SpinDown()
    If Val(CL_Velocidad_Inicial.Text) - 0.01 < 0 Then Exit Sub
    CL_Velocidad_Inicial.Text = Str(Val(CL_Velocidad_Inicial.Text) - 0.01)
End Sub

Private Sub SB_SLVinicial_SpinUp()
    CL_Velocidad_Inicial.Text = Str(Val(CL_Velocidad_Inicial.Text) + 0.01)
End Sub

'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SB_SPCoef_SpinDown()
    If Val(PI_Rozamiento.Text) - 0.01 < 0 Then Exit Sub
    PI_Rozamiento.Text = Str(Val(PI_Rozamiento.Text) - 0.01)
End Sub

'Scroll de incremento de cantidad en 0.01
Private Sub SB_SPCoef_SpinUp()
    PI_Rozamiento.Text = Str(Val(PI_Rozamiento.Text) + 0.01)
End Sub

'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SB_SPGravedad_SpinDown()
    If Val(PI_Gravedad.Text) - 0.01 < 0 Then Exit Sub
    PI_Gravedad.Text = Str(Val(PI_Gravedad.Text) - 0.01)
End Sub

'Scroll de incremento de cantidad en 0.01
Private Sub SB_SPGravedad_SpinUp()
    PI_Gravedad.Text = Str(Val(PI_Gravedad.Text) + 0.01)
End Sub

```

**'Scroll de decremento de cantidad en 0.01**

**'Si es menor a cero el resultado, queda igual que antes**

```
Private Sub SB_SPInclinacion_SpinDown()  
    If Val(PI_Inclinacion.Text) - 0.01 < 0 Then Exit Sub  
    PI_Inclinacion.Text = Str(Val(PI_Inclinacion.Text) - 0.01)  
End Sub
```

**'Scroll de incremento de cantidad en 0.01**

```
Private Sub SB_SPInclinacion_SpinUp()  
    PI_Inclinacion.Text = Str(Val(PI_Inclinacion.Text) + 0.01)  
End Sub
```

**'Scroll de decremento de cantidad en 0.01**

**'Si es menor a cero el resultado, queda igual que antes**

```
Private Sub SB_SPLongitud_SpinDown()  
    If Val(PI_Longitud.Text) - 0.01 < 0 Then Exit Sub  
    PI_Longitud.Text = Str(Val(PI_Longitud.Text) - 0.01)  
End Sub
```

**'Scroll de incremento de cantidad en 0.01**

```
Private Sub SB_SPLongitud_SpinUp()  
    PI_Longitud.Text = Str(Val(PI_Longitud.Text) + 0.01)  
End Sub
```

**'Scroll de decremento de cantidad en 0.01**

**'Si es menor a cero el resultado, queda igual que antes**

```
Private Sub SB_SPVelocidad_SpinDown()  
    If Val(PI_Velocidad_Inicial.Text) - 0.01 < 0 Then Exit Sub  
    PI_Velocidad_Inicial.Text = Str(Val(PI_Velocidad_Inicial.Text) - 0.01)  
End Sub
```

**'Scroll de incremento de cantidad en 0.01**

```
Private Sub SB_SPVelocidad_SpinUp()  
    PI_Velocidad_Inicial.Text = Str(Val(PI_Velocidad_Inicial.Text) + 0.01)  
End Sub
```

```
Private Sub SB_SSAngulo_SpinDown()
```

```
    If Val(PS_Angulo_Inicial.Text) - 0.01 < 0 Then Exit Sub  
    PS_Angulo_Inicial.Text = Str(Val(PS_Angulo_Inicial.Text) - 0.01)
```

```
End Sub
```

```
Private Sub SB_SSAngulo_SpinUp()
```

```
    PS_Angulo_Inicial.Text = Str(Val(PS_Angulo_Inicial.Text) + 0.01)
```

```
End Sub
```

```
Private Sub SB_SSGravedad_SpinDown()
```

```
    If Val(PS_Gravedad.Text) - 0.01 < 0 Then Exit Sub  
    PS_Gravedad.Text = Str(Val(PS_Gravedad.Text) - 0.01)
```

```
End Sub
```

```
Private Sub SB_SSGravedad_SpinUp()
```

```
    PS_Gravedad.Text = Str(Val(PS_Gravedad.Text) + 0.01)
```

```

End Sub

Private Sub SB_SSLongitud_SpinDown()
    If Val(PS_Longitud.Text) - 0.01 < 0 Then Exit Sub
    PS_Longitud.Text = Str(Val(PS_Longitud.Text) - 0.01)
End Sub

Private Sub SB_SSLongitud_SpinUp()
    PS_Longitud.Text = Str(Val(PS_Longitud.Text) + 0.01)
End Sub

Private Sub SB_SSMasa_SpinDown()
    If Val(PS_Masa.Text) - 0.01 < 0 Then Exit Sub
    PS_Masa.Text = Str(Val(PS_Masa.Text) - 0.01)
End Sub

Private Sub SB_SSMasa_SpinUp()
    PS_Masa.Text = Str(Val(PS_Masa.Text) + 0.01)
End Sub

'Scroll de decremento de cantidad en 0.01
'Si es menor a cero el resultado, queda igual que antes
Private Sub SpinButton1_SpinDown()
    If Val(CL_Altura.Text) - 0.01 < 0 Then Exit Sub
    CL_Altura.Text = Str(Val(CL_Altura.Text) - 0.01)
End Sub

'Scroll de incremento de cantidad en 0.01
Private Sub SpinButton1_SpinUp()
    CL_Altura.Text = Str(Val(CL_Altura.Text) + 0.01)
End Sub

'Verifica si se deben mostrar los gráficos de la simulación,
'el momento que se cambia de tipo de simulación. En caso de
'que se deba mostrar dichos gráficos, se desactiva el anterior
'y se muestra el actual
Private Sub SSTab1_Click(PreviousTab As Integer)
    Select Case SSTab1.Tab
        Case 0:
            Frame1.Height = 2835
            SSTab1.Height = 2800
            Iniciar.Top = 2885
            Detener.Top = 2885
            ToggleButton1.Top = 2885
            Ayuda.Top = 2885
            FSimulacion.Height = 3675
        Case 1:
            Frame1.Height = 3600
            Frame2.Height = 3550
            SSTab1.Height = 3660
            Iniciar.Top = 3665
            Detener.Top = 3665
    End Select
End Sub

```

```

    ToggleButton1.Top = 3665
    Ayuda.Top = 3665
    FSimulacion.Height = 4455
Case 2:
    Frame1.Height = 3075
    SSTab1.Height = 3045
    Iniciar.Top = 3120
    Detener.Top = 3120
    ToggleButton1.Top = 3120
    Ayuda.Top = 3120
    FSimulacion.Height = 3915
End Select
If ToggleButton1.Value = True Then
    Select Case PreviousTab
        Case 0:
            FCaidaLibre.Hide
        Case 1:
            FPlanoInclinado.Hide
        Case 2:
            FPendoloSimple.Hide
    End Select
    Select Case SSTab1.Tab
        Case 0:
            FCaidaLibre.Show
        Case 1:
            FPlanoInclinado.Show
        Case 2:
            FPendoloSimple.Show
    End Select
End If
End Sub

'Este evento es válido para Caída Libre.
'Cada vez que ocurre este evento (cada milisegundo
'como máximo), toma el tiempo inicial, calcula los
'datos a ser mostrados en pantalla y verifica si se
'debe terminar o no la simulación.
Private Sub Timer0_Timer()
    Dim mvarT As Double
    mvarT = Timer - mvarTo
    CL_Tiempo = mvarT
    CL_Distancia = CL_s(mvarG, mvarT, mvarVo)
    CL_Velocidad = CL_Vf(mvarG, mvarT, mvarVo)
    If Val(CL_Distancia) >= Val(CL_Altura) Then
        CL_Tiempo = CL_tt(mvarG, mvarS, mvarVo)
        CL_Velocidad = CL_Vt(mvarG, mvarS, mvarVo)
        CL_Distancia = CL_Altura
        q = 1
    End If
End Sub

```

```

    Detener_Click
End If
Call FCaidaLibre.Actualizar(Val(CL_Distancia))
End Sub

'Este evento es válido para Plano Inclinado.
'Cada vez que ocurre este evento (cada milisegundo
'como máximo), toma el tiempo inicial, calcula los
'datos a ser mostrados en pantalla y verifica si se
'debe terminar o no la simulación.
Private Sub Timer1_Timer()
    Dim mvarT As Double
    mvarT = Timer - mvarTo
    PI_Tiempo = mvarT
    PI_Distancia = PI_s(mvarG, mvarT, mvarI, mvarU, mvarVo)
    PI_Velocidad = PI_Vf(mvarG, mvarT, mvarI, mvarU, mvarVo)
    PI_Aceleracion = PI_At(mvarG, mvarU, mvarI)
    If Val(PI_Distancia) >= Val(PI_Longitud) Then
        PI_Tiempo = PI_tt(mvarG, mvarS, mvarI, mvarU, mvarVo)
        PI_Velocidad = PI_Vt(mvarG, mvarS, mvarI, mvarU, mvarVo)
        PI_Distancia = PI_Longitud
        q = 1
        Detener_Click
    End If
    Call FPlanoInclinado.Actualizar(Val(PI_Distancia))
End Sub

'Este evento es válido para Péndulo Simple.
'Cada vez que ocurre este evento (cada milisegundo
'como máximo), toma el tiempo inicial, calcula los
'datos a ser mostrados en pantalla y verifica si se
'debe terminar o no la simulación.
Private Sub Timer2_Timer()
    Dim mvarT As Double
    mvarT = Timer - mvarTo
    PS_Tiempo = mvarT
    PS_Angulo = PS_A(mvarAo, mvarL, mvarG, mvarT)
    If Val(PS_Periodo.Text) > 0 Then
        PS_Oscilacion.Text = Fix(mvarT / Val(PS_Periodo.Text))
    End If
    Call FPenduloSimple.Actualizar(Val(PS_Angulo))
End Sub

'Presenta u oculta la ventana del gráfico de simulación
'según se presione o no el botón de graficar.
Private Sub ToggleButton1_Click()
    If ToggleButton1.Value Then
        Select Case SSTab1.Tab
            Case 0:

```

```

        FCaidaLibre.Show
    Case 1:
        FPlanoInclinado.Show
    Case 2:
        FPendoloSimple.Show
End Select
Else
    Select Case SSTab1.Tab
        Case 0:
            FCaidaLibre.Hide
        Case 1:
            FPlanoInclinado.Hide
        Case 2:
            FPendoloSimple.Hide
    End Select
End If
End Sub

```

**'Escribe las unidades acorde al sistema con el que se está trabajando**

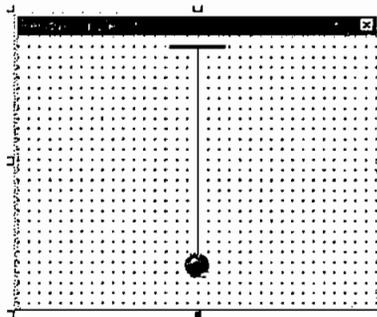
```

Public Sub Establecer_Sistema()
    Dim i As Integer
    If FPrincipal.Unidades(21).Checked Then
        For i = 0 To 4: LDistancia(i).Caption = "(m)": Next i
        For i = 0 To 3: LAceleracion(i).Caption = "(m/s" & Chr(178) & ")": Next i
        For i = 0 To 3: LVelocidad(i).Caption = "(m/s)": Next i
        LMasa.Caption = "(kg)"
    Else
        If FPrincipal.Unidades(22).Checked Then
            For i = 0 To 4: LDistancia(i).Caption = "(cm)": Next i
            For i = 0 To 3: LAceleracion(i).Caption = "(cm/s" & Chr(178) & ")": Next i
            For i = 0 To 3: LVelocidad(i).Caption = "(cm/s)": Next i
            LMasa.Caption = "(gr)"
        Else
            If FPrincipal.Unidades(23).Checked Then
                For i = 0 To 4: LDistancia(i).Caption = "(pies)": Next i
                For i = 0 To 3: LAceleracion(i).Caption = "(pies/s" & Chr(178) & ")":
            Next i
                For i = 0 To 3: LVelocidad(i).Caption = "(pies/s)": Next i
                LMasa.Caption = "(lb)"
            End If
        End If
    End If
End Sub

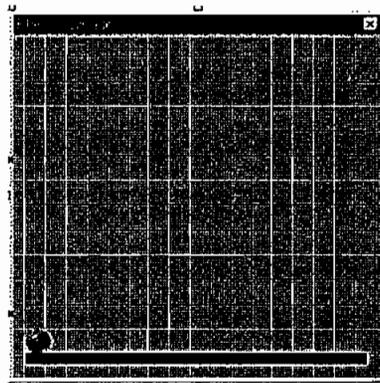
```

```
End If
End Sub
```

## Ventanas Gráficas para la Simulación



FPénduloSimple.frm



PlanoInclinado.frm



CaídaLibre.frm

### FPénduloSimple (code)

Private mvarLong As Double 'Longitud real de la línea que representa la cuerda

'Actualiza la imagen de acuerdo al ángulo en que se encuentra

Public Sub Actualizar(a As Double)

Line1.Y2 = Line1.Y1 + mvarLong \* Cos(a \* 3.1416 / 180)

Line1.X2 = Line1.X1 + mvarLong \* Sin(a \* 3.1416 / 180)

Image1.Top = Line1.Y2 + (320 / 2) \* Cos(a \* 3.1416 / 180) - 315 / 2

Image1.Left = Line1.X2 + (315 / 2) \* Sin(a \* 3.1416 / 180) - 315 / 2

End Sub

'Pone la imagen en su posición inicial

Public Sub Inicializar(a As Double)

mvarLong = 2400

Actualizar (a)

End Sub

'Fija la ventana en la izquierda de la ventana principal.

'Activa el boton de graficar cuando se abre la ventana

Private Sub Form\_Load()

FPenduloSimple.Left = FSimulacion.Left + FSimulacion.Width

FPenduloSimple.Top = FSimulacion.Top

FSimulacion.ToggleButton1.Value = True

End Sub

```
'Desactiva el botón de graficar cuando se cierra la ventana
Private Sub Form_Unload(Cancel As Integer)
    FSimulacion.ToggleButton1.Value = False
End Sub
```

### FPlanoInclinado (code)

```
Private mvarTop As Double 'Punto superior del objeto de experimento
Private mvarLeft As Double 'distancia del margen izquierdo del objeto del
experimento
Private mvarH As Double 'Longitud ingresada del plano
Private mvarI As Double 'Inclinación del plano
Private mvarLong As Double 'Longitud del plano mostrado en pantalla.
```

```
'Actualiza la posición del objeto, de acuerdo al espacio
'que éste haya recorrido.
'En caso de que la longitud del plano por el que cae
'sea igual a 0, no se realiza la actualización, para evitar
'la división para 0.
```

```
Public Sub Actualizar(s As Double)
    If mvarH <> 0 Then
        Image1.Top = mvarTop + (s * mvarLong / mvarH) * Sin(mvarI * 3.1416 /
180)
        Image1.Left = mvarLeft + (s * mvarLong / mvarH) * Cos(mvarI * 3.1416 /
180)
    End If
End Sub
```

```
'Inicializa las variables, de acuerdo a la longitud del plano
'por el que cae y la inclinación del mismo. Se fija la inclinación
'del plano mostrado en pantalla y se ubica el objeto al inicio de
'éste. Se fija el valor de la variable que guarda la longitud (en pantalla)
'que debe recorrer el objeto.
```

```
Public Sub Inicializar(h As Double, i As Double)
    mvarLong = 240
    mvarH = h
    mvarI = i
    Line1.X1 = 8 + mvarLong - mvarLong * Cos(i * 3.1416 / 180)
    Line1.Y1 = 8 + mvarLong - mvarLong * Sin(i * 3.1416 / 180)
    Image1.Top = Line1.Y1 - 21 * Cos(i * 3.1416 / 180)
    Image1.Left = Line1.X1
    mvarTop = Image1.Top
    mvarLeft = Image1.Left
    If mvarI <> 0 Then
```

```

        mvarLong = mvarLong - (Image1.Top + 21 - Line1.Y1) / Sin(mvarL * 3.1416
        / 180)
    End If
End Sub

```

**'Fija la ventana a la izquierda de la ventana principal.**

**'Activa el botón de graficar cuando se abre la ventana**

```

Private Sub Form_Load()
    FPlanoInclinado.Left = FSimulacion.Left + FSimulacion.Width
    FPlanoInclinado.Top = FSimulacion.Top
    FSimulacion.ToggleButton1.Value = True
End Sub

```

**'Desactiva el botón de graficar cuando se cierra la ventana**

```

Private Sub Form_Unload(Cancel As Integer)
    FSimulacion.ToggleButton1.Value = False
End Sub

```

## FCaídaLibre (code)

```

Private mvarH As Double 'Variable de altura del objeto
' Private p As Integer

```

**'Actualiza la posición del objeto, de acuerdo al espacio  
'que este haya recorrido.**

**'En caso de que la altura por la que cae el objeto sea  
'igual a 0, no se realiza la actualización, para evitar  
'la división para 0.**

```

Public Sub Actualizar(s As Double)
    If mvarH <> 0 Then
        Image1.Top = 120 + s * 4080 / mvarH
    End If
End Sub

```

**'Inicializa las variables, de acuerdo a la altura  
'por el que cae el objeto.**

```

Public Sub Inicializar(h As Double)
    mvarH = h
    Image1.Top = 120
    For a = 0 To 16
        Marca(a) = mvarH - a * mvarH / 17
    Next a

```

```
Frame1.Visible = True  
End Sub
```

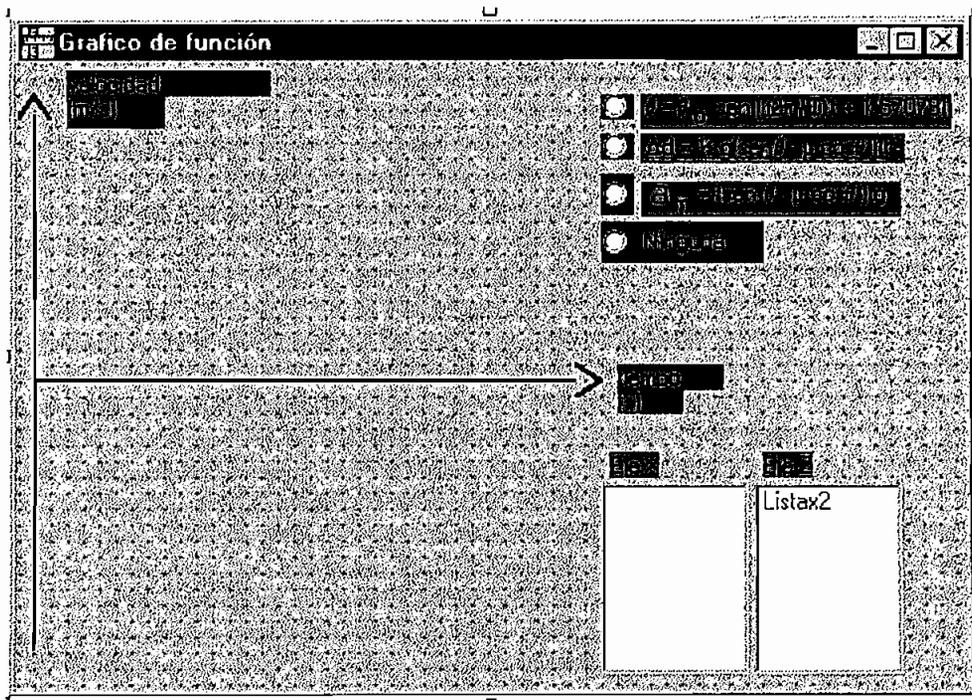
**'Fija la ventana a la izquierda de la ventana principal.  
'Activa el botón de graficar cuando se abre la ventana**

```
Private Sub Form_Load()  
    FCaidaLibre.Left = FSimulacion.Left + FSimulacion.Width  
    FCaidaLibre.Top = FSimulacion.Top  
    FSimulacion.ToggleButton1.Value = True  
End Sub
```

**'Desactiva el botón de graficar cuando se cierra la ventana**

```
Private Sub Form_Unload(Cancel As Integer)  
    FSimulacion.ToggleButton1.Value = False  
End Sub
```

### FGraf\_F.frm



### FGraf\_F (code)

```
Dim i As Double  
Dim m, j As Integer  
Dim ex1 As Double
```

```

Dim ey1 As Double
Dim ey2 As Double
Dim ex2 As Double

Private Function punto(X As Double, Y As Double)
    Line (X, Y)-(X + 15, Y + 15), , BF
End Function

Private Function fx(X As Double) As Double
    fx = Sin(X)
End Function

Private Function linea(X As Integer)
    If Option1.Value = True Then
        ForeColor = QBColor(5)
        Line ((120 + Val(Listax.List(X)) * ex1), (2400 - Val(Listay1.List(X)) * ey1))-
            ((120 + Val(Listax.List(X + 1)) * ex1), (2400 - Val(Listay1.List(X + 1)) *
            ey1))
        m = punto((120 + Val(Listax.List(X + 1)) * ex1), (2400 - Val(Listay1.List(X
            + 1)) * ey1))
    End If
    If Option2.Value = True Then
        ForeColor = QBColor(1)
        Line ((120 + Val(Listax.List(X)) * ex1), (2400 - Val(Listay2.List(X)) * ey2))-
            ((120 + Val(Listax.List(X + 1)) * ex1), (2400 - Val(Listay2.List(X + 1)) *
            ey2))
        m = punto((120 + Val(Listax.List(X + 1)) * ex1), (2400 - Val(Listay2.List(X
            + 1)) * ey2))
    End If
    If Option3.Value = True Then
        ForeColor = QBColor(14)
        If j = 0 Then
            Line ((120 + Val(Listay2.List(X)) * ex2), (2400 - Val(Listay1.List(X)) *
            ey1))-((120 + Val(Listay2.List(X + 1)) * ex2), (2400 - Val(Listay1.List(X +
            1)) * ey1))
            m = punto((120 + Val(Listay2.List(X + 1)) * ex2), (2400 -
            Val(Listay1.List(X + 1)) * ey1))
        Else
            Line ((120 + Val(Listax.List(X)) * ex1), (2400 - Val(Listax2.List(X)) *
            ex2))-((120 + Val(Listax.List(X + 1)) * ex1), (2400 - Val(Listax2.List(X +
            1)) * ex2))
            m = punto((120 + Val(Listax.List(X + 1)) * ex1), (2400 - Val(Listax2.List(X
            + 1)) * ex2))
        End If
    End If
End Function

Private Sub Form_Load()
    BackColor = QBColor(3)
    ForeColor = QBColor(4)

```

```

FGrafico_Funcion.Height = 5115
FGrafico_Funcion.Width = 7100
Option1.Value = False
Option2.Value = False
Option3.Value = False
Option4.Value = True
For m = 0 To 2
    Image1(m).Visible = False
    Image2(m).Visible = False
Next m
Image3(0).Visible = False
j = Val(FSimulacion.SSTab1.Tab)
If j = 0 Then
    For m = 0 To 2
        Image1(m).Visible = True
    Next m
End If
If j = 1 Then
    For m = 0 To 2
        Image2(m).Visible = True
    Next m
End If
If j = 2 Then
    Image3(0).Visible = True
    Option2.Visible = False
    Option3.Visible = False
End If
End Sub

'Private Sub Option1_Click()
Private Sub Option1_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    'cálculo de escalas
ex1 = 4000 / IIf(Val(Listax.List(17)) <> 0, Val(Listax.List(17)), 1)
ex2 = 4000 / IIf(Val(Listay2.List(17)) <> 0, Val(Listay2.List(17)), 1)
ey1 = 2000 / IIf(Val(Listay1.List(17)) <> 0, Val(Listay1.List(17)), 1)
ey2 = 2000 / IIf(Val(Listay2.List(17)) <> 0, Val(Listay2.List(17)), 1)
    ' cambio las etiquetas de ejes
If Option1.Value = True Then
    ' Caída libre y Plano Inclinado
    ejex.Caption = "tiempo"
    ejoy.Caption = "velocidad"
    Listax.Visible = True
    Listax2.Visible = False
    Listay1.Visible = True
    Listay2.Visible = False
    Lejex.Caption = "tiempo"

```

```

Lejey1.Caption = "velocidad"
If j = 2 Then
  ' Péndulo Simple
  ejex.Caption = "tiempo"
  ejey.Caption = "ángulo"
  Lejex.Caption = "tiempo"
  Lejey1.Caption = "ángulo"
End If

Lejex.Visible = True
Lejey1.Visible = True
Lejey2.Visible = False

If FPrincipal.Unidades(21).Checked = True Then
  u_ejex.Caption = "(s)"
  u_ejey.Caption = "(m/s)"
  If j = 2 Then
    u_ejey.Caption = "(grados)"
  End If
End If

If FPrincipal.Unidades(22).Checked = True Then
  u_ejex.Caption = "(s)"
  u_ejey.Caption = "(cm/s)"
  If j = 2 Then
    u_ejey.Caption = "(grados)"
  End If
End If

If FPrincipal.Unidades(23).Checked = True Then
  u_ejex.Caption = "(s)"
  u_ejey.Caption = "(pies/s)"
  If j = 2 Then
    u_ejey.Caption = "(grados)"
  End If
End If

End If
FGráfico_Funcion.Refresh
For i = 0 To 16
  línea (i)
Next i

End Sub

'Private Sub Option2_Click()
Private Sub Option2_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
  ex1 = 4000 / IIf(Val(Listax.List(17)) <> 0, Val(Listax.List(17)), 1)
  ex2 = 4000 / IIf(Val(Listay2.List(17)) <> 0, Val(Listay2.List(17)), 1)
  ey1 = 2000 / IIf(Val(Listay1.List(17)) <> 0, Val(Listay1.List(17)), 1)

```

```

ey2 = 2000 / IIf(Val(Listay2.List(17)) <> 0, Val(Listay2.List(17)), 1)
If Option2.Value = True Then
    Listay2.Left = 5460
    Lejey2.Left = 5520
    ejex.Caption = "tiempo"
    ejey.Caption = "distancia"
    Listax.Visible = True
    Listax2.Visible = False
    Listay1.Visible = False
    Listay2.Visible = True
    Lejex.Caption = "tiempo"
    Lejey2.Caption = "distancia"
    Lejex.Visible = True
    Lejey1.Visible = False
    Lejey2.Visible = True

    If FPrincipal.Unidades(21).Checked = True Then
        u_ejex.Caption = "(s)"
        u_ejey.Caption = "(m)"
    End If
    If FPrincipal.Unidades(22).Checked = True Then
        u_ejex.Caption = "(s)"
        u_ejey.Caption = "(cm)"
    End If
    If FPrincipal.Unidades(23).Checked = True Then
        u_ejex.Caption = "(s)"
        u_ejey.Caption = "(pies)"
    End If
End If
FGrafico_Funcion.Refresh
For i = 0 To 16
    linea (i)
Next i
End Sub

'Private Sub Option3_Click()
Private Sub Option3_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    ex1 = 4000 / IIf(Val(Listax.List(17)) <> 0, Val(Listax.List(17)), 1)
    ex2 = 4000 / IIf(Val(Listay2.List(17)) <> 0, Val(Listay2.List(17)), 1)
    ey1 = 2000 / IIf(Val(Listay1.List(17)) <> 0, Val(Listay1.List(17)), 1)
    ey2 = 2000 / IIf(Val(Listay2.List(17)) <> 0, Val(Listay2.List(17)), 1)
    If Option3.Value = True Then
        'mueve lista de datos para el grafico
        Listay2.Left = 4320
        Lejey2.Left = 4380
        ejex.Caption = "distancia"
        ejey.Caption = "velocidad"
    End If
End Sub

```



```

Listax.Visible = False
Listax2.Visible = False
Listay1.Visible = True
Listay2.Visible = True
Lejey1.Caption = "velocidad"
Lejey2.Caption = "distancia"

Lejex.Visible = False
Lejey1.Visible = True
Lejey2.Visible = True
If j = 1 Then
    Lejey2.Left = 5520
    ex2 = 1800 / IIf(Val(Listax2.List(17)) <> 0, Val(Listax2.List(17)), 1)
    ejex.Caption = "tiempo"
    ejey.Caption = "aceleración tan."
    Lejex.Caption = "tiempo"
    Lejey2.Caption = "ace.tan."
    Listax.Visible = True
    Listax2.Visible = True
    Listay1.Visible = False
    Listay2.Visible = False
    Lejex.Visible = True
    Lejey1.Visible = False
    Lejey2.Visible = True
End If

If FPrincipal.Unidades(21).Checked = True Then
    u_ejex.Caption = "(m)"
    u_ejey.Caption = "(m/s)"
    If j = 1 Then
        u_ejex.Caption = "(s)"
        u_ejey.Caption = "(m/s²)"
    End If
End If

If FPrincipal.Unidades(22).Checked = True Then
    u_ejex.Caption = "(cm)"
    u_ejey.Caption = "(cm/s)"
    If j = 1 Then
        u_ejex.Caption = "(s)"
        u_ejey.Caption = "(cm/s²)"
    End If
End If

If FPrincipal.Unidades(23).Checked = True Then
    u_ejex.Caption = "(pies)"
    u_ejey.Caption = "(pies/s)"
    If j = 1 Then

```

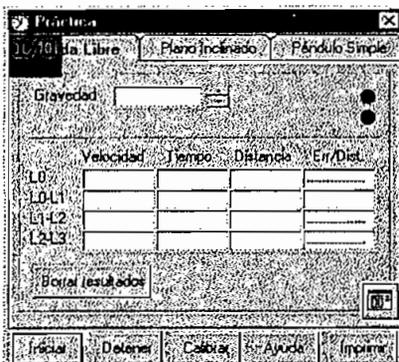
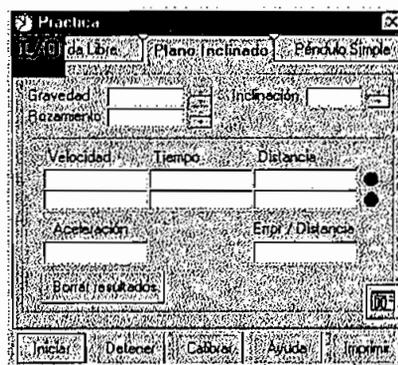
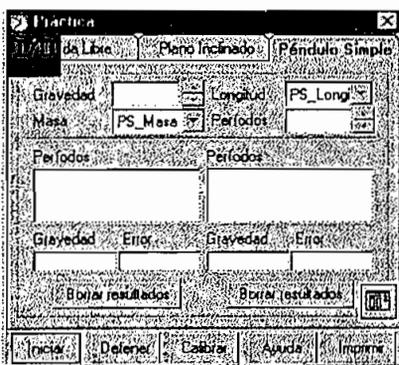


```

    u_ejex.Caption = "(s)"
    u_ejey.Caption = "(pies/s2)"
End If
End If
End If
FGráfico_Funcion.Refresh
For i = 0 To 16
    línea (i)
Next i
End Sub
Private Sub Option4_Click()
    Listax.Visible = False
    Listay1.Visible = False
    Listay2.Visible = False
    Lejex.Visible = False
    Lejey1.Visible = False
    Lejey2.Visible = False
    FGráfico_Funcion.Refresh
End Sub

```

## Fpráctica.frm



## Fpráctica (code)

Private mvar0 As Double 'Tiempo desde la posición inicial hasta que llega al primer sensor



```

Private mvarT1 As Double 'Tiempo desde que pasa el primer sensor hasta que
llega al segundo
Private mvarT2 As Double 'Tiempo desde que pasa el segundo sensor hasta que
llega al final del recorrido
Const mvarV0 As Double = 0 'Velocidad en la posición inicial
Private mvarV1 As Double 'Velocidad en el momento que llega al primer
sensor
Private mvarV2 As Double 'Velocidad en el momento que llega al segundo
sensor
Private mvarV3 As Double 'Velocidad al final del recorrido
Private mvarS0 As Double 'Espacio entre la posición inicial y el primer sensor
Private mvarA1 As Double 'Aceleración lineal
Private mvarE1 As Double 'Error calculado
Const mvarS1 As Double = 0.365 'Espacio entre el primer y segundo sensor
Const mvarS2 As Double = 0.1 'Espacio entre el segundo sensor y el final del
recorrido
Const mvarSL1 As Double = 0 'Espacio entre el primer y segundo sensor
Const mvarSL2 As Double = 0.4 'Espacio entre el segundo sensor y el final del
recorrido
Private mvarG As Double 'Gravedad
Private mvarU As Double 'Rozamiento
Private mvarI As Double 'Inclinación
Private mvarN As Double 'Número de oscilaciones
Private mvarT As Double 'Sumatoria de períodos
Private mvarTp As Double 'Promedio de períodos
Private mvarL As Double 'longitud de la cuerda
Private tempN As Integer 'variable temporal
Private tempI As Integer 'variable temporal 2
Private mvarIniciado As Boolean 'Indica si se tomó el primer tiempo de la
simulación
Private mvarEstado_Anterior As Boolean 'Indica el estado anterior del pin
Dim io2_valor As Integer
Dim fin As Boolean
Dim h_id As String ' el código de página del help
Public numcopies As Integer

Private Sub Ayuda_Click()
    If SSTab2.Tab = 0 Then
        h_id = "caida_libre"
    End If
    If SSTab2.Tab = 1 Then
        h_id = "plano_inclinado"
    End If
    If SSTab2.Tab = 2 Then
        h_id = "pendulo_simple"
    End If
    fp_impresora.HelpFile = ".\tesis.hlp"

```



```

    fp_impresora.HelpKey = h_id
    fp_impresora.HelpCommand = cdlHelpForceFile
    fp_impresora.ShowHelp
End Sub

Private Sub Borra01_Click()
    CL_Velocidad(0).Text = " "
    CL_Velocidad(1).Text = " "
    CL_Velocidad(2).Text = " "
    CL_Velocidad(3).Text = " "
    CL_Tiempo(0).Text = " "
    CL_Tiempo(1).Text = " "
    CL_Tiempo(2).Text = " "
    CL_Tiempo(3).Text = " "
    CL_Distancia(0).Text = " "
    CL_Distancia(1).Text = " "
    CL_Distancia(2).Text = " "
    CL_Distancia(3).Text = " "
    CL_Error(1).Text = " "
End Sub

Private Sub Borra02_Click()
    PI_Velocidad(0).Text = " "
    PI_Velocidad(1).Text = " "
    PI_Tiempo(0).Text = " "
    PI_Tiempo(1).Text = " "
    PI_Distancia(0).Text = " "
    PI_Distancia(1).Text = " "
    PI_CAceleracion.Text = " "
    PI_CError.Text = " "
End Sub

Private Sub Calibrar_Click()
    FCalibrar.Show
End Sub

Private Sub CL_SGravedad_SpinUp()
    CL_Gravedad.Text = Val(CL_Gravedad.Text) + 0.01
End Sub

Private Sub CL_SGravedad_SpinDown()
    If (Val(CL_Gravedad.Text) > 0) Then
        CL_Gravedad.Text = Val(CL_Gravedad.Text) - 0.01
    End If
End Sub

Private Sub Form_Load()
    fp_impresora.Flags = &H8

```



```

End Sub

Private Sub imprimir1_Click()
    FP_Imprimir.Show
End Sub

Private Sub Iniciar_Click()
    Fondo.Enabled = False
    Iniciar.Enabled = False
    Detener.Enabled = True
    Calibrar.Enabled = False
    Shape1.FillColor = &H80&
    Shape2.FillColor = &H80&
    Shape3.FillColor = &H80&
    Shape4.FillColor = &H80&
    Select Case SSTab2.Tab
        Case 0:
            MsgBox "Por Favor, ubique el móvil sobre el nivel superior del prototipo
                y presione Aceptar", 64, "Aviso"
            If Val(CL_Gravedad) <= 0 Then mvarG = 1
            mvarG = Val(CL_Gravedad)
            Timer1.Enabled = True
        Case 1:
            MsgBox "Por Favor, ubique el móvil sobre el sensor y presione Aceptar",
                64, "Aviso"
            mvarG = Val(PI_Gravedad)
            mvarU = Val(PI_Rozamiento)
            mvarI = Val(PI_Inclinacion)
            mvarI1 = 0
            Timer3.Enabled = True
        Case 2:
            MsgBox "Por Favor, ubique la masa pendular sobre el sensor y presione
                Aceptar", 64, "Aviso"
            mvarT = 0
            mvarL = 0
            mvarN = 0
            tempI = 0
            mvarIniciado = False
            mvarEstado_Anterior = False
            mvarG = Val(PS_Gravedad)
            If PS_Longitud.Text = "Pequeña" Then
                Timer6.Enabled = True
            Else
                Timer5.Enabled = True
            End If
        End Select
    End Select
End Sub

```

```

Private Sub Detener_Click()
    Select Case SSTab2.Tab
        Case 0:
            fin = True
            Timer1.Enabled = False
            Timer2.Enabled = False
            'Shape1.FillColor = &H80&
            'Shape2.FillColor = &H80&
        Case 1:
            Timer3.Enabled = False
            Timer4.Enabled = False
            'Shape3.FillColor = &H80&
            'Shape4.FillColor = &H80&
        Case 2:
            Timer5.Enabled = False
            Timer6.Enabled = False
            Mostrar_Resultados
    End Select
    Iniciar.Enabled = True
    Detener.Enabled = False
    Calibrar.Enabled = True
    Fondo.Enabled = True
End Sub

Private Sub PI_SGravedad_SpinUp()
    PI_Gravedad.Text = Val(PI_Gravedad.Text) + 0.01
End Sub

Private Sub PI_SGravedad_SpinDown()
    If (Val(PI_Gravedad.Text) > 0) Then
        PI_Gravedad.Text = Val(PI_Gravedad.Text) - 0.01
    End If
End Sub

Private Sub PI_SInclinacion_SpinUp()
    PI_Inclinacion.Text = Val(PI_Inclinacion.Text) + 0.01
End Sub

Private Sub PI_SInclinacion_SpinDown()
    If (Val(PI_Inclinacion.Text) > 0) Then
        PI_Inclinacion.Text = Val(PI_Inclinacion.Text) - 0.01
    End If
End Sub

Private Sub PI_SRozamiento_SpinUp()
    PI_Rozamiento.Text = Val(PI_Rozamiento.Text) + 0.01
End Sub

Private Sub PI_SRozamiento_SpinDown()

```

```

If (Val(PI_Rozamiento.Text) > 0) Then
    PI_Rozamiento.Text = Val(PI_Rozamiento.Text) - 0.01
End If
End Sub

Private Sub PS_Borrar_Click(Index As Integer)
    Do While PS_Periodos(Index).ListCount > 0
        PS_Periodos(Index).RemoveItem 0
    Loop
End Sub

Private Sub PS_SGravedad_SpinDown()
    If (Val(PS_Gravedad.Text) > 0) Then
        PS_Gravedad.Text = Val(PS_Gravedad.Text) - 0.01
    End If
End Sub

Private Sub PS_SGravedad_SpinUp()
    PS_Gravedad.Text = Val(PS_Gravedad.Text) + 0.01
End Sub

Private Sub PS_SNumero_de_Periodos_SpinDown()
    If (Val(PS_Numero_de_Periodos.Text) > 0) Then
        PS_Numero_de_Periodos.Text = Val(PS_Numero_de_Periodos.Text) - 1
    End If
End Sub

Private Sub PS_SNumero_de_Periodos_SpinUp()
    PS_Numero_de_Periodos.Text = Val(PS_Numero_de_Periodos.Text) + 1
End Sub

Private Sub SSTab2_Click(PreviousTab As Integer)
    Select Case SSTab2.Tab
        Case 0:
            SSTab2.Height = 3435
            Fondo.Height = 3435
            Iniciar.Top = 3480
            Detener.Top = 3480
            Calibrar.Top = 3480
            Ayuda.Top = 3480
            FPractica.Height = 4260
        Case 1:
            SSTab2.Height = 3435
            Fondo.Height = 3435
            Iniciar.Top = 3480
            Detener.Top = 3480
            Calibrar.Top = 3480
            Ayuda.Top = 3480
    End Select
End Sub

```

```

        FPractica.Height = 4260
    Case 2:
        SSTab2.Height = 3435
        Fondo.Height = 3435
        Iniciar.Top = 3480
        Detener.Top = 3480
        Calibrar.Top = 3480
        Ayuda.Top = 3480
        FPractica.Height = 4260
    End Select
End Sub

Private Sub Timer1_Timer()
    If Busy Then
        Shape1.FillColor = &H8000& 'verde
        mvar1 = Timer
        Timer1.Enabled = False
        Timer2.Enabled = True
    End If
End Sub

Private Sub Timer2_Timer()
    fin = False
    Do
        If Val(Ioport1.bitsize) = 8 Then
            io2_valor = Val(Ioport1.Value)
            If Not (io2_valor And 32) = 32 Then
                Shape2.FillColor = &H8000&
                mvar1 = Timer - mvar1
                Detener_Click
                Mostrar_Resultados
                Ioport1.address = 889
                Ioport1.bitsize = 8
                fin = True
            End If
        End If
    Loop Until fin = True
End Sub

Private Sub Timer3_Timer()
    If Not Busy Then
        Shape3.FillColor = &H8000&
        tempI = 1
        Timer3.Enabled = False
        Timer4.Enabled = True
    End If
End Sub

```

```

Private Sub Timer4_Timer()
    If tempI = 1 Then
        mvar1 = Timer
        tempI = 0
    End If
    If PageEnd Then
        Shape4.FillColor = &H8000&
        mvar1 = Timer - mvar1
        Detener_Click
        Mostrar_Resultados
    End If
End Sub

```

```

Private Sub Timer5_Timer()
    If Busy Then
        mvar1 = Timer
        If Not mvarEstado_Anterior And mvarIniciado Then
            PS_Periodos(IIf(PS_Masa.Text = "Pequeña", 1, 0)).AddItem (mvar1 -
            mvar0)
            mvarN = mvarN + 1
            If mvarN = Val(PS_Numero_de_Periodos.Text) Then Detener_Click
        End If
        mvarIniciado = True
        mvarEstado_Anterior = True
    Else
        If mvarEstado_Anterior Then
            mvar0 = mvar1
            mvarEstado_Anterior = False
        End If
    End If
End Sub

```

```

Private Sub Timer6_Timer()
    If Busy Then
        mvar1 = Timer
        If Not mvarEstado_Anterior And mvarIniciado Then
            PS_Periodos(IIf(PS_Masa.Text = "Pequeña", 1, 0)).AddItem (mvar1 -
            mvar0)
            'mvarT = mvarT + (mvar1 - mvar0)
            mvarN = mvarN + 1
            If mvarN = Val(PS_Numero_de_Periodos.Text) Then Detener_Click
        End If
        mvarIniciado = True
        mvarEstado_Anterior = True
    Else
        If mvarEstado_Anterior Then
            mvar0 = mvar1

```



```

        mvarEstado_Anterior = False
    End If
End If
End Sub
Private Sub Mostrar_Resultados()
    Select Case SSTab2.Tab
    Case 0:
        mvarV1 = CL_V1(mvarS1, mvarG, mvarT1)
        mvarT0 = CL_T0(mvarV1, mvarG)
        mvarS0 = CL_S0(mvarV1, mvarG)
        mvarV2 = CL_V2(mvarV1, mvarG, mvarT1)
        mvarV3 = CL_V3(mvarS2, mvarG, mvarV2)
        mvarT2 = CL_T2(mvarV2, mvarV3, mvarG)
        mvarE1 = err(mvarS0, mvarS2)
        CL_Velocidad(0) = mvarV0
        CL_Velocidad(1) = mvarV1
        CL_Velocidad(2) = mvarV2
        CL_Velocidad(3) = mvarV3
        CL_Tiempo(0) = 0
        CL_Tiempo(1) = mvarT0
        CL_Tiempo(2) = mvarT1 'mvarT0 + mvarT1
        CL_Tiempo(3) = mvarT2 ' mvarT0 + mvarT1 + mvarT2
        CL_Distancia(0) = 0
        CL_Distancia(1) = mvarS0
        CL_Distancia(2) = mvarS1 'mvarS0 + mvarS1
        CL_Distancia(3) = mvarS2 'mvarS0 + mvarS1 + mvarS2
        CL_Error(1) = mvarE1
    Case 1:
        tempI = 0
        ' cálculos
        mvarA1 = PI_At(mvarG, mvarU, mvarI)
        mvarV1 = PI_V1(mvarG, mvarT1, mvarU, mvarI)
        mvarS0 = PI_S0(mvarG, mvarT1, mvarU, mvarI)
        If FPrincipal.Unidades(21).Checked Then
            mvarL = mvarSL2
        End If
        If FPrincipal.Unidades(22).Checked Then
            mvarL = mvarSL2 * 100
        End If

        If FPrincipal.Unidades(23).Checked Then
            mvarL = mvarSL2 / 0.3048
        End If
        mvarE1 = err(mvarS0, mvarL)

        If (mvarV1 < 0) Or (mvarS0 < 0) Or (mvarA1 < 0) Then
            tempI = 1
        End If
    End Select
End Sub

```



```

End If

' despliegue de resultados
PI_Velocidad(0) = mvarV0
PI_Velocidad(1) = mvarV1
PI_Tiempo(0) = 0
PI_Tiempo(1) = mvarT1
PI_Distancia(0) = 0
PI_Distancia(1) = mvarS0
PI_CAceleracion.Text = mvarA1
PI_CError.Text = mvarE1

If tempI = 1 Then
    MsgBox "Datos Erroneos!! Por Favor, verifique que la medición del
    ángulo de inclinación y/o el coeficiente de rozamiento sean los
    adecuados", 64, "OK"
End If
Case 2:
tempN = IIf(PS_Masa.Text = "Pequeña", 1, 0)
tempI = 0

' suma de los resultados de los períodos de las oscilaciones
Do
    mvarT = mvarT + Val(PS_Periodos(tempN).List(tempI))
    tempI = tempI + 1
Loop Until tempI = mvarN
mvarTp = Tp(mvarT, mvarN)
mvarL = IIf(PS_Longitud.Text = "Pequeña", IIf(PS_Masa.Text =
"Pequeña", 0.159, 0.162), IIf(PS_Masa.Text = "Pequeña", 0.41, 0.415))

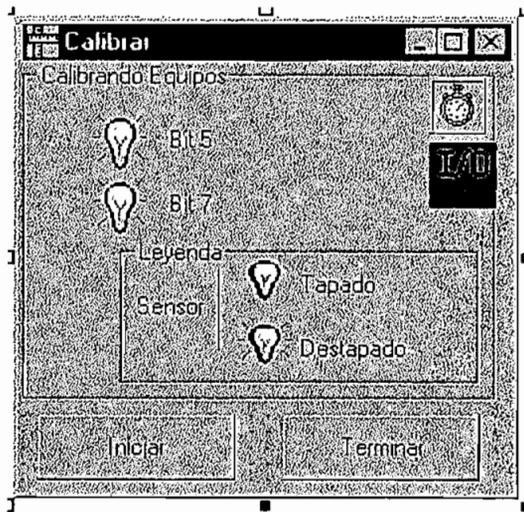
' verificación de unidades para la longitud
If FPrincipal.Unidades(21).Checked Then
    mvarL = mvarL
End If
If FPrincipal.Unidades(22).Checked Then
    mvarL = mvarL * 100
End If

If FPrincipal.Unidades(23).Checked Then
    mvarL = mvarL / 0.3048
End If

' presentación de los resultados de Gravedad y error
PS_Gravedad_calculada(tempN).Text = Val(PS_G(mvarL, mvarTp))
PS_Error(tempN).Text = err(Val(PS_Gravedad_calculada(tempN).Text),
Val(PS_Gravedad.Text))
End Select
End Sub

```

## Calibrar.frm



### FCalibrar (code)

Dim valor As Integer ' guarda el valor obtenido del puerto

' Al llamar al formulario, inicializa las variables  
' del puerto Lpt, para recibir los datos.

```
Private Sub Form_Load()  
    TimerC.Enabled = False  
    IoportC.address = 889  
    IoportC.bitsize = 8
```

' **Tamaño y posición de la ventana Calibrar**

```
frame.Height = 2535  
frame.Width = 3495  
FCalibrar.Top = Val(FPractica.Top)  
FCalibrar.Left = Val(FPractica.Left) + Val(FPractica.Width)  
FCalibrar.Width = 3720  
FCalibrar.Height = 3615  
TimerC.Enabled = True  
End Sub
```

' **Inicio de verificación del puerto**

```
Private Sub Iniciar_Click()  
    TimerC.Enabled = True  
End Sub
```

```

' Finaliza la verificación del puerto
' y esconde la ventana Calibrar

Private Sub Terminar_Click()
    TimerC.Enabled = False
    FCalibrar.Hide
End Sub

' Lee el puerto y verifica si corresponde
' al bit 5 o 7, para presentar los focos
' encendidos o apagados

Private Sub TimerC_Timer()
    If Val(IoportC.bitsize) = 8 Then
        valor = Val(IoportC.Value)

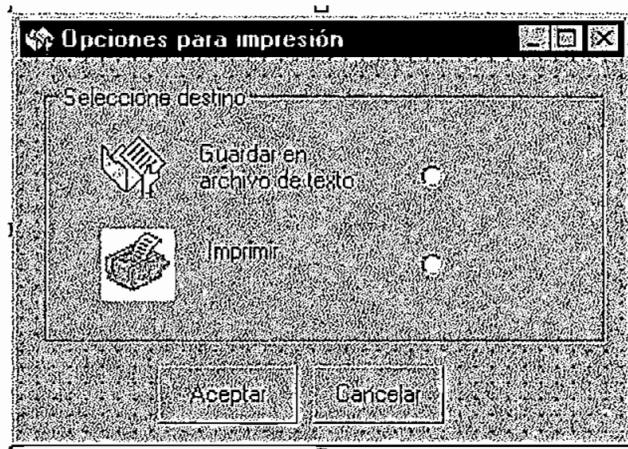
        If FPractica.SSTab2.Tab = 2 Then
            Image2(0).Visible = False
            Image1(0).Visible = False
            Label1.Visible = False
        Else
            Image2(0).Visible = True
            Image1(0).Visible = True
            Label1.Visible = True
            If (valor And 32) = 32 Then
                Image2(0).Visible = True 'Encendido
            Else
                Image2(0).Visible = False 'Apagado
            End If
        End If
        If (valor And 128) = 128 Then
            Image2(1).Visible = False 'Encendido
        Else
            Image2(1).Visible = True 'Apagado
        End If

    End If
    IoportC.address = 889
    IoportC.bitsize = 8

End Sub

```

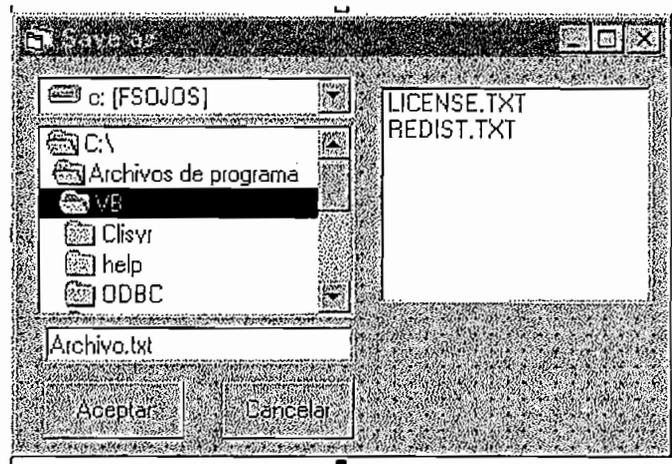
## FP\_Imprimir.frm



## FP\_Imprimir (code)

```
Private Sub Command1_Click()  
    If Option1.Value = True Then  
        FDiscos.Show  
    End If  
  
    If Option2.Value = True Then  
        imprimiresto (Val(FPractica.SSTab2.Tab))  
    End If  
    FP_Imprimir.Hide  
End Sub  
  
Private Sub Command2_Click()  
    FP_Imprimir.Visible = False  
End Sub  
  
Private Sub Form_Load()  
    FP_Imprimir.Top = FPractica.Top  
    FP_Imprimir.Left = FPractica.Left + FPractica.Width  
    FP_Imprimir.Width = 4590  
    FP_Imprimir.Height = 3225  
    Option1.Enabled = True  
    Option2.Enabled = True  
End Sub
```

## FDiscos.frm



## FDiscos (code)

Dim path1 As String '**Path en donde se va a guardar el archivo**

Dim archivo1 As String '**Nombre del archivo**

Dim x As Integer '**variable auxiliar**

' **Cancela el proceso de grabación y esconde**

' **la ventana de discos**

Private Sub Cancelar\_Click()

    FDiscos.Visible = False

End Sub

Private Sub Aceptar\_Click()

    path1 = FpFile1.Path

    archivo1 = path1 & "\" & Text1.Text

    '**llamada a la grabación**

    x = guardaresto(Val(FPractica.SSTab2.Tab), archivo1)

    FDiscos.Visible = False

    Dir1.Refresh

    FpFile1.Refresh

End Sub

Private Sub Dir1\_Change()

    FpFile1.Path = Dir1.Path

    FpFile1.Refresh

End Sub

Private Sub Drive1\_Change()

    Dir1.Path = Drive1.Drive



```

    FpFile1.Path = Dir1.Path
    Dir1.Refresh
    FpFile1.Refresh
End Sub

Private Sub Form_Load()
    FDiscos.Height = 3315
    FDiscos.Top = 0
    FDiscos.Width = 4890
End Sub

Private Sub FpFile1_Click()
    Text1.Text = FpFile1.filename
End Sub

```

### 3.3 FUNCIONES PUBLICAS UTILIZADAS EN EL PROGRAMA.

**EFC v1.0** utiliza módulos en donde se incluyen las funciones de uso general de todo el programa. Estos módulos almacenan los segmentos de código, a los cuales se accede cuando se llama a una función específica. Las funciones tienen característica de ser generales, es decir, son accedidas desde diferentes lugares del programa y arrojan resultados en función de los datos de entrada. En **EFC** se plantean tres grupos de funciones básicas, las mismas que se detallan con sus respectivos segmentos de código:

#### **Funciones\_Simulación (code)**

```

'Velocidad final transcurrido un tiempo en Caída Libre
Public Function CL_Vf(g As Double, t As Double, Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    CL_Vf = Vo + g * t
End Function

'Espacio recorrido transcurrido un tiempo en Caída Libre
Public Function CL_s(g As Double, t As Double, Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    CL_s = Vo * t + 0.5 * g * t ^ 2
End Function

```



**'Velocidad al final del recorrido en Caída Libre**

```
Public Function CL_Vt(g As Double, s As Double, Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    CL_Vt = (2 * g * s + Vo ^ 2) ^ 0.5
End Function
```

**'Tiempo al final del recorrido en Caída Libre**

**'En caso de que la gravedad=0 el tiempo retornado es 0 y  
'no se realiza el cálculo por medio de la ecuación para  
'evitar la división para 0.**

```
Public Function CL_tt(g As Double, s As Double, Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    If g = 0 Then
        CL_tt = 0
    Else
        CL_tt = (CL_Vt(g, s, Vo) - Vo) / g
    End If
End Function
```

**'Velocidad final transcurrido un tiempo en Plano Inclinado**

```
Public Function PI_Vf(g As Double, t As Double, i As Double, u As Double,
Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    PI_Vf = g * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180)) * t + Vo
    If PI_Vf < 0 Then
        PI_Vf = 0
    End If
End Function
```

**'Espacio recorrido transcurrido un tiempo en Plano Inclinado**

**'En caso de que la gravedad=0 el espacio retornado es 0 y  
'no se realiza el cálculo por medio de la ecuación para  
'evitar la división para 0.**

```
Public Function PI_s(g As Double, t As Double, i As Double, u As Double,
Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    If g = 0 Or i = 0 Then
        PI_s = 0
    Else
        PI_s = (PI_Vf(g, t, i, u, Vo) ^ 2 - Vo ^ 2) / (g * (Sin(i * 3.1416 / 180) - u *
Cos(i * 3.1416 / 180)) * 2)
        If PI_s < 0 Then
            PI_s = 0
        End If
    End If
End Function
```



**'Velocidad al final del recorrido en Plano Inclinado**

```
Public Function PI_Vt(g As Double, s As Double, i As Double, u As Double,
Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    PI_Vt = (g * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180)) * 2 * s + Vo ^
2) ^ 0.5
End Function
```

**'Tiempo al final del recorrido en Plano Inclinado.**

**'En caso de que la gravedad=0 el tiempo retornado es 0 y**

**'no se realiza el cálculo por medio de la ecuación para**

**'evitar la división para 0.**

```
Public Function PI_tt(g As Double, s As Double, i As Double, u As Double,
Optional Vo As Double)
    If IsMissing(Vo) Then Vo = 0
    If g = 0 Then
        PI_tt = 0
    Else
        PI_tt = (PI_Vt(g, s, i, u, Vo) - Vo) / (g * (Sin(i * 3.1416 / 180) - u * Cos(i *
3.1416 / 180)))
    End If
End Function
```

**'Período del Péndulo Simple**

```
Public Function PS_T(g As Double, l As Double)
    If g = 0 Then
        PS_T = 0
    Else
        PS_T = 2 * 3.1416 * (l / g) ^ 0.5
    End If
End Function
```

**'Frecuencia del Péndulo Simple**

```
Public Function PS_F(g As Double, l As Double)
    If g = 0 Then
        PS_F = 0
    Else
        PS_F = 1 / PS_T(g, l)
    End If
End Function
```

**'Angulo de abertura del Péndulo Simple en un tiempo dado**

```
Public Function PS_A(Ao As Double, l As Double, g As Double, t As Double)
    If l = 0 Then
        PS_A = 0
    Else
        PS_A = Ao * Sin(((g / l) ^ 0.5) * t + 1.5708)
    End If
End Function
```



## Funciones\_Práctica (code)

```
Dim io_valor As Integer
```

**'Abre la función que devuelve el estado del pin Busy del puerto de impresora.**

```
Public Function Busy() As Boolean
  If Val(FPractica.Ioport1.bitsize) = 8 Then
    io_valor = Val(FPractica.Ioport1.Value)
    If (io_valor And 128) = 128 Then
      ' cuando el bit 7 está en 1L
      Busy = True 'tapado
    Else
      ' cuando el bit 7 está en 0L
      Busy = False 'destapado
    End If
    FPractica.Ioport1.address = 889
    FPractica.Ioport1.bitsize = 8
  End If
End Function
```

**'Abre la función que devuelve el estado del pin PageEnd del puerto de impresora.**

```
Public Function PageEnd() As Boolean
  If Val(FPractica.Ioport1.bitsize) = 8 Then
    io_valor = Val(FPractica.Ioport1.Value)
    If (io_valor And 32) = 32 Then
      PageEnd = False
      FPractica.Ioport1.address = 889
      FPractica.Ioport1.bitsize = 8
    Else
      PageEnd = True
    End If
  End If
End Function
```

```
Public Function CL_V1(s1 As Double, a As Double, t1 As Double)
  CL_V1 = (s1 - 0.5 * a * t1 ^ 2) / t1
End Function
```

```
Public Function CL_V2(v1 As Double, a As Double, t1 As Double)
  CL_V2 = v1 + a * t1
End Function
```

```
Public Function CL_V3(s2 As Double, a As Double, v2 As Double)
  CL_V3 = (2 * a * s2 + v2 ^ 2) ^ 0.5
End Function
```

```
Public Function CL_S0(v1 As Double, a As Double)
  CL_S0 = (v1 ^ 2) / (2 * a)
```



```

End Function
Public Function CL_T0(v1 As Double, a As Double)
    CL_T0 = v1 / a
End Function

Public Function CL_T2(v2 As Double, v3 As Double, a As Double)
    CL_T2 = (v3 - v2) / a
End Function

Public Function PI_V1(a As Double, t1 As Double, u As Double, i As Double)
    PI_V1 = a * t1 * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180))
End Function

Public Function PI_At(a As Double, u As Double, i As Double)
    PI_At = a * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180))
End Function

Public Function PI_V3(s2 As Double, a As Double, v2 As Double, u As Double, i
As Double)
    PI_V3 = (2 * a * s2 * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180)) + v2 ^
2) ^ 0.5
End Function

Public Function PI_S0(a As Double, t1 As Double, u As Double, i As Double)
    PI_S0 = (t1 ^ 2) * 0.5 * a * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180))
End Function

Public Function PI_T0(v1 As Double, a As Double, u As Double, i As Double)
    PI_T0 = v1 / (a * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180)))
End Function

Public Function PI_T2(v2 As Double, v3 As Double, a As Double, u As Double, i
As Double)
    PI_T2 = (v3 - v2) / (a * (Sin(i * 3.1416 / 180) - u * Cos(i * 3.1416 / 180)))
End Function

Public Function Tp(t As Double, N As Double)
    If N > 0 Then
        Tp = t / N
    Else
        Tp = 0
    End If
End Function

Public Function PS_G(l As Double, t As Double)
    If t > 0 Then
        PS_G = ((2 * 3.1416 / t) ^ 2) * l
    Else
        PS_G = 0
    End If
End Function

```



```

End If
End Function
Public Function err(ve As Double, vt As Double)
    err = (Abs(ve - vt) * 100) / (vt)
End Function

```

### Impresión\_datos (code)

```

Public linea As String
Public narch As String
Public BeginPage, EndPage, numcopies
Dim i, j As Integer
Dim Tp As Double
Dim Uv, Ua, Ud, Up, L1, M1 As String

Public Function guardaresto(panel As Integer, nombre As String)
    If FPrincipal.Unidades(21).Checked = True Then
        Uv = " [m/s]"
        Ua = " [m/s" & Chr(178) & "]"
        Ud = " [m]"
        Up = " [Kg]"
        L1 = IIf(FPractica.PS_Longitud.Text = "Grande", " 0.395", " 0.15")
        M1 = IIf(FPractica.PS_Masa.Text = "Grande", " 0.08", " 0.03")
    End If
    If FPrincipal.Unidades(22).Checked = True Then
        Uv = " [cm/s]"
        Ua = " [cm/s" & Chr(178) & "]"
        Ud = " [cm]"
        Up = " [g]"
        L1 = IIf(FPractica.PS_Longitud.Text = "Grande", " 39.5", " 15")
        M1 = IIf(FPractica.PS_Masa.Text = "Grande", " 80", " 30")
    End If
    If FPrincipal.Unidades(23).Checked = True Then
        Uv = " [pies/s]"
        Ua = " [pies/s" & Chr(178) & "]"
        Ud = " [pies]"
        Up = " [libras]"
        L1 = IIf(FPractica.PS_Longitud.Text = "Grande", " 1.2959", " 0.4921")
        M1 = IIf(FPractica.PS_Masa.Text = "Grande", " 0.176", " 0.066")
    End If
    Open nombre For Output As #2
    linea = "Escuela Politécnica Nacional"
    Print #2, linea
    linea = "Facultad de Ingeniería Eléctrica"

```



```

Print #2, linea
linea = "Departamento de Electrónica y Telecomunicaciones"
Print #2, linea
linea = "Experimentos de Física en Computador"
Print #2, linea
Print #2, "_____ "

If panel = 0 Then
  linea = "Caída Libre:"
  Print #2, linea
  Print #2,
  linea = "Datos: "
  Print #2, linea
  linea = "Gravedad: " & FPractica.CL_Gravedad.Text + Ua
  Print #2, linea
  Print #2,
  linea = "Velocidad Tiempo Distancia Err/Dist."
  Print #2, linea
  linea = " " & Uv & " [s] " & Ud & " [%]"
  Print #2, linea
  linea = "L0 : " & FPractica.CL_Velocidad(0).Text & " " &
  FPractica.CL_Tiempo(0).Text & " " & FPractica.CL_Distancia(0).Text & "
  " & FPractica.CL_Error(0).Text
  Print #2, linea
  linea = "L0-L1: " & FPractica.CL_Velocidad(1).Text & " " &
  FPractica.CL_Tiempo(1).Text & " " & FPractica.CL_Distancia(1).Text & "
  " & FPractica.CL_Error(1).Text
  Print #2, linea
  linea = "L1-L2: " & FPractica.CL_Velocidad(2).Text & " " &
  FPractica.CL_Tiempo(2).Text & " " & FPractica.CL_Distancia(2).Text & "
  " & FPractica.CL_Error(2).Text
  Print #2, linea
  linea = "L2-L3: " & FPractica.CL_Velocidad(3).Text & " " &
  FPractica.CL_Tiempo(3).Text & " " & FPractica.CL_Distancia(3).Text & "
  " & FPractica.CL_Error(3).Text
  Print #2, linea
  Print #2, "_____ "
End If

If panel = 1 Then
  linea = "Plano Inclinado:"
  Print #2, linea
  Print #2,
  linea = "Datos: "
  Print #2, linea
  linea = "Gravedad : " & FPractica.PI_Gravedad.Text + Ua
  Print #2, linea

```



```

linea = "Rozamiento : " & FPractica.PI_Rozamiento.Text
Print #2, linea
linea = "Inclinación : " & FPractica.PI_Inclinacion.Text & " [grados]"
Print #2, linea
Print #2,
linea = "Velocidad      Tiempo      Distancia"
Print #2, linea
linea = " " & Uv & "      [s]      " & Ud
Print #2, linea
linea = FPractica.PI_Velocidad(0).Text & "      " &
FPractica.PI_Tiempo(0).Text & "      " & FPractica.PI_Distancia(0).Text
Print #2, linea
linea = FPractica.PI_Velocidad(1).Text & "      " &
FPractica.PI_Tiempo(1).Text & "      " & FPractica.PI_Distancia(1).Text
Print #2, linea
Print #2,
linea = "Aceleración: " & FPractica.PI_CAceleracion.Text + Ua
Print #2, linea
linea = "Error / Distancia: " & FPractica.PI_CError.Text & " %"
Print #2, linea
Print #2, " _____"
End If

If panel = 2 Then
Tp = 0
linea = "Péndulo Simple:"
Print #2, linea
Print #2,
linea = "Datos: "
Print #2, linea
linea = "Gravedad : " & FPractica.PS_Gravedad.Text + Ua
Print #2, linea
linea = "Masa      : " & FPractica.PS_Masa.Text + M1 + Up
Print #2, linea
linea = "Longitud : " & FPractica.PS_Longitud.Text + L1 + Ud
Print #2, linea
linea = "Períodos : " & FPractica.PS_Numero_de_Periodos
Print #2, linea
Print #2,
linea = "Oscilación Período"
Print #2, linea
j = If(FPractica.PS_Masa.Text = "Pequeña", 1, 0)
For i = 0 To Val(FPractica.PS_Numero_de_Periodos) - 1
linea = " O" & i + 1 & "      " & FPractica.PS_Periodos(j).List(i) & " [s]"
Tp = Tp + Val(FPractica.PS_Periodos(j).List(i))
Print #2, linea
Next i

```



```

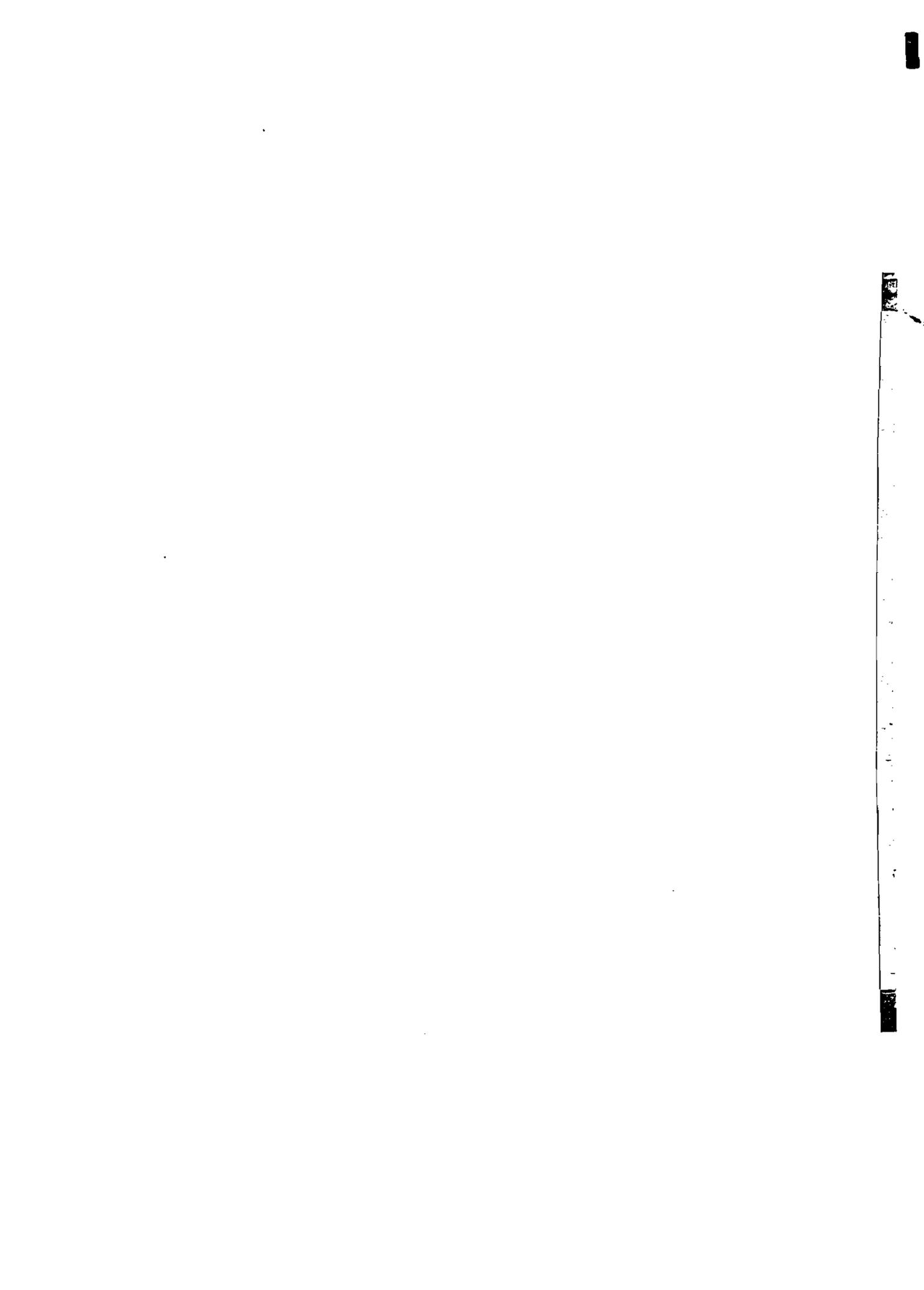
Tp = Tp / i
Print #2,
linea = "Resultados:"
Print #2, linea
linea = " Tp    = " & Tp & " [s]"
Print #2, linea
linea = " g     = " & FPractica.PS_Gravedad_calculada(j).Text + Ua
Print #2, linea
linea = " error = " & FPractica.PS_Error(j) & " %"
Print #2, linea
End If
Close #2
guardaresto = 1
End Function

Public Function imprimiresto(panel As Integer) As Integer
If FPrincipal.Unidades(21).Checked = True Then
    Uv = " [m/s]"
    Ua = " [m/s" & Chr(178) & "]"
    Ud = " [m]"
    Up = " [Kg]"
    L1 = IIf(FPractica.PS_Longitud.Text = "Grande", " 0.395", " 0.15")
    M1 = IIf(FPractica.PS_Masa.Text = "Grande", " 0.08", " 0.03")
End If
If FPrincipal.Unidades(22).Checked = True Then
    Uv = " [cm/s]"
    Ua = " [cm/s" & Chr(178) & "]"
    Ud = " [cm]"
    Up = " [g]"
    L1 = IIf(FPractica.PS_Longitud.Text = "Grande", " 39.5", " 15")
    M1 = IIf(FPractica.PS_Masa.Text = "Grande", " 80", " 30")
End If
If FPrincipal.Unidades(23).Checked = True Then
    Uv = " [pies/s]"
    Ua = " [pies/s" & Chr(178) & "]"
    Ud = " [pies]"
    Up = " [libras]"
    L1 = IIf(FPractica.PS_Longitud.Text = "Grande", " 1.2959", " 0.4921")
    M1 = IIf(FPractica.PS_Masa.Text = "Grande", " 0.176", " 0.066")
End If
Open "c:\temp.txt" For Output As #1
'FontSize = 14 ' Set FontSize.
Print #1,

```







```

If panel = 1 Then
  linea = "Plano Inclinado:"
  Print #1, Spc(40); linea
  Print #1,
  linea = "Datos: "
  Print #1, Spc(40); linea
  linea = "Gravedad  :" & FPractica.PS_Gravedad.Text + Ua
  Print #1, Spc(40); linea
  linea = "Masa      :" & FPractica.PS_Masa.Text + Up
  Print #1, Spc(40); linea
  linea = "Longitud  :" & FPractica.PS_Longitud.Text + Ud
  Print #1, Spc(40); linea
  linea = "Períodos  :" & FPractica.PS_Numero_de_Periodos
  Print #1, Spc(40); linea
  Print #1,
  linea = "Velocidad      Tiempo          Distancia"
  Print #1, Spc(40); linea
  linea = "          " & Uv & "          [s]          " & Ud
  Print #1, Spc(40); linea
  linea = FPractica.PI_Velocidad(0).Text & "          " &
  FPractica.PI_Tiempo(0).Text & "          " & FPractica.PI_Distancia(0).Text
  Print #1, Spc(40); linea
  linea = FPractica.PI_Velocidad(1).Text & "          " &
  FPractica.PI_Tiempo(1).Text & "          " & FPractica.PI_Distancia(1).Text
  Print #1, Spc(40); linea
  Print #1,
  linea = "Aceleración: " & FPractica.PI_CAceleracion.Text
  Print #1, Spc(40); linea
  linea = "Error / Distancia: " & FPractica.PI_CError.Text
  Print #1, Spc(40); linea
  Print
  "          #1,          Spc(40);
  "          "
End If

```

```

If panel = 2 Then
  'Opciones de impresión
  Tp = 0
  linea = "Péndulo Simple:"
  Print #1, Spc(40); linea
  Print #1,
  linea = "Datos: "
  Print #1, Spc(40); linea
  linea = "Gravedad  :" & FPractica.PS_Gravedad.Text + Ua
  Print #1, Spc(40); linea
  linea = "Masa      :" & FPractica.PS_Masa.Text + M1 + Up
  Print #1, Spc(40); linea
  linea = "Longitud  :" & FPractica.PS_Longitud.Text + L1 + Ud

```



```

Print #1, Spc(40); linea
linea = "Períodos  :" & FPractica.PS_Numero_de_Periodos
Print #1, Spc(40); linea
Print #1,
linea = "Oscilación  Período"
Print #1, Spc(40); linea
j = IIf(FPractica.PS_Masa.Text = "Pequeña", 1, 0)
For i = 0 To Val(FPractica.PS_Numero_de_Periodos) - 1
    linea = " O" & i + 1 & "      " & FPractica.PS_Periodos(j).List(i) & " [s]"
    Tp = Tp + Val(FPractica.PS_Periodos(j).List(i))
    Print #1, Spc(40); linea
Next i
Tp = Tp / i
Print #1,
linea = "Resultados:"
Print #1, Spc(40); linea
linea = " Tp      =" & Tp & " [s]"
Print #1, Spc(40); linea
linea = " g      =" & FPractica.PS_Gravedad_calculada(j).Text + Ua
Print #1, Spc(40); linea
linea = " error  =" & FPractica.PS_Error(j) & " %"
Print #1, Spc(40); linea
Print #1, Spc(40); " _____"
End If
Close #1

FPractica.fp_impresora.CancelError = True
FPractica.fp_impresora.Flags = &H8
On Error GoTo ErrHandler
FPractica.fp_impresora.ShowPrinter
If FPractica.fp_impresora.CancelError = False Then Exit Function

BeginPage = FPractica.fp_impresora.FromPage
EndPage = FPractica.fp_impresora.ToPage
numcopies = FPractica.fp_impresora.Copies
Printer.Copies = numcopies
Open "c:\temp.txt" For Input As #1
While Not EOF(1)
    Line Input #1, linea
    Printer.Print linea
Wend
Close #1
'MsgBox numcopies
Printer.EndDoc
'FontSize = 14 ' Set FontSize.
Exit Function
ErrHandler:

```



```
'FontSize = 10 ' Set FontSize.  
Exit Function  
End Function
```

### 3.4 UTILIZACION DE LA HERRAMIENTA IOPORT.OCX.

Aunque en el Internet no se muestra en mayor detalle el funcionamiento del control **IOPORT.OCX**, existe una página WEB en donde se puede obtener esta herramienta, allí se hace una reseña previa de los métodos de control directo del puerto paralelo usando lenguajes de programación basados en DOS, tal es el caso de los comandos de Qbasic **INP** y **OUT**.

Ya que VISUAL BASIC se ha ido transformando en un lenguaje de fácil utilización, en el cual se desarrollan muchos de los proyectos de hobistas, sigue siendo muy común que ciertas aplicaciones que tienen que ver con la comunicación externa, no son fácilmente programadas por la falta de recursos del sistema, existe un gran limitante que obliga al usuario a prescindir de la interacción directa con el puerto paralelo y recurrir al uso de otros lenguajes de programación, que en la mayoría de los casos, no los domina.

**IOPORT** se ha desarrollado específicamente con el fin de aumentar la capacidad de programación del Visual Basic, haciéndolo capaz de controlar en forma directa el acceso al puerto paralelo para comunicación externa. Se trata de un control que se inserta fácilmente en la caja de herramientas (ToolBox) del Visual Basic, y que de acuerdo a las especificaciones dadas en la página WEB, podrían manipularse **8** o **16** bits, tanto para lectura como para escritura. Esta diseñado como un control de 32 bits, para operar sobre **Windows 95** o **98**, y

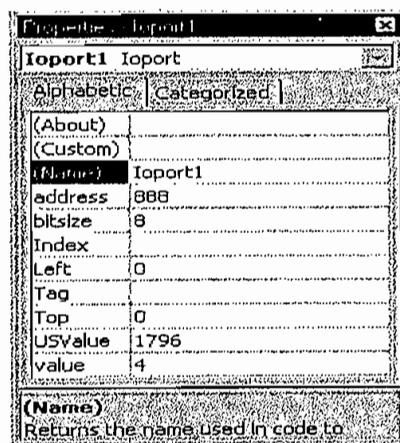


puede ser utilizado en las versiones 4.0 y 5.0 de Visual Basic. Para usuarios de NT existe la alternativa **NTPORT.OCX**.

Al acceder a las propiedades de esta herramienta, el usuario debe especificar el valor correspondiente al registro de entrada y/o salida que utiliza LPT1, (**EFC** usa el registro de entrada decimal **889**), además se debe especificar la cantidad de bits utilizados (bitsize) y el nombre con el que **IOPORT** será identificado al leerse el puerto.

### Propiedades de la Herramienta IOPORT.

Un ejemplo de utilización de este control en la programación, es el detallado en el código de programa que corresponde al módulo “Funciones\_práctica (code)”



```
Dim io_valor As Integer
Public Function Busy() As Boolean
    If Val(FPractica.Ioport1.bitsize) = 8 Then
        io_valor = Val(FPractica.Ioport1.Value)
        If (io_valor And 128) = 128 Then
            Busy = True 'tapado
        Else
            Busy = False 'destapado
        End If
        FPractica.Ioport1.address = 889
    End If
End Function
```

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and activities. It emphasizes that this is crucial for ensuring transparency and accountability in the organization's operations.

2. The second part of the document outlines the various methods and tools used to collect and analyze data. It highlights the need for consistent and reliable data collection processes to support informed decision-making.

3. The third part of the document focuses on the role of technology in modern data management. It discusses how advanced software solutions can streamline data collection, storage, and analysis, leading to more efficient and accurate results.

4. The fourth part of the document addresses the challenges associated with data management, such as data quality, security, and privacy. It provides strategies to mitigate these risks and ensure the integrity and confidentiality of the organization's data.

5. The fifth part of the document concludes by summarizing the key findings and recommendations. It stresses the importance of a proactive approach to data management to maximize the value of the organization's data assets.

### CONCLUSION

1. The document concludes by summarizing the key findings and recommendations. It stresses the importance of a proactive approach to data management to maximize the value of the organization's data assets.

```
FPractica.Ioport1.bitsize = 8
  End If
End Function
```

---

Nótese que primero se establece el dimensionamiento de la variable `io_valor`, como un valor entero (Integer). Luego es necesario verificar que el puerto tenga un tamaño de 8 bits. Esto se lo hace al obtener el valor que tiene la propiedad (bitsize) de la herramienta nombrada como `Ioport1`, y que se encuentra localizada en el formulario de nombre `Fpractica`. Si dicho valor es 8, entonces se almacena el valor leído del puerto en la variable `io_valor`. Así, el valor leído en el puerto corresponde al valor encontrado en la propiedad (value) de la herramienta `Ioport1` del formulario `Fpractica`.

En este punto, se realiza una comparación lógica de `io_valor`, con el valor decimal 128. Esto permite conocer si el bit 7 (Busy) está activado o no, para luego proceder con alguna actividad; por ejemplo, arrancar un cronómetro. Finalizada la verificación del puerto, es necesario la reubicación de los valores principales que hacen que `ioport1`, esté configurado como entrada: `address = 889, bitsize = 8`. Esto es necesario puesto que, existe la tendencia de que `ioport` adopte la configuración original: `address = 888, bitsize = 8`.



## CAPITULO IV

### RESULTADOS.

#### 4.1 CAMBIO DEL SISTEMA DE ILUMINACION DE LUZ VISIBLE A LUZ INFRARROJA.

Durante el diseño del interfaz general del proyecto y posteriormente, en las pruebas de funcionamiento, se detectaron problemas en la respuesta del circuito interfaz ante los cambios de luz detectados por los sensores. Puesto que los fototransistores trabajaban con luz visible o infrarroja, se utilizó luz visible proveniente de lámparas de 2.2 voltios, alimentadas por baterías de 1.5 voltios. En estas condiciones, el prototipo “Péndulo Simple” podía ser calibrado y trabajar adecuadamente; sin embargo, los prototipos “Caída Libre” y “Plano Inclinado” tenían inconvenientes, ya que luego de calibrarse uno de los dos sensores, el otro perdía sensibilidad.

En definitiva, se analizó más a fondo el funcionamiento del sistema y se concluyó lo siguiente:

1. Los fototransistores responden con luz visible, sin embargo, el funcionamiento es muy inestable debido a que en ciertos momentos, la lente del sensor recibe más o menos luz, dependiendo de las condiciones de luz ambientales. Esto influye sobre el mayor o menor grado de corriente entregada por el fototransistor utilizado, obligando al operador del interfaz a recalibrar constantemente todo el sistema. En el caso específico de los equipos de “Caída Libre” y “Plano Inclinado”, este tipo de inconveniente provocaba el constante



desequilibrio del sistema y el interfaz no trabajaba con la sensibilidad y la velocidad de detección requeridas.

2. Otro factor del cual dependía el funcionamiento del equipo, era la calidad de las lámparas utilizadas para la iluminación de los sensores. Aunque en un comienzo se usaban lámparas construidas en China, la mala calidad de éstos elementos se notaba en la apariencia y la vida útil que presentaban. Usualmente el máximo tiempo registrado de funcionamiento de las lámparas Chinas fue de aproximadamente 3 horas, tiempo dentro del cual, se veía una constante variación de la intensidad luminosa que decrecía constantemente, hasta que finalmente, la lámpara se dañaba.
3. Con esto se pensó definitivamente en reemplazar las lámparas utilizadas por lámparas Radio Shack, las cuales proveían de luz más estable y con las que se podría llegar a obtener mejores resultados al calibrar el circuito interfaz de los tres prototipos. Sin embargo, no se dejaba de sentir la sensación de que los dos sistemas electrónicos paralelos, que tenían los interfaces “Caída Libre” y “Plano Inclinado”, funcionaban aún inadecuadamente por el tipo de luz recibida en los sensores.
4. Con el transcurso del tiempo de uso y los transitorios soportados por las lámparas, éstas comenzaron a perder su calidad y definitivamente se debería buscar una alternativa de iluminación más estable, de mayor tiempo de vida útil y que sea capaz de excitar adecuadamente a los dos fototransistores de cada proyecto. Con estas premisas se cambia todo el sistema de iluminación normal (luz visible) a luz infrarroja, utilizando diodos LED en la banda de los 950 nm.



Con este tipo de luz y la potencia de iluminación que emiten estos dispositivos, independientemente de las condiciones de luz externas, los interfaces de cada prototipo funcionan adecuadamente.

El nuevo diseño requiere de cambios a nivel de la electrónica del circuito impreso y a nivel de las conexiones externas. Desaparecen las lámparas de luz visible y las baterías que las alimentan, además en los prototipos “Caída Libre” y “Plano Inclinado” se extrae el interruptor de encendido y apagado de las lámparas. En el “Péndulo simple” se eliminan los cables de las luces superior e inferior conectados en el conmutador (S1).

En los postes de luz de los prototipos “Péndulo Simple” y “Plano Inclinado”, se extraen las boquillas que se usaban para albergar los focos utilizados, y se reemplazan por los diodos LED infrarrojos, esta vez montados sobre unas bases de soporte y apuntamiento, construidas también en acrílico transparente de 3 líneas, (ver *ANEXOS literal A, Diagramas de los módulos físicos, pag. 218, 224, 227*).

Finalmente para completar la nueva estructura del proyecto, es necesario revisar los cambios en el diseño electrónico para que el circuito interfaz soporte el nuevo sistema de iluminación. Ya que los diodos LED infrarrojos son dispositivos que se alimentan con voltajes bajos (para este caso el voltaje de polarización de los diodos es de 1.6 voltios), puede utilizarse la misma fuente de alimentación de



+5 voltios DC del circuito. En el diseño se tendrá que utilizar una resistencia limitadora de corriente, la cual permitirá polarizar al diodo LED para su correcto funcionamiento. La *figura 4.1*, muestra el diagrama de bloques del nuevo circuito interfaz, que es muy similar al diagrama planteado en el diseño anterior (véase *capítulo II, figura 2.1, pag. 35*), la diferencia radica en el tipo de luz utilizada y la forma cómo se alimenta el sistema de iluminación. Adicionalmente en la *figura 4.2, pag. 142*, se indica el nuevo diagrama electrónico del sistema de iluminación, el que consta de un diodo LED  $D$  ( $\lambda = 950$  nm), y una resistencia  $R4$ , conectados hacia la fuente de alimentación principal.

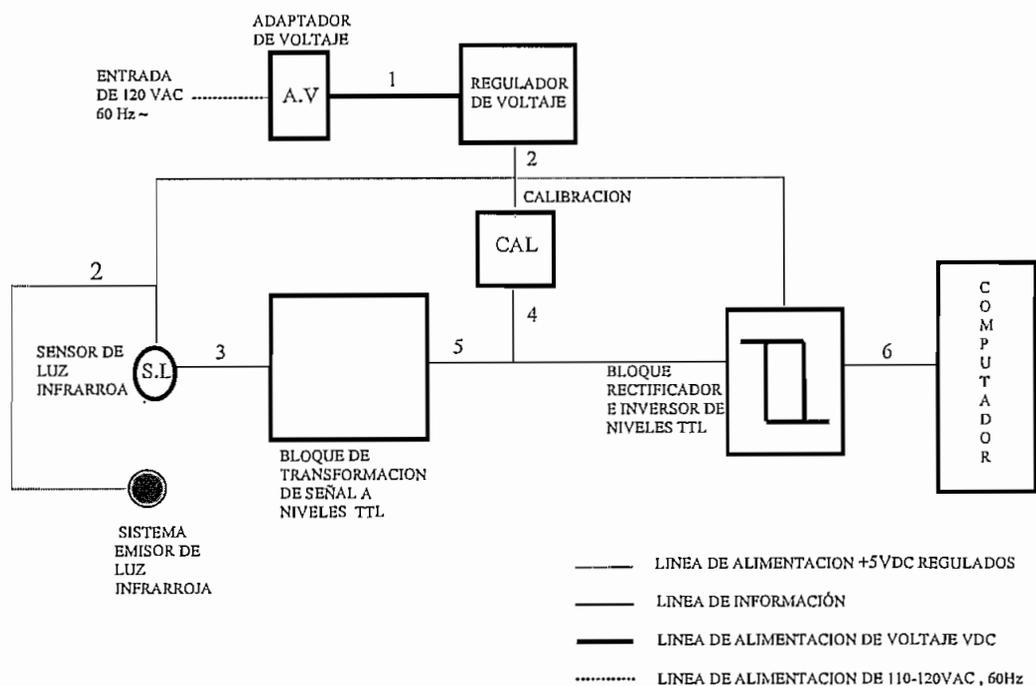


Figura 4.1.- Diagrama de bloques del circuito interfaz, con el sistema de luz infrarroja.



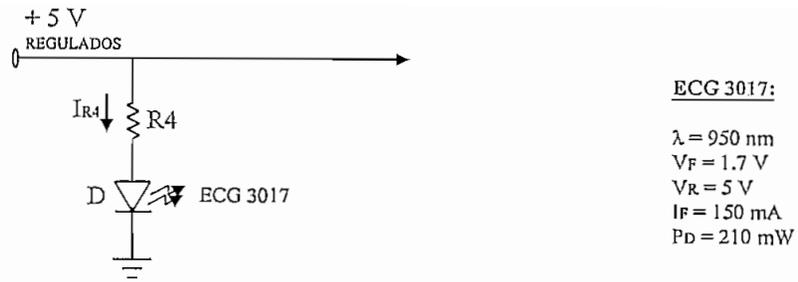


Figura 4.2.- Diagrama electrónico para el sistema de iluminación en el interfaz general.

Este nuevo sistema de iluminación consta de un diodo LED infrarrojo, el cual, es polarizado directamente mediante la misma fuente de alimentación de +5 VDC, que se utiliza en el circuito principal. R4 es usada para limitar la corriente en el diodo.

Para adaptar este pequeño esquema adicional a los tres prototipos, se deberá considerar que:

1. Para el prototipo "Péndulo Simple", se conectarán dos de estos sistemas para ser alimentados a través del interruptor conmutador (S1), como lo indica la figura 4.3.

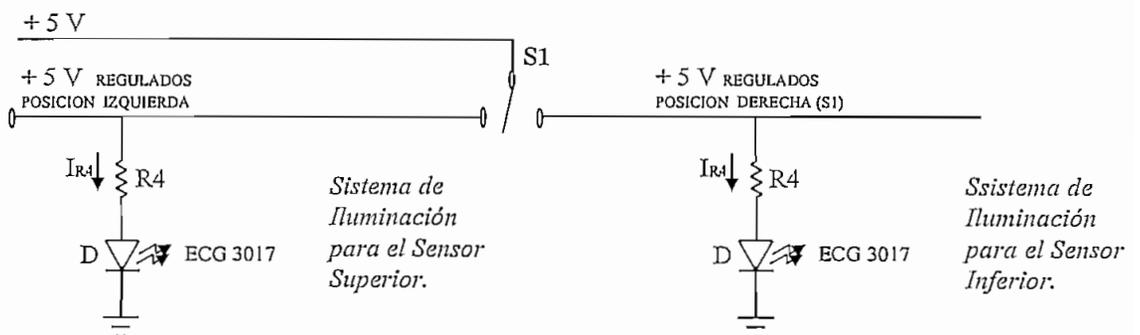


Figura 4.3.- Diagrama electrónico para el sistema de iluminación de "Péndulo Simple".



**NOTA:** Las funciones de los sensores superior e inferior son las de detectar el movimiento de un péndulo de longitud de cuerda pequeña y grande, respectivamente.

2. Para los prototipos “Caída Libre” y “Plano Inclinado”, los sistemas de iluminación se conectarán a la alimentación principal del circuito interfaz, es decir, a la salida del **7805C**, como lo indica la *figura 4.4*.

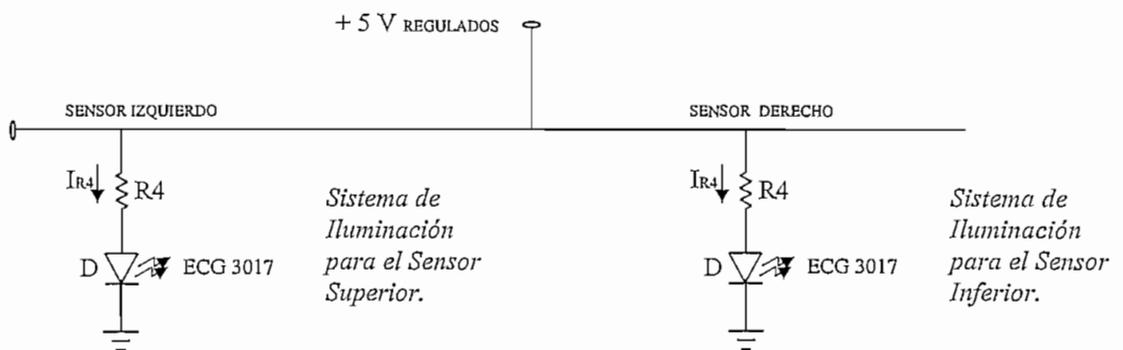


Figura 4.4.- Sistema de Iluminación para “Caída Libre” y “Plano inclinado”.

Para el diseño se debe considerar lo siguiente:

Sobre el diodo se tendrá un voltaje de 1.7 V, voltaje de polarización directa. Esto implica que sobre la resistencia **R4** caerían:

$$V_{R4} = V_{CC} - V_D = (5-1.7)V = 3.3V.$$

Debido a que R4 es una resistencia limitadora de corriente al diodo, ésta deberá ser dimensionada, de tal forma que la corriente que circula por el diodo sea adecuada y que no dañe al dispositivo. Por ejemplo:

Sea  $I_{R4} = 20 \text{ mA}$ , entonces:

$V_R = 3.3 \text{ V}$ , entonces:



$$R_4 = V_{R_4} / I_{R_4} = 3.3 \text{ V} / 20 \text{ mA} = 165 \Omega$$

Si se utiliza  $R_4 = 150 \Omega$ , la corriente  $I_{R_4}$  que circularía por la resistencia limitadora sería de 22 mA, y, la potencia de la luz emitida por el LED es mayor. Esto es conveniente puesto que el LED transmisor se encuentra alejado, una distancia aproximadamente de 10 cm del receptor (fototransistor). En resumen, no se necesita mayor distancia de separación entre transmisor y receptor, ni tampoco consumir más corriente de la fuente de alimentación. Volviendo a plantear las ecuaciones, se obtendría lo siguiente:

Para  $R_4 = 150 \Omega$ ;

$$I_{R_4} = 3.3 \text{ V} / 150 \Omega = 22 \text{ mA}$$

Y la potencia del LED, se calcula por:

$$P_D = (I_{R_4})^2 * R_4 = (22 \text{ mA})^2 * 150 \Omega = 72.6 \text{ mW}$$

Entonces, para el diseño se elige:

$$R_4 = 150 \Omega$$

Terminado el diseño del nuevo sistema de iluminación, se procedió con las pruebas de electrónica de todo el interfaz, obteniéndose como resultado las tablas que se muestran en el apartado 4.2 de este capítulo. Lo nuevos diagramas esquemáticos correspondientes a los circuitos interfaces de los prototipos “Péndulo Simple”, “Caída Libre” y “Plano Inclinado”, se muestran en la sección de *ANEXOS, literal D, pags. 237, 238 y 239.*



## 4.2 PRUEBAS DE ELECTRONICA.

Para la realización de este apartado se ha tomado en cuenta que:

1. El interfaz específico se encuentra calibrado.
2. El interfaz está conectado al computador a través del puerto paralelo.
3. El sistema de iluminación de los sensores trabaja con diodos LED infrarrojos.

### 4.2.1 TABLA DE RESULTADOS EXPERIMENTALES DEL INTERFAZ

#### “PENDULO SIMPLE”

Dentro del circuito interfaz de "Péndulo Simple" han sido ubicados varios puntos de medición, los cuales, proporcionan los principales valores de voltajes y corrientes del circuito. Estos puntos se encuentran también en el diagrama esquemático de Péndulo Simple (*ver ANEXOS, literal D, pag. 237*), y tienen el siguiente significado:

V<sub>A</sub>.- Indica el voltaje regulado proveniente del circuito integrado 7805 C.

V<sub>B</sub>.- Define el voltaje en la base del BJT.

V<sub>C</sub>.- Voltaje entre el potenciómetro de calibración R2 y la resistencia de protección R3.

V<sub>D</sub>.- Voltaje de colector del BJT.

V<sub>F</sub>.- Voltaje a la salida del “Schmitt Trigger”.

V<sub>G</sub>.- Voltaje de entrada al circuito integrado 7805C.

V<sub>H</sub>.- Voltaje del diodo LED infrarrojo.

I<sub>A</sub>.- Corriente que circula desde el emisor del fototransistor hasta el punto de voltaje V<sub>B</sub>.

I<sub>B</sub>.- Corriente en la resistencia limitadora y en el diodo LED.



Ic.- Corriente sobre la resistencia R1.

Id.- Corriente en el colector del fototransistor activo.

Ie.- Corriente que entra a la base del BJT.

If.- Corriente de entrada al puerto del computador.

Ig.- Corriente exigida al regulador de voltaje. Consumo del circuito interfaz.

Ih.- Corriente proporcionada por el adaptador de voltaje.

Ii.- Corriente en el potenciómetro R2 y en la resistencia R3.

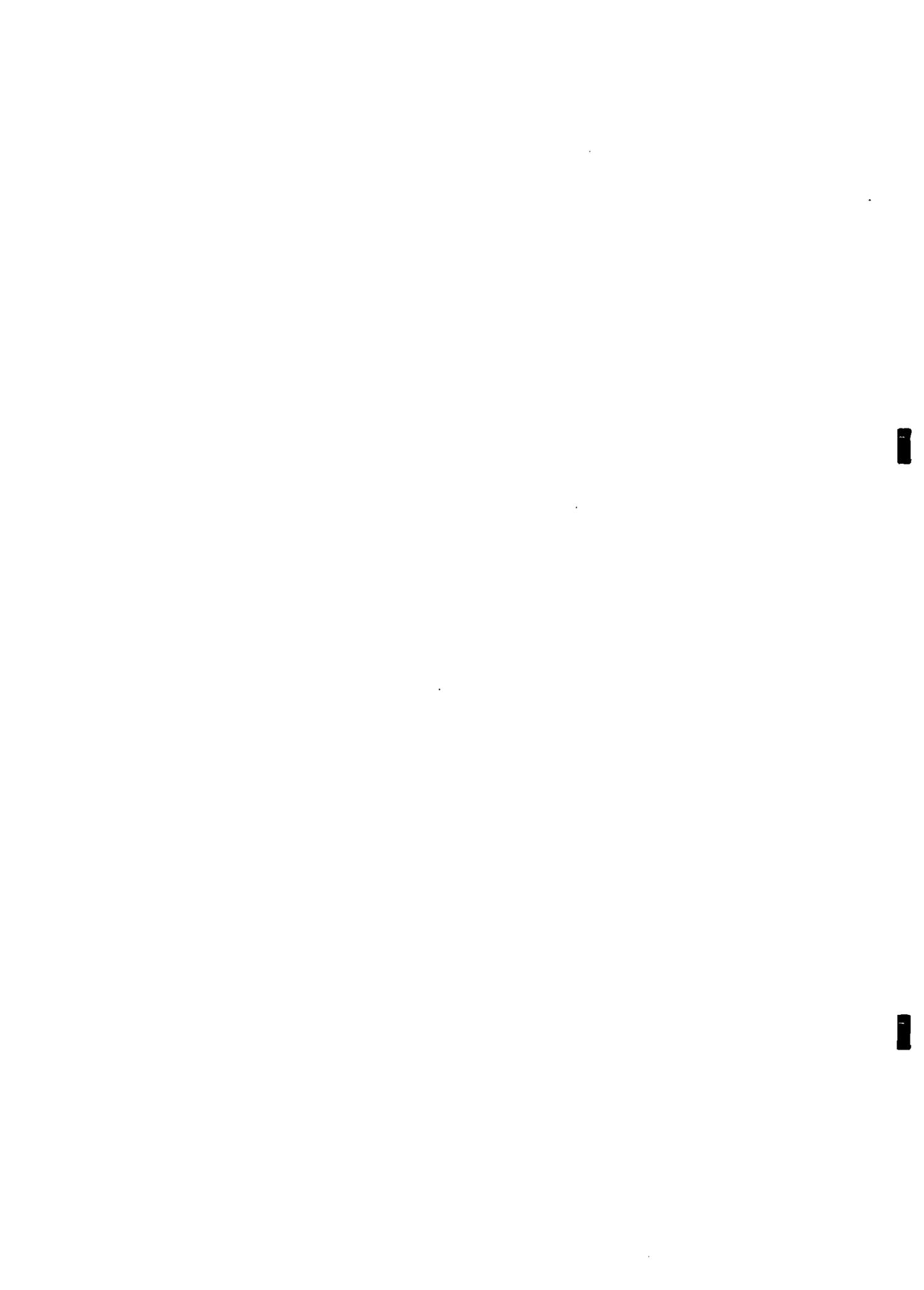
La siguiente tabla muestra los valores experimentales obtenidos en la práctica.

LUZ				OSCURIDAD			
Referencia	V [V]	Referencia	I [mA]	Referencia	V [V]	Referencia	I [mA]
VA	5.02	IA	0.4	VA	5.02	IA	0.0278
VB	0.79	IB	25.6	VB	0.71	IB	25.8
VC	4.35	IC	0.0062	VC	4.95	IC	0.0057
VD	0.21	ID	0.313	VD	4.47	ID	0.0116
VF	4.49	IE	0.402	VF	0.18	IE	0.0095
VG	9.79	IF	0.003	VG	9.86	IF	0
VH	1.24	IG	49.2	VH	1.23	IG	38.5
		Ih	54.7			Ih	43.2
		Ii	12.2			Ii	1.92

Tabla 4.2.1.- Datos experimentales medidos en el interfaz de “Péndulo Simple”.

De acuerdo a la *tabla 4.2.1*, se puede observar que:

- El voltaje de base del BJT ( $V_B$ ) es similar cuando el sensor está iluminado, como cuando no lo está.
- El voltaje de colector tiene un valor bajo, cuando el sensor está iluminado; y un valor alto, en condiciones de oscuridad.
- En el voltaje de salida del “Schmitt Trigger” ( $V_F$ ), los datos proporcionados por el interfaz concuerdan, puesto que cuando el sensor está iluminado, el voltaje de salida ( $V_F$ ) tiene un nivel alto, en caso contrario, ( $V_F$ ) tiene un nivel bajo.



- Nótese que la corriente que circula por el diodo LED infrarrojo ( $I_B$ ) es muy parecida a la corriente calculada en el diseño: (22 mA).
- El máximo consumo de corriente del circuito es de 54.7 mA ( $I_H$ ), en condiciones de luz sobre el sensor.
- En el pin #11 del puerto paralelo del computador, la máxima cantidad de corriente que se tiene es ( $I_F = 3 \text{ uA}$ ), en condiciones de luz sobre el sensor. Por lo tanto el puerto no corre riesgo alguno de daño en sus componentes internos.

#### 4.2.2 TABLA DE RESULTADOS EXPERIMENTALES DEL INTERFAZ “CAIDA LIBRE”

Para los interfaces de “Caída Libre” y “Plano Inclinado”, se han dispuesto también de puntos de medición de voltaje y corriente. Estos valores experimentales obtenidos se muestran en las *tablas 4.2.2 y 4.2.3, pags. 149 y 150*, respectivamente y se encuentran ubicados dentro de los respectivos diagramas esquemáticos (*ver ANEXOS, literal D, pags. 238 y 239*):

VA.- Indica el voltaje regulado proveniente del circuito integrado 7805 C.

VB.- Define el voltaje en la base del BJT (Q2).

VC.- Define el voltaje en la base del BJT (Q4).

VD.- Voltaje de colector del BJT (Q2).

VE.- Voltaje de colector del BJT (Q4).

VF.- Voltaje a la salida del “Schmitt Trigger” U1A.

VG.- Voltaje a la salida del “Schmitt Trigger” U1B.

VH.- Voltaje de entrada a regulador de voltaje 7805C.

VI.- Voltaje en el punto entre el diodo D1 y la resistencia R4.



- V<sub>J</sub>.- Voltaje en el punto entre el potenciómetro R2 y la resistencia R3, sensor superior.
- V<sub>K</sub>.- Voltaje en el punto entre el diodo D2 y la resistencia R4.
- V<sub>L</sub>.- Voltaje en el punto entre el potenciómetro R2 y la resistencia R3, sensor inferior.
- I<sub>A</sub>.- Corriente que circula desde el emisor del fototransistor (Q3) hasta el punto de voltaje V<sub>B</sub>.
- I<sub>B</sub>.- Corriente que circula desde el emisor del fototransistor (Q1) hasta el punto de voltaje V<sub>C</sub>.
- I<sub>C</sub>.- Corriente sobre la resistencia R1, sensor superior.
- I<sub>D</sub>.- Corriente sobre la resistencia R1, sensor inferior.
- I<sub>E</sub>.- Corriente que entra a la base del BJT (Q2).
- I<sub>F</sub>.- Corriente que entra a la base del BJT (Q4).
- I<sub>G</sub>.- Corriente exigida al regulador de voltaje. Consumo del circuito interfaz.
- I<sub>H</sub>.- Corriente proporcionada por el adaptador de voltaje.
- I<sub>I</sub>.- Corriente en el potenciómetro R2 y en la resistencia R3, sensor superior.
- I<sub>J</sub>.- Corriente en el potenciómetro R2 y en la resistencia R3, sensor inferior.
- I<sub>K</sub>.- Corriente en el colector del fototransistor (Q1).
- I<sub>L</sub>.- Corriente en el diodo D2, sensor inferior.
- I<sub>M</sub>.- Corriente en el colector del fototransistor (Q3).
- I<sub>N</sub>.- Corriente en el diodo D1, sensor superior.
- I<sub>O</sub>.- Corriente de entrada al "Schmitt Trigger" U1A.
- I<sub>P</sub>.- Corriente de entrada al "Schmitt Trigger" U1B.



I<sub>Q</sub>.- Corriente de salida hacia el computador U1A.

I<sub>R</sub>.- Corriente de salida hacia el computador U1B.

La tabla de valores experimentales se detalla a continuación:

LUZ				OSCURIDAD			
Referencia	V [v]	Referencia	I [mA]	Referencia	V [v]	Referencia	I [mA]
VA	4.96	IA	0.0322	VA	4.99	IA	0.008
VB	0.78	IB	0.303	VB	0.07	IB	0.0009
VC	0.73	IC	0.0064	VC	0.05	IC	0.0008
VD	0.2	ID	0.0075	VD	4.98	ID	0.0006
VE	0.2	IE	0.314	VE	4.98	IE	0
VF	4.46	IF	0.283	VF	0.14	IF	0
VG	4.42	IG	99.6	VG	0.14	IG	63.1
VH	9.51	IH	107.4	VH	10.1	IH	69.1
VI	1.19	II	19.7	VI	1.19	II	0
VJ	3.88	IJ	19.2	VJ	4.97	IJ	0
VK	1.12	IK	0.321	VK	1.14	IK	0.0031
VL	4.02	IL	26	VL	4.98	IL	26
		IM	0.4			IM	0.0055
		IN	25.1			IN	25.3
		IO	0.215			IO	0
		IP	0.186			IP	0
		IQ	0.0004			IQ	0
		IR	0.0004			IR	0

Tabla 4.2.2.- Datos experimentales medidos en el interfaz de “Caída Libre”.

En la tabla anterior, se puede observar que:

- A diferencia de la *tabla 4.2.1, pag. 146*, el voltaje de alimentación regulado es ahora de 4.96 voltios.
- El consumo máximo del circuito electrónico total es de 107.4 mA cuando hay luz sobre los dos sensores. Esto se debe a que, en este caso, se tienen dos circuitos sensores en paralelo, asistidos por la misma fuente de alimentación.
- Los voltajes de los colectores de los BJT ( $V_D$ ) y ( $V_E$ ), son niveles bajos cuando los sensores reciben luz, y niveles altos cuando éstos se encuentran apagados. Los valores obtenidos de 0.2 voltios, corresponden a los valores ubicados cuando se calibró el equipo.
- De la misma forma, los voltajes de salida de los “Schmitt Trigger” ( $V_F$ ) y ( $V_G$ ) son concordantes, pues indican un nivel alto cuando lo sensores están



iluminados; y un nivel bajo, cuando los sensores se encuentran en condiciones de oscuridad. Los niveles lógicos resultantes son totalmente reconocidos por el computador como niveles TTL.

- Las corrientes que ingresan al computador ( $I_Q$ ) y ( $I_R$ ), en el peor de los casos tienen un valor de 0.4  $\mu A$ , por lo tanto el computador no puede sufrir daños durante la conexión del circuito interfaz al puerto paralelo.
- Las corrientes que circulan a través de los diodos LED ( $I_N$ ) y ( $I_L$ ), tienen un valor de 25.1 mA y 26 mA respectivamente, similar a lo calculado en el diseño electrónico. Por tanto, se asegura que los diodos trabajen adecuadamente.

#### 4.2.3 TABLA DE RESULTADOS EXPERIMENTALES DEL INTERFAZ “PLANO INCLINADO”.

En el prototipo “Plano Inclinado”, los resultados experimentales son muy parecidos a los expuestos en la *tabla 4.2.2, pag. 149*, esto se debe a que los interfaces de “Caída Libre” y “Plano Inclinado” son iguales en su diseño electrónico. La *tabla 4.2.3*, muestra los valores obtenidos para “Plano Inclinado”:

LUZ				OSCURIDAD			
Referencia	V [v]	Referencia	I [mA]	Referencia	V [v]	Referencia	I [mA]
VA	4.95	IA	0.0482	VA	4.98	IA	0.009
VB	0.77	IB	0.3	VB	0.075	IB	0.0009
VC	0.75	IC	0.0078	VC	0.054	IC	0.0007
VD	0.20	ID	0.0066	VD	4.98	ID	0.0006
VE	0.20	IE	0.328	VE	4.97	IE	0
VF	4.51	IF	0.29	VF	0.12	IF	0
VG	4.50	IG	99.6	VG	0.12	IG	63.4
VH	9.52	IH	110.54	VH	10.2	IH	69.85
VI	1.19	II	19.7	VI	1.2	II	0
VJ	3.73	IJ	19.6	VJ	4.97	IJ	0
VK	1.12	IK	0.32	VK	1.16	IK	0.0045
VL	4.00	IL	26	VL	4.98	IL	26.4
		IM	0.42			IM	0.006
		IN	25			IN	25.32
		IO	0.22			IO	0
		IP	0.178			IP	0
		IQ	0.0004			IQ	0
		IR	0.0004			IR	0

Tabla 4.2.3.- Resultados experimentales obtenidos en el interfaz de “Plano Inclinado”.



De la tabla de resultados experimentales, se observa lo siguiente:

- Los puntos de referencia son los mismos expuestos en "Caída Libre".
- El voltaje de alimentación regulado es ahora de 4.95 voltios.
- El consumo máximo del circuito electrónico total es de 110.54 mA, cuando hay luz sobre los dos sensores. De igual forma, este valor de corriente es el demandado por dos circuitos sensores, conectados en paralelo a la fuente alimentación.
- Los voltajes de los colectores de los BJT ( $V_D$ ) y ( $V_E$ ), son niveles bajos cuando los sensores reciben luz, y niveles altos cuando éstos se encuentran apagados. Los valores obtenidos de 0.2 voltios, corresponden a los valores ubicados cuando se calibró el equipo.
- De la misma forma, los voltajes de salida de los "Schmitt Trigger" ( $V_F$ ) y ( $V_G$ ) son concordantes, pues indican un nivel alto cuando los sensores están iluminados; y un nivel bajo, cuando los sensores se encuentran en condiciones de oscuridad. Los niveles lógicos resultantes son totalmente reconocidos por el computador como niveles TTL.
- Las corrientes que ingresan al computador ( $I_Q$ ) y ( $I_R$ ), en el peor de los casos tienen un valor de 0.4  $\mu A$ , por lo tanto el computador no puede sufrir daños durante la conexión del circuito interfaz al puerto paralelo.
- Las corrientes que circulan a través de los diodos LED ( $I_N$ ) y ( $I_L$ ), tienen un valor de 25.0 mA y 26 mA respectivamente, similar a lo calculado en el diseño electrónico. Por tanto, se asegura que los diodos trabajen adecuadamente.



### 4.3 PRUEBAS DE LOS INTERFACES ELECTRÓNICOS MEDIANTE EL USO DE RUTINAS DE PROGRAMA (SOFTWARE).

Para la realización de las pruebas de los circuitos interfaces se ha utilizado un programa creado en QBASIC, el cual, utiliza el puerto paralelo como puerto de comunicaciones con el exterior.

Este programa, mediante las líneas de programa 220 a 245, realiza previamente un estudio del puerto que usa el computador,. Una vez que el computador reconoce el puerto, procede con la lectura de los pines de entrada, esto es, toma los bits de los registros de entrada (BASE+1 y BASE+2). En el caso del registro (BASE+1) solo se toman los 4 bits más significativos, mientras que para el registro de entrada (BASE+2), se utilizan los 4 bits menos significativos. Para hacer que el registro (BASE+2) sea utilizado como entrada, es necesario programar los bits de ese registro con niveles altos, para esto se utiliza la línea de programa 740.

Se denominan “**high nibble**” y “**low nibble**”, a los conjuntos de bits más significativos y menos significativos, que conforman los dos registros de entrada (BASE+1) y (BASE+2), respectivamente, tal como se muestran en las líneas 760 a 790. El programa efectúa una operación lógica OR entre los dos registros de acuerdo a la línea 800, consiguiéndose de esta forma habilitar un byte de datos (8 bits), que pueden ser utilizados para la entrada de datos desde el exterior.

Finalmente, se muestra en la pantalla el byte de entrada, empezando por el bit menos significativo (D0) hasta el bit más significativo (D7); y los respectivos estados actuales “**HIGH**” o “**LOW**”, además la equivalencia decimal del byte de



entrada y el valor decimal de la dirección (BASE) en uso, correspondiente al puerto paralelo LPT1, (véase figura 4.5).

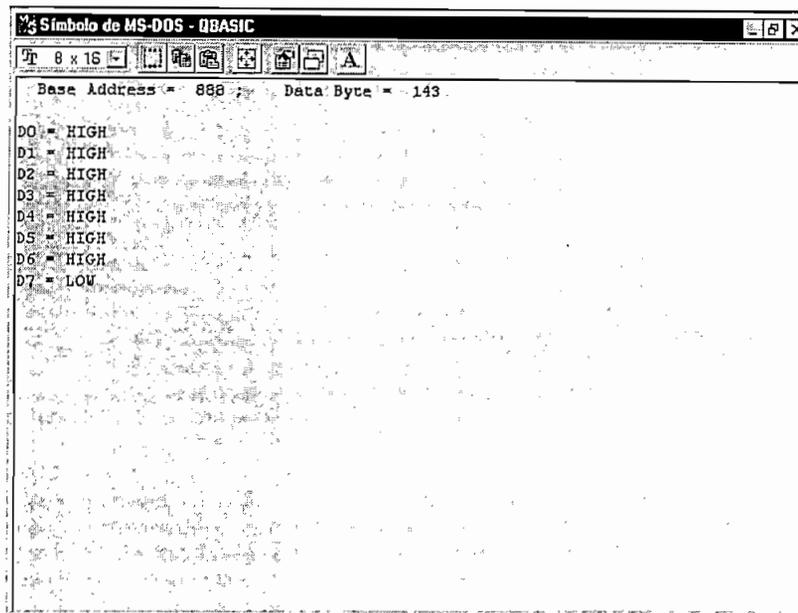


Figura 4.5.- Ejecución del programa en QBASIC para monitoreo del puerto paralelo.

Nótese que cuando se conectan los interfaces al puerto paralelo del computador y puesto que el bit (D7) "Busy" es un bit complementado en el puerto, el programa tendrá éste bit en estado de **LOW**, a diferencia del bit (D5) "Paper End", el cual se encuentra en estado de **HIGH**. El programa muestra todo el tiempo el byte de entrada y terminará su exposición cuando se presione alguna tecla.

**NOTA 1:** Únicamente para este proyecto, se tiene interés en el funcionamiento de los bits (11) y (12), que corresponden a "busy" y "paper end", respectivamente, puesto que se controlan dos sensores, los cuales han sido conectados a estos dos



pinos del puerto paralelo. Los estados lógicos de los bits restantes del octeto no tienen significado, puesto que, los mismos no tienen relación con alguna aplicación, e inclusive, existe la posibilidad de que éstos estados se encuentren oscilando en la pantalla debido a una posible autoinducción de los bits utilizados en el puerto.

NOTA 2: Este programa se ha utilizado para realizar análisis del tiempo de respuesta de los sensores, además de servir como medio de calibración de los circuitos diseñados. Similar lógica se ha empleado en el diseño del paquete computacional que se utiliza en este proyecto.

Cuando se realiza la calibración de los circuitos electrónicos, se observa la variación en el estado del bit que corresponde al sensor utilizado. Normalmente se deberá partir del hecho de que, en condiciones de iluminación, los sensores relacionados con el bit (D7) y (D5), “Busy” y “Paper End”, deberán estar con los estados **LOW** y **HIGH**, respectivamente. En el caso de no estarlo, se recurre al movimiento de la perilla de calibración propia de cada sensor, moviéndola hasta conseguir ubicar los estados normales antes mencionados. Así el circuito electrónico estará calibrado.

### LISTADO DEL PROGRAMA DE PRUEBAS EN “QBASIC”

```
1 REM
2 REM FILE: SW_IN.BAS
3 REM
100 REM basic 8-bit switch input from parallel printer port
105 REM
110 REM input BUSY, ACK, PAPER END, INPUT, STROBE, AUTO FEED, ERROR,
120 REM and SELECT INPUT signals from IBM type parallel printer port.
130 REM
140 REM written in QBASIC
150 LPT = 0: BaseAddress = 0
160 DataByte = 0: HighNibble = 0: LowNibble = 0: OffSet = 0
170 REM
```



```

200 REM -=[ begin program ]=-
210 CLS 0
220 REM find base address of LPT1, use 10 for LPT2, and 12 for LPT3
230 BaseAddress = &H400 + 8
240 DEF SEG = 0
245 LPT = PEEK(BaseAddress) + (PEEK(BaseAddress + 1) * 256)
250 REM
260 GOSUB 700: REM read switches
270 GOSUB 860: REM display data
280 IF INKEYS = "" THEN GOTO 250
290 END: REM -=[ program end ]=-
300 REM
310 REM
699 REM
700 REM -=[ Subroutine, Read Switches ]=-
710 REM
720 HighNibble = 0: LowNibble = 0
730 REM first set all open collector output pins HIGH to read from base + 2
735 REM
740 OUT (LPT + 2), (INP(LPT + 2) OR 15)
750 REM FOR E = 1 TO 50: NEXT E: REM delay dependent on your machine
760 LowNibble = INP(LPT + 2)
770 HighNibble = INP(LPT + 1)
780 LowNibble = (LowNibble AND &HF)
790 HighNibble = (HighNibble AND &HF0)
800 DataByte = (HighNibble OR LowNibble): REM combine nibbles into byte
810 RETURN: REM end of subroutine, read switches
850 REM
860 REM -=[ subroutine, display data ]=-
870 REM
880 LOCATE 1, 3
890 PRINT "Base Address = "; LPT; "; Data Byte = "; DataByte; " "
895 PRINT
900 PRINT "D0 = ";
902 IF (DataByte AND 1) = 1 THEN PRINT "HIGH" ELSE PRINT "LOW "
904 PRINT "D1 = ";
906 IF (DataByte AND 2) = 2 THEN PRINT "HIGH" ELSE PRINT "LOW "
908 PRINT "D2 = ";
910 IF (DataByte AND 4) = 4 THEN PRINT "HIGH" ELSE PRINT "LOW "
912 PRINT "D3 = ";
914 IF (DataByte AND 8) = 8 THEN PRINT "HIGH" ELSE PRINT "LOW "
916 PRINT "D4 = ";
918 IF (DataByte AND 16) = 16 THEN PRINT "HIGH" ELSE PRINT "LOW "
920 PRINT "D5 = ";
922 IF (DataByte AND 32) = 32 THEN PRINT "HIGH" ELSE PRINT "LOW "
924 PRINT "D6 = ";
926 IF (DataByte AND 64) = 64 THEN PRINT "HIGH" ELSE PRINT "LOW "
928 PRINT "D7 = ";
930 IF (DataByte AND 128) = 128 THEN PRINT "HIGH" ELSE PRINT "LOW "
932 RETURN: REM end of subroutine, display data
934 SYSTEM

```



#### 4.4 CONCLUSIONES Y RECOMENDACIONES.

Luego de la realización de éste proyecto, se pueden detallar las siguientes conclusiones generales, las cuales fueron muy trascendentales para la consecución de los objetivos de este proyecto:

1. Es muy relativo el uso del puerto paralelo para comunicaciones con el exterior, y dependerá de la velocidad de procesamiento de información requerida, de la distancia del cable usado para la interconexión de los dispositivos con el computador, del tipo de aplicación a conectarse al computador, etc. Ultimamente, muchos experimentadores aficionados de la electrónica, interactúan con el puerto paralelo, y, en el mercado existe una gran información bibliográfica que trata de temas relacionados con este puerto de comunicaciones.
2. No siempre el primer diseño realizado es la solución definitiva en la obtención del objetivo propuesto. En este caso, se tuvo que cambiar el sistema de iluminación del proyecto debido a que el circuito inicial con luz visible, mostraba deficiencias de estabilidad y comportamiento, frente al circuito con luz infrarroja. Por otro lado, el uso de las lámparas también fue un limitante por el tiempo de vida útil que presentaban y por la necesidad de una fuente de energía alternativa (baterías) para alimentarlas. Con el montaje de los diodos LED, los gastos de operación de los prototipos son totalmente reducidos, pues sólo se necesita utilizar la misma fuente del circuito interfaz.
3. Al realizar las primeras pruebas de los prototipos con luz visible, el circuito interfaz de "Péndulo Simple", funcionaba regularmente. Con el tiempo de uso



del equipo, las lámparas iban envejeciendo y por último se dañaban. Este envejecimiento producía que el equipo pierda paulatinamente sensibilidad. En el caso de los circuitos de “Caída Libre” y “Plano Inclinado” se notaba una descalibración continua de los interfaces y fallas en la estabilidad de los mismos.

4. El trabajo de los fototransistores es mucho más efectivo con luz infrarroja puesto que la respuesta de estos no depende de las condiciones de luz externas del ambiente.
5. Fue necesario realizar cambios físicos en los circuitos impresos de cada interfaz electrónico. En algunos casos, se realizaron perforaciones adicionales en la tarjeta para albergar elementos adicionales, requeridos para la adaptación del nuevo sistema de luz infrarroja. Debido a esto, se procedió a rediseñar las tarjetas de circuito impreso para los tres prototipos.
6. La obtención de los datos experimentales, esto es, valores de corriente y voltaje, se lo realizó en los puntos más importantes de los circuitos interfaces de cada prototipo. Los resultados presentados en las tablas 4.2.1, 4.2.2 y 4.2.3, pags. 146, 149 y 150, respectivamente, indican valores conseguidos cuando los interfaces están calibrados y conectados al puerto paralelo del computador.
7. En el proceso de calibración de los circuitos interfaces, es importante obtener alrededor de 0.2 voltios en la entrada del “Schmitt Trigger”. De esta forma, se asegura un disparo rápido desde un nivel bajo a un nivel alto, cuando el sensor vuelve a recibir luz, habiendo permanecido en oscuridad. Esto se explica con el hecho de que el circuito integrado 74LS14, según datos del fabricante, tiene un



voltaje de umbral mínimo (**VH-**) de 0.6 voltios. Así, es necesario que el nivel bajo obtenido en el colector del BJT o a la entrada del disparador, sea menor a (**VH-**) para que el circuito no presente errores en la respuesta entregada y el puerto paralelo del computador reciba el dato correcto.

8. El circuito integrado 74LS14 utiliza disparadores de Schmitt complementados. Esta lógica hace que la respuesta del integrado sea igualmente complementada.
9. La evolución del lenguaje BASIC durante los últimos 15 años ha tendido a la programación enfocada a objetos. Sin embargo, aunque muchas ventajas han surgido en cuanto a facilidades en la programación, no ha dejado de ser cierto que en la programación gráfica actual, se han perdido las opciones de control del puerto paralelo en forma directa. Tal es el caso de las versiones de Visual BASIC, las cuales requieren de ayudas del tipo **[.DLL]** (*Dynamic Link Library*, "librerías de enlace dinámico") u **[.OCX]** (*OLE Control, Object Linking and embedding Control* ó "Control de Incrustación y Enlace de Objetos"), para poder manejar el puerto paralelo. No obstante, en este proyecto se ha planteado como posible alternativa, el uso de rutinas creadas en lenguaje C. Esto implica que en ciertos momentos, Visual BASIC pasa el control del programa al lenguaje C, se da lectura directamente al puerto, y finalmente se devuelve el control al Visual BASIC, corriéndose el riesgo de perder los datos en la recuperación del mando por parte del programa fuente. Con esta situación, se eligió el uso de una herramienta para Visual BASIC, la cual fue bajada desde una dirección del Internet (véase *BIBLIOGRAFIA*, referencia #27) y es una herramienta destinada al manejo del puerto paralelo sin tener que abandonar el



control del programa por parte del lenguaje gráfico. El nombre de esta herramienta es **IOPORT.OCX**.

**NOTA:** [DLL].- Función o funciones compiladas a 32 bits con características de código reutilizable. Se puede utilizar en varios programas. (Funciones).

[OCX].- Objeto compilado a 32 bits con característica de código reutilizable en binario. Consiste en una aplicación completa.

10. **IOPORT.OCX**, permite al Visual BASIC tener un control del puerto paralelo en forma directa. Con esta opción, se tiene la posibilidad de configurar el puerto paralelo tanto para entrada, como para salida de datos, teniéndose para ambos casos 8 bits disponibles al usuario.

11. El programa principal del proyecto generó inconvenientes al momento de probarse con el prototipo de “Caída Libre”, puesto que el computador no detectaba la rapidez del paso del cuerpo a través de los dos sensores, durante la caída. Luego del análisis respectivo, se indagó en la rutina de lectura del puerto (rutina general para la lectura del puerto paralelo en los tres prototipos), y se detectó que existían varias líneas de comandos que le tomaban al microprocesador demasiado tiempo de ejecución, entonces la detección del paso del cuerpo al cruzar el sensor inferior, era ineficiente. La solución consistió en ubicar una rutina propia de lectura del puerto para la ventana de “Caída Libre”, con lo que se consigue mejorar la detección del cuerpo en la parte inferior del equipo.



12. En el análisis de los tres fenómenos físicos, el computador actúa como un cronómetro, es decir, se toma el tiempo de duración del movimiento específico y se lo almacena como una variable  $t$ , la cual, es utilizada en las ecuaciones del movimiento.
13. Dentro de los cálculos efectuados por el programa principal, existen los cálculos de error. Se ha visto conveniente mostrar este tipo de resultados puesto que, al medirse variables físicas, se está involucrando la interacción del ser humano y los móviles no son puntuales. El mayor error considerado, es el introducido por el mismo ser humano, no así en el proceso de medición del tiempo por el computador, éste último es muy preciso.

### **RECOMENDACIONES.**

Dentro de las recomendaciones, es importante mencionar lo siguiente:

- Los equipos pueden tener mayor cantidad de sensores y así volverlos más inteligentes.
- Para “Péndulo Simple” sería óptimo pensar en un sistema móvil montado sobre el panel vertical, que permita ubicar el sensor y la fuente de luz para estudiar cualquier longitud pendular.
- En “Plano Inclinado”, se utilizó un bloque de madera con base de cuero, para soporte de elevación del plano móvil. Basta con moverlo conforme se levanta o baja el plano móvil, para que éste quede sujeto. Sin embargo, sería recomendable utilizar un sistema mecánico capaz de regular el ángulo de inclinación del plano superior (móvil), mediante el uso de un tornillo con



tuerca tipo mariposa. Este sistema vuelve cómoda la búsqueda del ángulo de inclinación adecuado para provocar el movimiento del móvil. Se recomienda adicionalmente ubicar un graduador sobre el plano fijo, y así obtener lecturas directas del ángulo de inclinación del plano móvil.

- Ampliando el alcance de la experiencia de "Plano Inclinado", se puede sugerir la obtención del coeficiente de rozamiento dinámico, el cual es relativo a dos superficies en contacto, utilizando un equipo que permita variar las superficies de rozamiento.
- El prototipo de "Caída Libre" podría disponer de un sistema que detecte la partida del cuerpo, desde que parte del reposo, hasta su llegada al suelo.
- Es posible que se utilicen otros materiales como superficies de contacto en el plano inclinado y el móvil. Así se experimentaría el movimiento en el plano inclinado con otros resultados.
- Conforme el estudiante realiza la práctica experimental, es recomendable que otra persona realice los cálculos utilizando las ecuaciones del movimiento respectivo. El objetivo de esto, es llegar a determinar los errores sucedidos y donde se generaron, es decir, discutir el problema teniéndose bases teóricas y experimentales.
- Debido a que EFC (Experimentos de Física en Computador) fue diseñado en Visual Basic v5.0, las pruebas de instalación realizadas sobre computadores con Windows 95(B) o 98, fueron satisfactorias. Sin embargo, sobre versiones anteriores, la instalación no se pudo realizar puesto que ciertas librerías del sistema operativo requerían ser actualizadas. Para esto, se recurre a la



utilización de parches del sistema operativo, que permiten que el programa pueda ser instalado y ejecutado satisfactoriamente. Además, es necesario que se cumplan los siguientes requerimientos de hardware adicionales:

1. 16 Mb de RAM.
  2. 20 Mb libres en disco duro.
  3. Monitor VGA o SVGA con capacidad de resolución a 256 colores.
- En un futuro podría ampliarse el alcance de este proyecto al estudio de otros fenómenos físicos, y también a otras ramas de la ciencia, por ejemplo: Biología, Química, Relatividad, Matemática, etc. Actualmente ya existen casas comerciales Americanas que se dedican a la fabricación de equipos para la interacción de las ciencias con el computador. En éstas, se construyen interfaces que permiten la comunicación del mundo analógico con el digital.



## CAPITULO V

### MANUAL DEL USUARIO.

#### 5.1 OPERACIÓN DE LOS PROTOTIPOS POR EL USUARIO.

##### 5.1.1 PENDULO SIMPLE.

El prototipo “Péndulo Simple” consiste en una caja de acrílico transparente, que contiene el circuito de interfaz hacia el computador, y un sistema físico diseñado para la demostración del fenómeno en estudio. En la *figura 5.1*, se muestra la ubicación del interfaz en el sistema general.

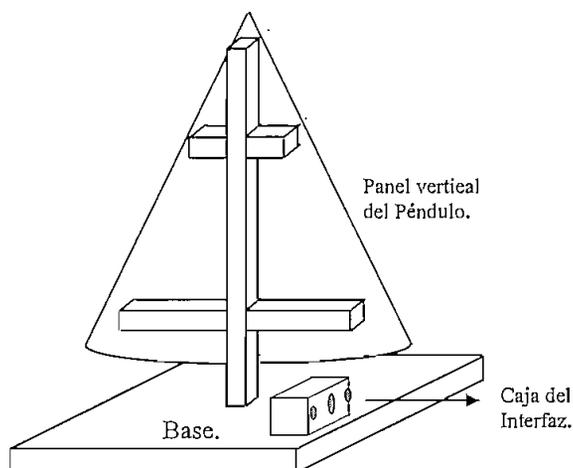


Figura 5.1.- Ubicación de la caja interfaz en el módulo físico.

En la parte frontal de la caja, se encuentran los interruptores que controlan la alimentación, tanto para el circuito impreso, como para los sensores y los diodos LED infrarrojos. Adicionalmente se tiene un potenciómetro de control por perilla, puesto específicamente para la calibración del circuito.



En “Péndulo Simple”, el interruptor S1 (figura 5.2), es el dispositivo que comanda la activación individual de los sensores y los respectivos diodos LED. Estos LED son los dispositivos que proveen luz infrarroja a los sensores. S1 se encuentra situado al lado izquierdo de la parte frontal de la caja interfaz, y opera de la siguiente manera:

1. Interruptor S1 en *posición izquierda*, acciona el fototransistor superior y también el encendido del LED superior.
2. Interruptor S1 en la *posición derecha*, acciona el fototransistor inferior y también el encendido del LED inferior.
3. Interruptor S1 en la *posición central*, significa que el circuito está en modo de espera (*Stand-By*), tanto para los sensores como para los diodos LED.

El interruptor S2 controla el encendido y apagado (ON/OFF) de todo el circuito, y está situado en la parte frontal derecha de la caja. El usuario puede verificar que S2 está señalizado, es decir, que se indica la posición del estado de encendido (ON) y del estado de apagado (OFF) del circuito general.

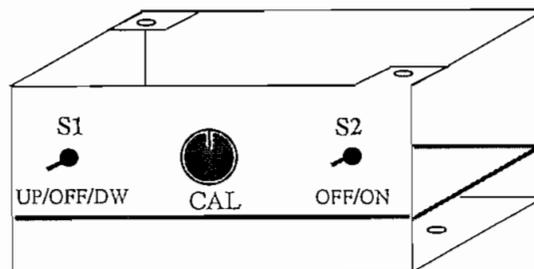


Figura 5.2.- Dispositivos de control del circuito interfaz de “Péndulo Simple”.



En la parte central del frente de la caja interfaz, se ubica el potenciómetro, que sirve como dispositivo de calibración (**CAL**) de los dos sensores, basta que se calibre uno de ellos, para que el otro se encuentre en las mismas condiciones. La *figura 5.2, pag. 164*, muestra la ubicación de los dispositivos de control en la caja del interfaz.

En la parte posterior y en el centro de la caja interfaz, se sitúa el cable de datos hacia el computador (**COMP**), el mismo que está configurado como:

**PIN #1:** NC.- No existe conexión.

**PIN #2:** Cable verde .- Al pin #11 (**Busy**) del conector DB-25 H del computador.

**PIN #3:** Cable negro .- Conexión a tierra, que corresponde a los pines 18-25 (**Gnd a-h**) del conector DB-25 H del computador.

**NOTA:** Los números a los que corresponden los pines del cable de datos, están identificados en el conector “macho” que se inserta en la placa del circuito impreso.

Al lado izquierdo, en la parte posterior de la caja, se encuentra instalado un terminal tipo “jack” para la conexión de la alimentación del sistema, proporcionado por el adaptador universal de voltaje. Este tipo de conectores tienen la facilidad de ser instalados exteriormente, es decir, mediante una tuerca de sujeción, se fijan en el panel posterior de la caja.

Finalmente, dos perforaciones conjuntas permiten la salida de los cables de conexión de los diodos LED infrarrojos y de los sensores (fototransistores) hacia el exterior. Estos cables son dirigidos a los puntos de análisis del péndulo, es decir, dos



pares de cables se dirigen hacia los diodos o LED's de iluminación y dos pares hacia los sensores, respectivamente. La *figura 5.3*, muestra la parte posterior de la caja interfaz.

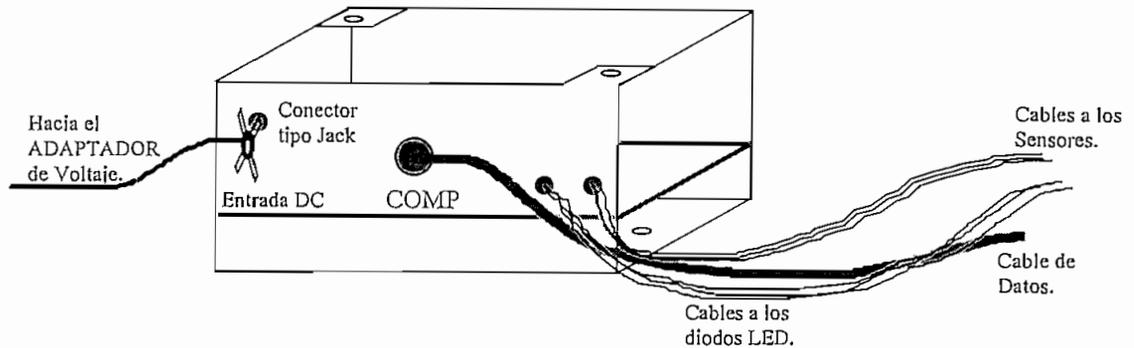


Figura 5.3.- Conexiones en la parte posterior de la caja interfaz.

El interfaz "Péndulo Simple" trabajará correctamente cuando haya sido totalmente calibrado, es decir cuando los fototransistores tengan buena sensibilidad. En este proyecto, se plantean dos maneras básicas de realizar un proceso de calibración: Una consiste en remover la tapa de la caja interfaz y, mediante el uso de un multímetro, evaluar el pin #1 del CI 74LS14, instalado en la placa. Así, si el sensor se encuentra iluminado por el correspondiente diodo LED, la respuesta de este pin deberá ser de un valor fijo de 0.2 voltios, de lo contrario se ajusta la perilla del potenciómetro hasta obtener dicho valor (*ver apartado 5.2, pag. 191*). La otra forma de calibrar el sistema, consiste en utilizar el programa computacional del proyecto



(ver apartado 5.4, pag. 209), en donde se define un icono gráfico, que indica el estado de sensibilidad del sensor. Este icono se muestra en pantalla cuando se ha seleccionado la opción **CALIBRACION** del programa computacional, e indica por ejemplo, mediante un foco iluminando, cuando el sensor se encuentra destapado; y con un foco apagado, cuando el sensor ha sido tapado. De igual forma, se tendrá que modificar la posición de la perilla del potenciómetro para lograr que el icono responda a las variaciones de la excitación luminosa que recibe el sensor. Para esto, el equipo deberá estar conectado necesariamente al computador. Mayores referencias de este procedimiento, se encuentran en los apartados 5.2.1 y 5.4, pags. 191 y 209, respectivamente.

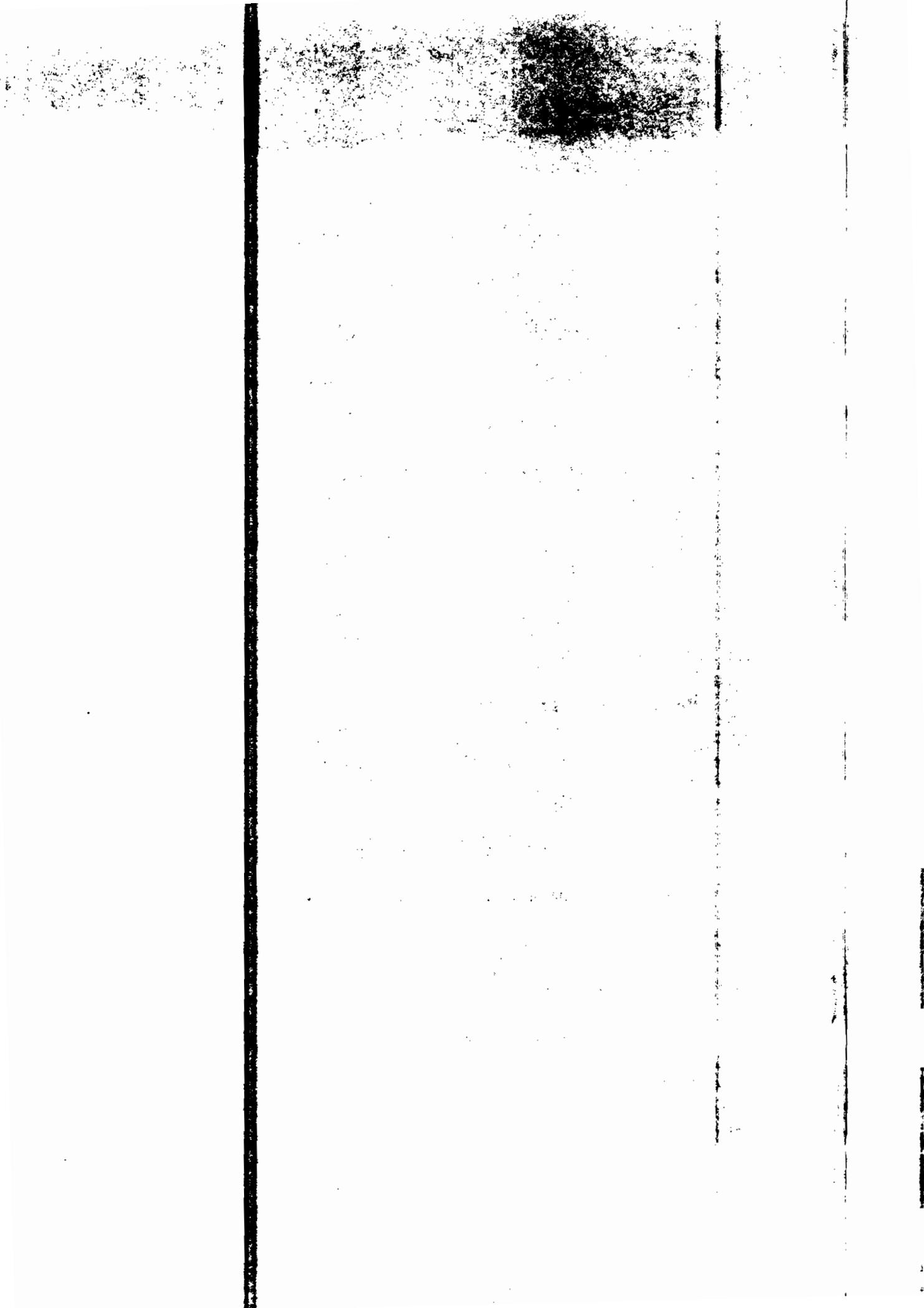
### 5.1.2 PROCEDIMIENTOS PARA REALIZAR LA PRACTICA DE “PENDULO SIMPLE”.

El procedimiento para una práctica, utilizando el prototipo “Péndulo Simple”, se detalla a continuación:

1. Verificar que el equipo esté activado. Si no lo está, proceder con la conexión de la alimentación del sistema, insertando el conector “macho” del adaptador universal, en el terminal tipo “jack” ubicado en la parte posterior de la caja.

**NOTA:** Es importante no cambiar la posición del interruptor incluido en el adaptador de voltaje, pues dicha posición permite que la fuente proporcione aproximadamente 9 voltios al circuito interfaz.

2. Accionar el interruptor principal (**MAIN**) en la posición de encendido (**ON**).



3. Verificar que el cable de datos esté correctamente conectado al puerto paralelo del computador.
4. En el caso de que el equipo haya sido previamente calibrado, pasar al numeral 5. De otro modo, proceder con la opción de calibración, ya sea directamente utilizando un equipo de medición (*multímetro*), o seleccionando dentro del programa computacional, la opción de *Calibración* del sistema.
5. Elegir, en el menú de selección principal, la opción *Práctica* y seleccionar *Péndulo Simple*.
6. Elegir, mediante el accionamiento del interruptor S1 (*UP/OFF/DW*), el sensor superior o inferior que se utilizará en la práctica, optando así por el uso de una longitud de cuerda corta (**I**) o larga (**L**), respectivamente.
7. Insertar los datos solicitados por el computador. En la práctica se puede seleccionar la masa grande (**M**) o la masa pequeña (**m**), así como también el tipo de longitud de cuerda a utilizarse. Para el uso correcto del programa computacional es necesario la revisión del apartado 5.4, pag. 198.
8. Hacer click en **INICIAR**. El computador mostrará en la pantalla un mensaje indicando que se proceda con la colocación de la masa pendular en estudio sobre la posición del sensor, es decir, la masa deberá fijarse interrumpiendo el haz infrarrojo que incide sobre el sensor. Este haz no es visible para el ser humano, pues la luz infrarroja emitida por el LED, tiene una longitud de onda  $\lambda = 950 \text{ nm}$ , y no es captada por el ojo humano, sin embargo basta con situar la masa pendular



entre el hueco que corresponde al sensor, ubicado en el panel vertical del prototipo, y el diodo emisor de luz (LED).

- Una vez ubicada la masa sobre el sensor, el usuario debe indicar al computador que está listo para comenzar con la experiencia. Para esto, se hace click en el botón **ACEPTAR** del mensaje que aparece en la pantalla (*ver punto 8*). Inmediatamente la computadora empieza a estudiar el puerto, y el valor que entrega el interfaz, mientras el sensor está tapado, es un valor de **0L**. Todo esto sucede hasta que el operador suelta la masa para comenzar las oscilaciones, entonces se rompe la condición de oscuridad y el sensor recibe luz otra vez. Para ese momento el interfaz entrega al computador un valor de **1L**. Mientras el péndulo realiza su primera oscilación en un tiempo **T1**, el interfaz continúa entregando al puerto un valor de **1L**. El tiempo de oscilación se mide en el momento que el computador detecta una transición positiva en el puerto de entrada, es decir, un cambio de nivel de **0L** a **1L**. Para ello, el programa arranca un temporizador interno (**timer**), cuya medición terminará cuando la masa pendular vuelva a tapar el sensor, es decir, cuando el péndulo regresa a su posición inicial, luego de cumplir con la primera oscilación y volviendo a las condiciones iniciales de oscuridad **0L**. La *figura 5.4*, muestra la lógica utilizada en la medición del periodo de una oscilación.

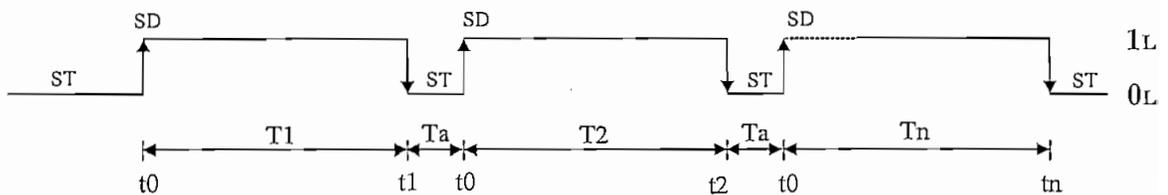
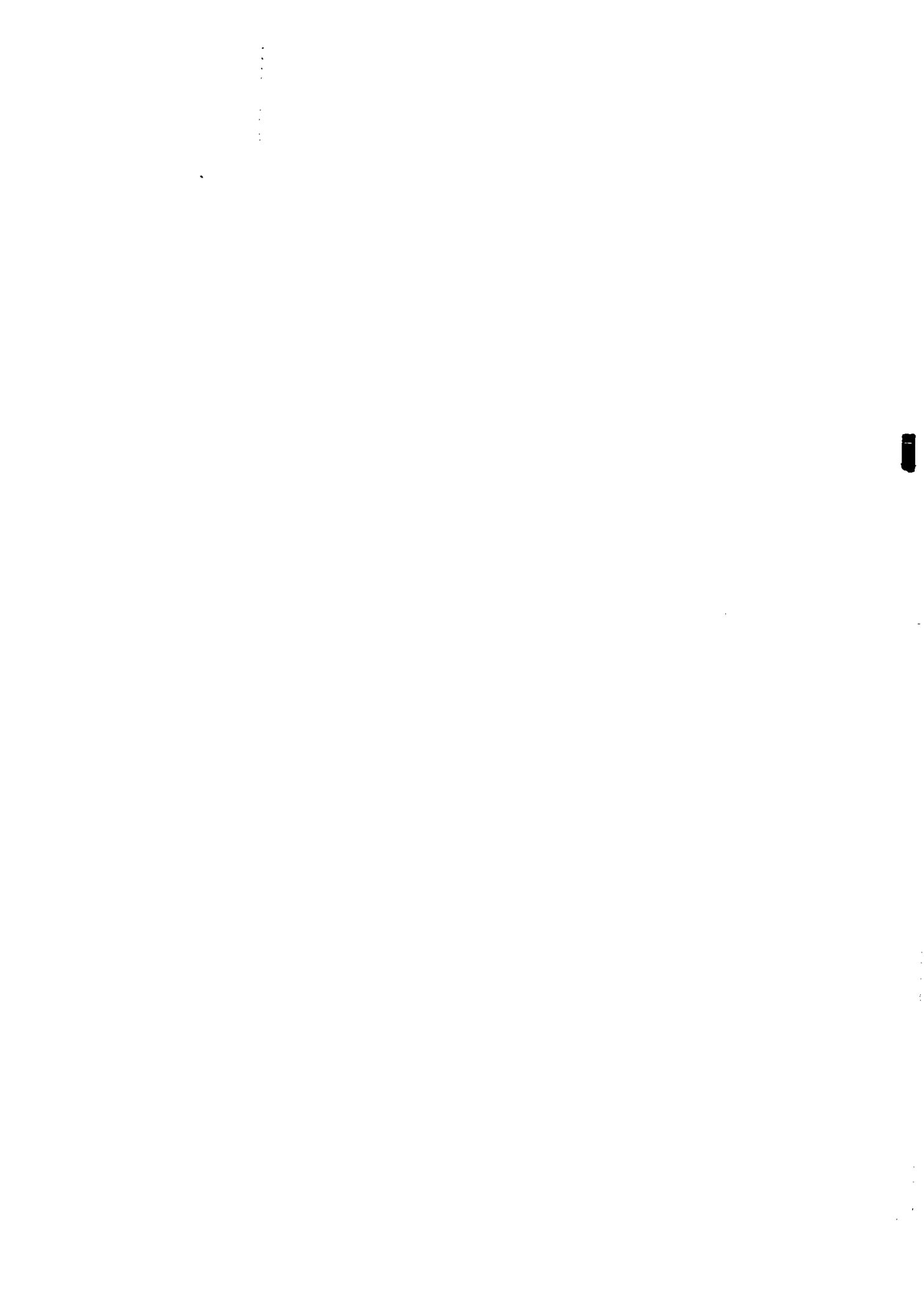


Figura 5.4.- Forma de onda en el pin de entrada al puerto. Medición del tiempo de oscilación.



ST = Sensor tapado (0L.).

SD = Sensor destapado (1L.).

T1 = Duración de la primera oscilación pendular en segundos [s].  $T1 = t1 - t0$

t0 = Tiempo actual tomado del reloj interno en seg [s], cuando el sensor fue destapado y empieza la primera oscilación.

t1 = Tiempo actual tomado del reloj interno en seg [s], cuando el sensor fue tapado por primera vez, luego de cumplirse la primera oscilación.

T2 = Duración de la segunda oscilación pendular en segundos [s].  $T2 = t2 - t0$

t2 = Tiempo actual tomado del reloj interno en seg [s], cuando el sensor fue tapado por segunda vez, luego de cumplirse la segunda oscilación.

Tn = Duración de la n-ésima oscilación pendular en segundos [s].  $Tn = tn - t0$

tn = Tiempo actual tomado del reloj interno en seg [s], cuando el sensor fue tapado por n-ésima vez, luego de cumplirse la n-ésima oscilación.

Ta = Tiempo de apagado cuando la masa pendular se encuentra cruzando por el sensor. El valor de Ta es relativo a las dimensiones de la masa en estudio. A mayores dimensiones de la masa pendular, mayor es el tiempo Ta.

10. Mientras el programa realiza los cálculos pertinentes, es recomendable que el usuario realice por separado el mismo estudio, aplicando las ecuaciones del movimiento. Así se enriquece la experiencia mediante conclusiones más acertadas, al momento de establecerse comparaciones entre los resultados prácticos y los teóricos.

11. El programa mostrará en la matriz de resultados, todos los valores obtenidos de acuerdo al número de oscilaciones pendulares producidas ( $T1, T2, \dots, Tn$ ). Para la



longitud de cuerda (**L**), los resultados se muestran en la matriz izquierda de la pantalla del monitor; mientras que para la longitud de cuerda (**l**), la matriz de resultados es la derecha. Se calcula un periodo de oscilación promedio **T<sub>p</sub>**, y con este valor se obtiene la aceleración de la gravedad experimental (**g<sub>exp</sub>**), valor que será comparado con el valor de gravedad teórica o conocida (**g<sub>teo</sub>**), es decir, el valor proporcionado en la bibliografía. Se aplica la siguiente ecuación matemática:

$$\mathbf{g_{exp} = (2\pi/T_p)^2 \cdot L} \quad (1)$$

Donde:

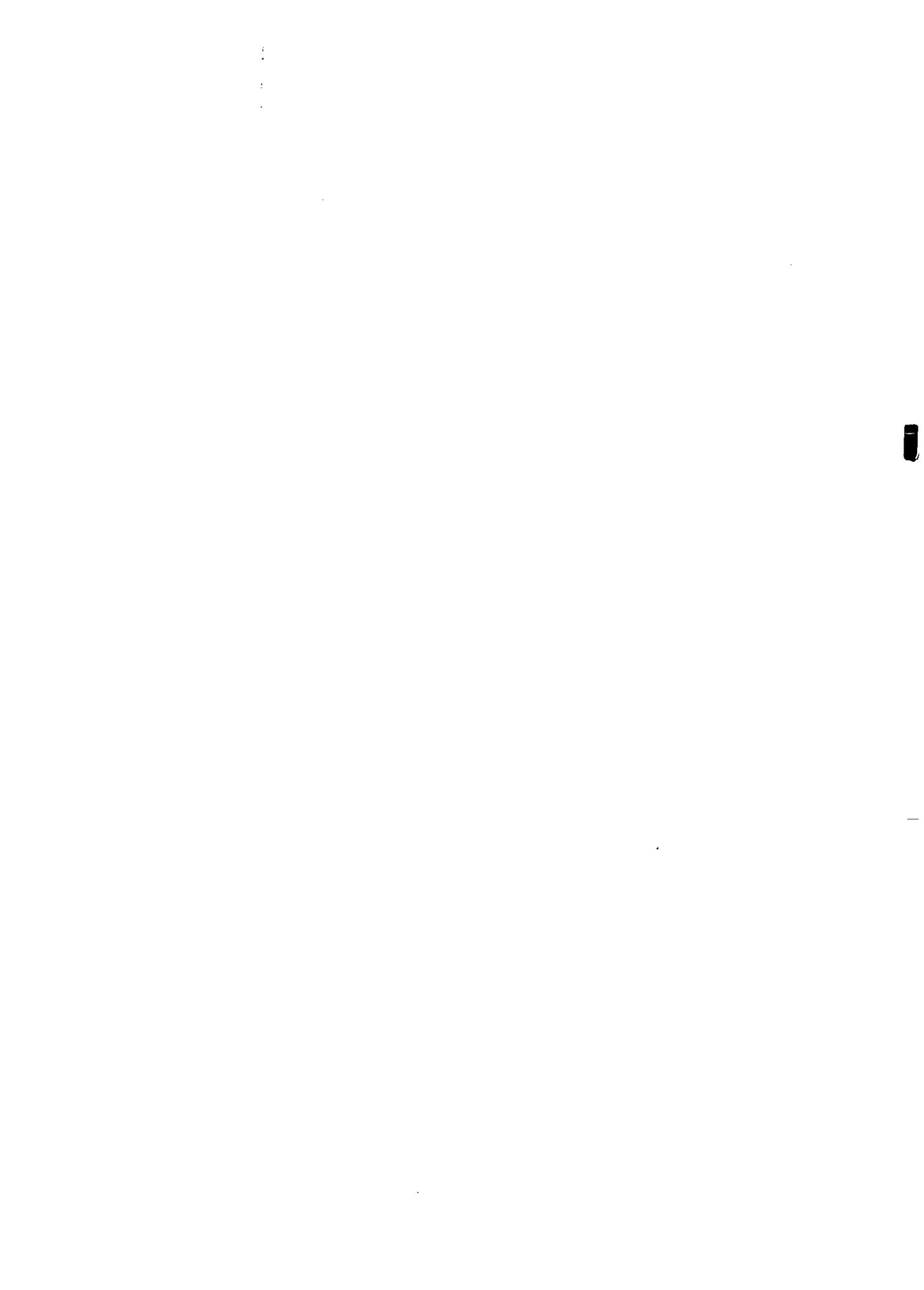
**T<sub>p</sub>** = Periodo de oscilación promedio.

**L** = Longitud de cuerda del péndulo. Pequeña(**l**), Grande(**L**)

**g<sub>exp</sub>** = Valor de la aceleración de la gravedad calculada (experimental)

12. Finalmente, se obtiene el error generado en el cálculo de (**g**), mediante la comparación del valor obtenido experimentalmente (*ecuación #1*), con el valor teórico de **g** puesto inicialmente como dato en el programa computacional (ver *paso 7, pag. 168*). El programa calcula el error producido, mediante la siguiente ecuación:

$$\mathbf{Err_g = | g_{exp} - g_{teo} | / g_{teo} * 100\%} \quad (2)$$



13. En la pantalla de resultados de “Péndulo Simple”, el usuario tiene acceso a los iconos *Borrar Resultados* e *Iniciar*, los cuales permiten el reseteo de la matriz de resultados, dejándola en blanco para reiniciar la práctica, si así lo desea.
14. Se puede llegar a demostrar que el período de oscilación pendular no depende de la masa suspendida en la cuerda. Esto se puede verificar en la ecuación (1). Sería muy valioso realizar la práctica, utilizando las dos masas proporcionadas en el equipo: Masa grande (**M**), y masa pequeña (**m**). El resultado en el cálculo de los periodos de oscilaciones, no cambiará.

NOTA: Mientras se realiza la experiencia con una longitud de cuerda determinada, la frecuencia de oscilación pendular, es la misma con las dos masas, además, *“a mayor longitud de cuerda, menor es la frecuencia de oscilación del péndulo y mayor el tiempo en cumplirse la oscilación completa”*.

15. Terminada la práctica, se pueden almacenar los datos resultantes en un archivo de texto para imprimirlo posteriormente, ya que la conexión de la impresora temporalmente está utilizada. En el apartado 5.4.2.2.2, pag. 209, se explica en detalle la opción de impresión de los resultados experimentales.



### 5.1.3 CAIDA LIBRE.

El prototipo “Caída Libre”, consiste en un sistema físico compuesto por un panel vertical en forma de “U”, montado sobre una base de madera. Los sensores y diodos LED infrarrojos están ubicados en las paredes laterales del panel vertical. Se ha tenido especial cuidado en la instalación de estos elementos, pues es muy importante que tanto el sensor como el LED infrarrojo se encuentren alineados. De esta forma, se realiza adecuadamente la detección del paso de un cuerpo en caída libre. Adicionalmente, se ha dispuesto el circuito interfaz dentro de una caja de acrílico. La caja se encuentra ubicada a un lado de la base del prototipo, como se muestra en la *figura 5.5*. Así la operación de ésta, es más sencilla.

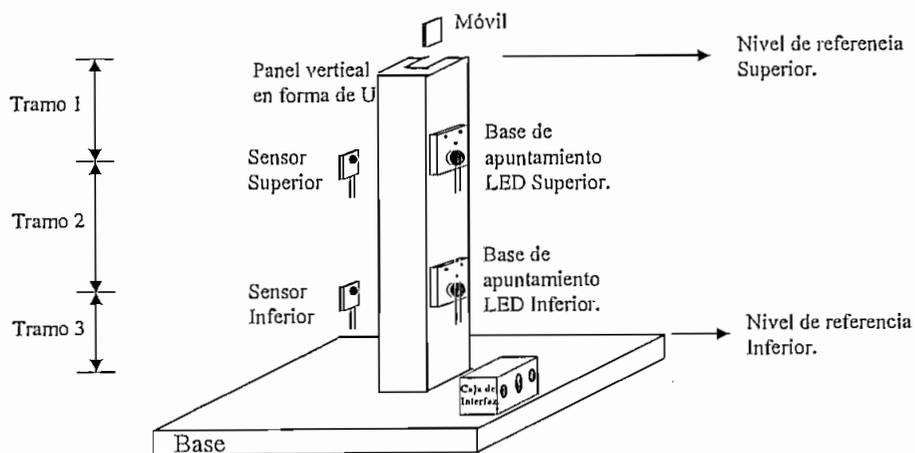


Figura 5.5.- Ubicación de la caja interfaz en el módulo físico.

Al lado izquierdo, en la parte frontal de la caja, se encuentra el interruptor S1 que maneja la alimentación, tanto para el circuito impreso total, como para los sensores y diodos LED infrarrojos, que se activan al mismo tiempo. Esto es



importante, puesto que mientras el cuerpo cae, un sensor superior estará listo para detectar el paso del móvil en el nivel alto, y un sensor inferior detectará el paso del mismo en el nivel bajo. S1 es un interruptor señalizado, es decir, que se indica la posición del estado de encendido (ON) y del estado de apagado (OFF) del circuito.

Para realizar la calibración del circuito, el interfaz dispone de dos potenciómetros montados en la parte frontal de la caja de protección. El primero, ubicado a la izquierda, calibra la sensibilidad del fototransistor superior, mientras que el segundo, ubicado a la derecha, controla la sensibilidad del fototransistor inferior. Los potenciómetros utilizados realizan la calibración de los sensores en forma independiente. (ver calibración del interfaz “Caída Libre” y “Plano Inclinado”, apartado 5.2.2, pag. 194). La figura 5.6, muestra la ubicación de los dispositivos de control en la parte frontal de la caja interfaz.

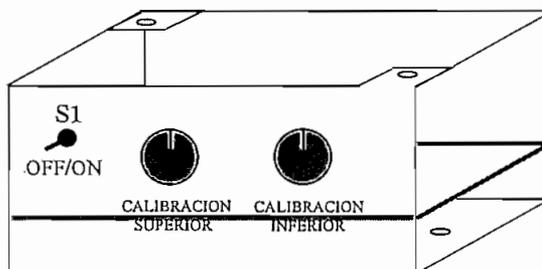


Figura 5.6.- Dispositivos de control del circuito interfaz de “Caída Libre”.

En la parte posterior central de la caja interfaz, se sitúa el cable de datos hacia el computador (COMP), y está configurado en el circuito impreso de la siguiente manera:

**PIN #1:** Cable rojo.- Al pin #12 (**Paper End**) del conector DB-25 H del computador.

**PIN #2:** Cable verde .- Al pin #11 (**Busy**) del conector DB-25 H del computador.

**PIN #3:** Cable negro .- Conexión a tierra, que corresponde a los pines 18-25 (**Gnd a-h**) del conector DB-25 H del computador.

**NOTA:** Los números que corresponden a los pines del cable de datos, están identificados en el conector insertado en la placa del circuito impreso.

En la parte posterior izquierda de la caja, se ha ubicado un terminal tipo “jack” para la conexión de la fuente de alimentación de todo el circuito electrónico. Este conector está instalado en el panel posterior de la caja, mediante una tuerca de sujeción. Aquí se inserta el conector “macho” más grande, del multiconector del adaptador universal de voltaje.

También atrás de la caja existen dos perforaciones juntas, y que conducen los cables de conexión de los diodos LED y sensores hacia el exterior. La *figura 5.7*, muestra la parte posterior de la caja interfaz.

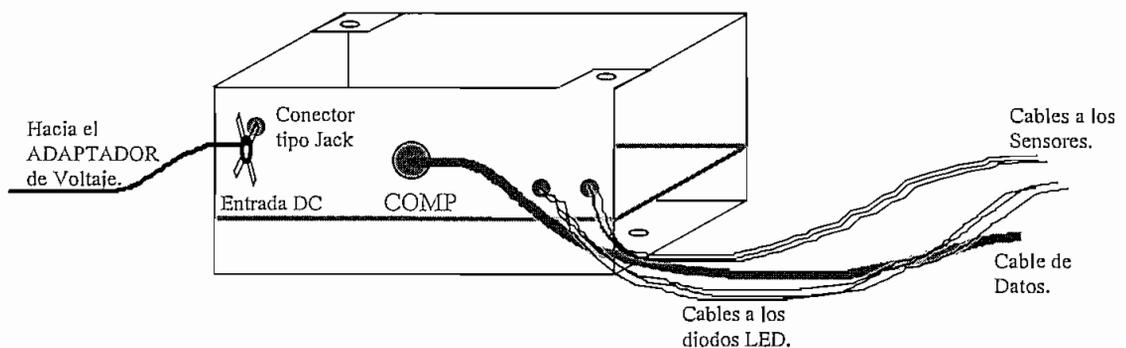


Figura 5.7.- Conexiones en la parte posterior de la caja interfaz.

Los cables de conexión a los sensores y diodos LED, son enrutados a través de los lados laterales del panel vertical en forma de “U”, y terminan conectados en los puntos de análisis del prototipo, es decir, dos pares de cables se dirigen hacia las bases de apuntamiento de los diodos LED, y, dos pares, hacia los respectivos fototransistores ubicados al frente de cada emisor infrarrojo, (ver figura 5.5, pag. 173).

**NOTA :** El sistema trabaja adecuadamente cuando se encuentra calibrado. El interfaz de “Caída Libre” se puede afinar manipulando las dos perillas frontales de la caja, mientras que, en forma paralela, se hace el monitoreo del voltaje sobre los pines #1 y #3 del circuito integrado principal (CI: 74LS14). De otra manera, se puede recurrir al programa computacional del proyecto y utilizar la herramienta de *Calibración*. En el apartado 5.2.2, pag. 194, se describe el proceso de calibración electrónica para los circuitos de “Caída Libre” y “Plano Inclinado”.

#### **5.1.4 PROCEDIMIENTOS PARA REALIZAR LA PRACTICA DE “CAIDA LIBRE”.**

El procedimiento que deberá seguirse, se detalla a continuación:

1. Verificar que el equipo esté activado. Si no lo está, proceder a conectar la alimentación, insertando el conector “macho” del adaptador universal en el terminal tipo “jack”, ubicado en la parte posterior de la caja, ver figura 5.7, pag. 175.

**NOTA:** Se sugiere no cambiar la posición del interruptor incluido en el adaptador de voltaje, pues dicha posición permite que la fuente proporcione aproximadamente 9 voltios al circuito interfaz.

2. Accionar el interruptor principal (**MAIN**) del sistema, a la posición de encendido (**ON**).
3. Verificar que el cable de datos esté correctamente conectado al puerto paralelo del computador.
4. En caso de que el equipo haya sido previamente calibrado, pasar al numeral 5. De otro modo, proceder con la opción de calibración, ya sea directamente utilizando un equipo de medición (*multímetro*), o utilizando la opción de **Calibración**, incluida en el programa principal.
5. Seleccionar dentro del programa, en el menú principal, la opción *Práctica*, luego elegir *Caída Libre*.
6. Insertar los datos solicitados por el computador, esto es: Distancias ( $\Delta x$ ) de los tramos 1, 2, y 3, valor de la aceleración de la gravedad (**g**). El programa automáticamente reconoce que el móvil cae desde el reposo, es decir, con ( $V_0 = 0$ ), y realiza los cálculos con este valor de velocidad inicial. Si el cuerpo no es soltado desde el reposo ( $V_0 \neq 0$ ), se generarán errores en el estudio del movimiento y los resultados no serán válidos.

**NOTA:** Para la correcta utilización del programa computacional es necesario revisar el apartado 5.4, pag. 198.

7. Hacer click en **INICIAR**. El computador mostrará en la pantalla un mensaje indicando que se proceda a colocar el cuerpo de prueba, sobre la parte alta de la vertical en forma de “U” del prototipo (ver “nivel de referencia superior”, figura 5.5, pag. 173).
8. Hacer click en **ACEPTAR**. En este punto, el programa se encuentra listo para detectar al cuerpo en caída libre.
9. Para evitar errores en la experiencia, antes de efectuarse la caída, es conveniente ubicar al móvil en una posición central dentro de la “U” y que asegure que el cuerpo caerá y será detectado por los dos sensores, sin que el cuerpo golpee las paredes verticales del módulo físico.
10. El cálculo que realiza el programa se rige por las siguientes ecuaciones:

$$\Delta x = V_0 \cdot t + \frac{1}{2} \cdot g \cdot t^2 \quad (1)$$

$$V_f = V_0 + g \cdot t \quad (2)$$

$$V_f^2 = V_0^2 + 2 \cdot g \cdot \Delta x \quad (3)$$

11. En caso de soltar el móvil más abajo o más arriba del nivel de referencia superior del módulo físico, el programa estará en condiciones de calcular el error generado por desplazamiento en el primer tramo (ver figura 5.5, tramo 1, pag. 173), es decir, se obtiene el error producido durante el recorrido desde el punto donde se

suelta el cuerpo (*nivel superior*), hasta el punto de cruce por el primer sensor (superior). Para esto, el computador utiliza la *ecuación #1*, y la aplica en el segundo tramo de caída (*tramo 2*), es decir, en el tramo de recorrido entre el sensor superior y el sensor inferior. Además se toma en cuenta que:

- $\Delta x$  en el tramo 2, es un valor conocido e introducido en el punto 6.
- En la *ecuación #1*:  $t_{\text{Tramo 2}} = t2 - t1$ , donde  $t1$  es el tiempo registrado por el computador cuando el móvil cruza el sensor superior, y  $t2$  es el tiempo registrado por el computador cuando el móvil ha cruzado el sensor inferior. La diferencia ( $t_{\text{Tramo 2}}$ ), proporciona el tiempo o la duración de la caída, en el tramo 2.
- Si se despeja la variable ( $V_0$ ) de la *ecuación #1*, se obtiene la velocidad inicial del móvil en el tramo 2 y ésta es igual a la velocidad final del cuerpo en el tramo 1.

$$V_{0\text{Tramo 2}} = (\Delta x_{\text{Tramo 2}} - \frac{1}{2} \cdot g \cdot t_{\text{Tramo 2}}^2) / t \quad (4)$$

- El computador aplica la *ecuación #2* en el tramo 1, y en éste se conoce que la velocidad inicial es cero ( $V_0 = 0$ ), y la velocidad final ( $V_f$ ) es igual a la velocidad inicial ( $V_0$ ) del tramo 2. Con esto, se despeja el tiempo ( $t$ ) que corresponde al tiempo de caída del cuerpo en el tramo 1, desde que parte del nivel superior hasta que cruza por el sensor superior.

$$t_{\text{Tramo 1}} = V_{f\text{Tramo 1}} / g \quad (5)$$

- El siguiente paso consiste en obtener el desplazamiento del móvil en el tramo 1. Para esto, el programa calcula ( $\Delta x$ ) con la *ecuación #3*. De igual forma, se mantiene la condición de que el móvil parte del reposo ( $V_0 = 0$ ).

$$\Delta x_{\text{Tramo 1}} = V_{f_{\text{Tramo 1}}}^2 / 2 \cdot g \quad (6)$$

- Mediante la utilización de la *ecuación #2*, se obtiene la velocidad final en el tramo 2, a nivel del sensor inferior.

$$V_{f_{\text{Tramo 2}}} = V_{0_{\text{Tramo 2}}} + g \cdot t_{\text{Tramo 2}} \quad (7)$$

- En el tramo 3 no se dispone de un sensor que determine la llegada del móvil al suelo, por lo tanto el computador no puede contabilizar el tiempo que demora el móvil en desplazarse desde el sensor inferior hasta la base. Sin embargo, se puede aplicar la *ecuación #3* y calcular la velocidad final ( $V_f$ ) en este tramo. Como datos adicionales, se utiliza la distancia ( $\Delta x$ ) del tramo 3, valor insertado en el punto 6 y el valor de la velocidad final ( $V_f$ ) del tramo 2, la misma que corresponde a la velocidad inicial ( $V_0$ ) del móvil en el tramo 3.

$$V_{f_{\text{Tramo 3}}}^2 = V_{0_{\text{Tramo 3}}}^2 + 2 \cdot g \cdot \Delta x_{\text{Tramo 3}} \quad (8)$$

- Finalmente, el computador calcula el tiempo transcurrido de caída en el tramo 3 y luego el error generado en el desplazamiento del móvil por el tramo 1, mediante la comparación del desplazamiento calculado en la ecuación #6 ( $\Delta x_{\text{Tramo 1}}$ ), con el valor teórico de distancia del tramo 1 ( $\Delta x_{\text{teo}}$ ), el cual fue introducido en el punto 6.

$$t_{\text{Tramo 3}} = (V_{f_{\text{Tramo 3}}} - V_{o_{\text{Tramo 3}}}) / g \quad (9)$$

$$\text{Err} \Delta x_{\text{Tramo 3}} = | \Delta x_{\text{Tramo 3}} - \Delta x_{\text{teo}} | / \Delta x_{\text{teo}} * 100\% \quad (10)$$

12. Mientras el programa ejecuta los cálculos pertinentes, es recomendable que el usuario realice por separado el mismo estudio, aplicando las ecuaciones del movimiento. Así se enriquece la experiencia con conclusiones más acertadas.
13. El programa mostrará los resultados obtenidos durante la práctica. Además el usuario tiene acceso, mediante los iconos *Borrar Resultados* e *Iniciar*, a resetear los resultados obtenidos en la pantalla, poniendo los casilleros en blanco, y a reiniciar la práctica, respectivamente.
14. Terminada la práctica, se pueden almacenar los datos resultantes en un archivo de texto para imprimirlo posteriormente, ya que la conexión de la impresora temporalmente está utilizada. En el apartado 5.4.2.2.2, pag. 209, se explica en detalle la opción de impresión de los resultados experimentales.

1  
2  
3

### 5.1.5 PLANO INCLINADO.

El prototipo “Plano Inclinado”, es un sistema físico compuesto por dos planos contruidos en madera y unidos mediante una bisagra. El plano superior dispone de una plataforma de madera lisa, que sirve como pista de deslizamiento del móvil en estudio, además este plano tiene una libertad de giro de aproximadamente 100 grados con respecto al plano inferior que se ha ubicado como base del prototipo. Los sensores están instalados debajo del plano móvil (plano superior), mientras que los diodos LED infrarrojos, están montados sobre soportes de acrílico, directamente sobre los sensores. Tanto el sensor como el LED infrarrojo se encuentran alineados. De esta forma, se realiza adecuadamente la detección del paso de un cuerpo deslizando por el plano inclinado.

Adicionalmente, se ha dispuesto el circuito interfaz dentro de una caja de acrílico, la cual se encuentra ubicada a un lado de la base del prototipo, como se muestra en la *figura 5.8*. Esto permite que la caja tenga una operación adecuada.

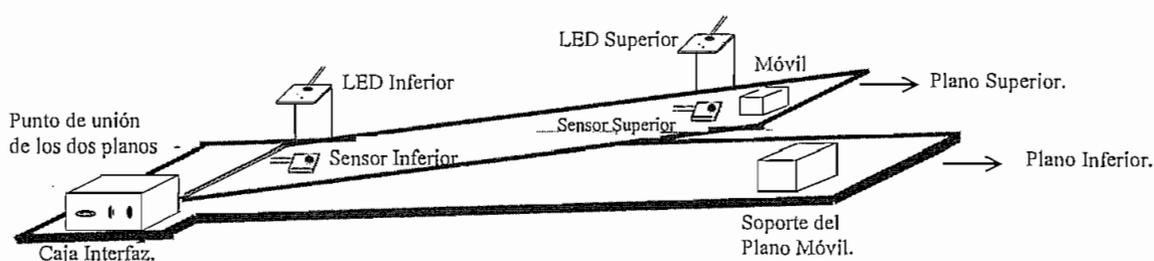


Figura 5.8.- Plano Inclinado. Ubicación de los sensores, diodos LED y caja interfaz.



Al lado izquierdo, en el frente de la caja, se encuentra el interruptor S1 que maneja la alimentación (*ver figura 5.9*), tanto para el circuito impreso total, como para los sensores y diodos LED infrarrojos, que se activan en forma simultánea. Mientras el cuerpo empieza a deslizarse, pasa por un sensor ubicado en la parte superior del plano móvil, luego, casi al terminar el recorrido, otro sensor detectará el paso del cuerpo en la parte inferior del plano. S1 es un interruptor señalizado, es decir, que indica la posición del estado de encendido (ON) y del estado de apagado (OFF) del circuito total.

Al igual que el prototipo de “Caída Libre”, para realizar la calibración del circuito interfaz, se dispone de dos potenciómetros montados en el frente de la caja interfaz, el primero, ubicado a la izquierda, calibra la sensibilidad del fototransistor montado en la parte superior del plano móvil, mientras que el segundo, ubicado a la derecha, controla la sensibilidad del fototransistor de la parte inferior del mismo plano. Los potenciómetros utilizados realizan la calibración de los sensores en forma independiente. (*ver calibración del interfaz “Caída Libre” y “Plano Inclinado”, apartado 5.2.2, pag. 194*). La *figura 5.9*, muestra la ubicación de los dispositivos de control en el frente de la caja interfaz.

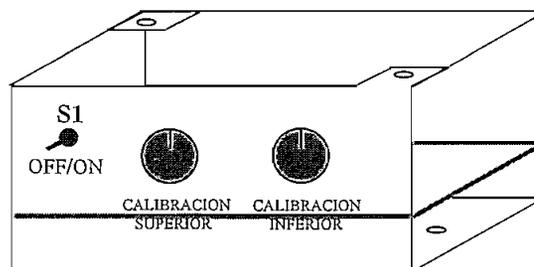


Figura 5.9.- Dispositivos de control del circuito interfaz de “Plano Inclinado”.



En la parte posterior de la caja interfaz, se tiene el cable de datos hacia el computador (**COMP**), y está configurado en el circuito impreso de la siguiente manera:

**PIN #1:** Cable rojo.- Al pin #12 (**Paper End**) del conector DB-25 H del computador.

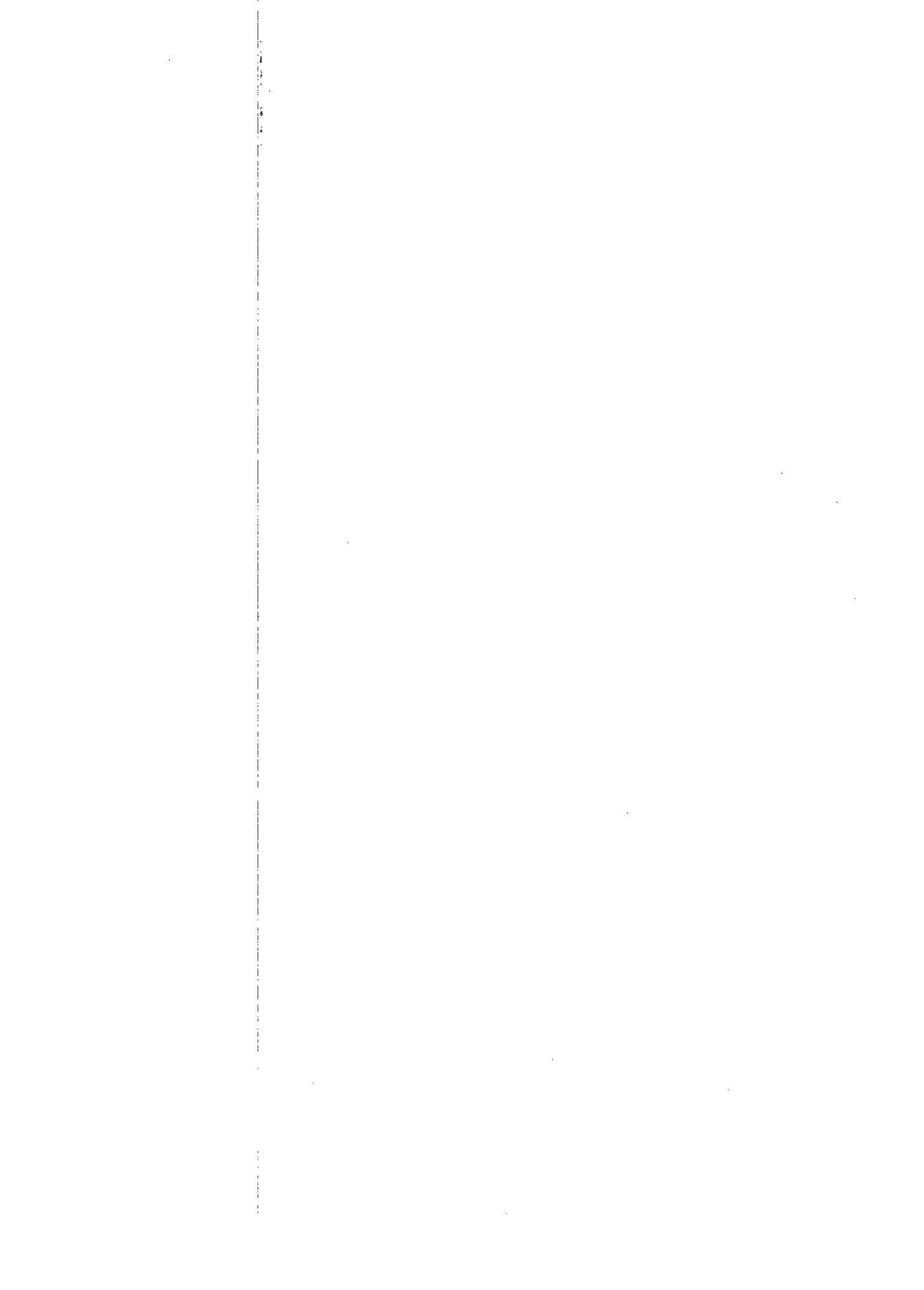
**PIN #2:** Cable verde .- Al pin #11 (**Busy**) del conector DB-25 H del computador.

**PIN #3:** Cable negro .- Conexión a tierra, que corresponde a los pines 18-25 (**Gnd a-h**) del conector DB-25 H del computador.

**NOTA:** Los números que corresponden a los pines del cable de datos, están identificados en el conector insertado en la placa del circuito impreso.

A la izquierda, en la parte posterior de la caja, se ha montado un terminal tipo “jack” para la conexión de la fuente de alimentación de todo el circuito electrónico. Este conector está instalado en el panel posterior de la caja, mediante una tuerca de sujeción. Aquí se inserta el conector “macho” más grande, del multiconector del adaptador universal de voltaje.

Atrás de la caja existen dos perforaciones juntas que conducen los cables de conexión de los diodos LED y de los sensores hacia el exterior. La *figura 5.10, pag. 185*, muestra la parte posterior de la caja interfaz.



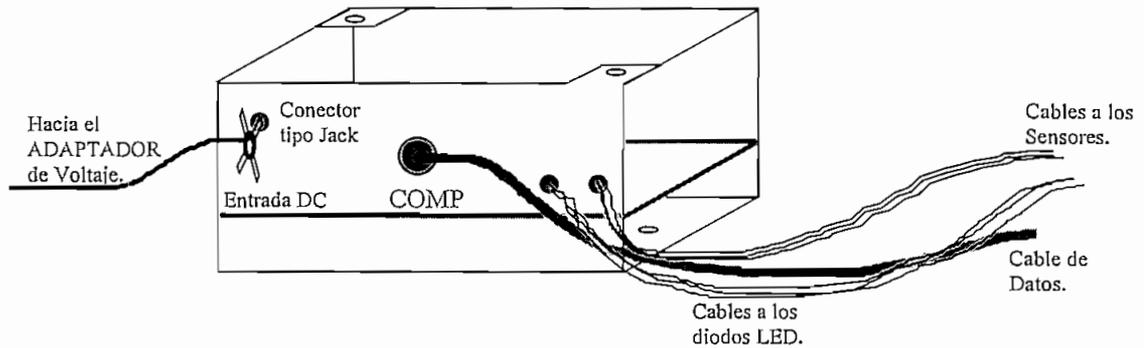


Figura 5.10.- Conexiones en la parte posterior de la caja interfaz.

Los cables de conexión a los sensores y diodos LED están instalados por el lado lateral del plano móvil, y terminan conectados a los puntos de análisis del prototipo, es decir, dos pares de cables se dirigen hacia los postes de luz, y dos pares hacia los respectivos fototransistores ubicados al frente de cada emisor infrarrojo, por debajo del plano móvil.

**NOTA:** El sistema “Plano Inclinado” trabaja adecuadamente cuando se encuentra calibrado. El interfaz se puede afinar manipulando las dos perillas frontales de la caja, mientras que, en forma paralela, se hace el monitoreo del voltaje sobre los pines #1 y #3 del circuito integrado principal (CI: 74LS14). De otra manera, se puede recurrir al programa computacional del proyecto y utilizar la herramienta de *Calibración*. En el apartado 5.2.2, pag. 194, se describe el proceso de calibración electrónica para los circuitos de “Caída Libre” y “Plano Inclinado”.



### 5.1.6 PROCEDIMIENTOS PARA REALIZAR LA PRACTICA DE “PLANO INCLINADO”.

El procedimiento que debe seguirse, es detallado a continuación:

1. Verificar que el equipo esté activado. Si no lo está, proceder a conectar la alimentación, insertando el conector “macho” del adaptador universal en el terminal tipo “jack”, ubicado en la parte posterior de la caja.

**NOTA:** Se sugiere no cambiar la posición del interruptor incluido en el adaptador de voltaje, pues dicha posición permite que la fuente proporcione aproximadamente 9 voltios al circuito interfaz.

2. Accionar el interruptor principal (**MAIN**) del sistema, a la posición de encendido (**ON**).
3. Verificar que el cable de datos esté correctamente conectado al puerto paralelo del computador.
4. En el caso de que el equipo haya sido previamente calibrado, pasar al numeral 5. De otro modo, proceder a la opción de calibración, ya sea directamente utilizando un equipo de medición (*multímetro*), o utilizando la opción de **Calibración**, incluida en el programa principal.
5. Seleccionar dentro del programa, en el menú principal, la opción *Práctica*, luego elegir *Plano Inclinado*.
6. Insertar los datos solicitados por el computador, esto es:
  - Distancia ( $\Delta x$ ) establecida entre los dos sensores a lo largo del plano móvil.

- Valor de la aceleración de la gravedad ( $g$ ).
- El programa automáticamente reconoce que el móvil parte desde el reposo, es decir, con  $V_0 = 0$ , y realiza los cálculos con este valor de velocidad inicial. Si el cuerpo no es soltado desde el reposo ( $V_0 \neq 0$ ), se generarán errores en el estudio del movimiento y los resultados no serán acertados.
- Valor del ángulo de inclinación ( $\theta$ ) del plano móvil (*plano superior*), respecto al plano fijo (*plano inferior*). Este valor define el ángulo con el cual, el móvil rompe el estado de reposo y empieza a deslizarse con movimiento acelerado.

**NOTA:** Es posible que se requiera mover el plano superior, aumentando paulatinamente el ángulo de inclinación ( $\theta$ ), hasta ver que el cuerpo vence el estado de reposo y empieza a moverse. En ese momento es recomendable tomar un graduador y medir el ángulo de inclinación del plano para insertarlo como dato en la práctica.

- Valor del coeficiente de rozamiento dinámico ( $\mu$ ), que dependerá del tipo de material existente en las dos superficies en contacto (móvil y plano inclinado). Este valor se obtiene de la tabla de coeficientes de rozamiento dinámico, al momento de seleccionar la ventana correspondiente a ( $\mu$ ).
- Masa del cuerpo ( $M$ ), la cual es indiferente en la práctica puesto que, para este ejemplo, esta variable se simplifica y desaparece en la deducción de las ecuaciones del movimiento.

**NOTA:** Para la correcta utilización del programa computacional es necesario revisar el apartado 5.4, pag. 200.

7. Hacer click en **INICIAR**. El computador mostrará en la pantalla un mensaje indicando que se proceda a colocar el cuerpo de prueba sobre el sensor superior, en la parte alta del plano vertical, (ver “*sensor superior*”, figura 5.8, pag. 182).
8. Hacer click en **ACEPTAR**. En este punto, el programa se encuentra listo para estudiar el movimiento del cuerpo deslizándose por el plano inclinado.
9. Para evitar errores en la experiencia, antes de efectuarse la práctica, es conveniente ubicar el móvil, antes de efectuarse la práctica, en la parte central de la pista de deslizamiento. Así se asegura que el cuerpo, mientras desciende, es detectado por los sensores sin desviarse de la trayectoria.
10. El computador realiza los cálculos respectivos, mediante la utilización de las ecuaciones que se deducen del análisis físico de un plano inclinado. La figura 5.11, muestra el ejemplo característico de un sistema de este tipo. Generalmente, para la deducción de las ecuaciones, se parte del diagrama de fuerzas sobre el móvil y se aplican las leyes de “*Newton*”. Así, se tiene que:

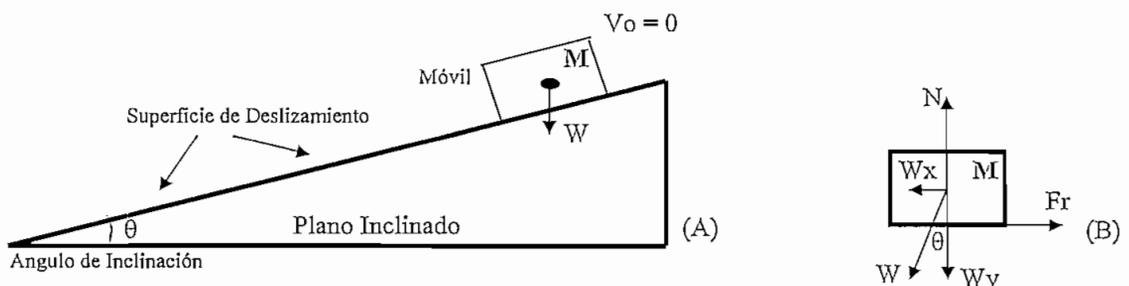


Figura 5.11.- (A) Plano Inclinado. (B) Diagrama de fuerzas.

Donde:

$M$  = Masa del cuerpo (móvil).

$W$  = Peso del cuerpo.

$W_x$  = Componente del peso en el eje x.

$W_y$  = Componente del peso en el eje y.

$F_r$  = Fuerza de fricción.

$N$  = Fuerza normal.

$\theta$  = Angulo de inclinación del plano inclinado.

$g$  = Valor de la aceleración de la gravedad.

En el eje Y, se tiene que:

$$N - W_y = 0 \Rightarrow N = W_y \Rightarrow N = M \cdot g \cdot \cos \theta \quad (1)$$

En cambio, en el eje X:

$$W_x - F_r = M \cdot a_T \Rightarrow M \cdot g \cdot \sin \theta - \mu \cdot N = M \cdot a_T ; \quad (F_r = \mu \cdot N) \quad (2)$$

(1) en (2):

$$M \cdot g \cdot \sin \theta - \mu \cdot M \cdot g \cdot \cos \theta = M \cdot a_T \quad (3)$$

Simplificando la masa ( $M$ ) y ordenando:

$$\boxed{a_T = g(\sin \theta - \mu \cdot \cos \theta)} \quad (4)$$

Como  $a_T = (V_f - V_o) / t$ , se despeja  $V_f$  y se tiene que:

$$\boxed{V_f = g \cdot t(\sin \theta - \mu \cdot \cos \theta) + V_o} \quad (5)$$

Para el cálculo del desplazamiento se tiene la ecuación:

$$\Delta x = V_0 \cdot t + \frac{1}{2} \cdot a_T \cdot t^2 \quad (6)$$

Entonces,  $V_0 = 0$  y reemplazando la ecuación (4) en la (6), se tiene que:

$$\Delta x_{EXP} = \frac{1}{2} \cdot g(\text{sen } \theta - \mu \cdot \text{cos } \theta) \cdot t^2 \quad (7)$$

Donde:

$\Delta x_{EXP}$  = Desplazamiento calculado experimentalmente.

Y el error generado por el desplazamiento del cuerpo, se calcula así:

$$\text{Err} \Delta x = \left| \frac{\Delta x_{EXP} - \Delta x_{TEO}}{\Delta x_{TEO}} \right| * 100\% \quad (8)$$

11. Mientras el programa realiza los cálculos pertinentes, es recomendable que el usuario realice por separado el mismo estudio, aplicando las ecuaciones del movimiento. Así se enriquece la experiencia con conclusiones más acertadas.
12. El programa mostrará los resultados obtenidos durante la práctica. Esto es, se mostrará el resultado de la aceleración adquirida por el cuerpo (*ecuación 4*), la velocidad final del móvil cuando el cuerpo ha cruzado al sensor inferior (*ecuación 5*), el desplazamiento efectuado por el móvil entre el sensor superior y el sensor inferior (*ecuación 7*), y el error generado durante ese desplazamiento (*ecuación 8*). Además el usuario tiene acceso, mediante los iconos *Borrar Resultados e Iniciar*,

a resetear los resultados obtenidos en la pantalla, poniendo los casilleros en blanco, y, a reiniciar la práctica, respectivamente.

13. Terminada la práctica, se puede almacenar los datos resultantes en un archivo de texto para luego imprimirlo, ya que la conexión de la impresora está temporalmente utilizada. En el apartado 5.4.2.2.2, pag. 209, se explica en detalle la opción de impresión de los resultados experimentales.

## **5.2 CALIBRACION ELECTRONICA DEL EQUIPO.**

### **5.2.1 CALIBRACION DEL INTERFAZ “PENDULO SIMPLE”.**

Para efectuar la calibración electrónica del circuito interfaz, se realizarán los siguientes pasos:

1. Remover la tapa de la caja interfaz. Utilizar la herramienta adecuada para el tipo de tornillo de la caja.
2. Verificar que el interfaz tenga el interruptor (S2) en la posición de encendido (ON).
3. Verificar que el interruptor conmutador (S1) se encuentre indicando el funcionamiento del sensor superior o inferior.

**NOTA:** Recuérdese que debido a la configuración del circuito electrónico de “Péndulo Simple”, si se calibra el sensor superior, el sensor inferior quedará en similar condición, y viceversa.

4. Utilizar un multímetro con una escala adecuada para medir voltios DC. Ubicar las puntas, negra de referencia (GND) y la roja (V/ $\Omega$ ), sobre los pines #7 y #1 respectivamente del circuito integrado CI 74LS14, de acuerdo a la *figura 5.12*.

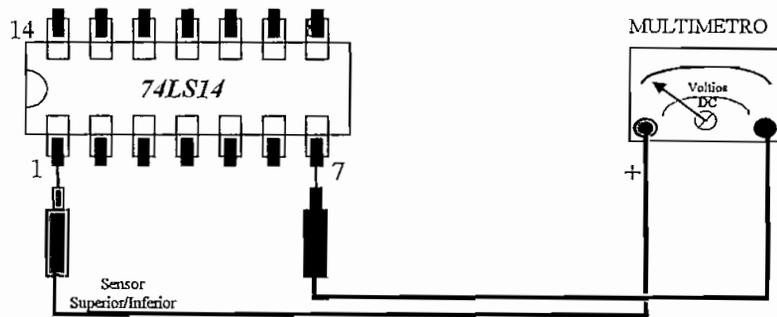


Figura 5.12.- Verificación del voltaje de entrada al “Schmitt Trigger”.

5. Cuando los sensores están en contacto directo con la luz emitida por los LED infrarrojos, el valor óptimo de voltaje en el pin #1 debe ser de 0.2 voltios. Esto se debe a lo siguiente:

- Según el circuito electrónico mostrado en el *diagrama de “Péndulo Simple” dentro de ANEXOS, literal D, pag. 236*, cuando el fototransistor (Q1/Q3) recibe luz, se genera suficiente corriente en la base del transistor BJT (Q2), capaz de saturarlo. Cuando sucede esto, se tiene un voltaje de colector  $V_{CE}$  de aproximadamente 0.2 voltios. Como este valor es menor al voltaje  $V_{H-}$  que especifica el fabricante del integrado, se asegurará la transición positiva en el lazo de histéresis del circuito, de acuerdo a la *figura 5.13, pag. 193*.

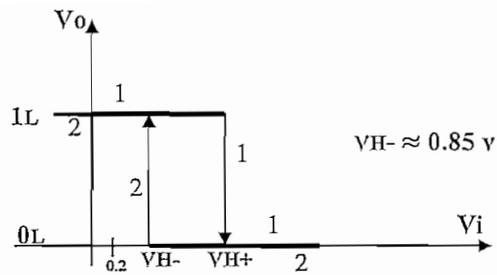


Figura 5.13.- Lazo de histéresis del disparador de Schmitt 74LS14.

**NOTA:** El gráfico de la *figura 5.13*, muestra: 1) Ruta del disparador cuando el sensor inicialmente estuvo iluminado y luego fue tapado. 2) Ruta del disparador cuando el sensor inicialmente estuvo tapado y luego fue iluminado.

- Si  $V_{CE} \approx V_{H-}$  y si el disparador sigue la ruta 2, puede suceder que no ocurra el disparo desde un nivel bajo a un nivel alto, y la salida ( $V_o$ ) siga manteniéndose en un nivel bajo. En estas condiciones el circuito estará funcionando inadecuadamente. Para esto, se tratará de poner un valor de 0.2 voltios en el colector del BJT, mediante la manipulación del potenciómetro de calibración, paralelamente al monitoreo del pin #1 del circuito integrado 74LS14, como se observa en la *figura 5.12*, pag. 192, obteniéndose finalmente una respuesta adecuada del disparador de Schmitt y la calibración definitiva del circuito electrónico.

6. Ubicar la tapa de la caja interfaz en su posición y ajustarla.

**NOTA:** Si el equipo no responde a los cambios de luz sobre el sensor, verificar las conexiones del circuito y luego los niveles de voltaje en los puntos principales de medición, para compararlos con los resultados presentados en el *capítulo IV, apartado 4.2.1, tabla 4.2.1, pag. 146*. Además, revisar que los interruptores del interfaz se encuentren activados.

### 5.2.2 CALIBRACION DEL INTERFAZ “CAIDA LIBRE” Y “PLANO INCLINADO”.

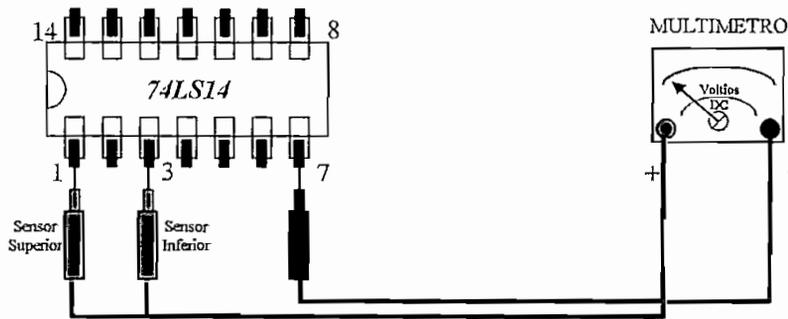
Para realizar la calibración de los circuitos electrónicos, tanto en “Caída Libre” como en “Plano Inclinado”, son necesarios los siguientes pasos:

1. Remover la tapa de la caja interfaz. Utilizar la herramienta adecuada para el tipo de tornillo de la caja.
2. Verificar que el interfaz tenga el interruptor (S1) en la posición de encendido (ON).

**NOTA:** Para la puesta a punto del circuito electrónico, es necesario proceder con la calibración de los sensores individualmente, es decir, cada sensor tiene su propio sistema de control de calibración.

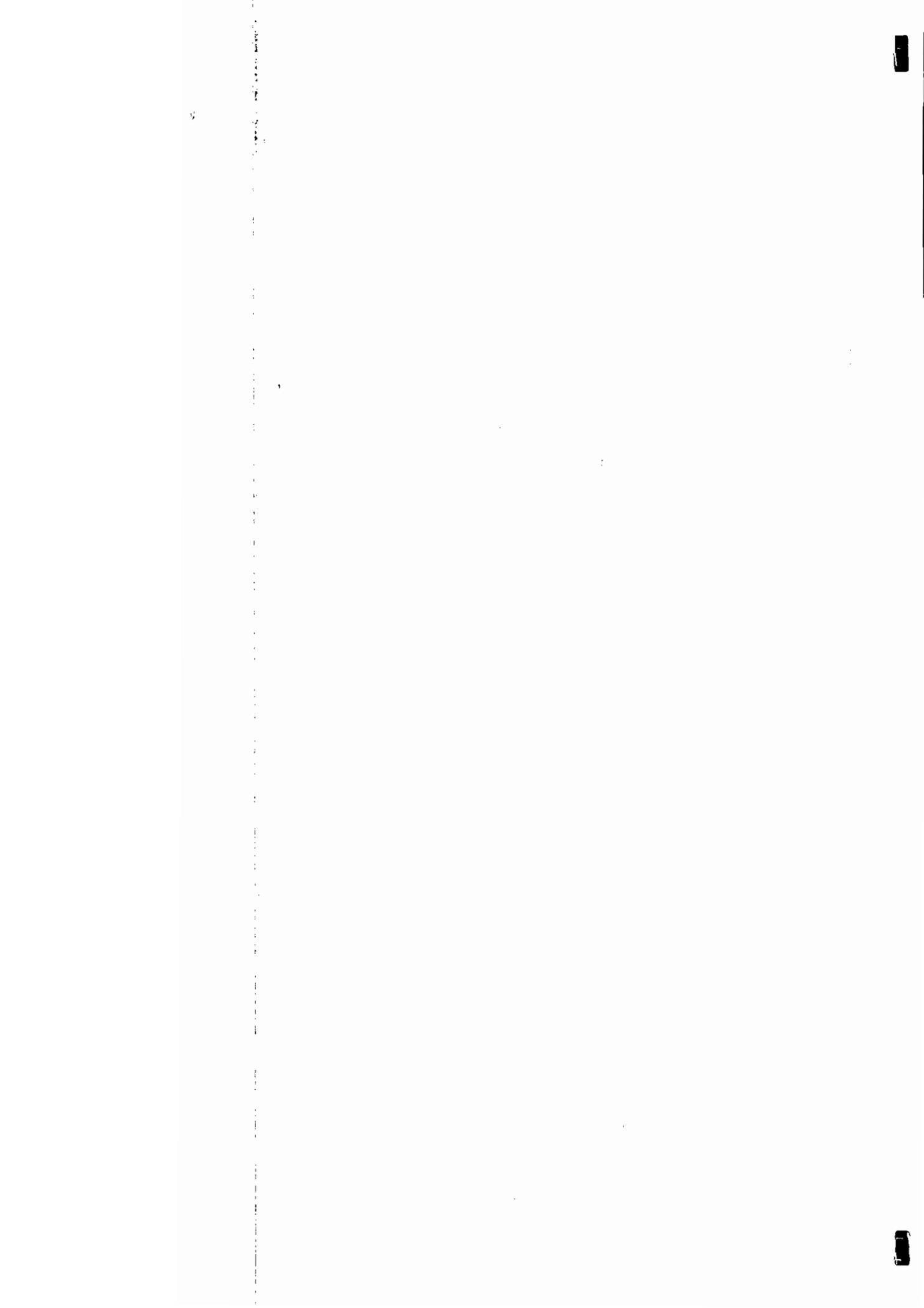
3. Utilizar un multímetro con una escala adecuada para medir voltios DC. Ubicar las puntas, negra de referencia (GND) y la roja (V/ $\Omega$ ) en el circuito integrado CI

74LS14, sobre los pines #7 y #1 respectivamente, para la calibración del sensor superior; y sobre los pines #7 y #3 respectivamente, para la calibración del sensor inferior, de acuerdo a la *figura 5.14*.



**Figura 5.14.- Verificación del voltaje de entrada a los “Schmitt Trigger”, sensor superior/inferior.**

4. Cuando los sensores están en contacto directo con la luz emitida por los LED infrarrojos, el valor óptimo de voltaje en el pin #1 y el pin #3, debe ser de 0.2 voltios. Si el voltaje obtenido es mayor que dicho valor e inclusive mayor al voltaje **VH-** (típicamente de 0.85 voltios), dado por el fabricante del integrado, entonces el disparo por transición positiva no se realiza y el interfaz no responderá adecuadamente ante los cambios de luz suscitados sobre el sensor. En resumen, si el sensor, inicialmente tapado, vuelve a ser iluminado, el voltaje de entrada del “Schmitt Trigger” disminuye considerablemente, y se realizará la transición positiva en el momento de que se asegure un valor de voltaje menor al voltaje umbral **VH-** establecido por el circuito integrado, *ver figura 5.13, pag. 193*. Por esta razón, resulta adecuado que se manipulen las perillas de calibración de los sensores paralelamente con la lectura del voltaje de los



sensores paralelamente con la lectura del voltaje de los pines #1 y #3, utilizando un multímetro, con el objeto de ubicar el valor de 0.2 voltios en las entradas de los “Schmitt Trigger”, consiguiéndose que la respuesta del integrado sea rápida y segura.

5. Terminado el proceso de calibración, volver a tapar la caja interfaz, ajustando los tornillos de sujeción.

**NOTA:** Si el equipo no responde a los cambios de luz sobre el sensor, verificar las conexiones del circuito y luego los niveles de voltaje en los puntos principales de medición, para compararlos con los resultados presentados en el *capítulo IV, apartado 4.2.2, tabla 4.2.2, pag. 149* y *apartado 4.2.3, tabla 4.2.3, pag. 150*. Además revisar que el interruptor del interfaz se encuentre activado.

### 5.3 MANTENIMIENTO DE LOS EQUIPOS.

Es importante que se realice el mantenimiento de los equipos, debido a que algunas veces durante las pruebas, los interfaces electrónicos pueden presentar otros valores de respuesta hacia la entrada del computador, es decir, luego de varias horas de prueba de estabilidad térmica, en ciertos casos se notó que el voltaje de entrada a los “Schmitt Trigger”, fue de aproximadamente 0.67 voltios. Aunque este valor sigue siendo menor que el voltaje VH- establecido por el fabricante del circuito integrado 74LS14, se corre el riesgo de que el interfaz no responda adecuadamente al momento de la realización de la práctica, perdiéndose de esta forma, la verdadera información registrada sobre los sensores. Para esto, se recomienda realizar el proceso de



calibración de los interfaces electrónicos antes de cada experiencia y mediante las perillas de calibración, ubicar el voltaje de entrada al disparador en 0.2 voltios, cuando el sensor se encuentre iluminado.

En el caso de daño de algún dispositivo electrónico en la placa del circuito impreso, existe la facilidad de remover toda la tarjeta principal con solo extraer los conectores minicom, montados en los terminales tipo macho de la tarjeta.

El circuito integrado principal del interfaz, es un dispositivo TTL que se alimenta con 5 voltios DC. Sin embargo, este dispositivo es susceptible de dañarse cuando la fuente de alimentación sobrepasa el límite de voltaje permitido por el integrado. Si esto sucede, es posible el reemplazo inmediato de la pieza, puesto que ésta se encuentra ubicada sobre un zócalo de 14 pines. Adicionalmente como parte del mantenimiento preventivo y en caso de fallas, es necesario mencionar los siguientes parámetros:

- Revisar que los diodos LED infrarrojos se encuentren siempre alineados contra los sensores de luz.
- Limpiar eventualmente los huecos donde se albergan los sensores. El lente del fototransistor puede estar tapado por partículas de basura, provocándose errores en la respuesta entregada al computador.
- Con el tiempo de uso, la vida útil de los interruptores principales del interfaz llegará a su fin. En ese caso proceda con el reemplazo del interruptor averiado.
- En el diseño inicial, el sistema de detección utilizaba lámparas de luz visible asistidas por baterías. Sin embargo, aunque el funcionamiento del circuito era



relativamente óptimo, las lámparas llegaban a degradarse y posteriormente a quemarse. Por esta razón se utilizan diodos LED infrarrojos, los cuales son fácilmente reemplazables y sobre todo demuestran estabilidad y larga duración.

- Evítese usar móviles pesados en el experimento de “Caída Libre”, pues la base del prototipo es de madera y tiene un único recubrimiento de cuero.
- Tratar siempre de desconectar primero el adaptador de voltaje, antes de desconectar el otro extremo (multiconector en forma de cruz). Conectado el adaptador de voltaje, puede llegar a suceder que la fuente entre en corto circuito, cuando se desconecte el multiconector del terminal tipo “jack”, de la parte posterior de la caja interfaz. Esto se debe a que, al extraer el multiconector, el terminal “vivo” de éste último topa con la referencia (GND) de todo el circuito. En estas condiciones la fuente puede sufrir daños severos.
- Limpiar regularmente el polvo que se asienta sobre los prototipos físicos.
- Se han reforzado los soportes de iluminación en los proyectos físicos. Sin embargo, en caso de rotura de uno de ellos, se sugiere utilizar una extensión metálica, para soldar las partes separadas. No puede romperse la pieza a la altura de las esquinas o dobleces.

## **5.4 MANEJO DEL PAQUETE COMPUTACIONAL DEL PROYECTO.**

### **5.4.1 INTRODUCCION.**

El uso de la computadora en el convivir diario, se ha convertido en algo indispensable en todas las actividades, y no podía ser de otra manera en el campo

.....



científico. En las grandes universidades del mundo, la computadora ha sido desde sus inicios una herramienta poderosa, que ha permitido sacar adelante muchos proyectos para el cálculo, medición, simulación, diseño, etc.

En nuestro país, el uso de la computadora en las universidades ha tenido una importancia histórica en todos los campos. Es por eso que, este trabajo ha sido pensado para beneficio del estudio de la física en particular, como un plan piloto para la enseñanza práctica de ésta rama de las ciencias.

Este manual será una guía para la utilización adecuada del programa computacional, y es indispensable que se obtenga el mayor provecho al conocer de las múltiples opciones con las que el programa cuenta.

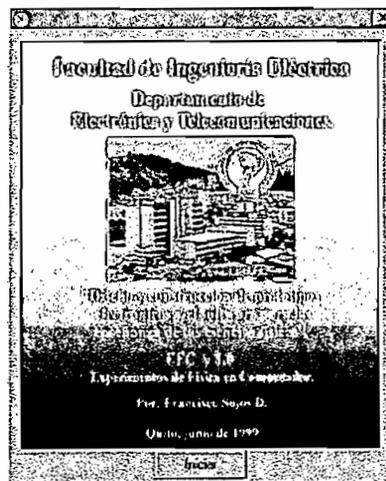


Figura 5.15.- Pantalla de presentación de EFC.

El primer contacto del usuario al ejecutar el programa, es una pantalla de presentación del proyecto en donde, se solicita el acceso mediante el botón **INICIAR**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

(ver figura 5.15, pag. 199). Una vez que se ingresa, el programa desaparecerá la ventana de presentación y se mostrará la ventana del **MENU DE OPCIONES**.

#### 5.4.2 MENU DE OPCIONES.

El programa **EFC v1.0** (*Experimentos de Física en Computador*), cuenta con un menú de opciones, en donde se encuentran: **Experimento, Unidades, Ventana y Ayuda**. Con este menú, se puede manejar el programa y sus múltiples opciones como por ejemplo, las unidades en las que se trabaja, la forma de ordenar el escritorio de trabajo, encontrar un respaldo teórico de los datos obtenidos en las prácticas y/o simulación (**Ayuda**), y el escogimiento de la parte del programa al cual se desea ingresar.

##### 5.4.2.1 VENTANA DE ACCESO DIRECTO

Esta ventana fue creada pensando en la comodidad que requiere el usuario para el ingreso a las diferentes opciones (ver figura 5.16, pag. 201), y consta de las siguientes partes:

**Iconos.**- Son botones con dibujos representativos, con los cuales se puede ingresar a la ventana buscada.

**Títulos Superiores.**- Indican si los iconos pertenecen a "*Caída Libre*", "*Plano Inclinado*" o "*Péndulo Simple*".



**Títulos Laterales.-** Indican en qué modo de ejecución el usuario estaría inmerso. Esto es **Simulación** o **Práctica**. Adicionalmente, existe un título de **Ayuda**, que señala a un botón adjunto, el cual accionará el menú de ayuda del programa.

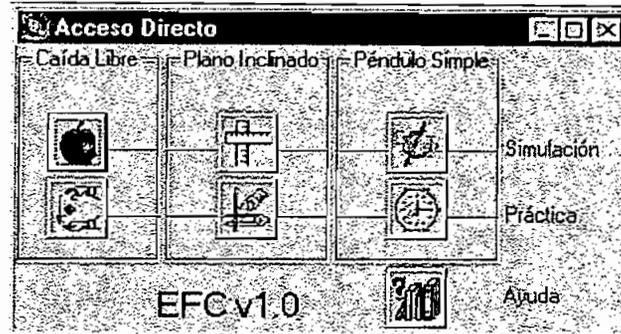


Figura 5.16.- Ventana de acceso directo a las opciones de EFC.

Esta ventana se activará al iniciar el programa, luego de presionar el botón **INICIAR** de la ventana de presentación del programa (ver figura 5.15, pag. 199).

#### 5.4.2.2 OPCION "EXPERIMENTO".

Dentro del menú de opciones general, este es el primero. Se ubica primero en la izquierda de la barra de opciones. Esta opción muestra adicionalmente las siguientes opciones:

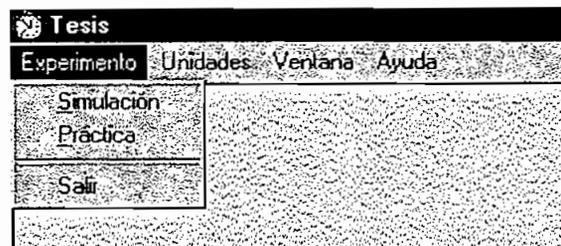


Figura 5.17.- Elecciones sobre la opción "Experimento".



**SIMULACION.-** Conduce a una ventana en donde se elegirá una actividad de simulación de "Caída Libre", "Plano Inclinado" y "Péndulo Simple", a través de un submenú adicional. Además cuenta con varias opciones para cada uno de los tres fenómenos físicos. Los valores obtenidos de la simulación son valores teóricos, y para obtener resultados fiables, es necesario también ingresar datos teóricos.

**PRACTICA.-** Lleva a una ventana en donde se encuentra, al igual que en Simulación, un submenú de "Caída Libre", "Plano Inclinado" y "Péndulo Simple". La diferencia radica en que en esta parte se puede interactuar con los equipos construidos para el propósito. Los resultados obtenidos en esta parte, se obtienen con la ayuda de mediciones de tiempo y datos teóricos de distancia, masa, velocidad inicial, coeficientes de rozamiento o gravedad, etc., es por eso que podrían existir diferencias con los resultados obtenidos en la parte de Simulación.

**SALIR.-** Esta opción o el icono con forma de X, en la parte superior derecha de la ventana, permiten que el programa termine y se volverá a trabajar con el entorno del sistema operativo.

#### 5.4.2.2.1 SIMULACIÓN.

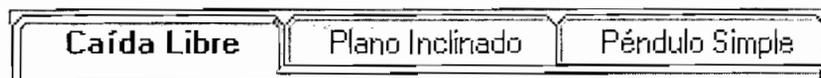


Figura 5.18.- Opciones para la elección de Simulación.

Consta, como antes se mencionó, de una ventana con tres opciones. Es muy útil para resolver problemas de física, si se tienen los datos adecuados que se solicitan

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

en cada una de las ventanas. Las ecuaciones utilizadas, son conocidas y se las encuentra en cualquier bibliografía sobre física (Mecánica) y M.A.S (Movimiento Armónico Simple).

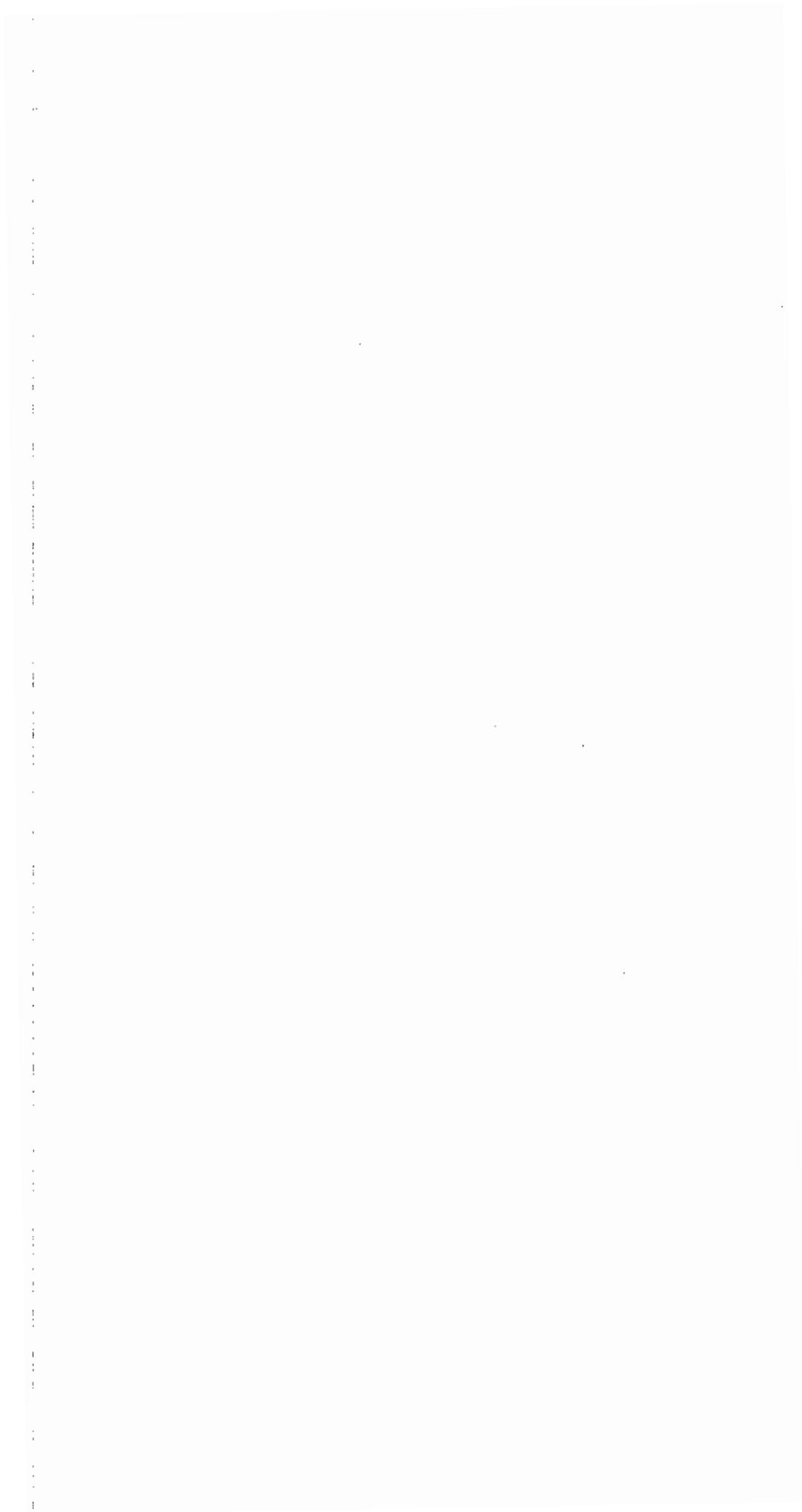
Además de los datos solicitados y los cuadros de resultados, **Simulación** consta de cuatro botones dispuestos en cada una de las tres ventanas: "**Caída Libre**", "**Plano Inclinado**" y "**Péndulo Simple**", y permiten interactuar al usuario con el programa. Estos son:

**Iniciar.**- Luego de ingresar los datos en la ventana, este botón permite comenzar la simulación del fenómeno en tiempo real, mostrando al móvil en pleno movimiento y cuya rapidez de acción depende de la magnitud de los valores insertados como datos previos.

**Detener.**- Una vez iniciada la simulación, es la forma de parar de forma anormal el proceso. También es empleado en "Péndulo Simple" para presentar los resultados de la simulación.

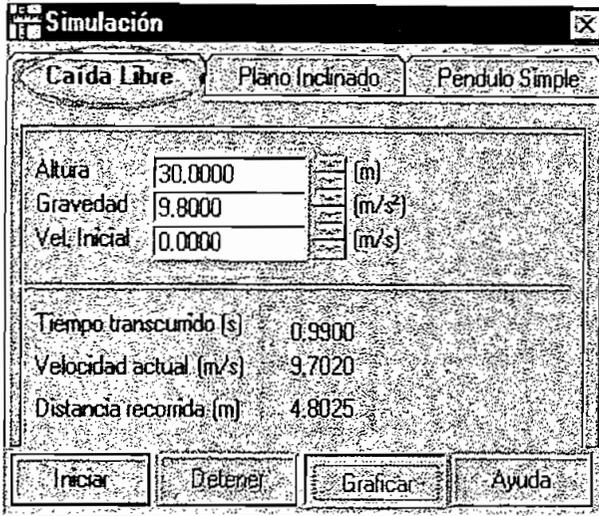
**Graficar.**- Este botón permite enseñar y ocultar las ventanas auxiliares de gráficos, en donde se simula el desplazamiento de un móvil en tiempo real.

**Ayuda.**- Muestra la ventana de ayuda.



## CAIDA LIBRE.

Para la simulación de "Caída Libre", se ingresarán varios datos como:

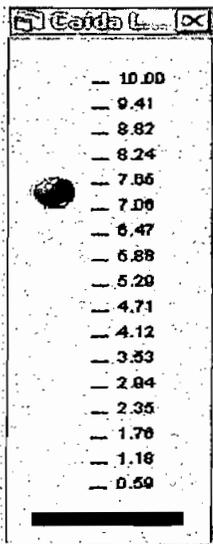


**Altura.**- Representa la distancia desde el punto más alto, con respecto al suelo, y es el punto donde se suelta el objeto de prueba. Está sujeta a las unidades seleccionadas.

**Gravedad.**- Es un dato teórico, en las unidades adjuntas, de la aceleración de la gravedad.

**Velocidad Inicial.**- Es la velocidad a la que se encuentra el cuerpo en el instante de pasar por el punto más alto, puede ser también que parta del reposo.

Figura 5.19.- Simulación de "Caída Libre".

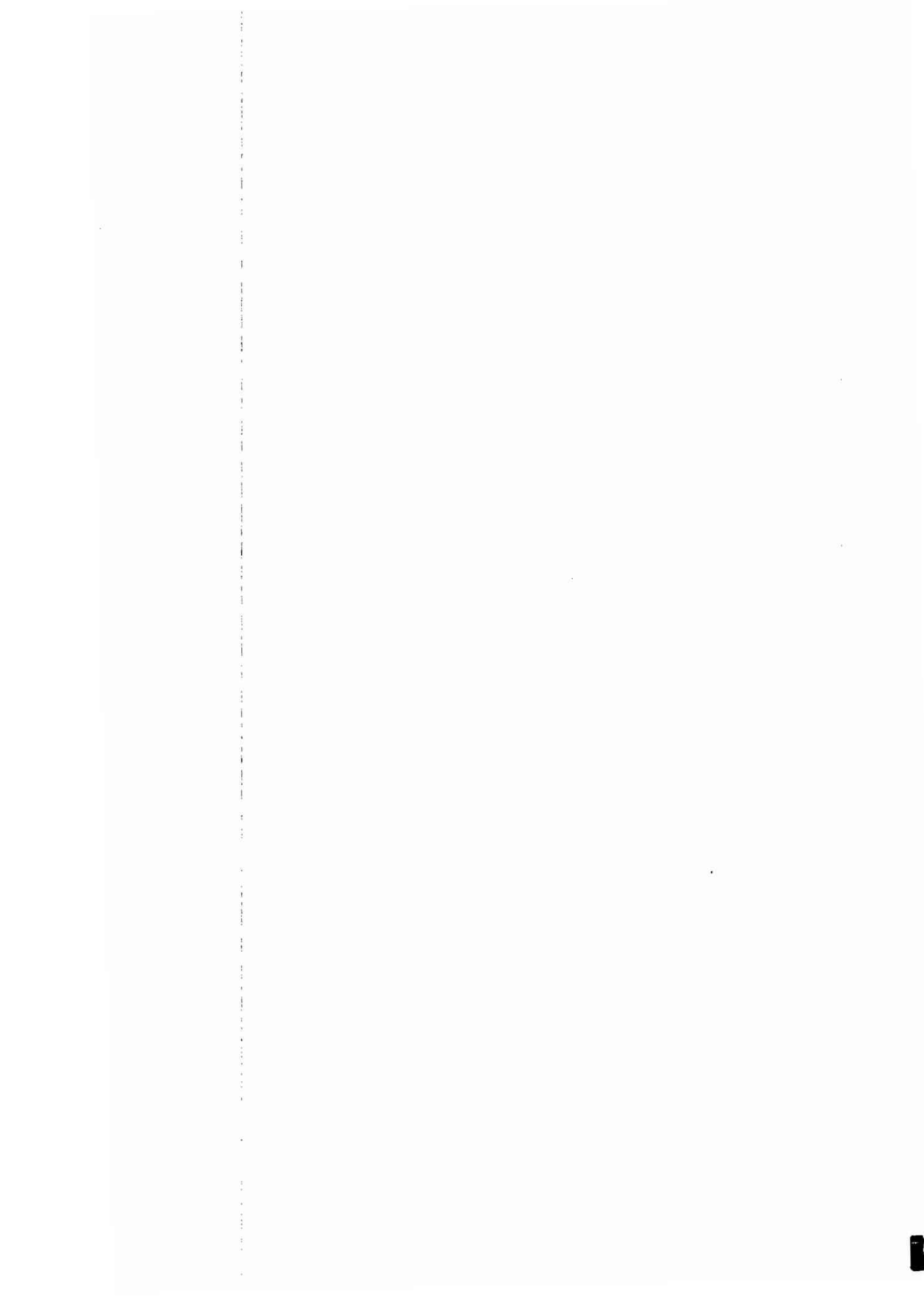


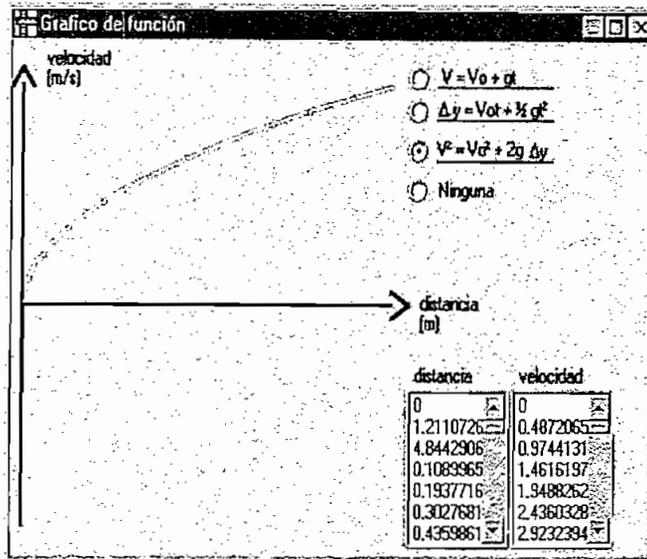
Como resultado, al pulsar el botón **Iniciar**, se obtendrá un cambio en el **Tiempo**, **Velocidad** y **Distancia** (*resaltados con el cuadro de la figura 5.19*), hasta que el objeto llegue al suelo, o el usuario detenga el proceso.

Además, como se ve en la *figura 5.20*, se obtiene una ventana en la cual se visualiza claramente el desplazamiento del móvil en tiempo real.

Durante la caída, el cuerpo en estudio pasa por 17 niveles de referencia. En cada uno de éstos se calculan los parámetros de tiempo, velocidad y distancia antes mencionados.

Figura 5.20.- Gráfico de "Caída Libre".



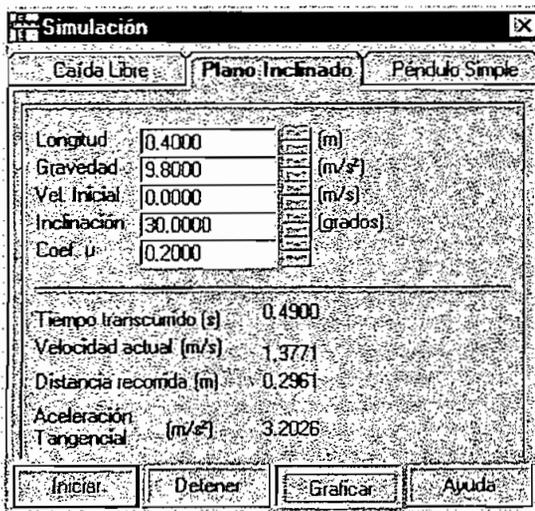


Como resultado de los valores obtenidos en la simulación, es posible realizar la gráfica del movimiento que realiza el móvil. Para "Caída Libre" es necesario graficar y analizar tres ecuaciones, las cuales se exponen en la *figura 5.21*.

Esta ventana consta de los ejes, las ecuaciones y los cuadros de datos para la realización de la gráfica. A cada valor del cuadro, le corresponde el valor respectivo del otro cuadro.

Figura 5.21.- Ventana del gráfico del movimiento "CL".

### PLANO INCLINADO.



Aquí se necesitan los siguientes datos: **Longitud, Gravedad, Velocidad Inicial, Inclinación** y el **Coefficiente de rozamiento**.

Este coeficiente de rozamiento, se lo encuentra en la ventana de ayuda del programa, en la tabla de coeficientes.

Los resultados obtenidos son: Tiempo transcurrido, Velocidad final (Actual al finalizar la práctica), distancia recorrida y Aceleración Tangencial.

Figura 5.22.- Simulación de "Plano Inclinado".



La simulación del movimiento aquí obtenida, se realiza mediante un cuerpo que se desliza por una línea con cierta pendiente, y cuya inclinación es igual al valor de ángulo ingresado por el usuario como dato de entrada. El móvil se comportará como indica la teoría.

Al pulsar el botón Iniciar, se producirán cambios en los valores de tiempo, velocidad y distancia recorrida. La aceleración tangencial es constante.

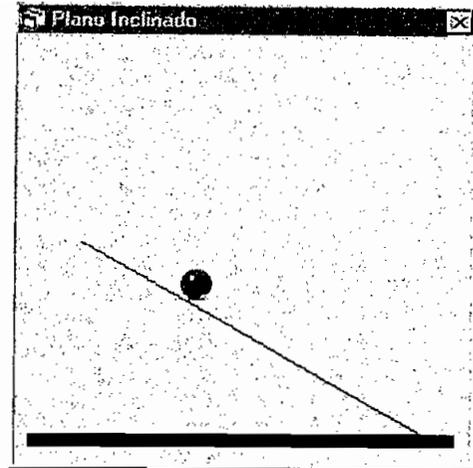


Figura 5.23.- Gráfico de "Plano Inclinado".

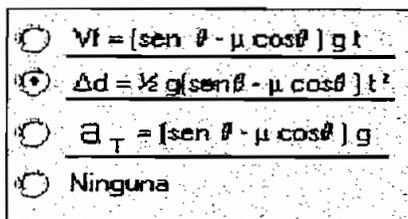


Figura 5.24.- Ecuaciones en "Plano Inclinado"

Estas son las funciones disponibles para ser graficadas en este movimiento, así también, se puede observar una opción que dice **Ninguna**, y desaparece los gráficos, dejando la pantalla en blanco.

**NOTA:** De igual forma es posible graficar las curvas del movimiento de "Plano Inclinado".

### PENDULO SIMPLE.

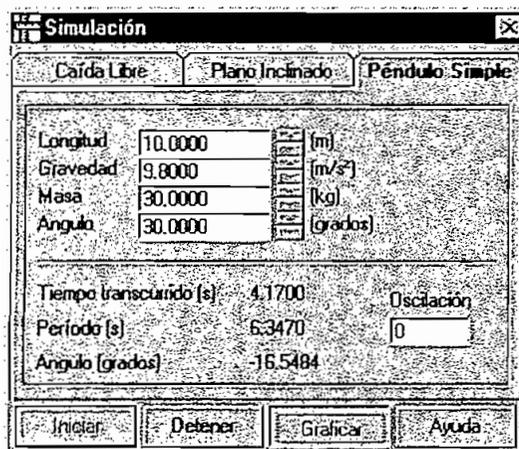


Figura 5.25.- Simulación de "Péndulo Simple".

Para la simulación de esta parte, se necesita que el usuario ingrese los siguientes datos: **Longitud, Gravedad, Masa y Angulo.**

Nótese que a la derecha de los datos existen unas barras de incremento y decremento de los valores, esto permite facilitar el manejo y precisión de los datos.

Además, a la derecha de los mismos, se encuentran las unidades, que pueden cambiarse dentro del menú de **Unidades.**



Entre los resultados, se tiene: **Tiempo**, **Período** y **Ángulo**. Además es posible visualizar el número de oscilaciones que va cumpliendo el péndulo durante su movimiento.

El ángulo está tomado respecto al eje vertical, para lo cual se adoptó como signo positivo los ángulos recorridos a la derecha y, signo negativo, los recorridos hacia la izquierda de la vertical de referencia.

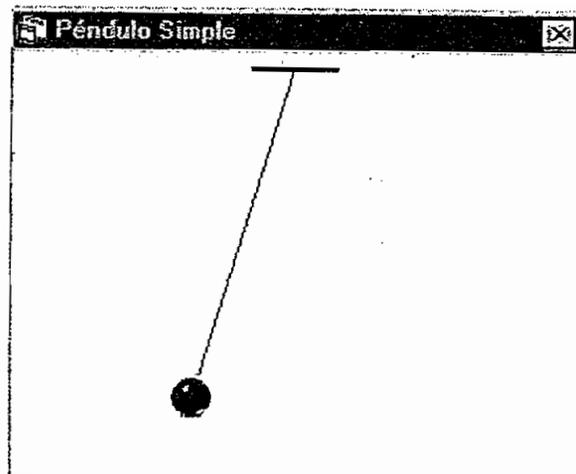


Figura 5.26.- Gráfico de "Péndulo Simple".

$$\theta = \theta_0 \text{ sen } \left[ \left( \frac{2\pi}{T} \right) t + 1.57079 \right]$$

Figura 5.27.- Ecuación de "Péndulo Simple".

En este caso, la única ecuación que se puede graficar es una función angular que depende del período pendular **T**.

El gráfico de la *figura 5.28*, se genera a partir de los datos señalados en las tablas de tiempo y ángulo. A cada valor de tiempo le corresponde al valor frontal de ángulo.

Los ejes contienen los nombres y las unidades a los que corresponde cada tabla.

La opción **Ninguna**, desaparece el gráfico de la pantalla, y se muestran únicamente los ejes cartesianos.

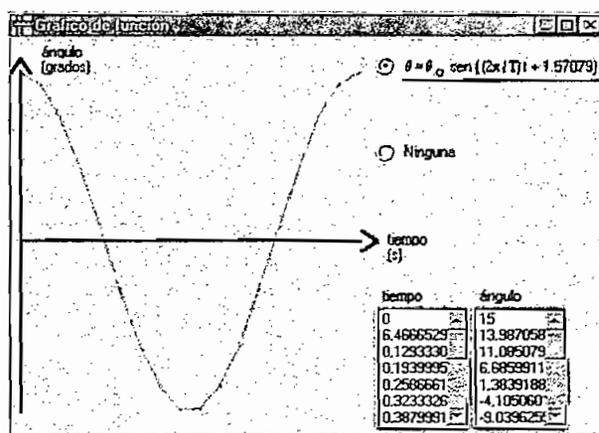


Figura 5.28.- Ventana gráfica del movimiento "PS".



#### 5.4.2.2.2 PRÁCTICA

Como parte principal de la concepción de este programa, la práctica consta de tres experiencias, sin embargo, la interacción se realiza ahora mediante la inserción de datos "reales", es decir datos tomados por los sensores instalados en los equipos. Estos equipos interactúan y envían datos a la computadora, mediante el puerto paralelo utilizado por la impresora (DB25).



Figura 5.29.- Menú de selecciones correspondientes.

De forma general, las ventanas pertenecientes a la selección **PRACTICA**, constan de los siguientes botones de acceso:

**Iniciar.**- Permite la inicialización de la práctica. Luego de activar este botón se recibirá un mensaje de aviso para ubicar el móvil en la posición inicial de análisis.

**Detener.**- Una vez iniciada la práctica, este botón permite detener de forma anormal el curso de la misma.

**Calibrar.**- Permite conocer el estado de los sensores, controlar la sensibilidad de los mismos y poner a punto el equipo. Mediante las perillas de control de cada interfaz se puede manipular el estado de los bits de datos que recibe la computadora. Estos datos



son ceros o unos (0L o 1L), e indican si los sensores están tapados o destapados (ver figura 5.30).

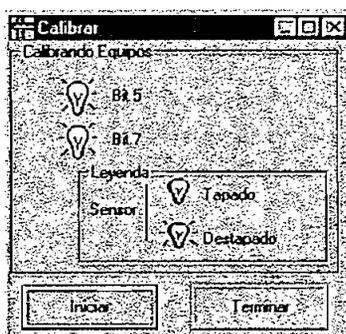


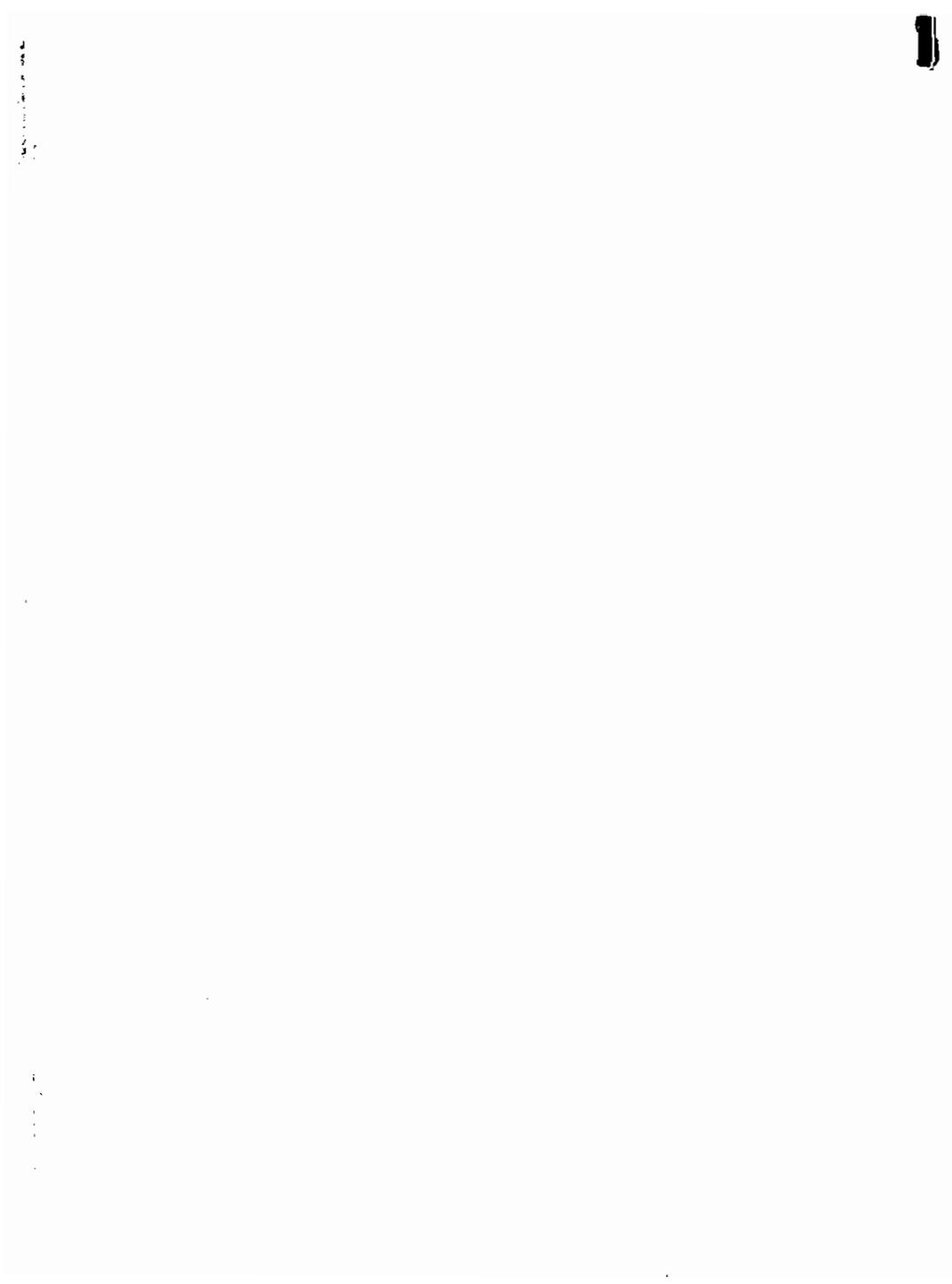
Figura 5.30.- Calibración del sistema bajo software.

**Ayuda.**- Mediante este botón, se accede al menú de ayuda en donde se encuentra un apartado teórico de los tres casos físicos en estudio.

**Imprimir.**- Presenta las opciones para impresión. Aquí se guardan los resultados en un archivo de texto o a su vez, se los envía a la impresora.



Figura 5.31.- Opciones de salida de los resultados prácticos.



**NOTA:** Se recomienda llevar los datos resultantes de la práctica a un archivo para posteriormente editarlos y prepararlos para la impresión final. Esto se lo hace porque el puerto paralelo está ocupado por el prototipo, y la impresión de resultados sólo puede ser posterior a la finalización de las prácticas. La opción de impresión directa será de utilidad siempre y cuando se disponga de una red de computadoras en las cuales se configure una impresora.

### CAIDA LIBRE.

Aquí se solicita el dato de la **gravedad** teórica. Así, se calcula el error generado en el desplazamiento del móvil, mediante la comparación de la distancia (L0-L1) calculada durante la práctica y el valor de distancia física del tramo (L0-L1), el mismo que es un valor fijo.

Los círculos rojos, cambiarán a verdes cuando los sensores detecten el paso del cuerpo.

La práctica consta de tres tramos (L0-L1, L1-L2 y L2-L3) y del punto inicial (L0). Para estos tramos, se calcula la velocidad, tiempo y distancia recorrida por el móvil. El error que se calcula en el tramo (L0-L1) es el único calculado, puesto que solo en este tramo se obtiene un valor de desplazamiento práctico.



Figura 5.32.- Práctica de "Caída Libre".

Esta ventana cuenta con el botón **Borrar resultados**, el cual permite borrar y limpiar la pantalla de resultados al momento de comenzar otra vez la práctica.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

## PLANO INCLINADO.

Los datos que esta ventana solicita son: **Gravedad** (teórica), **Rozamiento** (dado en tablas) e **Inclinación** del plano respecto a la horizontal.

De igual forma existen dos círculos rojos, los mismos que cambiarán a verdes cuando los sensores detecten el paso del cuerpo.

Los valores resultantes calculados de esta ventana son la **Velocidad**, **Tiempo** y **Distancia**.

Esta ventana cuenta con un botón (**Borrar resultados**), y sirve para mantener limpia la matriz de datos al momento de iniciar nuevamente la práctica.

El error que se calcula en el desplazamiento resulta de la comparación

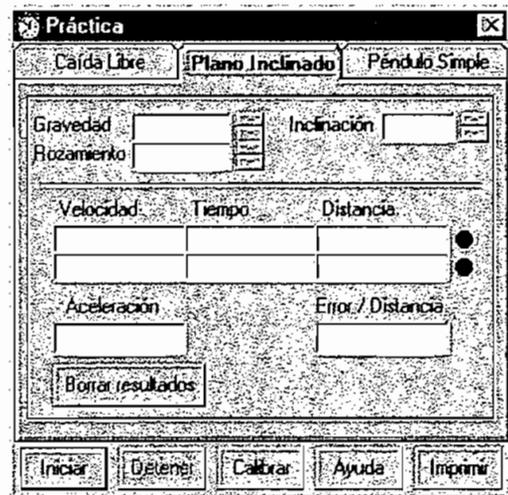


Figura 5.33.- Práctica en "Plano Inclinado".

de la distancia recorrida por el móvil, entre los dos sensores, y el valor de distancia física entre los mismos.

## PÉNDULO SIMPLE.

Para la práctica de péndulo simple, es necesario que el usuario ingrese la **gravedad** (teórica), el número de **Períodos** a ser medidos, y que escoja entre una **Masa** grande y una pequeña, así como una **Longitud** grande y una pequeña.

Para los resultados, se dispone de dos ventanas, en donde se desplegará la información de los tiempos obtenidos en cada oscilación. Dichos datos aparecen en una u otra ventana, dependiendo de la longitud de la cuerda utilizada, ya que se analizan dos sensores diferentes.

Además, una vez concluida la experiencia, se despliega la gravedad práctica resultante y el error generado al comparar esta última con el valor de gravedad teórica.

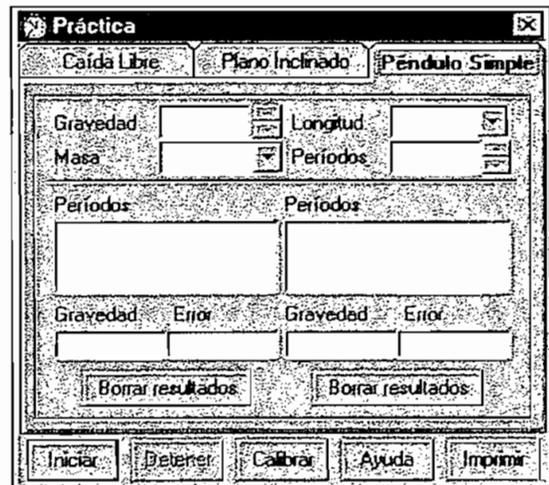


Figura 5.34.- Práctica de "Péndulo Simple".

Debajo de cada ventana de resultados, se encuentran los botones de Borrar resultados, para limpiar la matriz de datos.

4

1

1

### 5.4.2.3 OPCION "UNIDADES".

Este menú sirve para seleccionar el sistema de unidades con el que se va a trabajar. Esas opciones afectan tanto a la **Simulación**, como a la **Práctica**.

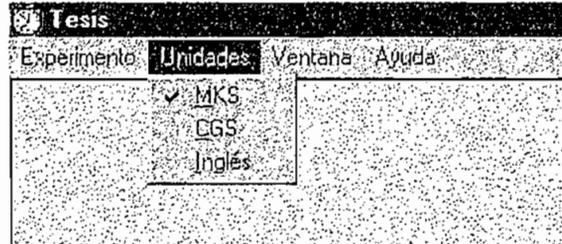


Figura 5.35.- Selección del sistema de unidades.

#### 5.4.2.3.1 SISTEMA DE UNIDADES MKS (Metro-Kilogramo-Segundo).

Con esta opción, el sistema trabajará en las unidades del sistema **MKS** (*Metros-Kilogramos-Segundos*). Las unidades son:

Longitud	Metros (m)
Masa	Kilogramos (kg)
Tiempo	Segundos (s)
Velocidad	(m/s)
Aceleración	(m/s <sup>2</sup> )
Angulo	Grados

Tabla 5.1.- Sistema de unidades MKS.

#### 5.4.2.3.2 SISTEMA DE UNIDADES CGS (Centímetro-Gramo-Segundo).

Con esta opción, el sistema trabajará en las unidades del sistema **CGS** (*Centímetros-Gramos-Segundos*). Las unidades son:

Longitud	Centímetros (cm)
Masa	Gramos (g)
Tiempo	Segundos (s)
Velocidad	(cm/s)
Aceleración	(cm/s <sup>2</sup> )
Angulo	Grados

Tabla 5.2.- Sistema de unidades CGS.



#### 5.4.2.3.3 SISTEMA DE UNIDADES INGLES (Pie-Libra-Segundo).

Este sistema es uno de los más antiguos, y se lo denominó así, porque nació en el Reino Unido. Las unidades a emplearse son:

Longitud	Pies
Masa	Libras (lb)
Tiempo	Segundos (s)
Velocidad	(pies/s)
Aceleración	(pies/s <sup>2</sup> )
Angulo	Grados

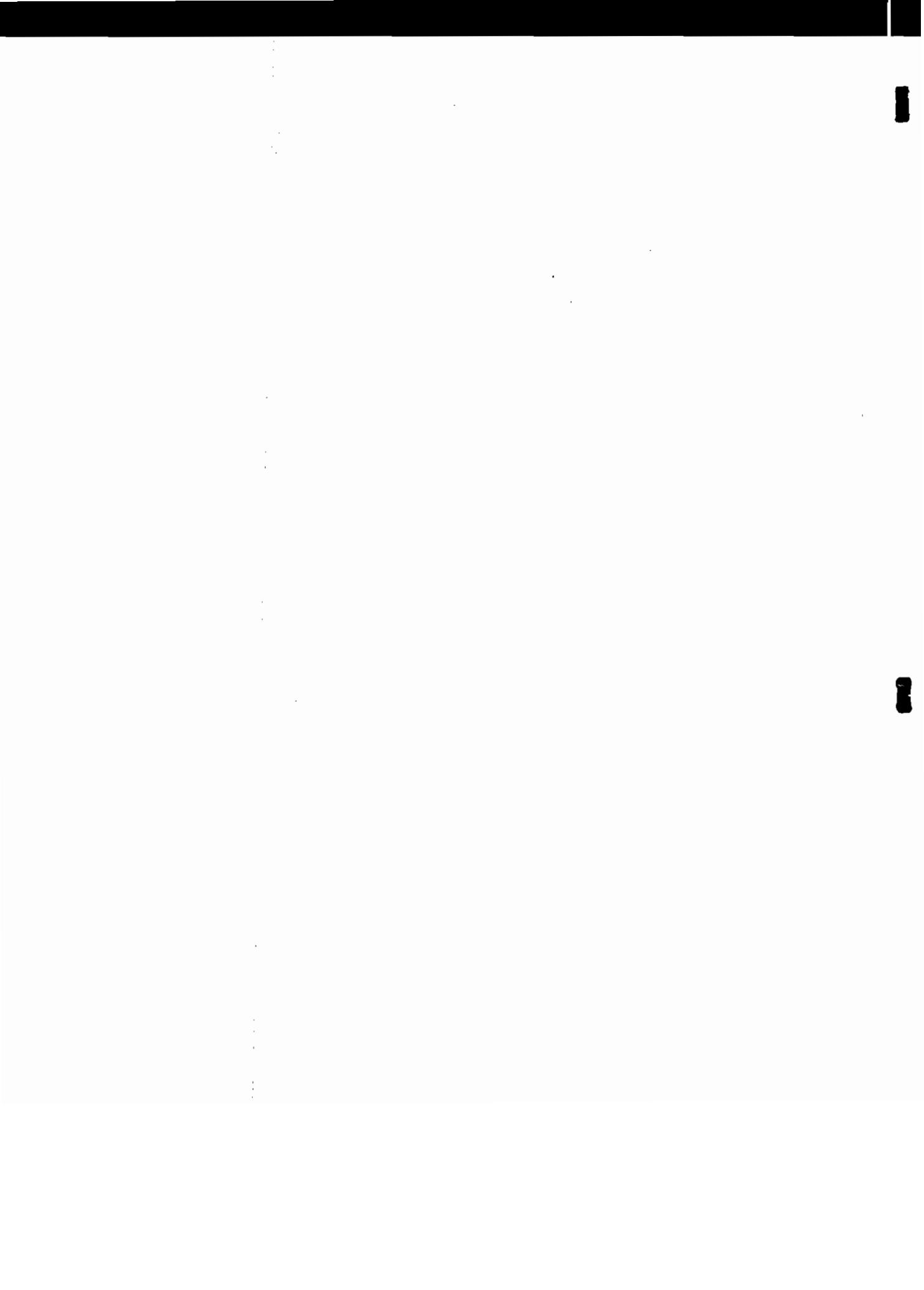
Tabla 5.3.- Sistema de unidades Inglés.

#### 5.4.2.4 OPCION "VENTANA".

Esta opción permite ordenar las ventanas de dos formas: **Cascada** y **Mosaico**. Es de gran utilidad cuando se tienen abiertas varias ventanas y existe dificultad en localizar alguna de ellas. De esta manera, el panorama se verá ordenado.



Figura 5.36.- Selección del tipo de visualización de ventana.



### 5.4.2.5 OPCION "AYUDA".

Esta opción mostrada en la *figura 5.37*, activa la ayuda del programa. Se trata de una pantalla, donde se expone el índice de temas, los cuales contienen la teoría requerida para cada una de las tres prácticas (*ver figura 5.38*).



Figura 5.37.- Acceso a la ventana de ayuda del programa.

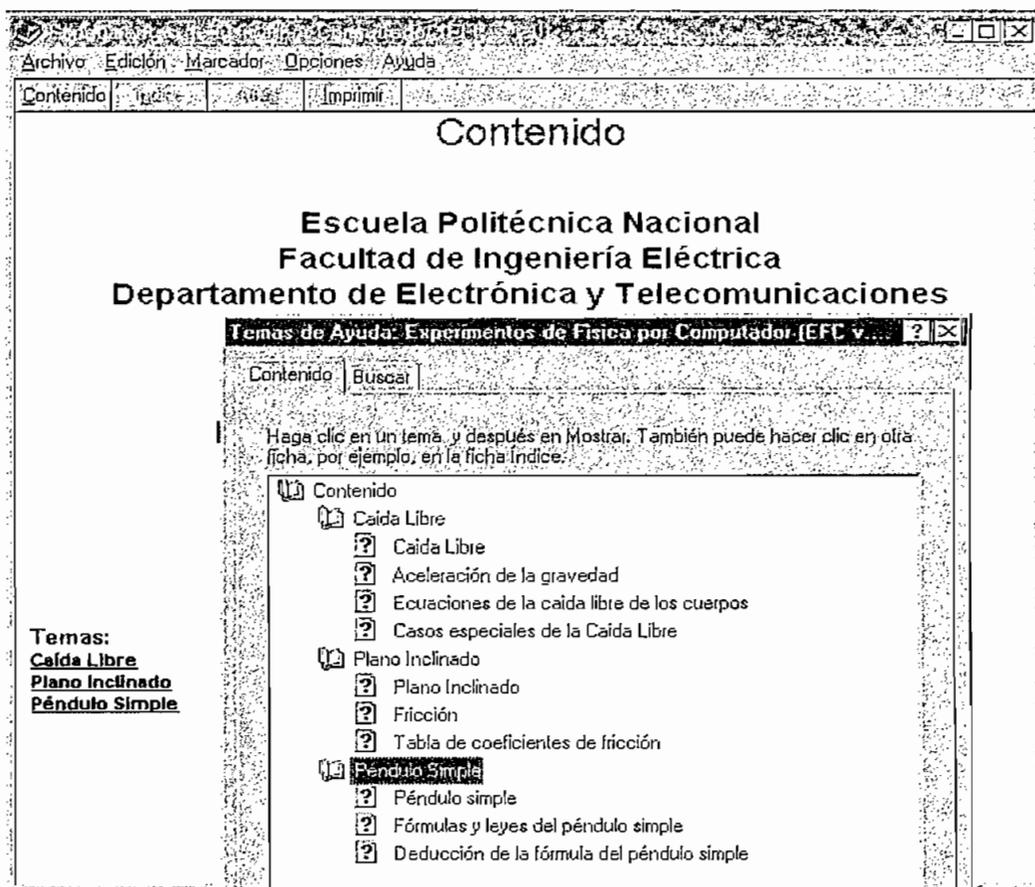


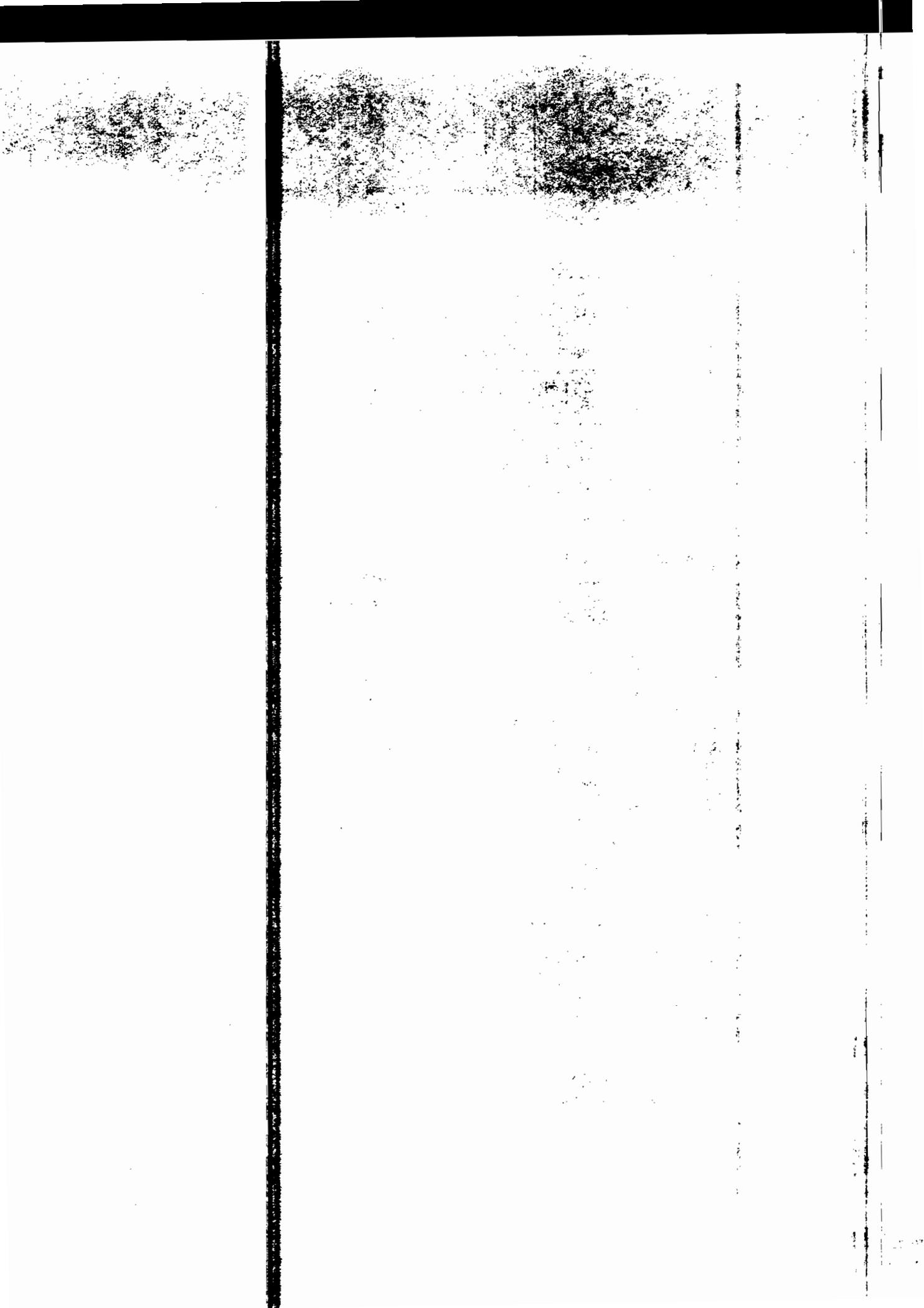
Figura 5.38.- Ventana de ayuda del programa EFC v1.0.



## ANEXOS

A. DIAGRAMAS DE LOS MODULOS FISICOS .....	215
A.1 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO	
“PENDULO SIMPLE” .....	215
A.1.1 Esquemas Gráficos .....	215
A.1.2 Tabla de longitudes físicas en [cm] .....	220
A.2 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO	
“CAIDA LIBRE” .....	220
A.2.1 Esquemas Gráficos .....	220
A.2.2 Tabla de longitudes físicas en [cm] .....	223
A.3 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO	
“PLANO INCLINADO” .....	224
A.3.1 Esquemas Gráficos .....	224
A.3.2 Tabla de longitudes físicas en [cm] .....	228
B. DISEÑO DE LOS CIRCUITOS IMPRESOS .....	229
B.1 DISEÑO DEL CIRCUITO IMPRESO DE “PENDULO SIMPLE” .....	229
B.2 DISEÑO DEL CIRCUITO IMPRESO DE “CAIDA LIBRE” .....	230
B.3 DISEÑO DEL CIRCUITO IMPRESO DE “PLANO INCLINADO” .....	231
C. MATERIALES UTILIZADOS .....	232
C.1 LISTA DE MATERIALES DEL PROTOTIPO “PENDULO SIMPLE” .....	232
C.2 LISTA DE MATERIALES DEL PROTOTIPO “CAIDA LIBRE” .....	233
C.3 LISTA DE MATERIALES DEL PROTOTIPO “PLANO INCLINADO” .....	234
D. DIAGRAMAS ELECTRONICOS FINALES .....	235
D.1 PENDULO SIMPLE .....	236
D.2 CAIDA LIBRE .....	237
D.3 PLANO INCLINADO .....	238
E. SITIOS WEB IMPORTANTES PARA VISITAR .....	239
E.1 SITIOS WEB PARA OBTENER EL "IOPORT.OCX" .....	239
E.2 INFORMACION ADICIONAL PARA EL PUERTO PARALELO .....	241
F. AYUDA DEL PROGRAMA COMPUTACIONAL "EFC" .....	245
G. DATOS TECNICOS DE LOS SEMICONDUCTORES .....	258

*Los ANEXOS E, F y G constan de información obtenida de libros, artículos, folletos y páginas WEB, la cual puede estar sujeta a errores. No es responsabilidad del autor cualquier error encontrado en estos literales.*



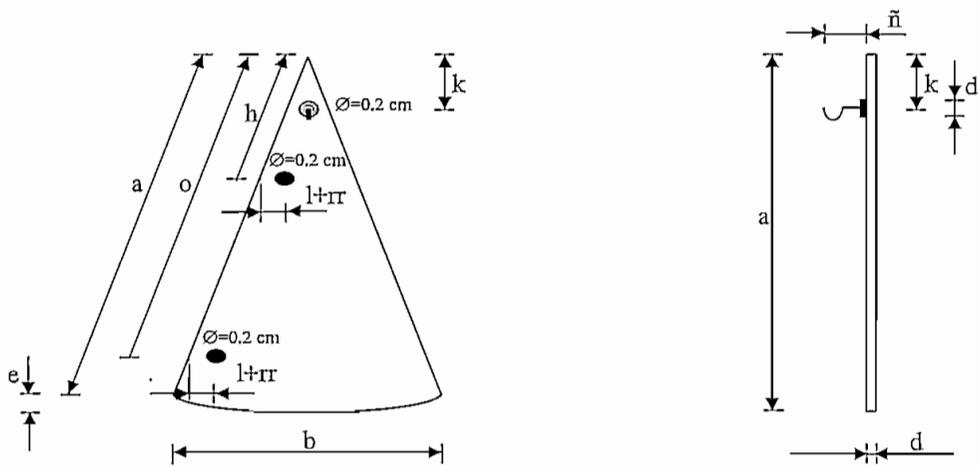
# ANEXOS

## A. DIAGRAMAS DE LOS MODULOS FISICOS.

### A.1 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO

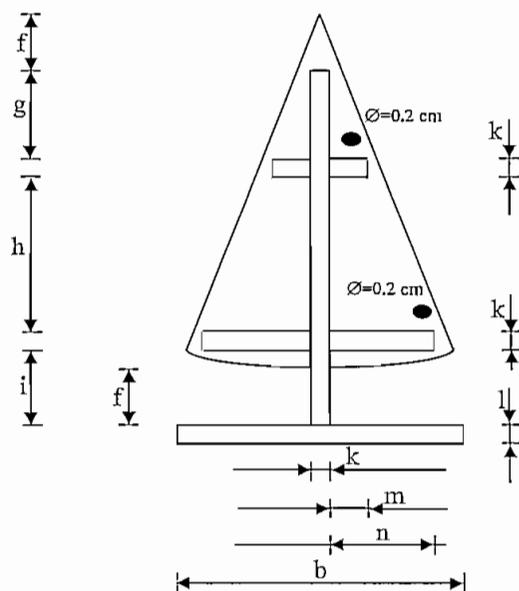
#### “PENDULO SIMPLE”.

##### A.1.1 Esquemas Gráficos.

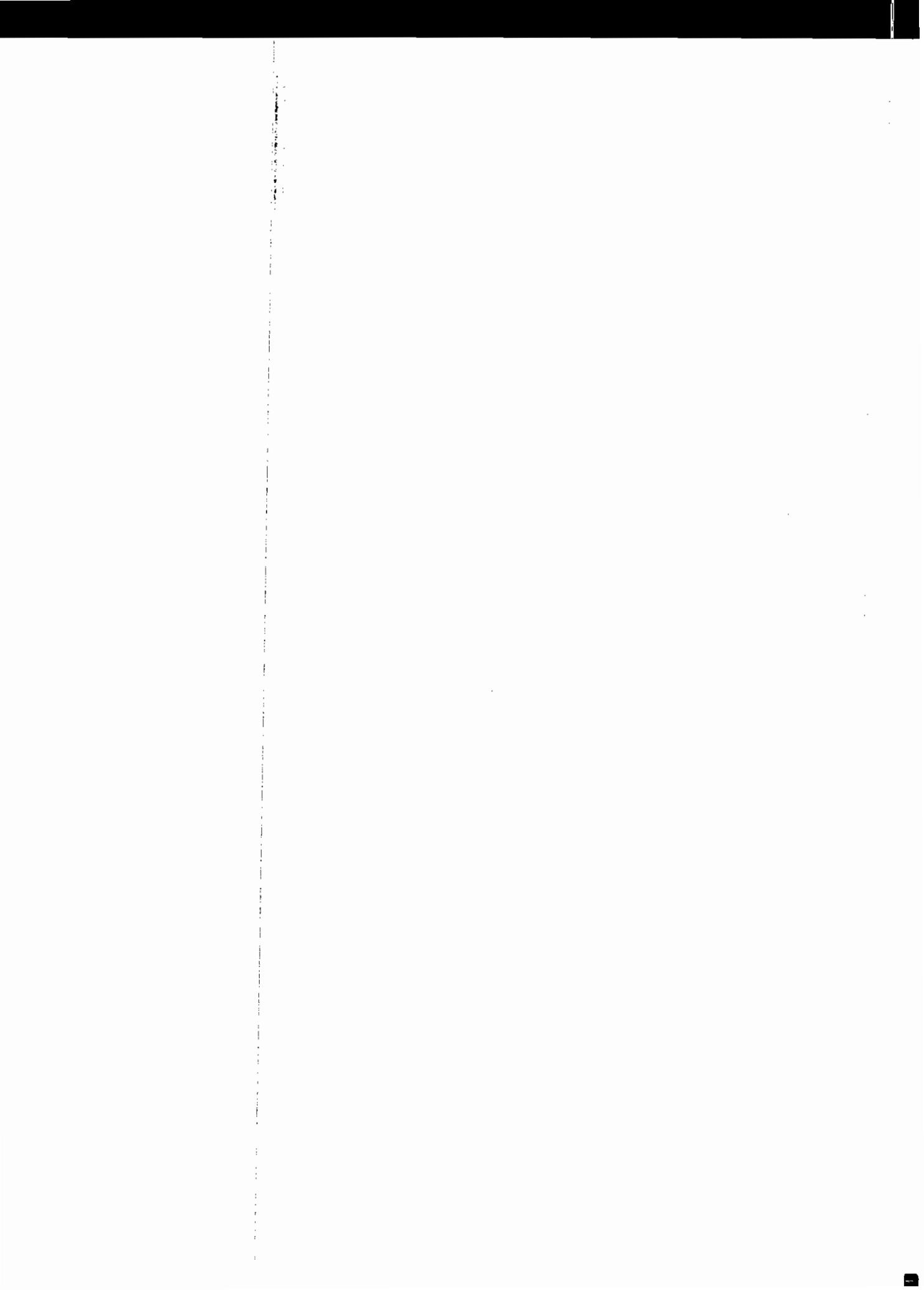


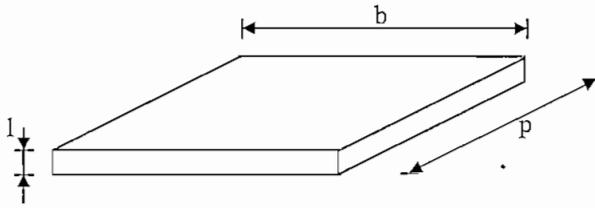
**Vista Frontal:  
Panel Vertical**

**Vista Lateral:  
Panel Vertical**

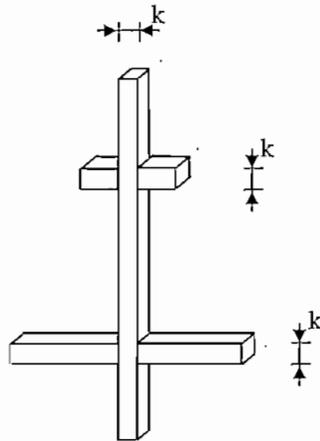


**Vista Posterior:  
Estructura de soporte  
del Péndulo.**

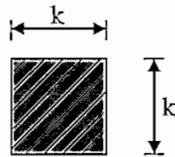




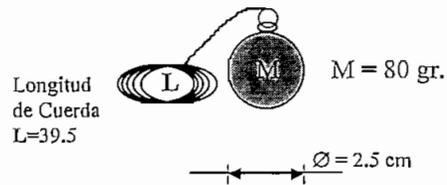
Base para soporte del Péndulo.



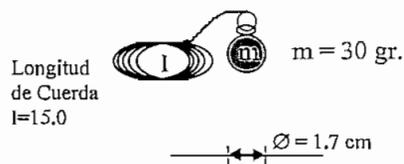
Estructura de Soporte del Péndulo.



Corte Transversal: Soporte del Péndulo.



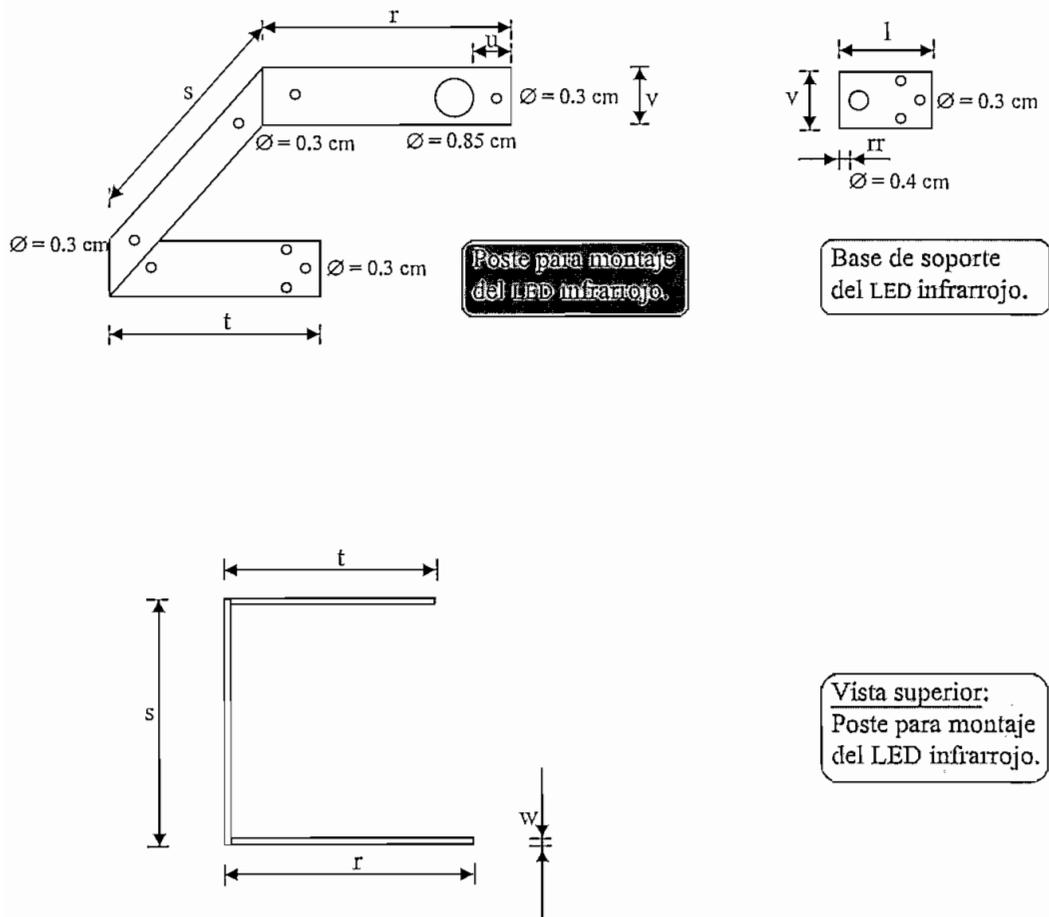
Esfera de acero de masa  $M$ .



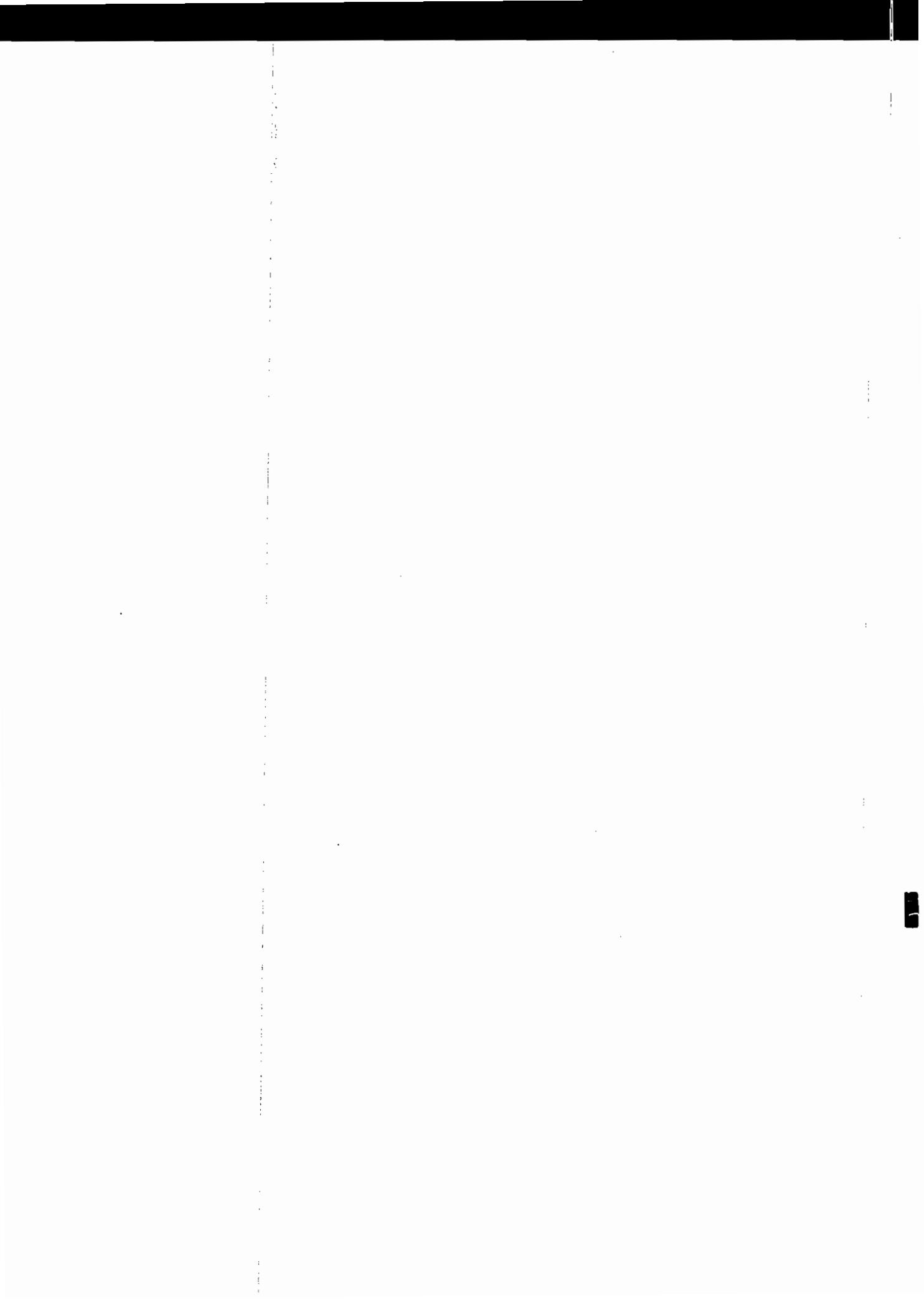
Esfera de acero de masa  $m$ .



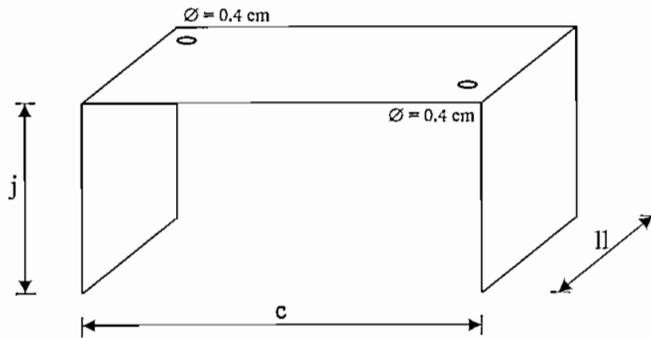
**NOTA:** El par de masas indicadas son esferas constituidas por rulimanes de acero adaptados a las necesidades del prototipo diseñado. Cada esfera dispone de un broche soldado en la periferia, en donde se coloca la cuerda que sujetará la masa contra el pivot, ubicado en la parte superior del panel vertical del prototipo de “péndulo simple”.



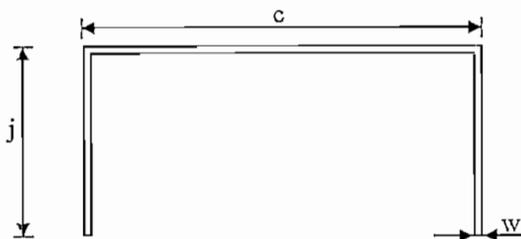
**NOTA:** Los postes de Iluminación están reforzados mediante el uso de “Brackets” (ángulos metálicos), en los sitios esquineros. La utilización de estas



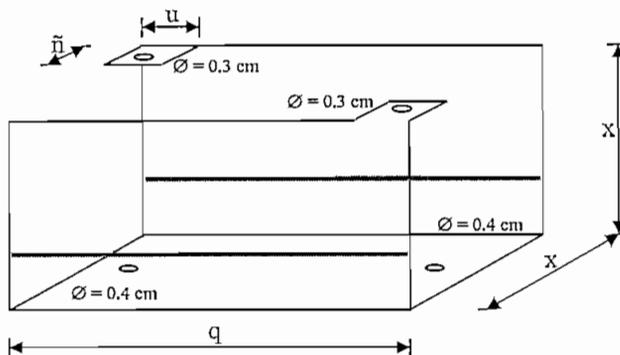
piezas, es para volver a los postes más compactos, debido a que éstos son  
 contruidos en acrílico de tres líneas y, en el sitio del dobléz, el material es más  
 propenso a roturas.



Tapa de la Caja  
 del Circuito  
 Interfaz.



Vista Frontal:  
 Tapa de la Caja  
 del Circuito Interfaz



Base de la caja del  
 circuito Interfaz.

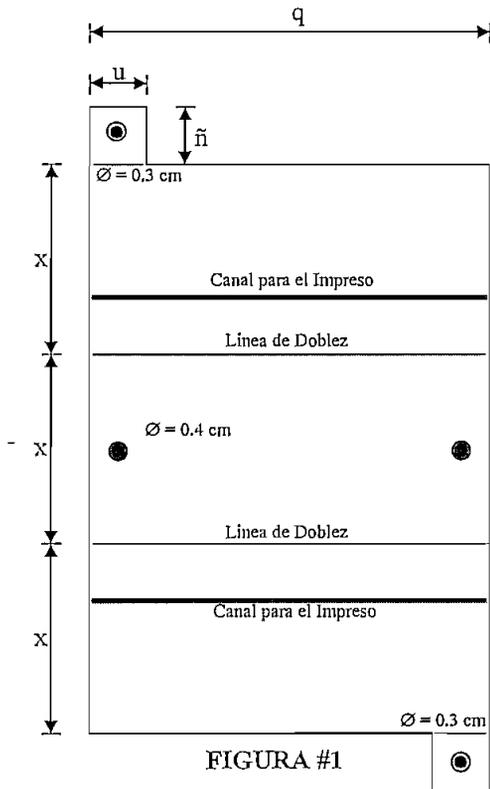


FIGURA #1

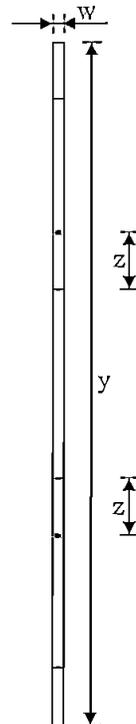
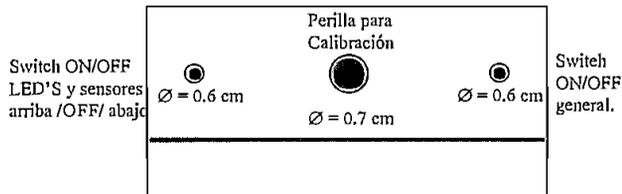
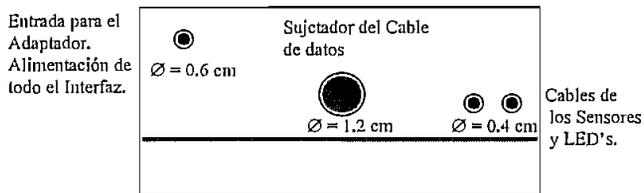


FIGURA #2

1. Vista de la base desdoblada de la Caja Interfaz.  
2. Vista Lateral.



Vista Frontal:  
Posicionamiento de la perilla e interruptores de control. Diámetros.



Vista Posterior:  
Diámetros de los huecos para los cables de los LED's, sensores, cable de datos y conexión de la alimentación

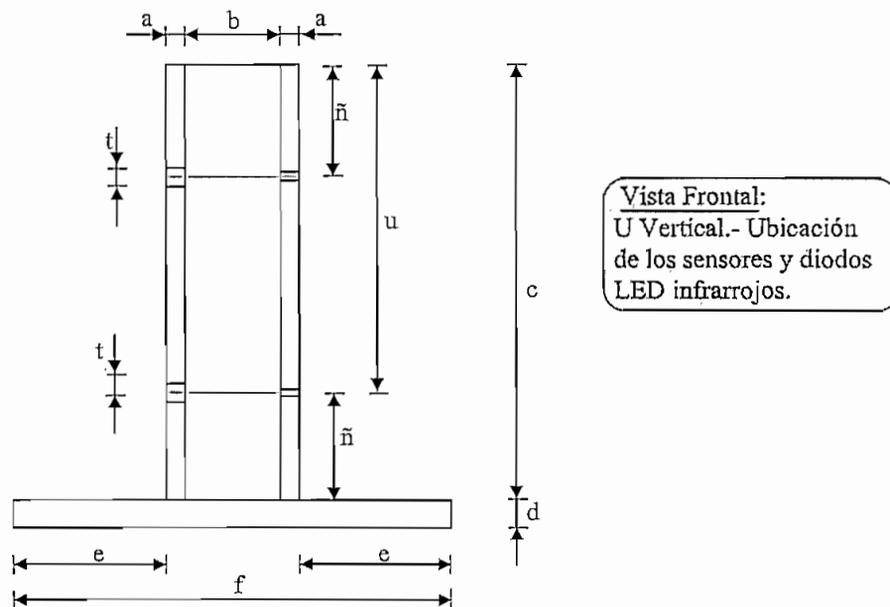
A.1.2 Tabla de longitudes físicas en [cm].

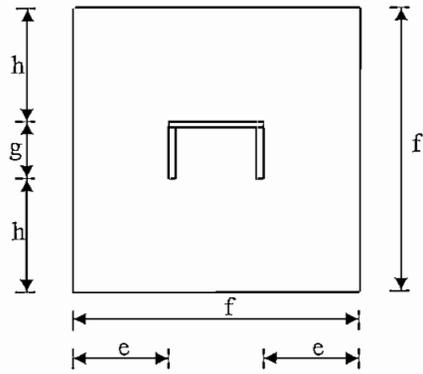
a	50.00	ñ	2.00
b	40.00	o	45.30
c	11.50	p	30.00
d	0.90	q	11.00
e	4.00	r	7.50
f	6.00	rr	0.20
g	14.00	s	9.30
h	20.10	t	7.40
i	10.00	u	1.50
j	6.80	v	2.10
k	2.90	w	0.30
l	2.30	x	6.30
ll	6.40	y	18.90
m	6.50	z	2.20
n	16.50		

A.2 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO

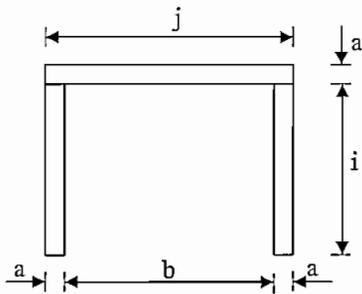
“CAIDA LIBRE”.

A.2.1 Esquemas gráficos.

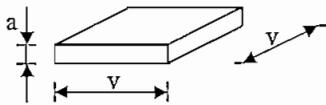




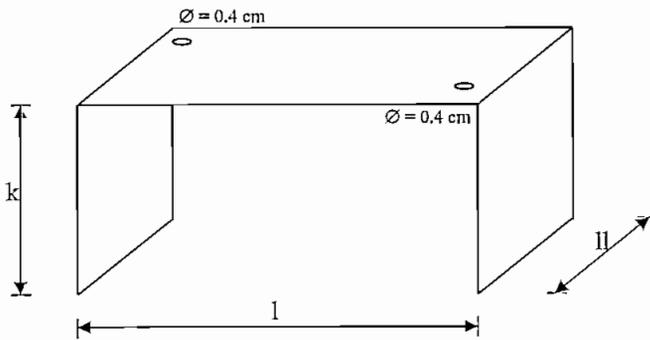
Vista Superior:  
U vertical y  
ubicación en la Base.



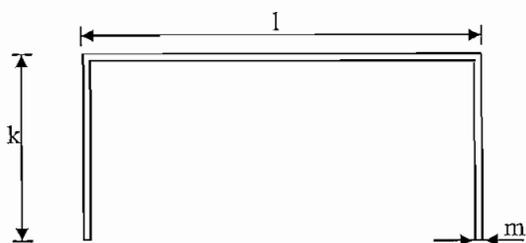
Vista Superior:  
U vertical.



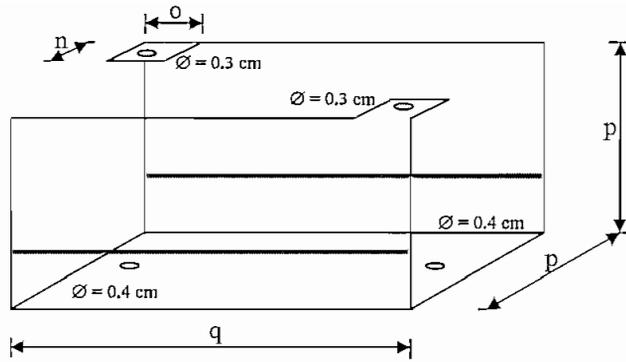
Cuerpo  
experimental.



Tapa de la caja  
del Circuito  
Interfaz.



Vista Frontal:  
Tapa de la Caja del  
Circuito Interfaz.



Base de la caja del circuito Interfaz.

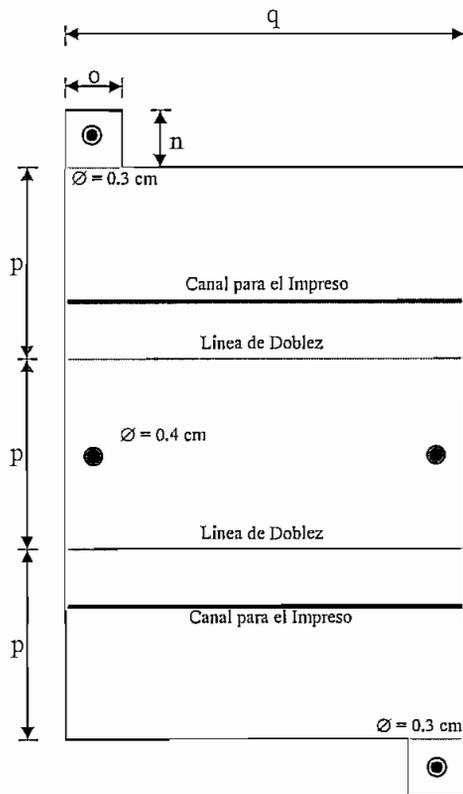
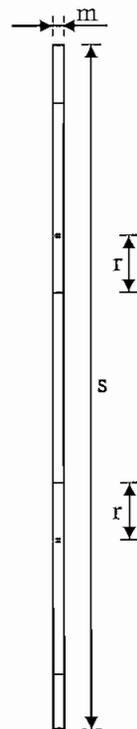
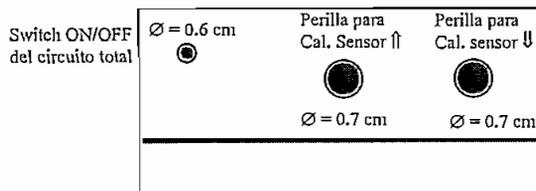


FIGURA #1



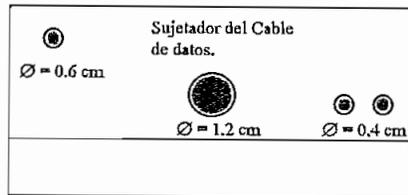
1. Base desdoblada de la caja Interfaz.  
2. Vista Lateral.

FIGURA #2



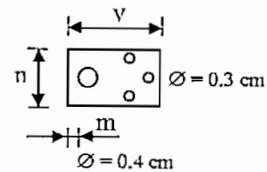
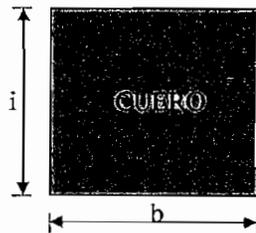
Vista Frontal:  
Posicionamiento de las perillas y el interruptor de control. Diámetros

Entrada de la Alimentación al Circuito Interfaz.



Orificios para cables de los Sensores y LED's.

**Vista Posterior:**  
Diámetros de los huecos para los cables de los LED's, sensores, cable de datos y conector para la alimentación del circuito.

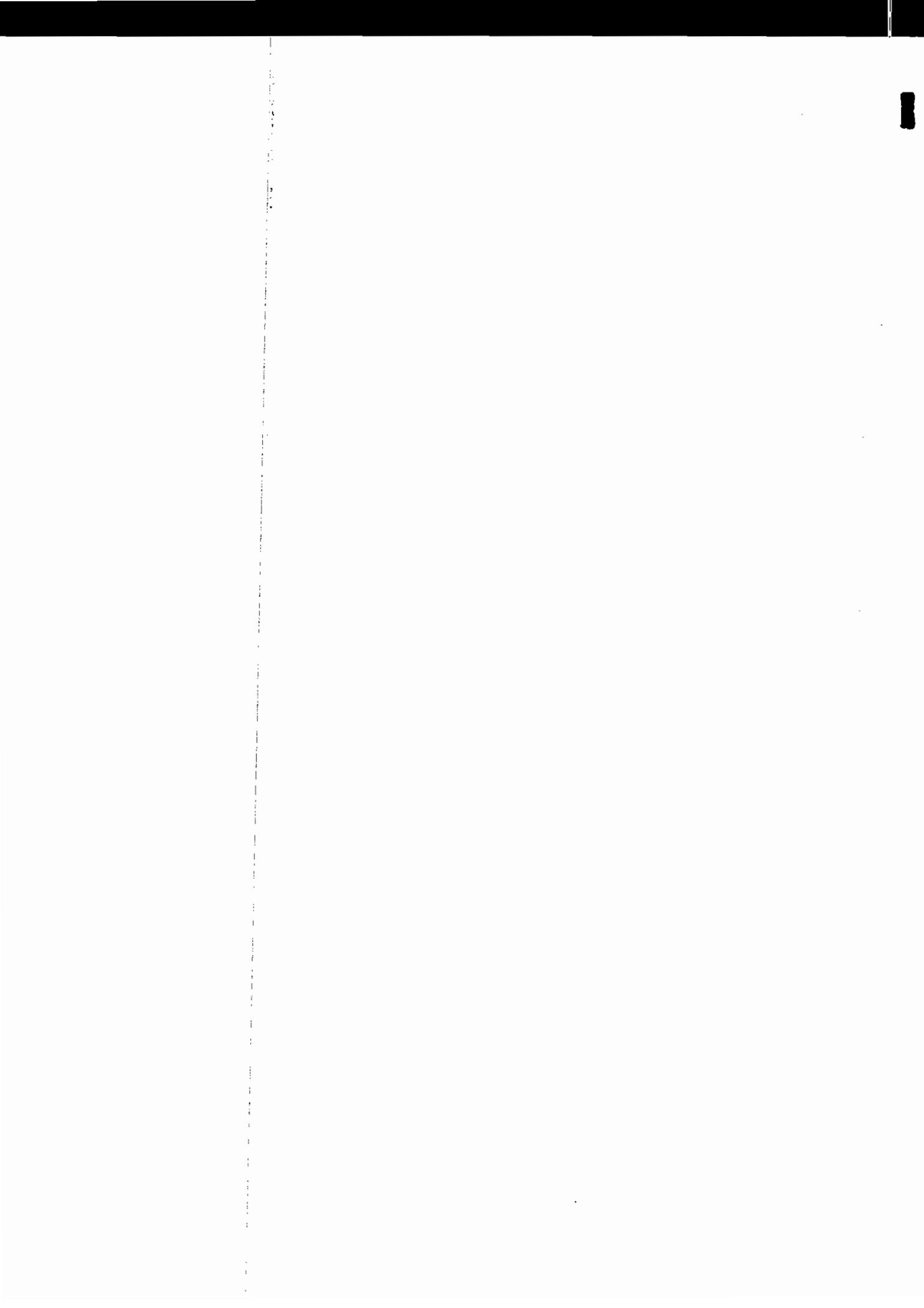


Pedazo de cuero para protección de la base del prototipo.

Soporte de apuntamiento y ajuste del LED infrarrojo.

**A.2.2 Tabla de longitudes físicas en [cm].**

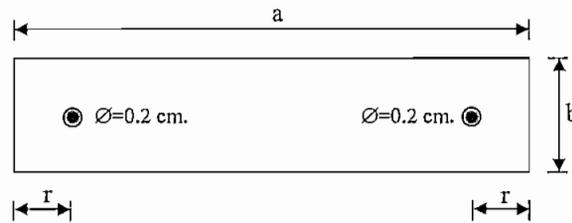
a	0.85	ll	6.40
b	5.60	m	0.30
c	56.00	n	2.00
d	2.20	ñ	10.00
e	11.25	o	1.50
f	30.00	p	6.30
g	6.00	q	11.00
h	12.00	r	2.20
i	5.00	s	18.90
j	7.30	t	0.40
k	6.80	u	46.00
l	11.50	v	3.00



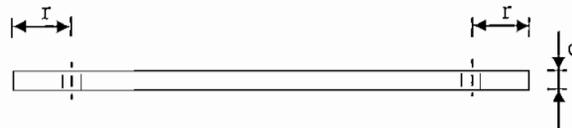
### A.3 DIAGRAMAS PARA LA CONSTRUCCION DEL PROTOTIPO

#### “PLANO INCLINADO”.

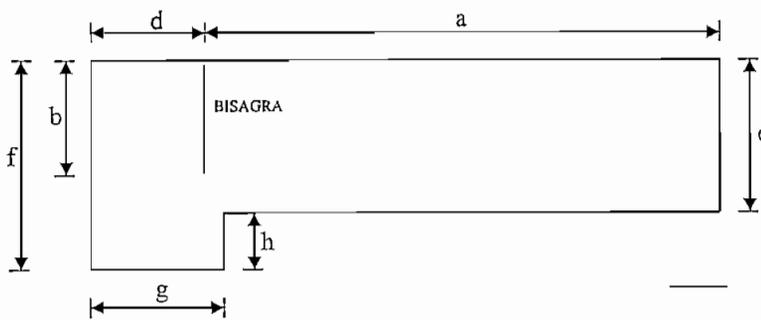
##### A.3.1 Esquemas gráficos.



Vista Superior:  
Pista del Móvil.

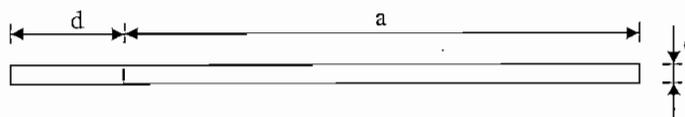


Vista Lateral:  
Pista del Móvil.

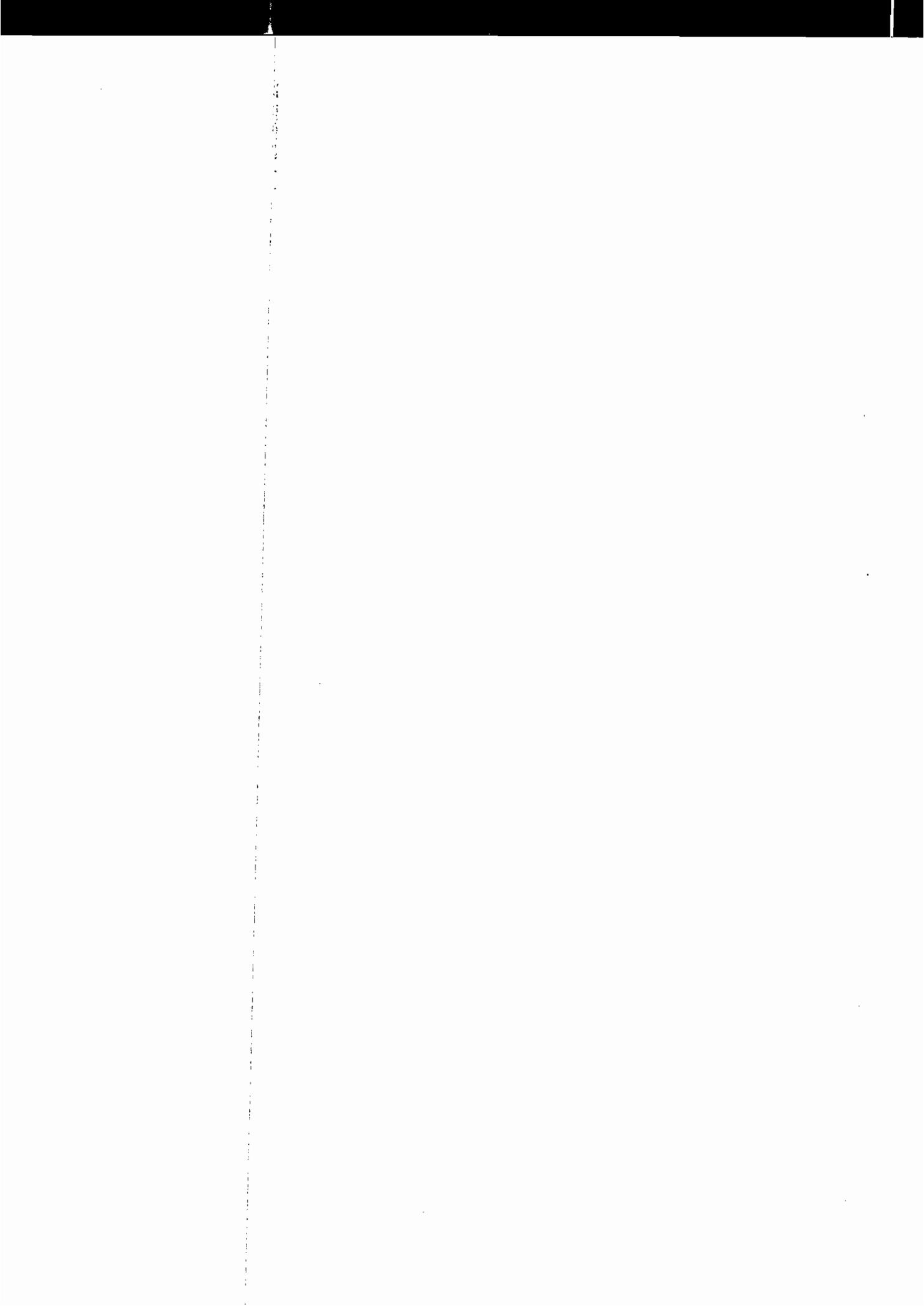


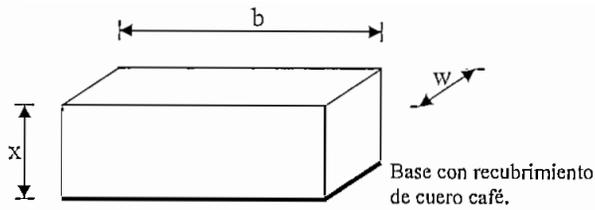
Vista Superior:  
Base del Plano.

Línea punteada que indica la posición de la bisagra.

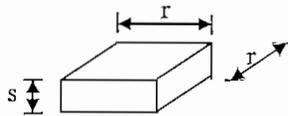


Vista Lateral:  
Base del Plano.



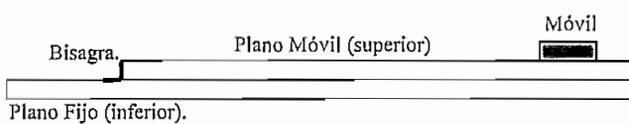


Bloque para apoyo del plano móvil.

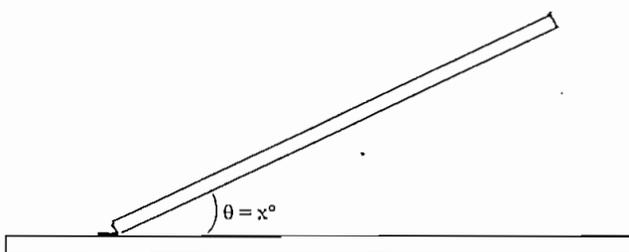


Móvil del Plano.  
Las dimensiones de éste, dependerán del usuario. (Ej:  $r=5\text{ cm}$ )

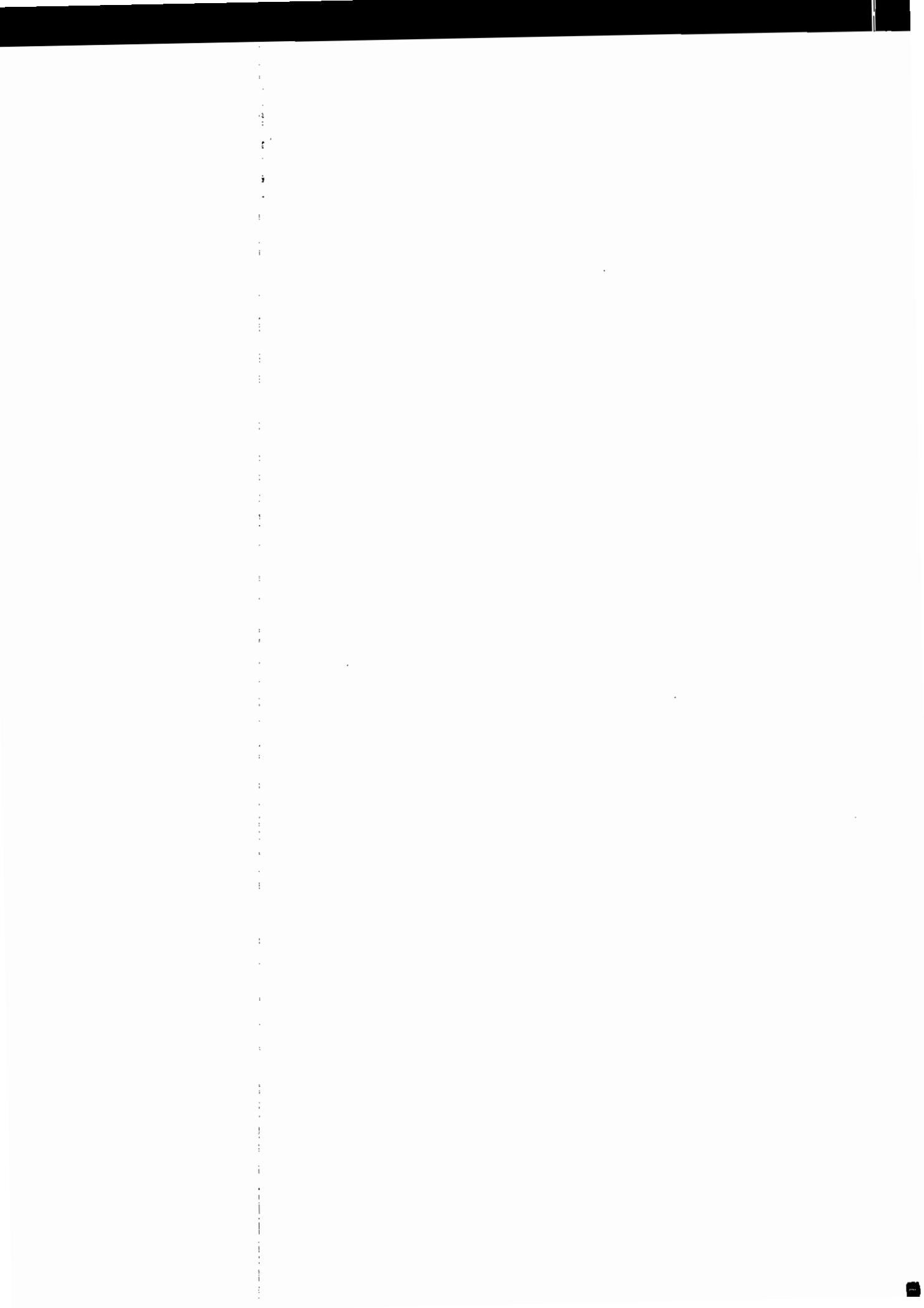
**NOTA:** El móvil utilizado será cualquier objeto cuya superficie sea plana para un mejor contacto con el plano inclinado. De acuerdo al coeficiente de rozamiento  $\mu$ , valor que es característico de los materiales, se tendrá diferentes ángulos de inclinación para que el móvil rompa el estado de reposo.

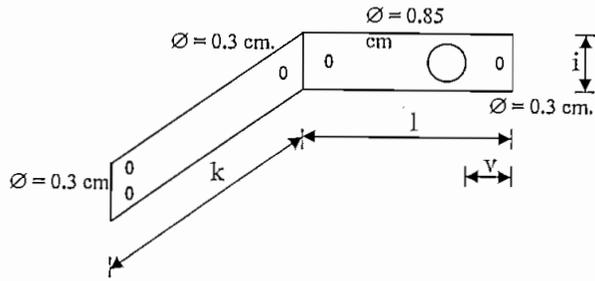


Vista del plano inclinado cuando el ángulo de inclinación es de  $\theta = 0^\circ$ .

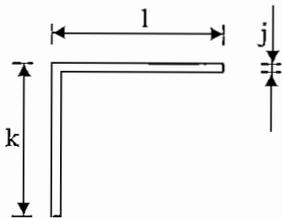


Vista del Plano Inclinado cuando el ángulo de inclinación es de  $\theta = x^\circ$

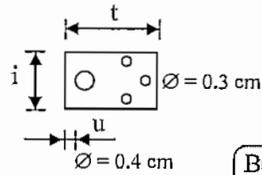




Poste para montaje del LED infrarrojo.

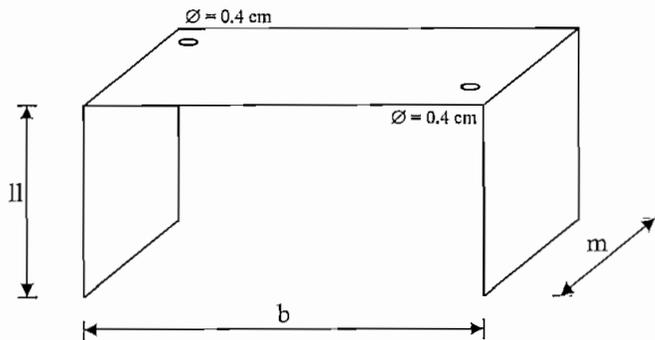


Vista Lateral:  
Poste para montaje del LED infrarrojo.

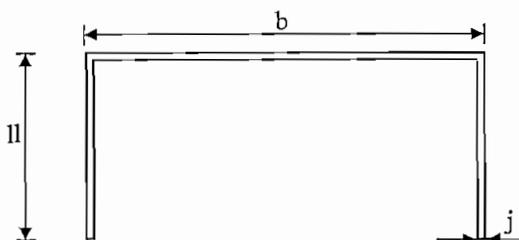


Base de soporte del LED infrarrojo.

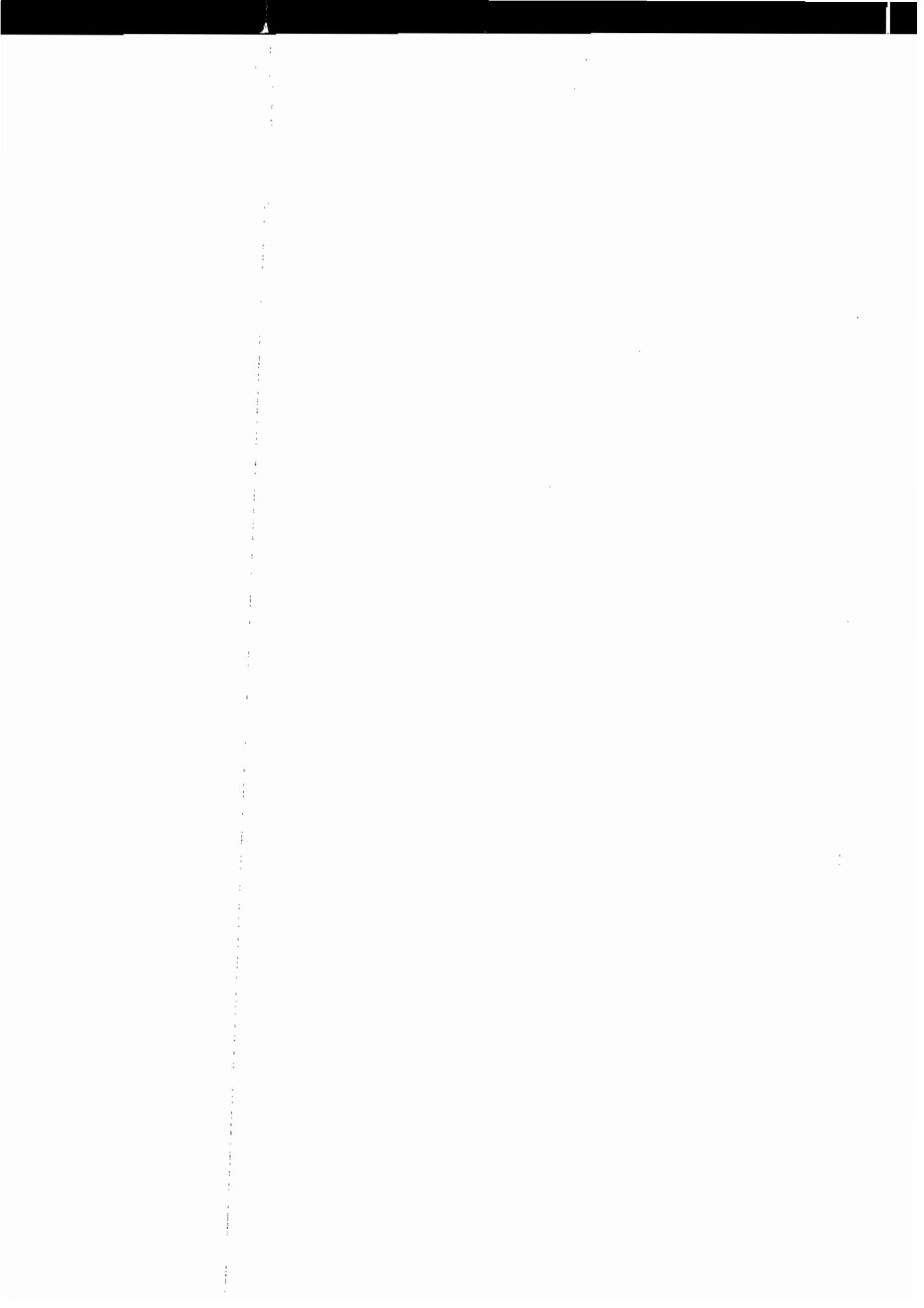
**NOTA:** Los postes de Iluminación se encuentran reforzados en sus esquinas con piezas angulares (Brackets).

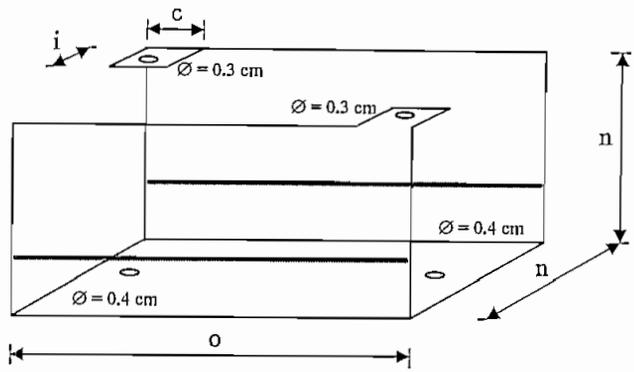


Tapa de la caja del Circuito Interfaz.



Vista Frontal:  
Tapa de la caja del circuito interfaz.





Base de la caja del circuito Interfaz.

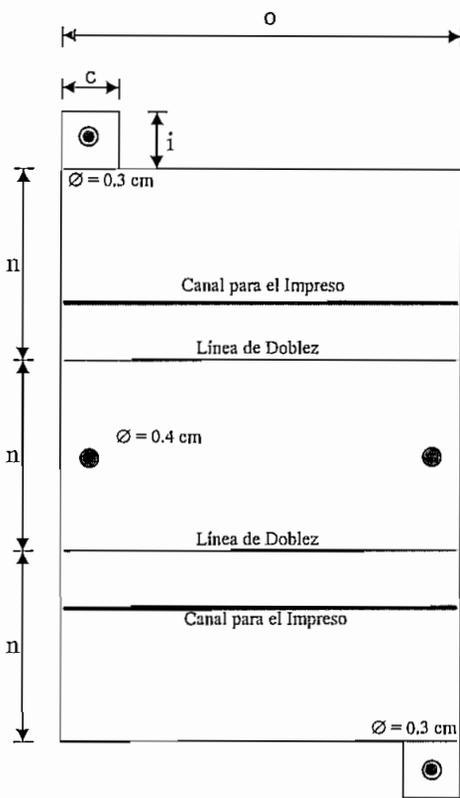
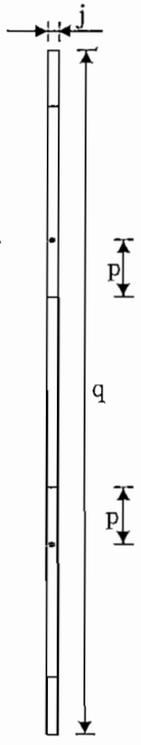
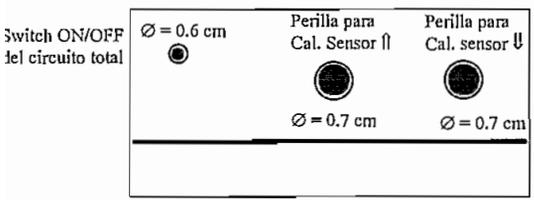


FIGURA #1

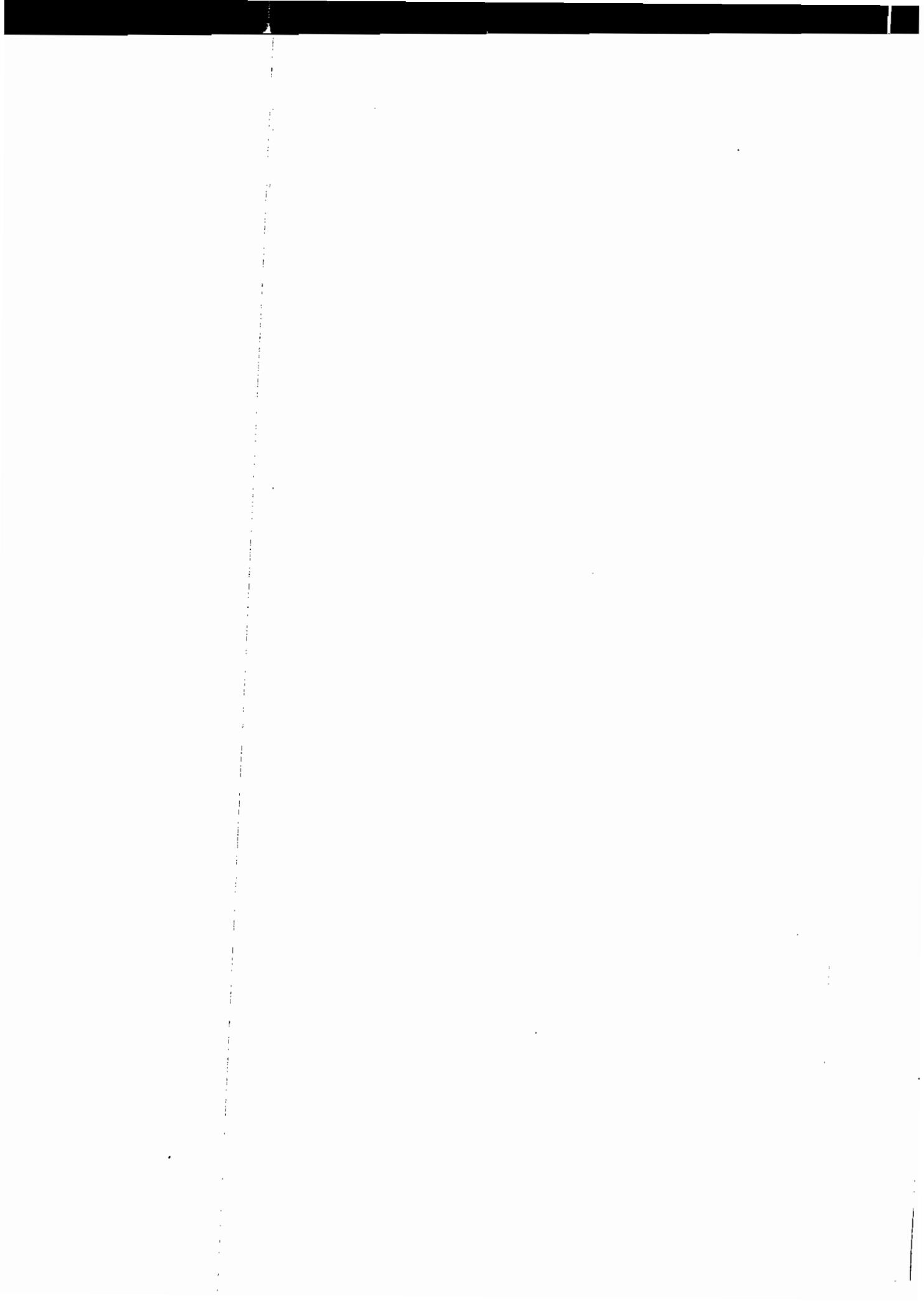


1. Base desdoblada de la caja Interfaz.  
2. Vista lateral.

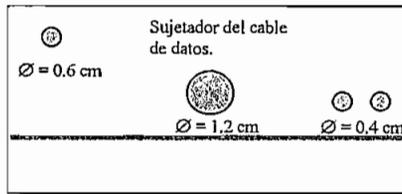
FIGURA #2



Vista Frontal:  
Posicionamiento de la perilla e interruptores de control. Diámetros



Entrada para el Adaptador o Fuente de Alimentación.

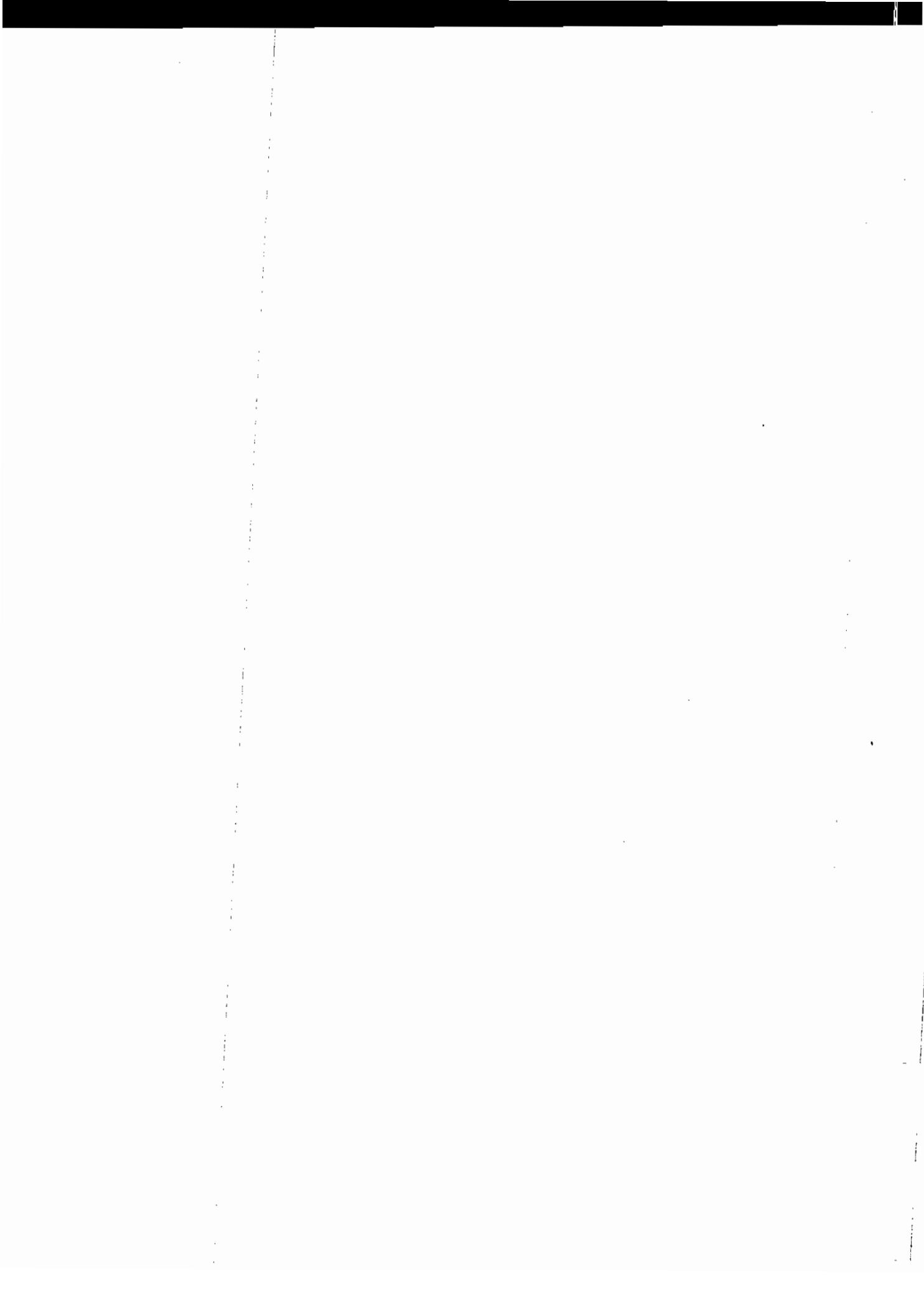


Salidas de los Sensores y LED's.

**Vista Posterior:**  
Diámetros de los huecos para los cables de los sensores, LED's, cable de datos y conector para la alimentación.

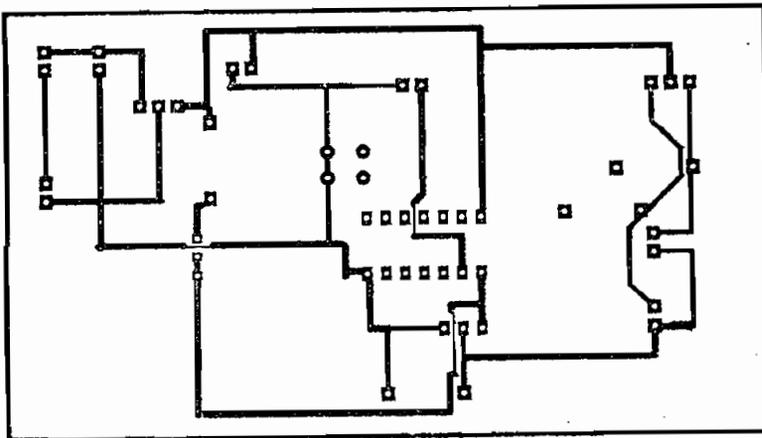
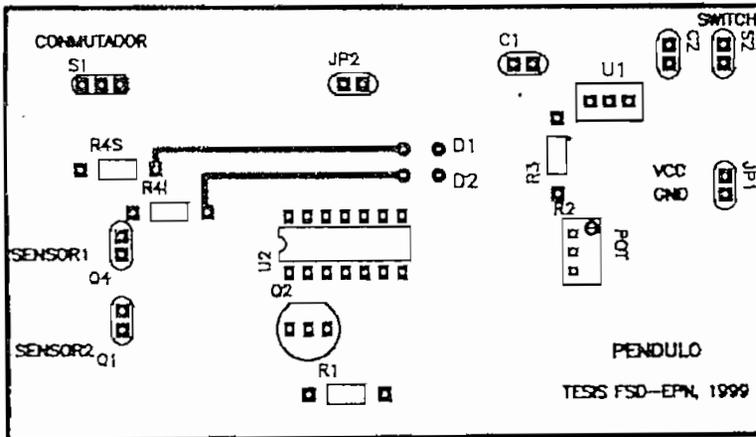
### A.3.2 Tabla de longitudes físicas en [cm].

a	50.00	m	6.40
b	11.40	n	6.30
c	1.20	o	11.00
d	10.00	p	2.20
e	14.50	q	18.90
f	22.50	r	5.00
g	13.50	s	0.85
h	8.00	t	2.40
i	2.10	u	0.20
j	0.30	v	1.50
k	8.40	w	4.00
l	7.60	x	3.00
ll	6.80		



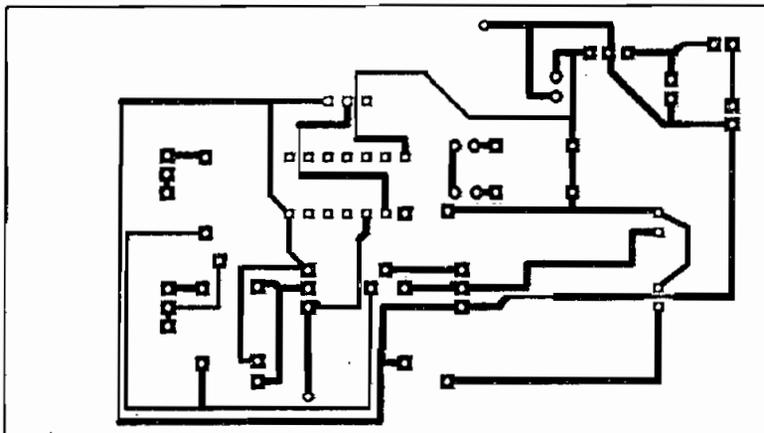
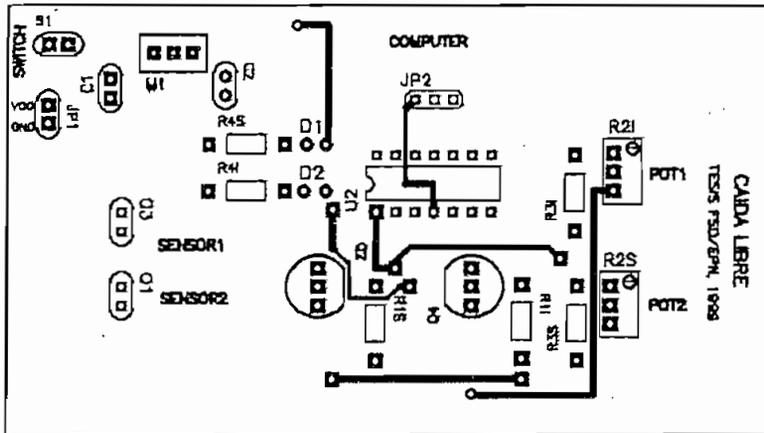
## B. DISEÑO DE LOS CIRCUITOS IMPRESOS.

### B.1 DISEÑO DEL CIRCUITO IMPRESO DE "PENDULO SIMPLE".



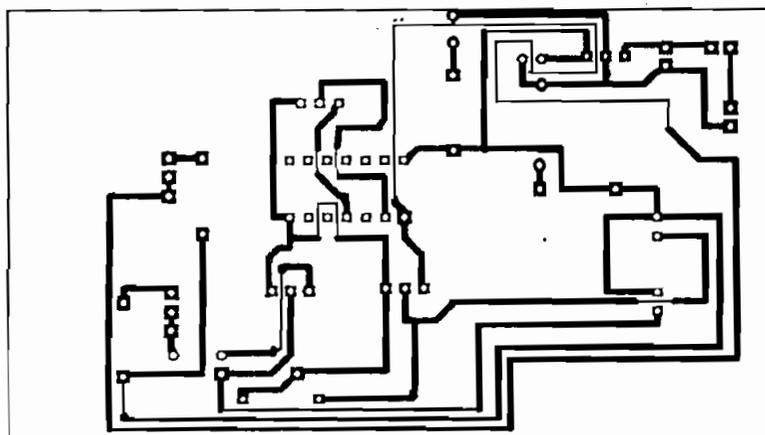
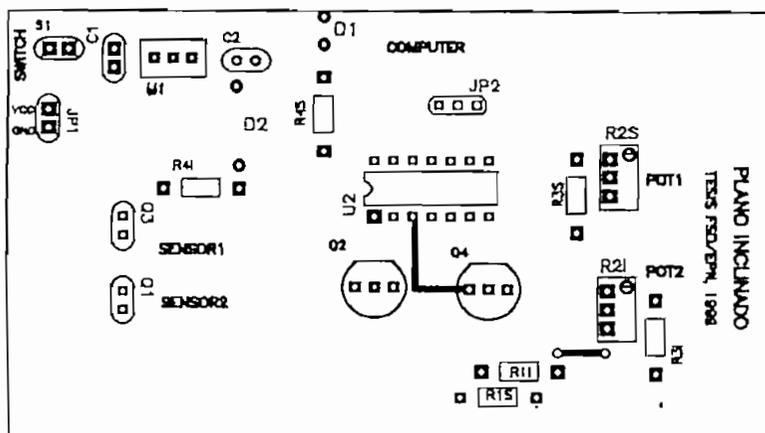


## B.2 DISEÑO DEL CIRCUITO IMPRESO DE "CAIDA LIBRE".





### B.3 DISEÑO DEL CIRCUITO IMPRESO DE "PLANO INCLINADO".





## C. MATERIALES UTILIZADOS.

### C.1 LISTA DE MATERIALES DEL PROTOTIPO "PENDULO SIMPLE".

TIPO	CANTIDAD	ELEMENTO(S)	CARACTERISTICAS
RESISTENCIA	1	R1	120 K, 1/4 W
	1	R2	POTENCIOMETRO 1 K, EJE LARGO PARA PANEL
	1	R3	56, 1/4W
	2	R4	150, 1/4 W
CAPACITOR	2	C1, C2	0.01 uF.
SEMICONDUCTORES	1	U1	7805C, 1 A
	1	U2	SN74LS14 HEX SCHMITT TRIGGER
	2	D1, D2	ECG 3017 DIODO LED INFRARROJO
	2	Q1, Q3	NTE 3034A FOTOTRANSISTOR DE LUZ VISIBLE/INFRARROJA
	1	Q2	2N3904 BJT DE PROPOSITO GENERAL TIPO NPN
VARIOS	1		ADAPTADOR AC/DC UNIVERSAL in: 110/220V AC 60/50 HZ, out: 3, 4.5, 6, 7, 5, 9, 12, V DC, 500 mA.
	1	S1	INTERRUPTOR DOBLE DE 3 POSICIONES
	1	S2	INTERRUPTOR SIMPLE
	1	J1	CONECTOR TIPO "JACK" HEMBRA PARA PANEL
	1		PERILLA
	1		ZOCALO DE 14 PINES
	1		CIRCUITO IMPRESO DE 60mmX100mm
	14		ESPADINES PARA CIRCUITO IMPRESO
	2		CONECTORES EN LINEA DE 3 PINES
	4		CONECTORES EN LINEA DE 2 PINES
	1		CONECTOR MINICOM HEMBRA DE 3 PINES
	1		CONECTOR MINICOM MACHO DE 3 PINES PARA CIRCUITO IMPRESO
	3		METROS DE CABLE FTP DE 4 HILOS
	1		SUJETADOR DE CABLE PARA PANEL
	1		CUERDA METÁLICA DE LONGITUD 39.5 cm
	1		CUERDA METÁLICA DE LONGITUD 15.0 cm
	1		RULIMAN DE ACERO DE 2.5 cm DE DIAMETRO
	1		RULIMAN DE ACERO DE 1.7 cm DE DIAMETRO
	4		BRACKETS (ANGULOS METALICOS) PARA EL REFORZAMIENTO DE ESQUINAS PLASTICAS DE 2.5cm X 2.5cm X 1.2cm
	1		CAJA DE ACRILICO DE 11.6cm X 6.8cm X 6.5cm (3 LINEAS)
	2		POSTES DE ACRILICO DE 7.5cm X 9.3cm X 7.5cm X 2.1 cm (3 LINEAS)
	4		TORNILLOS PARA SUJECION DE LA CAJA DE ACRILICO
	10		TORNILLOS CON TUERCA PARA LOS POSTES
	6		TORNILLOS PARA SOPORTE DE LOS POSTES
	1		CONECTOR DB-25 MACHO CON TAPA
	2		FORROS PROTECTORES PLASTICOS PARA LOS DIODOS LED.
	1		MODULO FISICO DE MADERA (ESPECIFICACIONES POR DISEÑO)
	---		AMARRAS PLASTICAS PEQUEÑAS
	---		GRAPAS
	---		CINTA AUTOFUNDENTE 3M
	5		ARANDELAS
	1		GANCHO METALICO
	---		SOLDADOR, SUELDA DE ESTAÑO Y PASTA LIMPIADORA.



## C.2 LISTA DE MATERIALES DEL PROTOTIPO "CAIDA LIBRE".

TIPO	CANTIDAD	ELEMENTO(S)	CARACTERISTICAS
RESISTENCIA	2	R1	120 K, 1/4 W
	2	R2	POTENCIOMETRO 1 K, EJE LARGO PARA PANEL
	2	R3	56, 1/4W
	2	R4	150, 1/4 W
CAPACITOR	2	C1, C2	0,01 uF.
SEMICONDUCTORES	1	U1	7805C, 1 A
	1	U2	SN74LS14 HEX SCHMITT TRIGGER
	2	D1, D2	ECG 3017 DIODO LED INFRARROJO
	2	Q1, Q3	NTE 3034A FOTOTRANSISTOR DE LUZ VISIBLE/INFRARROJA
	2	Q2, Q4	2N3904 BJT DE PROPOSITO GENERAL TIPO NPN
VARIOS	1		ADAPTADOR AC/DC UNIVERSAL in: 110/220V AC 60/50 Hz, out: 3, 4.5, 6, 7, 5, 9, 12, V DC, 500 mA.
	1	S1	INTERRUPTOR SIMPLE
	1	J1	CONECTOR TIPO "JACK" HEMBRA PARA PANEL
	2		PERILLAS
	1		ZOCALO DE 14 PINES
	1		CIRCUITO IMPRESO DE 60mmX100mm
	14		ESPADINES PARA CIRCUITO IMPRESO
	2		CONECTORES EN LINEA DE 3 PINES
	4		CONECTORES EN LINEA DE 2 PINES
	1		CONECTOR MINICOM HEMBRA DE 3 PINES
	1		CONECTOR MINICOM MACHO DE 3 PINES PARA CIRCUITO IMPRESO
	3		METROS DE CABLE FTP DE 4 HILOS
	1		SUJETADOR DE CABLE PARA PANEL
	1		MOVIL DE MADERA (ESPECIFICACIONES POR DISEÑO)
	1		TROZO DE CUERO DE 5.5cm X 5.0cm
	1		CAJA DE ACRILICO DE 11.6cm X 6.8cm X 6.5cm (3 LINEAS)
	2		SOPORTES DE ACRILICO DE 3.0cm X 201cm (3 LINEAS)
	4		TORNILLOS PARA SUJECION DE LA CAJA DE ACRILICO
	6		TORNILLOS PARA LOS SOPORTES DE ACRILICO
	1		CONECTOR DB-25 MACHO CON TAPA
2		FORROS PROTECTORES PLASTICOS PARA LOS DIODOS LED.	
1		MODULO FISICO DE MADERA (ESPECIFICACIONES POR DISEÑO)	
---		AMARRAS PLASTICAS PEQUEÑAS	
---		GRAPAS	
---		CINTA AUTOFUNDENTE 3M	
---		SOLDADOR, SUELDA DE ESTAÑO Y PASTA LIMPIADORA.	

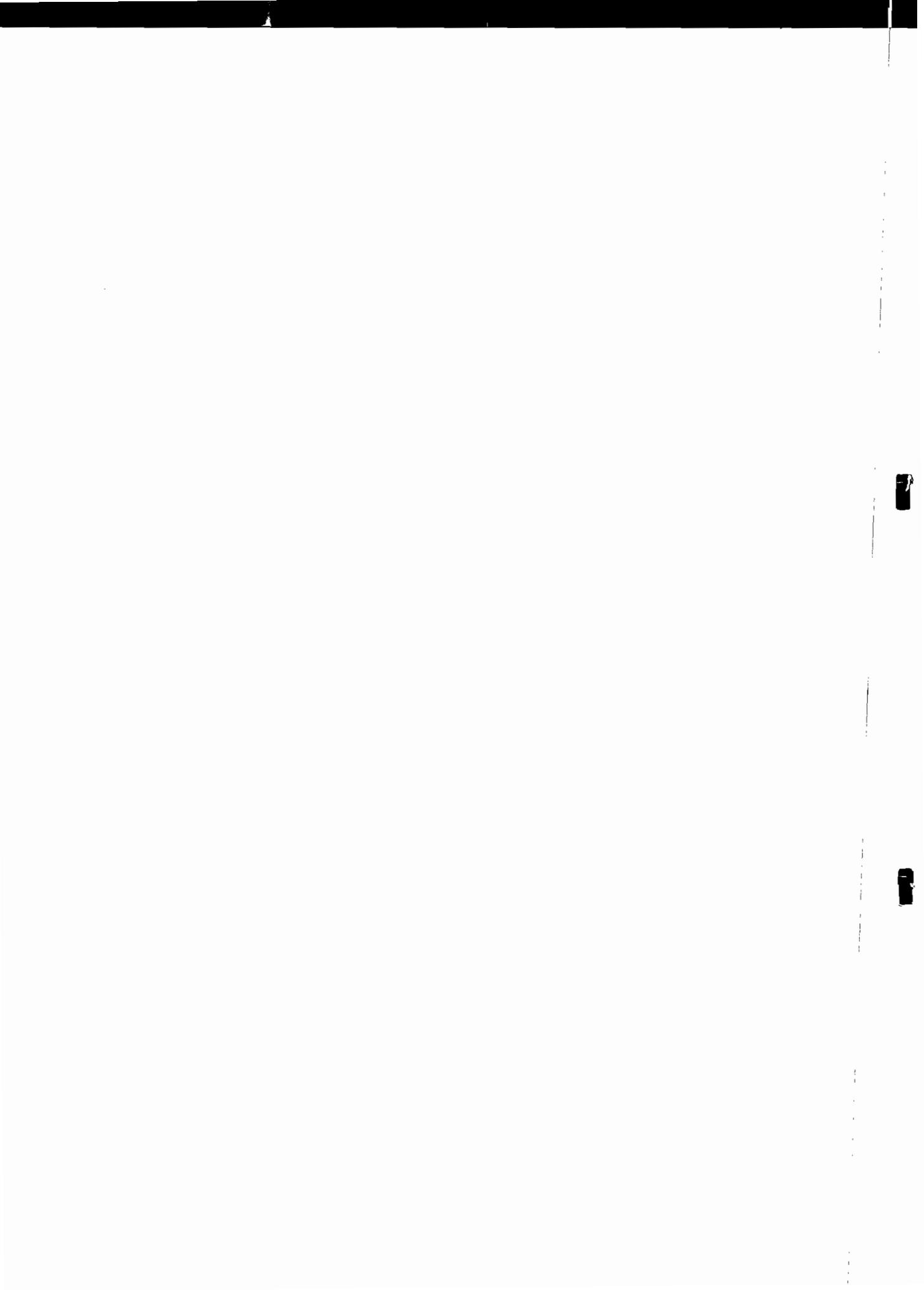


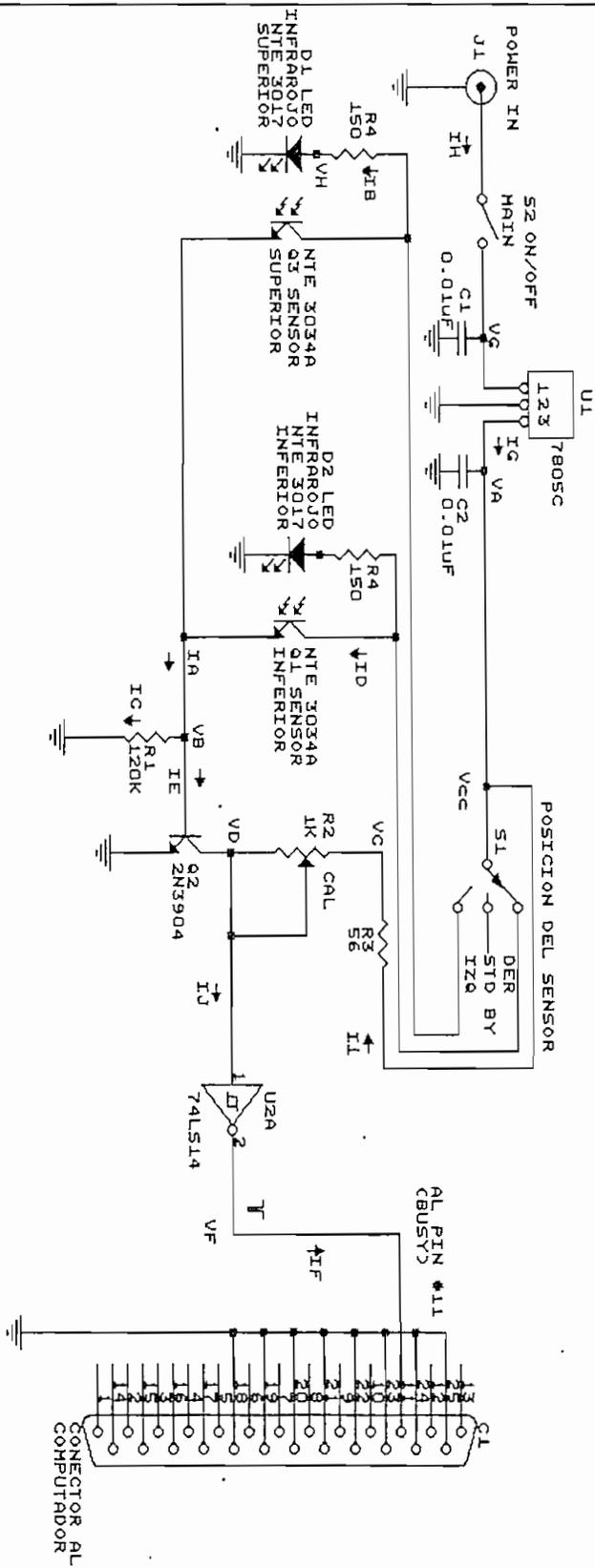
### C.3 LISTA DE MATERIALES DEL PROTOTIPO "PLANO INCLINADO".

TIPO	CANTIDAD	ELEMENTO(S)	CARACTERISTICAS
RESISTENCIA	2	R1	120 K, 1/4 W
	2	R2	POTENCIOMETRO 1 K, EJE LARGO PARA PANEL
	2	R3	56, 1/4W
	2	R4	150, 1/4 W
CAPACITOR	2	C1, C2	0,01 uF.
SEMICONDUCTORES	1	U1	7805C, 1 A
	1	U2	SN74LS14 HEX SCHMITT TRIGGER
	2	D1, D2	ECG 3017 DIODO LED INFRARROJO
	2	Q1, Q3	NTE 3034A FOTOTRANSISTOR DE LUZ VISIBLE/INFRARROJA
	2	Q2, Q4	2N3904 BJT DE PROPOSITO GENERAL TIPO NPN
VARIOS	1		ADAPTADOR AC/DC UNIVERSAL in: 110/220V AC 60/50 Hz, out: 3, 4.5, 6, 7, 5, 9, 12, V DC, 500 mA.
	1	S1	INTERRUPTOR SIMPLE
	1	J1	CONECTOR TIPO "JACK" HEMBRA PARA PANEL
	2		PERILLAS
	1		ZOCALO DE 14 PINES
	1		CIRCUITO IMPRESO DE 60mmX100mm
	14		ESPADINES PARA CIRCUITO IMPRESO
	2		CONECTORES EN LINEA DE 3 PINES
	4		CONECTORES EN LINEA DE 2 PINES
	1		CONECTOR MINICOM HEMBRA DE 3 PINES
	1		CONECTOR MINICOM MACHO DE 3 PINES PARA CIRCUITO IMPRESO
	3		METROS DE CABLE FTP DE 4 HILOS
	1		SUJETADOR DE CABLE PARA PANEL
	2		BRACKETS (ANGULOS METALICOS) PARA EL REFORZAMIENTO DE ESQUINAS PLASTICAS DE 2.5cm X 2.5cm X 1.2cm
	1		CAJA DE ACRILICO DE 11.6cm X 6.8cm X 6.5cm (3 LINEAS)
	2		POSTES DE ACRILICO DE 7.7cm X 8.4cm X 2.1cm (3 LINEAS)
	4		TORNILLOS PARA SUJECION DE LA CAJA DE ACRILICO
	6		TORNILLOS CON TUERCA PARA LOS POSTES
	4		TORNILLOS PARA SOPORTE DE LOS POSTES
	1		CONECTOR DB-25 MACHO CON TAPA
	2		FORROS PROTECTORES PLASTICOS PARA LOS DIODOS LED.
	1		MODULO FISICO DE MADERA (ESPECIFICACIONES POR DISEÑO)
	1		SOPORTE DE MADERA CON BASE DE CUERO (ESPECIFICACIONES POR DISEÑO)
	—		AMARRAS PLASTICAS PEQUEÑAS
	1		GRADUADOR
	—		GRAPAS
	—		CINTA AUTOFUNDENTE 3M
	2		ARANDELAS
	1		MOVIL DE MADERA PARA LAS PRUEBAS (ESPECIFICACIONES POR DISEÑO).
	—		SOLDADOR, SUELDA DE ESTAÑO Y PASTA LIMPIADORA.



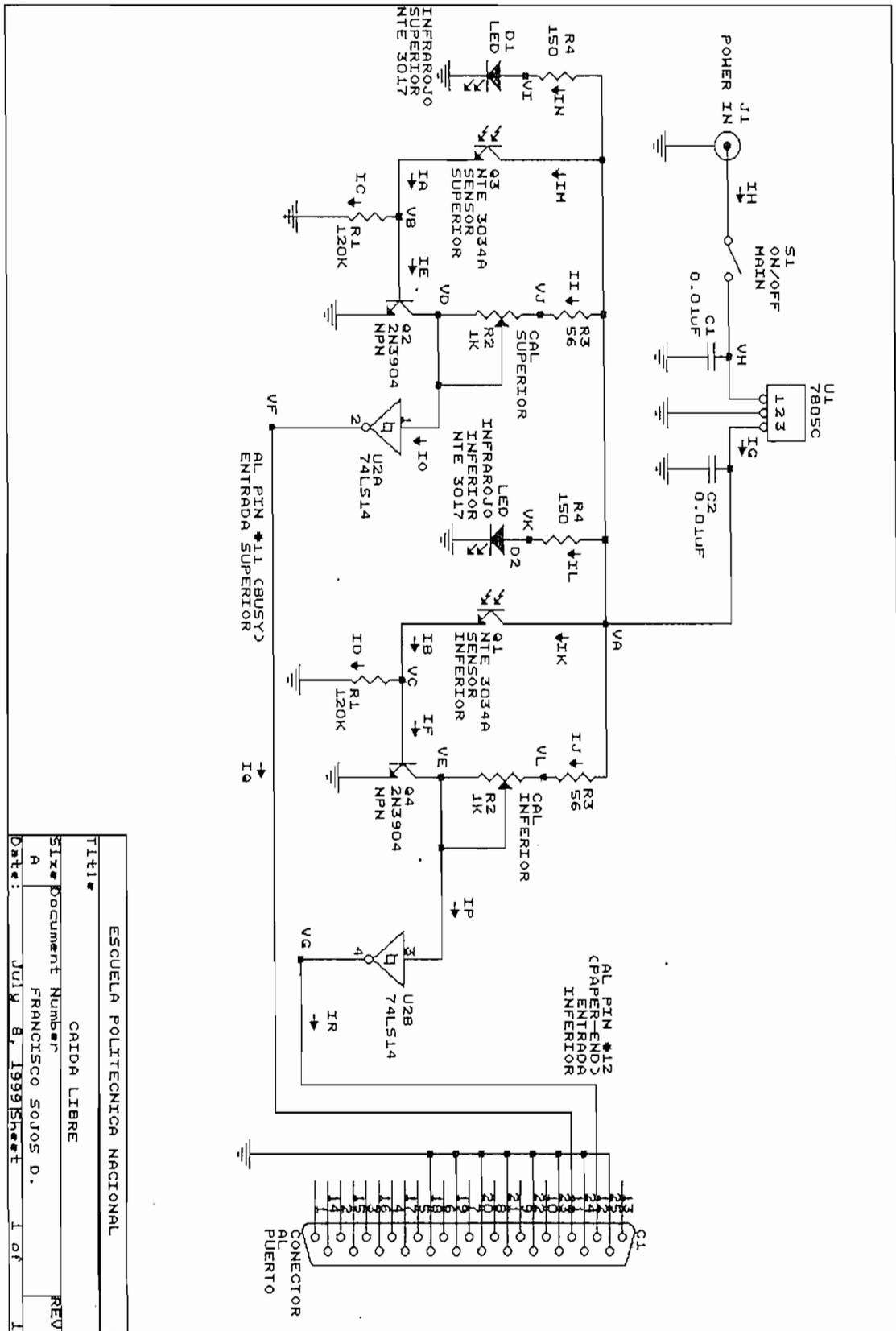
## **D. DIAGRAMAS ELECTRONICOS FINALES.**





ESCUELA POLITECNICA NACIONAL	
Titulo	PENDULO SIMPLE
Size	Document Number
A	FRANCISCO SOJOS D.
Date:	August 19, 1993
Sheet	1 of 1
REV	





ESCUELA POLITECNICA NACIONAL

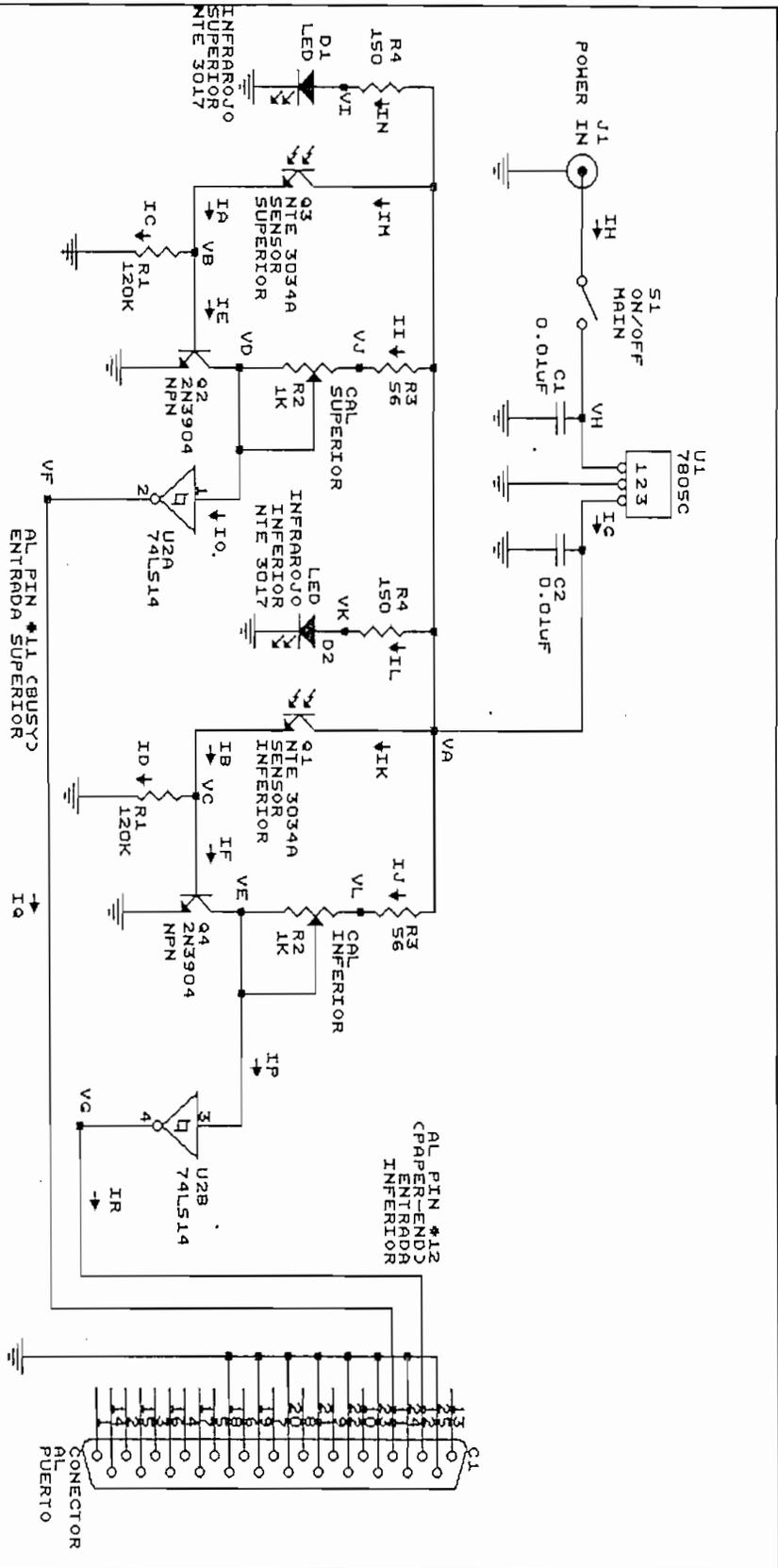
TITULO CAIDA LIBRE

Size Document Number FRANCISCO SOJOS D.

A

Date: JULY 8, 1999 Sheet 1 of 1

REV



ESCUELA POLITECNICA NACIONAL	
Title	PLANO INCLINADO
Size Document Number	FRANCISCO SOJOS D.
Date:	JULY 8, 1999 Sheet 1 of 1
REV	

## E. SITIOS WEB IMPORTANTES PARA VISITAR.

### E.1 SITIOS WEB PARA OBTENER EL "IOPORT.OCX".

The screenshot shows the Netscape browser window displaying the homepage of Upper Canada Technologies (UCT). The browser title is "UPPER CANADA TECHNOLOGIES - Netscape". The address bar shows the URL "http://www.uct.ca". The page content includes the UCT logo, the text "UPPER CANADA TECHNOLOGIES", and a section titled "What we're about...". This section describes the site as a resource for hobbyists interested in computer-based control topics, projects, and software. It lists several key areas of focus:

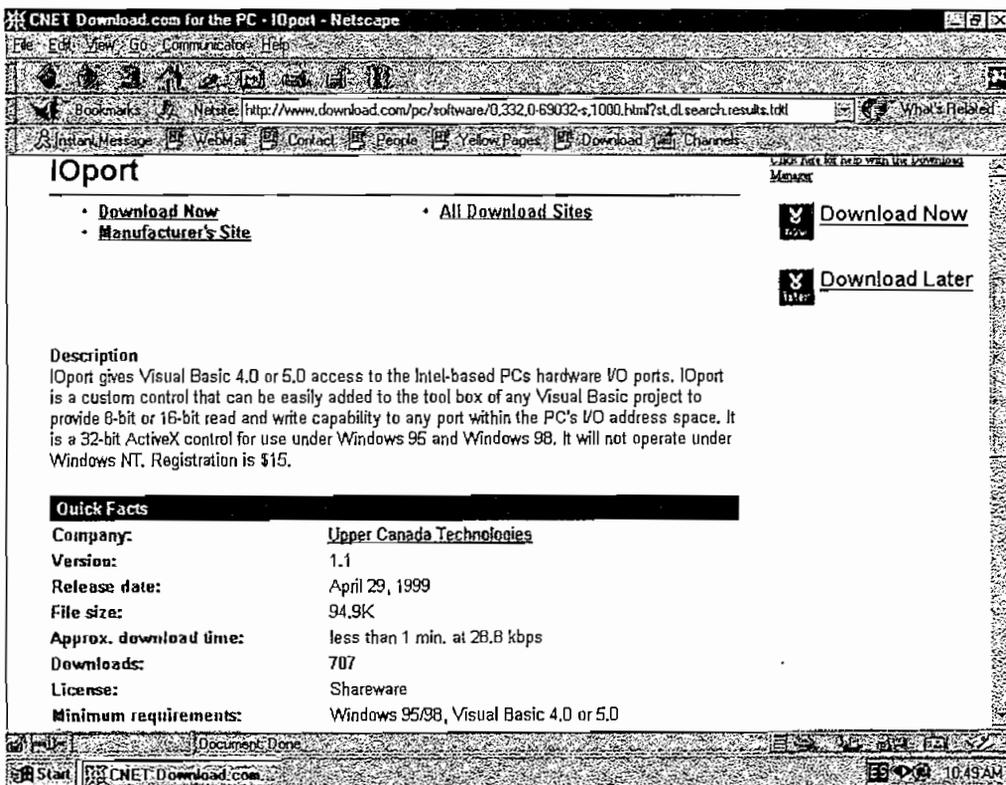
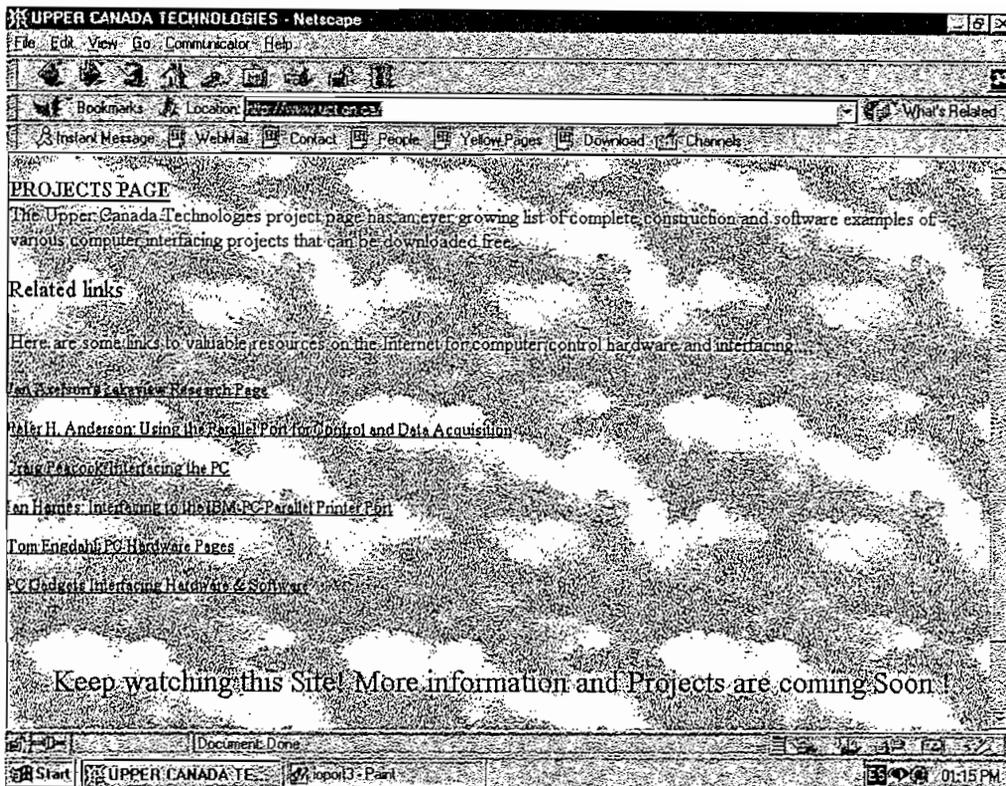
- Interfacing through the computers Printer Ports.
- Interfacing through the Computers Serial Ports.
- Designing interfacing cards for direct insertion into the computers expansion bus.
- Writing software for computer interfacing.

Below this list, the text states: "As a beginning, a key to interfacing software development using Visual Basic is the IOPORT.OCX v1.1 Custom Control for Windows 95/98 or NTPORT.OCX v1.0 for Windows NT." At the bottom of the page, there is a button labeled "I/O" and a link that says "Download IOPORT.OCX". The browser's status bar at the bottom shows "Document Done" and the time "01:11 PM".

The screenshot shows the Netscape browser window displaying a page titled "Why would I use IOPORT?". The browser title is "UPPER CANADA TECHNOLOGIES - Netscape". The address bar shows the URL "http://www.uct.ca". The page content includes the text "Why would I use IOPORT?" and a paragraph explaining the utility of IOPORT in DOS-based applications. It states that IOPORT is a custom control that can be easily added to the tool box of any Visual Basic project to provide 8-bit or 16-bit READ and WRITE capability to any port within the PC's I/O address space. It also mentions that IOPORT is a 32-bit ActiveX/OCX control designed to be used with Visual Basic 4.0 or 5.0 under Windows 95 and Windows 98, but it will not operate under Windows NT. Below this text, there is a button labeled "NTPORT" and a link that says "Download NTPORT.OCX".

Below this section, the page is titled "When would I use NTPORT?". The text explains that many programmers are migrating to the Windows NT platform for its security and network capabilities. Part of NT's stability comes from its requirement that user-mode programs may not access the computer hardware directly but must make requests for hardware access through a "kernel-mode" driver. For many Visual Basic programmers, this places a road block to performing the kinds of hardware I/O functions that were simply achieved in DOS. IOPORT provided a return to that capability for Windows 95/98 users. Now NTPORT provides that ability in Windows NT by incorporating a kernel-mode driver. NTPORT works and acts in the same way as IOPORT did so that code written for IOPORT can be easily migrated to NTPORT. NTPORT will work equally well under Windows 95, Windows 98, or Windows NT.

The browser's status bar at the bottom shows "Document Done" and the time "01:14 PM".



## E.2 INFORMACION ADICIONAL PARA EL PUERTO PARALELO.

### Parallel Port Central

A collection of files and links to material about the PC's parallel port, including ECP, EPP, bidirectional, and IEEE-1284 modes (and other I/O ports as well).  
brought to you by Jan Axelson

Basics | ECP & EPP | Chips | Cables | Programming

### Parallel Port Complete

For all you ever wanted to know about the parallel port, order a copy of Parallel Port Complete. Read Chapter 1 on-line.

#### The Basics

- Jan's Parallel Port FAQ has answers to frequently asked questions about using, interfacing, and programming the parallel port in all of its modes. Last update: 4/17/99.
- The parallel port FAQ . Basics about the parallel port and how to use it, by Zhahai Stewart. (50K)
- The Computer Peripherals Guide at the Mining Co. is a good place to start if you're looking for help in using standard peripherals such as printers and drives. Hosted by Rick Russell.
- Control and data acquisition information and projects, from Peter H. Anderson and his students. Code examples in C.
- Para14.zip. Parallel-port diagnostic and information utility from Parallel Technologies. (88K)
- Various LPT documents. Steve Walz has collected a useful set of parallel-port-related FAQs and information.
- Craig Peacock's Interfacing the PC. Includes a section on the parallel port.
- Interfacing to the IBM-PC Parallel Printer Port. General information, plus several projects.
- Tom Engdahl's PC Hardware pages. Many links.
- PC Webopaedia. Definitions and many links.
- The Connect Pages have information on how to connect two PCs running various combinations of operating systems, using serial and parallel ports. From Kime.Net.
- External Parallel Port devices and Linux. Many links and info about the Linux-parport mailing list. From Grant Guenther.
- For printer problems under Windows, try Microsoft's on-line support. Search for "lpt.vxd."
- I've written a variety of magazine articles on parallel-port-related topics.

### Enhanced and Extended Ports

Including the bidirectional (PS/2) port, extended capabilities port (ECP), and enhanced parallel port (EPP), and the IEEE-1284 standard that describes them all.

#### General Information

- A table showing the pinouts of all three of the commonly used parallel-port connectors, including the 25-pin D-sub, the 36-contact Centronics connector, and the new IEEE-1284C connector. (6K PDF (Acrobat) file) Warp Nine Engineering. Detailed descriptions of the EPP, ECP, and other IEEE-1284 modes.
- Warp Nine's products include ECP/EPP parallel-port cards, EPP and ECP drivers, and a peripheral interface controller that provides the logic between a general-purpose microcontroller and a PC's parallel port, to enable a peripheral to support EPP and ECP

modes. Also check here for the latest information on IEEE standards relating to the parallel port. And there's a link to Microsoft's document describing ECP mode. (325k, PDF file).

- Various files relating to the IEEE-1284 standard, from Lexmark's FTP site Information on ordering standards from IEEE. The title of the parallel port standard is 1284-1994 IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers.
- Paul Rademacher has a page describing his experience with programming an EPP link using one of Far Point's parallel-port cards.
- Standard Microsystems Corp. has preliminary versions of the daisy-chain specification and EPP BIOS, plus drivers and source code for SMC chips.

## Chips

The data sheets for parallel-port controller chips include timing diagrams, register assignments, and other useful details about accessing and using the new modes. PC-side chips are for use in PCs; peripheral-side chips are for use in peripherals.

- AMD's Am29202 processor, for use in EPP/ECP (peripheral-side).
- Cirrus Logic's parallel I/O chips.
- Holtek's HT6535 SPP/EPP/ECP Controller. (PC-side)
- Intel's 82091 advanced integrated peripheral (PC side).
- National Semiconductor has a variety of Super I/O chips that include an IEEE 1284-compatible parallel port with EPP and ECP modes. (PC-side)
- National's 74VHC161284 Transceiver contains eight bidirectional data buffers and eleven control/status buffers to implement an IEEE-1284 Level 2 interface. Outputs on the cable side can be configured to be either open drain or push-pull.
- Standard Microsystems Corp., (SMSC), another manufacturer of EPP/ECP parallel-port chips. (PC- and peripheral-side).
- Exar/Startech's ST78C34 and ST78C36 are EPP/ECP chips. Look under Products/Communications/General Purpose. (PC-side).
- Texas Instruments' TL16PIR552 includes a PC-side ECP+EPP, plus a dual UART and IrDa interface.
- Warp Nine's W91284PIC supports all modes, plus daisy-chaining and negotiating. (peripheral-side)
- Winbond has PC-side chips.

## Cables

- Denis Kondakov has figured out how to do Direct Cable Connection ECP transfers between PCs using a simple, home-brewed cable configuration. It requires a patch to `paralink.vxd`.
- Parallel Technologies' Universal Cable contains active circuits for high-speed parallel-port communications using Windows 95/98's Direct Cable Connection.

## Programming

There are several ways to access the parallel port and other I/O ports in PCs, including direct I/O, system-level device drivers, and Windows API calls.

Under Windows 3.x/95/98, applications can read and write directly to port addresses. Use your compiler's built-in functions (`inp` and `out` or similar) or in-line assembly code. (See the source code in `inout32.zip` below for an example.) This method is simple, but it's slow, it can't protect the port from access by other applications, and it doesn't work at all under NT. If you use Visual Basic or another language that doesn't have functions for port I/O or allow in-line assembly, you can use a DLL or a custom control that adds port I/O functions to an application.

A system-level device driver enables faster port access and can manage access by multiple applications. Driver types include VxD (virtual device driver) for Windows 95/98, WDM for Windows 98/NT, and kernel-mode driver for Windows NT. Hardware interrupts must use a system-level driver under Windows 95/98/NT. If you don't want to write your own driver, there are custom controls and other tools that enable applications to access ports and respond to interrupts via a driver.

A third way to access ports is to use Windows API calls or other built-in controls. However, Windows has no functions for generic port access, only functions tied to specific uses. For example, there are API calls for accessing printers and serial ports controlled by UARTs. In Visual Basic, the Printer object and MSComm control are other options for parallel and serial-port access. Built-in functions and controls like these are usually the best solution when their abilities match what you're trying to accomplish.

Below are links to tools that you can use for port access. To make deciding easier, I've grouped them according to whether they are for use with 16-bit or 32-bit compilers, and which operating systems they support. The list includes freeware, shareware, and commercial products.

## Programming Tools for Port I/O and Interrupts

### 32-bit-only Tools

Use these in 32-bit programs created with products such as 32-bit VB4, VB5, or VB6.

*For Windows 95/98 and NT*

- DriverLinx Port I/O. Freeware. Supports: port I/O. From Scientific Software Tools. John Pappas has written TDLPortIO, a freeware interface for DriverLinx for use with C++ Builder, Delphi, and other languages.
- DriverX. Supports: port I/O and interrupts. From Tetradyne Software.
- Ioport & NTPort. Shareware. Supports: port I/O. From Robert Woudsma at Upper Canada Technologies.
- Parallel Port Direct I/O Access package. Shareware. Supports: port I/O. Supports access under NT by changing the I/O permission map for the process that claims the port. From Peter Shoebridge at Zeecube Software
- TVicHW32 & TVicPort. Shareware. Supports: port I/O and hardware interrupts. From EnTech Taiwan
- W95pio and Ntpio. Freeware. Supports: port I/O. Intended for use with HP's VEE visual programming language, but adaptable to other uses.
- WinRT, WinRT-VB, WinDK. Supports: port I/O and hardware interrupts. A variety of products for many uses. From BlueWater Systems

*For Windows 95/98 only*

- Inpout32.zip. Freeware. Supports: port I/O. The file contains inpout32.dll. I wrote inpout32 in Delphi 2. The zip file includes the DLL, Visual-Basic declarations for inp and out, documentation, a test program, and the Delphi source code, which includes assembly code for port I/O. This file was a download of the week in Visual Basic Programmer's Journal's VB Zone Development Exchange. Operating systems: Windows 95/98. From Lakeview Research.
- Lptaccess. Freeware. This is a VxD that enables an application to reserve port addresses for its exclusive use. Details are in Karen Hazzah's VxD Advisor column in the January 1998 issue of Windows Developer's Journal. The magazine's website has the program code and info about the magazine.
- Win95io.zip. Freeware. Supports: port I/O. Another inpout DLL. The DLL is just 2048 bytes. From Jonathan Wood at Softcircuits.

*For Windows NT only*

- MapMemPlus. Freeware. Supports: port I/O. From Gabriele Mondada. Tinyport. Shareware. Supports: port I/O. A ready-to-run kernel-mode device driver. Tinyport complies with the rules that Microsoft has specified for NT drivers, so it doesn't undermine NT's stability and can be used in critical applications. From Manfred Keul.

## 16-bit-only Tools

Use these with products such as Visual Basic 3 or 16-bit Visual Basic 4.

- inport16.zip Freeware. Supports: port I/O. Includes an inport DLL, source code in PowerBasic, and an example VB project. From Lakeview Research.
- inport.zip. Freeware. Supports: port I/O. Another inport DLL. Includes assembly-language source code. From Jay Munro.
- vbasm.zip. Freeware. Supports: port I/O. A DLL that adds Inp, Out, Peek, Poke, Call Interrupt, and more for accessing ports and memory. From Jonathan Wood at Softcircuits.

## For 16- and 32-bit Programs (Windows 3.x, Windows 95/98)

- Port16 and Port32. Shareware. Supports: port I/O. From Hone Software.
- PowerBasic is the source for several products that are useful for applications that access ports. PowerBasic's 16-bit and 32-bit DLL compilers enable you to write and compile DLLs in Basic. The Basic syntax is nearly identical to classic QuickBasic. Because the DLLs are compiled, not interpreted like Visual Basic code, they're fast. The 16-bit edition includes inp and out for port access. The 32-bit edition allows port access under Windows 95/98 via inline assembly code. There's also a 32-bit Basic console compiler for text-only Windows applications and even a DOS compiler.
- "Using 16-bit DLLs with 32-bit VB." This article explains, step-by-step, a method for using 16-bit DLLs in 32-bit VB applications. The example described is port I/O with Vbasm (see above). To use this method, you must have both a 16-bit and a 32-bit edition of Visual Basic. From COOL.STF.

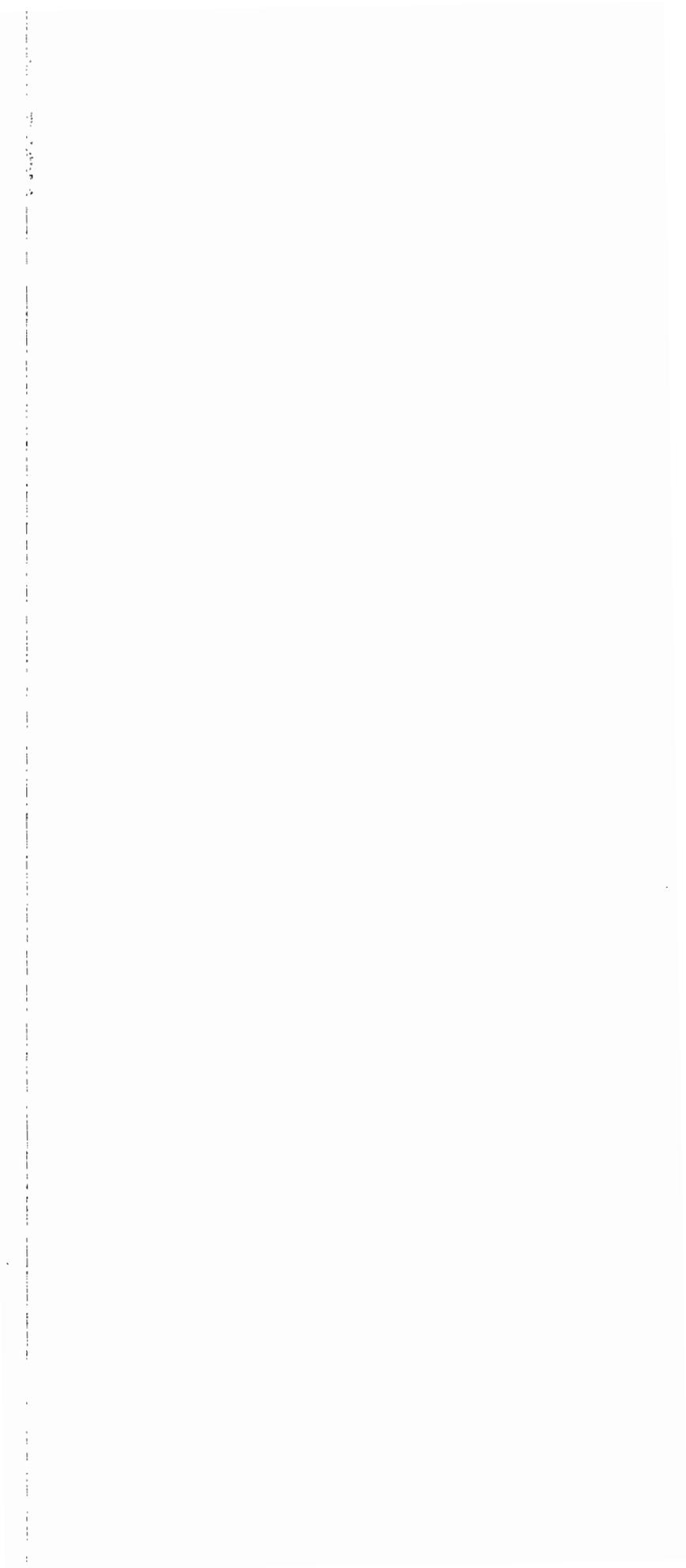
## Other Driver Information and Sources

- Dan Norton has an FAQ and many links relating to Device Driver Development
- Warp Nine Engineering has IEEE-1284 drivers.

Tomado de: [www.uct.on.ca](http://www.uct.on.ca)



## **F. AYUDA DEL PROGRAMA COMPUTACIONAL "EFC".**



# Contenido

**Escuela Politécnica Nacional  
Facultad de Ingeniería Eléctrica  
Departamento de Electrónica y Telecomunicaciones**

“Diseño y construcción de prototipos electrónicos y  
el software para la enseñanza de las ciencias físicas”

## **Experimentos de Física en Computador EFC v1.0**

Por: Francisco Sojos D.

Quito – Ecuador  
1.999

**Temas:**

Caída Libre

Plano Inclinado

Péndulo Simple

## CAIDA LIBRE

Es un hecho, y se observa repetidamente, que todos los cuerpos tienden a caer sobre la superficie terrestre. Este fenómeno se debe a la atracción que la Tierra ejerce sobre los cuerpos próximos a su superficie y que recibe el nombre de gravedad. Esto es, sólo un caso particular de una propiedad general de la materia llamada Gravitación Universal.

Como resultado de numerosos experimentos se puede enunciar la siguiente ley de la caída de los cuerpos:

Como una consecuencia inmediata de la igualdad de la aceleración se concluye que: “En el vacío todos los cuerpos emplean el mismo tiempo en caer una misma distancia, si parten en las mismas condiciones.”

Es necesario especificar en el vacío, porque si la caída se verifica en un medio como el aire, esta substancia se opone al movimiento, y esta oposición o resistencia depende principalmente de la forma geométrica del cuerpo. Así por ejemplo: Si se sueltan desde una misma altura una piedra y una pluma, obviamente la pluma llegará más tarde que la piedra debido a que presentó más resistencia a caer dentro de este medio físico. Si el mismo experimento se lo hace en un medio como el vacío, el experimentador visualizará que los dos móviles caen por igual y al mismo tiempo.

Temas relacionados:

Contenido

Aceleración de la gravedad

Ecuaciones de la caída libre de los cuerpos

Casos especiales en caída libre

## ACELERACION DE LA GRAVEDAD

La aceleración con la que caen los cuerpos se llama aceleración de la gravedad, y se la designa por la letra  $g$ . Su valor depende de las condiciones locales de cada lugar de la Tierra. Debido a que ésta no es perfectamente redonda ya que es achatada en los polos, es necesario indicar que por estas circunstancias el valor que tiene la aceleración de la gravedad es diferente en cada lugar del planeta.

Incorrectamente y por rapidez de expresión la aceleración de la gravedad es llamada gravedad o fuerza de gravedad, cuando en realidad el término gravedad designa el fenómeno por el que la Tierra (todas las masas en general) atrae a los cuerpos. Fuerza de gravedad en cambio, es la fuerza que provoca que los cuerpos se aceleren en su un medio gravitacional.

En caída libre, el valor máximo de  $g$  es de  $9.83 \text{ m/s}^2$  en el polo norte y disminuye a medida que nos acercamos al ecuador, donde su valor baja a ser de  $9.78 \text{ m/s}^2$ . En París (Francia), el valor de  $g$  se encuentra en los  $9.809 \text{ m/s}^2$  y que corresponden en el sistema inglés a  $32 \text{ pies/s}^2$ .

### Contenido

## ECUACIONES DE LA CAIDA LIBRE DE LOS CUERPOS

Las ecuaciones que se aplican a la caída libre de los cuerpos son las mismas del M.R.U.V (Movimiento Rectilíneo Uniformemente Variado), aunque se acostumbra representar la altura o espacio recorrido por  $h$  y la aceleración lineal por  $g$ . Entonces para la caída libre de los cuerpos, o sea, cuando se parte del reposo (sin velocidad inicial), se tiene:

$$v = g*t, \quad h = \frac{1}{2}g*t^2, \quad v^2 = 2g*h \quad [1]$$

En el caso general de caída libre, es decir, cuando  $v_0 \neq 0$ , se tienen las siguientes ecuaciones:

$$v = v_0 + g*t \quad [2]$$

$$h = v_0*t + \frac{1}{2}g*t^2 \quad [3]$$

$$v^2 = v_0^2 + 2g*h \quad [4]$$

En el caso de que el cuerpo es lanzado verticalmente hacia arriba, el movimiento de subida será retardado, con lo que la aceleración de la gravedad  $g$  tendrá ahora un valor negativo:  $a = -g$ , y las ecuaciones del movimiento serán:

$$v = v_0 - g*t \quad [5]$$

$$h = v_0*t - \frac{1}{2}g*t^2 \quad [6]$$

$$v^2 = v_0^2 - 2g*h \quad [7]$$

### Contenido

## CASOS ESPECIALES EN CAIDA LIBRE

**Altura Máxima:** A medida que un cuerpo es lanzado hacia arriba, éste experimenta una desaceleración en su movimiento vertical, es decir, en un principio arranca con velocidad inicial  $v_0 \neq 0$  hasta que llegará a un instante en el que la velocidad final adquirida es  $v = 0 \text{ m/s}^2$ . Si de la ecuación (7) hacemos  $v^2 = 0$ , se tiene que:

$v_0^2 = 2g \cdot h$ , de donde despejando la variable  $h$ , se tendrá que la altura máxima  $h_m$  es:

$$h_m = v_0^2 / 2g \quad (8)$$

**Tiempo de Subida:** De igual forma, el tiempo de subida se lo puede deducir de la ecuación (5). De la misma forma, si el cuerpo sube, llegará en cierto instante a detenerse, con velocidad final  $v = 0 \text{ m/s}^2$ . En ese momento se tiene que:

$v_0 = g \cdot t$ , de donde despejando la variable  $t$ , veremos que este valor corresponde al tiempo de subida  $t_s$ , que finalmente se lo calculará mediante la siguiente ecuación:

$$t_s = v_0 / g \quad (9)$$

### Particularidades de la Caída Libre.

En un análisis de movimiento del móvil, puede suceder los casos siguientes:

- El cuerpo arranca desde el reposo, es decir con velocidad inicial  $v_0 = 0 \text{ m/s}^2$ , entonces el análisis del movimiento se deberá realizarlo con el grupo de ecuaciones (1).
- Si el caso es que el cuerpo ya estuvo con cierta velocidad antes de cruzar el punto de referencia inicial, entonces servirán las ecuaciones (2), (3) y (4).
- Las ecuaciones (5), (6) y (7) se deberán utilizar sólo si se trata de un movimiento hacia arriba. No debe olvidarse el hecho de que un cuerpo en subida, experimenta una desaceleración igual a:  $-g$  y que además en el punto máximo de subida la velocidad final del móvil será de cero ( $v=0$ ).
- El punto de análisis en la vertical de caída es relativo y los cálculos dependerán de éste. Dicho con un ejemplo: En la figura #1 se tiene un móvil descendiendo en caída libre. Nótese que se han situado 3 puntos de observación: A, B y C.

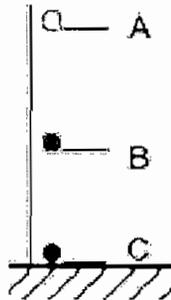


FIGURA #1. Puntos de observación en Caída Libre.

Para los tres puntos situados en análisis, se tienen las siguientes consideraciones:

- En el punto A el cuerpo puede o no arrancar con velocidad inicial  $v_0 = 0 \text{ m/s}^2$ .
- En el punto B el móvil en estudio tiene una velocidad inicial cuyo valor corresponde al valor de la velocidad final adquirida en el tramo AB.
- Finalmente, en el punto C, el cuerpo adquiere una velocidad final del tramo BC. Esta puede calcularse teniendo en cuenta que la velocidad inicial ( $v_0$ ) del tramo BC es igual a la velocidad final ( $v$ ) del tramo AB.
- Entre puntos de referencia se tienen los recorridos de cada tramo y son medidas longitudinales que pueden reemplazarse en las variables de espacio recorrido de las ecuaciones aplicadas.

### Contenido

## PLANO INCLINADO

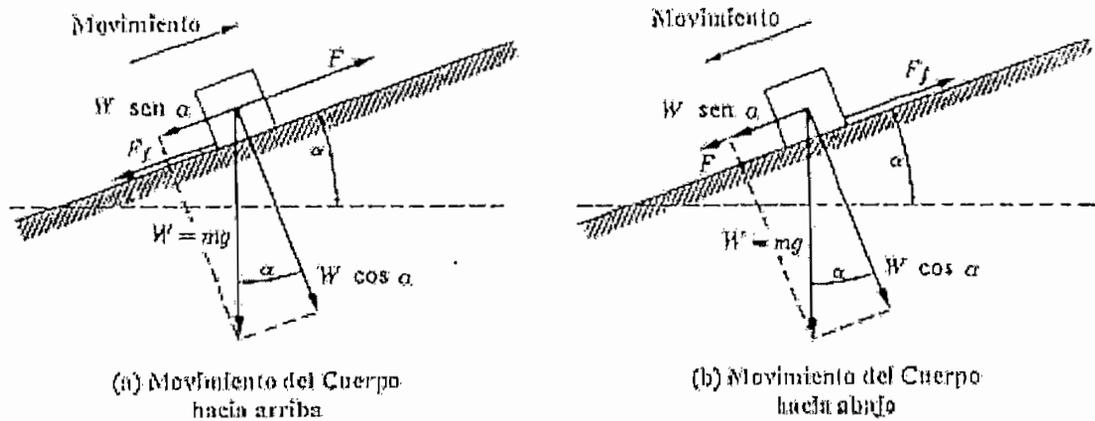
El movimiento de un cuerpo sobre una superficie con un cierto grado de inclinación se denomina "Plano Inclinado". Este movimiento está regido por las leyes de la fricción por desplazamiento.

Si a un cuerpo en equilibrio que se encuentra sobre un plano inclinado, se le aplica una fuerza mayor a la fuerza de rozamiento, entonces el cuerpo se moverá en el sentido de la fuerza aplicada.

Por ejemplo, se plantea el siguiente problema:

Un cuerpo de masa  $m$  se encuentra sobre un plano inclinado de ángulo  $\alpha$ . Qué fuerza debe aplicarse al cuerpo de modo que se mueva:

- (a) Hacia arriba.
- (b) Hacia abajo.



Solución:

(a) Las fuerzas que actúan sobre el cuerpo son el peso  $W = mg$ , dirigido hacia abajo, la fuerza aplicada  $F$  y la fuerza de fricción  $F_f$ , la cual es siempre opuesta al movimiento, en este caso hacia abajo. Así mismo, se descompone el peso del cuerpo en sus dos componentes  $F_x$  y  $F_y$ . Aplicando la segunda ley de Newton, la cual indica, que la sumatoria de las fuerzas aplicadas sobre el cuerpo en movimiento (Eje X) es igual a la masa de este por su aceleración, se tiene que:

$$\Sigma F_x = m a$$

tenemos

$$F - mg \sin \alpha - F_f = ma$$

En el eje y, el cuerpo no se mueve, y por lo tanto está en reposo, por lo tanto se deduce que: La sumatoria de las fuerzas aplicadas sobre el cuerpo en movimiento es igual a cero, por lo tanto,

$$\Sigma F_y = 0$$

$$N - F_y = 0$$

$$N = F_y$$

De donde:

$$N = W \cos \alpha.$$

De acuerdo a la ecuación  $F_f = \mu_k N$  (donde  $\mu_k$  es el coeficiente de fricción y  $N$  es la Normal), de la figura (a) vemos que la fuerza normal ( $N$ ) del cuerpo contra el plano es:  $mg \cos \alpha$ . Así  $F_f = \mu_k mg \cos \alpha$ . Y la ecuación total del movimiento se transforma en:

$$F - mg (\sin \alpha + \mu_k \cos \alpha) = ma.$$

Esta ecuación sirve para dos propósitos. Si conocemos la aceleración  $a$ , podemos encontrar la fuerza aplicada  $F$ . Por otro lado, si conocemos la fuerza  $F$  aplicada al móvil, podemos encontrar la aceleración  $a$  con la que se mueve el mismo. En el primer caso tenemos:

$$F = m [ a + g (\sin \alpha + \mu_k \cos \alpha)].$$

- ( ) En la figura (b) se ilustran las fuerzas cuando el cuerpo se mueve hacia abajo. Ahora hemos supuesto que la fuerza  $F$  es hacia abajo. Sin embargo, la fuerza de fricción  $F_f$ , debe ser hacia arriba para oponerse al movimiento. Considerando la dirección hacia abajo como positiva, el estudiante puede verificar que las ecuaciones del movimiento son ahora:

$$\Sigma F_x = m a$$

$$F + W \sin \alpha - F_f = m a$$

$$\Sigma F_y = 0$$

$$N = W \cos \alpha.$$

$$F + W \sin \alpha - \mu_k W \cos \alpha = m a$$

$$F + mg \sin \alpha - \mu_k mg \cos \alpha = m a$$

$$F + mg (\sin \alpha - \mu_k \cos \alpha) = ma.$$

ó

$$F = m [ a - g (\sin \alpha - \mu_k \cos \alpha)].$$

El signo negativo en cada caso significa que la fuerza  $F$  es hacia arriba en lugar de hacia abajo como habíamos supuesto. Sugerimos que el estudiante determine el movimiento del cuerpo cuando no se aplica ninguna fuerza  $F$ , y en vista del resultado obtenido justifique el signo negativo obtenido previamente para  $F$ .

Temas relacionados:

Contenido

Fricción

## FRICCIÓN

Es un hecho experimental bien conocido que siempre que tratamos de mover un cuerpo en contacto con otro encontramos cierta resistencia. Ej. Si tenemos un libro sobre una mesa y lo empujamos, vemos que se detiene después de recorrer cierta distancia, puesto que ha estado sometido a una fuerza que se opuso al movimiento.

Esta fuerza se denomina fuerza de fricción por deslizamiento y se debe a la interacción entre las moléculas de los dos cuerpos, algunas veces llamada fuerza de cohesión o adhesión, dependiendo si los cuerpos son del mismo o diferente material. El fenómeno es algo complejo y depende de muchos factores tales como la condición y la naturaleza de las superficies, la velocidad relativa, etc.

“Fricción es la fuerza que aparece en la superficie de contacto de dos cuerpos diferentes en movimiento relativo, oponiéndose siempre a dicho movimiento”.

En general hay dos clases de coeficientes de fricción. El coeficiente estático de fricción ( $\mu_s$ ), que al multiplicarse por la fuerza normal, nos da la fuerza mínima necesaria para poner en movimiento relativo dos cuerpos que están inicialmente en contacto y en reposo, y el coeficiente dinámico de fricción o por deslizamiento ( $\mu_k$ ).

### Fricción por deslizamiento

Si se trata de dos cuerpos sólidos y uno se desliza sobre el otro, se tiene la fricción por deslizamiento. Este es el caso de un libro que se desliza sobre una mesa y que explicamos anteriormente. Si se desea que continúe deslizándose es necesario ejercer sobre el libro una fuerza para vencer la fuerza de fricción. Como la superficie de los cuerpos, aún cuando parezcan muy pulimentadas, presentan muchas rugosidades o irregularidades al examinarse microscópicamente, la fricción por deslizamiento se debe en parte, a que al reposar un cuerpo sobre otro las irregularidades de la superficie del primero se entrelazan o se traban con las del segundo, dificultándose así el movimiento relativo de ambos, lo que dinámicamente equivale a una fuerza opuesta a dicho movimiento. La fricción depende, además de la fuerza de adhesión que se produce entre las moléculas de la superficies en contacto.

Para que el cuerpo se deslice con movimiento uniforme es necesario aplicarle una fuerza igual y contraria a la de fricción. Este es el modo experimental de medir la fricción por deslizamiento.

La fuerza de fricción por deslizamiento siempre se opone al movimiento del cuerpo, y por ende tiene una dirección opuesta a la velocidad, además esta resulta al multiplicar el coeficiente dinámico de fricción ( $\mu_k$ ) por la fuerza normal.

### Leyes de la Fricción por deslizamiento

Las leyes de la fricción por deslizamiento son:

- 1.- La fricción por deslizamiento depende de las sustancias en contacto.
- 2.- La fricción por deslizamiento depende del estado en que se encuentren las superficies (grado de pulimentación, barnizado, grasa, etc.).
- 3.- La fricción por deslizamiento es independiente de la forma y el área de la superficie de contacto.
- 4.- La fricción por deslizamiento es proporcional a la fuerza que se ejerce perpendicularmente a las superficies de contacto. O sea, si ( $N$ ) es la fuerza normal a las superficies y ( $F_f$ ) la fuerza de fricción:

$$F_f = \mu_k N \quad (1)$$

donde  $\mu_k$  ( $\mu$ ) es el coeficiente de fricción.

El coeficiente de fricción depende de la naturaleza de las sustancias en contacto. Para cedro sobre cedro oscila entre 0.4 y 0.5 y para un metal sobre otro está comprendido entre 0.15 y 0.3.

Además, la fricción por deslizamiento disminuye al aumentar la velocidad relativa de los cuerpos en contacto, lo cual se comprende porque entonces tienen menos oportunidad de trabarse las irregularidades.

### Temas Relacionados:

[Contenido](#)

[Tabla de coeficientes de fricción](#)

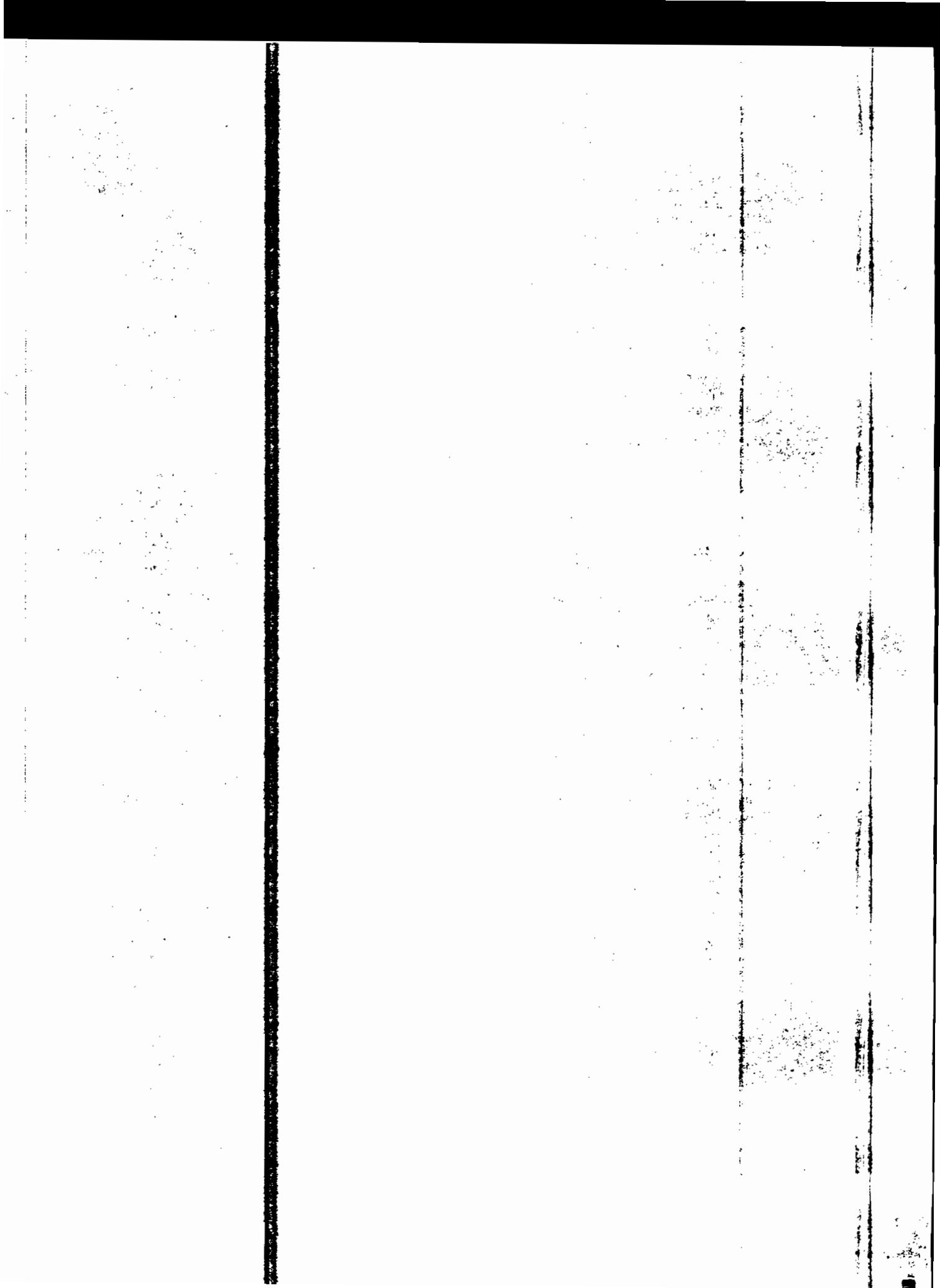
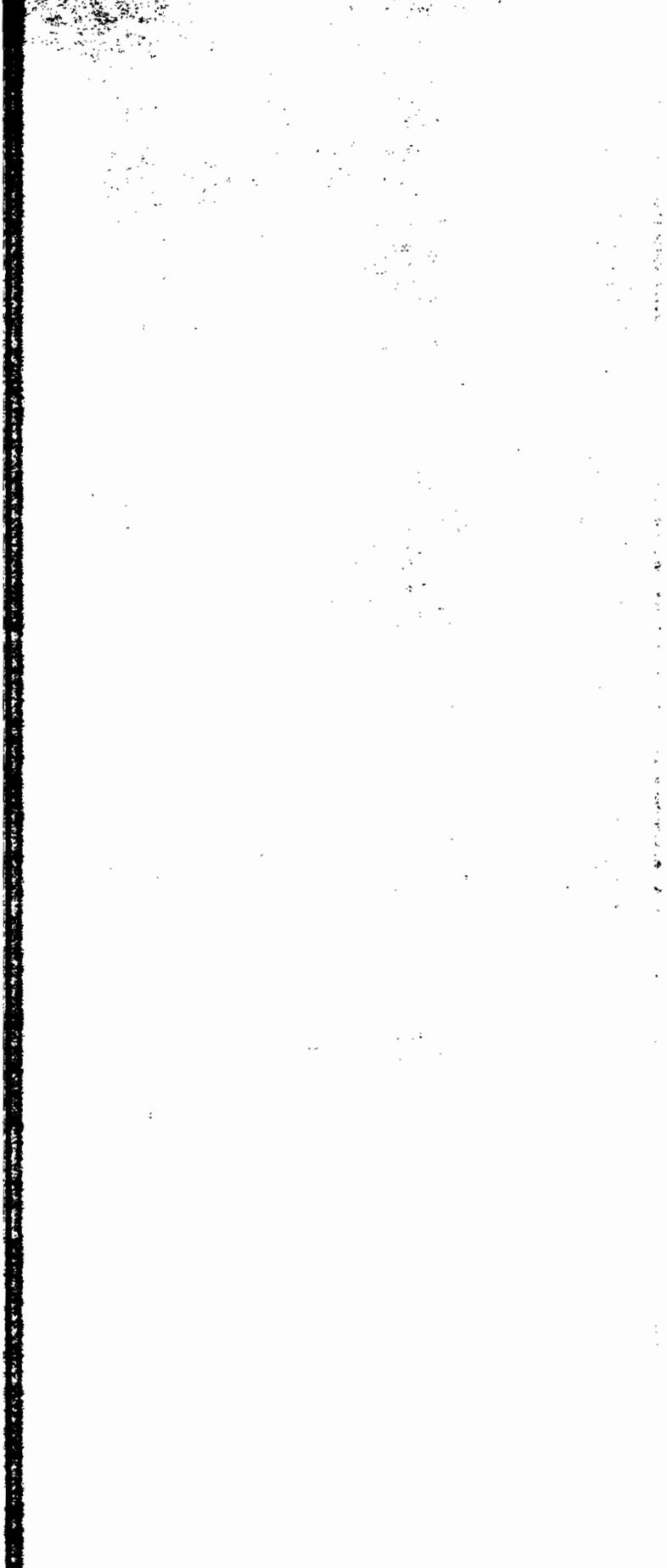


Tabla de coeficientes de fricción (Todas las superficies secas)

Materiales	Estático	Dinámico
	$\mu_s$	$\mu_k$
Acero duro sobre acero duro	0.78	0.42
Acero dulce sobre acero dulce	0.74	0.57
Acero duro sobre grafito	0.21	...
Acero duro sobre metal Babbit (ASTM No. 1)	0.7	0.33
Acero duro sobre metal Babbit (ASTM No. 8)	0.42	0.35
Acero dulce sobre bronce fosforado	...	0.34
Acero dulce sobre cobre	0.53	0.36
Acero dulce sobre hierro fundido	...	0.23
Acero dulce sobre plomo	0.95	0.95
Níquel sobre acero dulce	...	0.64
Aluminio sobre acero dulce	0.61	0.47
Magnesio sobre acero dulce	...	0.42
Magnesio sobre magnesio	0.6	...
Teflón sobre teflón	0.04	0.04
Teflón sobre acero	0.04	...
Carburo de tungsteno sobre carburo de tungsteno	0.2	...
Carburo de tungsteno sobre acero	0.5	...
Carburo de tungsteno sobre cobre	0.35	...
Carburo de tungsteno sobre hierro	0.8	...
Carburo de ligado sobre cobre	0.35	...
Carburo de ligado sobre hierro	0.8	...
Cadmio sobre acero dulce	...	0.46
Cobre sobre acero dulce	0.53	0.36
Níquel sobre níquel	1.1	0.53
Bronce sobre acero dulce	0.51	0.44
Zinc sobre hierro fundido	0.85	0.21
Magnesio sobre hierro fundido	...	0.25
Cobre sobre hierro fundido	1.05	0.29
Estaño sobre hierro fundido	...	0.32
Plomo sobre hierro fundido	...	0.43
Aluminio sobre aluminio	1.05	1.4
Vidrio sobre vidrio	0.94	0.4
Carbón sobre vidrio	...	0.18
Granate sobre acero dulce	...	0.39
Vidrio sobre níquel	0.78	0.56
Cobre sobre vidrio	0.68	0.53
Hierro fundido sobre hierro fundido	1.1	0.15
Bronce sobre hierro fundido	...	0.22
Encino sobre encino (paralelo a la vena)	0.62	0.48
Encino sobre encino (perpendicular a la vena)	0.54	0.32
Piel sobre encino (paralelo)	0.61	0.52
Hierro fundido sobre encino	...	0.49
Piel sobre hierro fundido	...	0.56
Plástico laminado sobre acero	...	0.35

Nota: Estos valores deben considerarse como promedios, ya que los coeficientes de fricción son cantidades macroscópicas que dependen de las propiedades microscópicas de ambos materiales, y fluctúan bastante.

## Contenido



## PENDULO SIMPLE

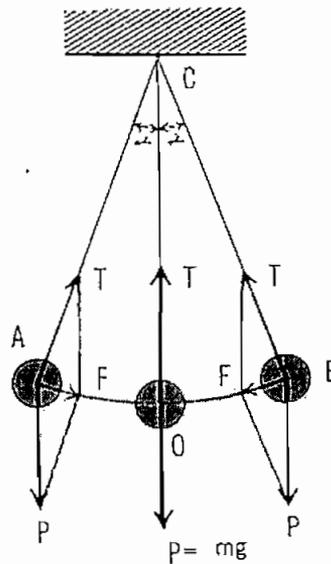
El péndulo simple consiste en una masa de dimensiones muy pequeñas, o sea, una partícula suspendida de un punto fijo mediante un hilo inextensible y de peso despreciable.

El péndulo se encuentra en equilibrio cuando el hilo  $CO$  está vertical. Si el péndulo se desplaza hasta  $B$ , de modo que el hilo forme el ángulo  $\alpha$  con la vertical y se lo deja libre, comienza a oscilar entre  $B$  y una posición simétrica  $A$ , al otro lado de la vertical bajo la acción combinada de su peso  $P$  y la tensión  $T$  del hilo, que producen una resultante  $F$ .

La elongación en el péndulo simple se mide por el ángulo que el hilo del péndulo forma con la vertical en un momento cualquiera.

La amplitud del péndulo es el mayor ángulo que se separa a uno u otro lado de la vertical.

Puede probarse que el péndulo está animado de m.a.s. solamente cuando su amplitud es pequeña, es decir inferior a unos  $45^\circ$ .



Temas relacionados:

[Contenido](#)

[Formulas y leyes del péndulo Simple](#)

[Deducción de las fórmulas de péndulo Simple](#)



CLASSIFICATION

CONFIDENTIAL

SECRET

TOP SECRET

SECRET

CONFIDENTIAL

## FORMULAS Y LEYES DEL PENDULO SIMPLE

Si  $l$  es la longitud del péndulo, se demuestra que en el caso de oscilaciones de pequeña amplitud, el período  $T$  está dado por la fórmula:

$$T = 2 \pi \sqrt{l/g}$$

Donde,  $g$  corresponde a la aceleración de la gravedad.

De la fórmula anterior se desprenden las siguientes conclusiones:

El período del péndulo simple es:

- 1) directamente proporcional a la raíz cuadrada de su longitud.
- 1) Inversamente proporcional a la raíz cuadrada de la aceleración de la gravedad.
- 1) Independiente de la masa pendular.
- 1) Independiente de la amplitud mientras ésta sea pequeña, 0 45.

La primera conclusión se obtiene comparando los períodos de varios péndulos de longitudes diferentes por ejemplo  $L$ ,  $4L$ ,  $9L$ , etc. Se encuentra entonces que los períodos respectivos resultan ser  $T$ ,  $2T$ ,  $3T$ , etc.

Para verificar la segunda conclusión, es necesario transportar el péndulo de un lugar a otro en la Tierra, donde la gravedad sea diferente, puesto que no se puede modificar a voluntad este parámetro. Fue descubierta en 1671 por Richltter al observar que el período de un péndulo cambiaba al transportarlo de París a Cayena y viceversa.

La tercera conclusión es una consecuencia de la ausencia de la masa en la fórmula

$$T = 2 \pi \sqrt{l/g}$$

La cuarta deducción es también consecuencia de la ausencia de la amplitud en la fórmula del período y suele llamarse ley del isocronismo (Griego, isos: igual, cronos: tiempo). Fue descubierta en 1583 por Galileo al observar que el período de las oscilaciones de las lámparas de la Catedral de Pisa que se movían al encenderlas, no cambiaba al ir disminuyendo su amplitud a medida que se amortiguaba su movimiento. Galileo descubrió también la primera y la tercera deducción, años más tarde.

Un péndulo bate segundos cuando tarda un segundo en dar una oscilación sencilla y por tanto, tiene un período de dos segundos.

Temas Relacionados:

Contenido

Péndulo Simple

Deducción de la fórmula de péndulo simple



## DEDUCCION DE LA FORMULA DE PENDULO SIMPLE

La fuerza que produce el movimiento armónico simple es:

$$F = -m\omega^2 x$$

En el caso de un péndulo la fuerza responsable del movimiento oscilatorio es,

$$F = -mg \sin \theta$$

Ahora bien; si la amplitud es pequeña podemos substituir  $\sin \theta$  por  $\theta$ , cuando el ángulo está en radianes, o sea:  $\theta = x / l$ .

Luego:

$$\sin \theta = \theta = x / l$$

Por tanto:

$$F = - (mgx) / l$$

Comparando las dos expresiones para la fuerza:

$$m\omega^2 = (mg) / l \Rightarrow \omega = \sqrt{(g / l)}$$

$$\omega^2 = g / l \Rightarrow \omega = \sqrt{(g / l)}$$

$$T = (2 \pi) / \omega = 2 \pi \sqrt{(l / g)}$$

La ecuación diferencial del movimiento es:

$$(d^2 \theta) / (dt^2) + (g / l) \sin \theta = 0$$

Dado que el la amplitud de las oscilaciones es pequeña, podemos escribir  $\sin \theta \sim \theta$  y reemplazarlo en la ecuación anterior para obtener:

$$(d^2 \theta) / (dt^2) + (g / l) \theta = 0$$

y dada la ecuación:

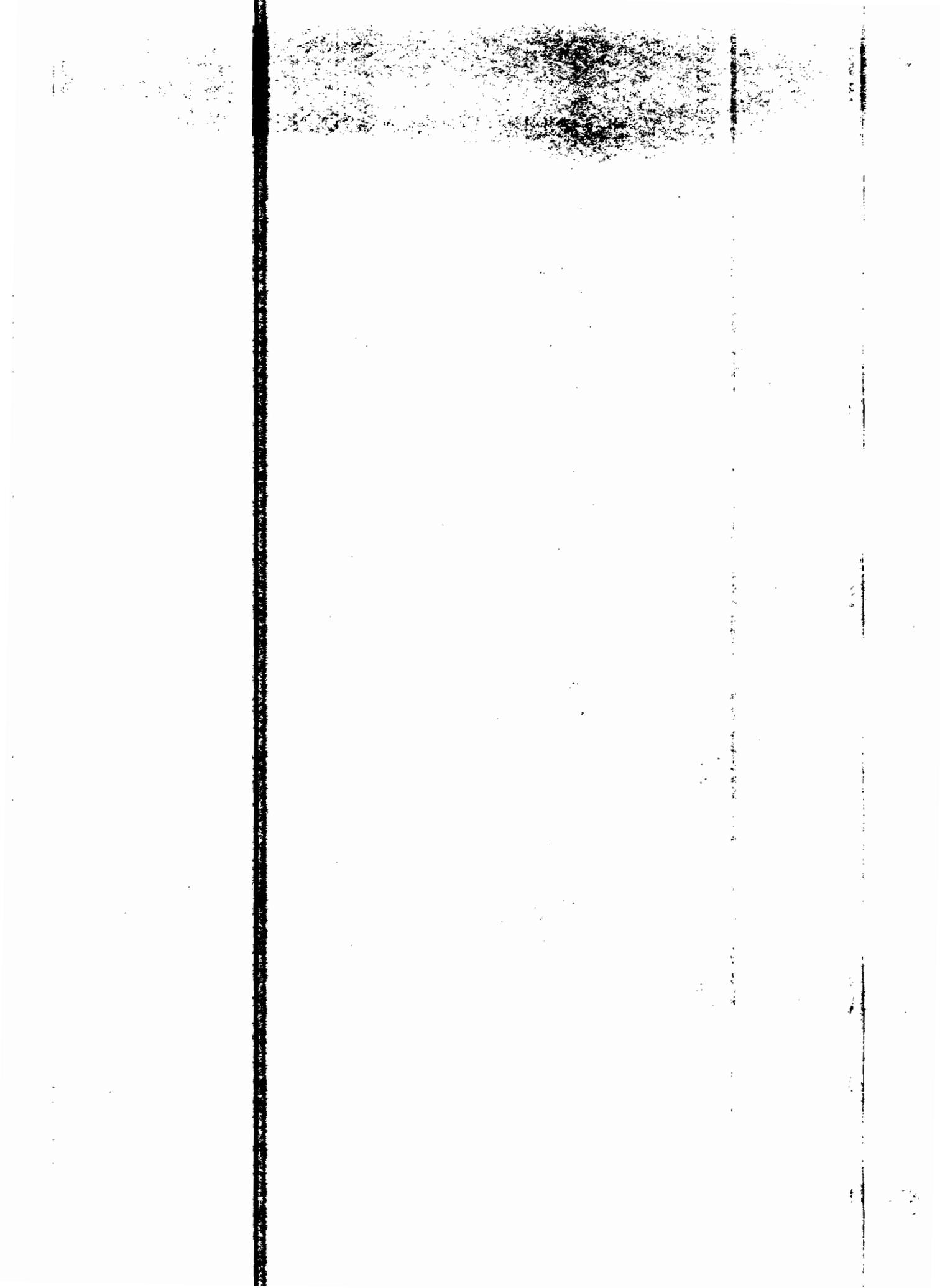
$$(d^2 x) / (dt^2) + \omega^2 x = 0$$

entonces podemos llegar a la conclusión que, dentro de nuestra aproximación, el movimiento angular del péndulo es armónico simple con  $\omega^2 = g / l$ . El ángulo  $\theta$  puede así expresarse en la forma:

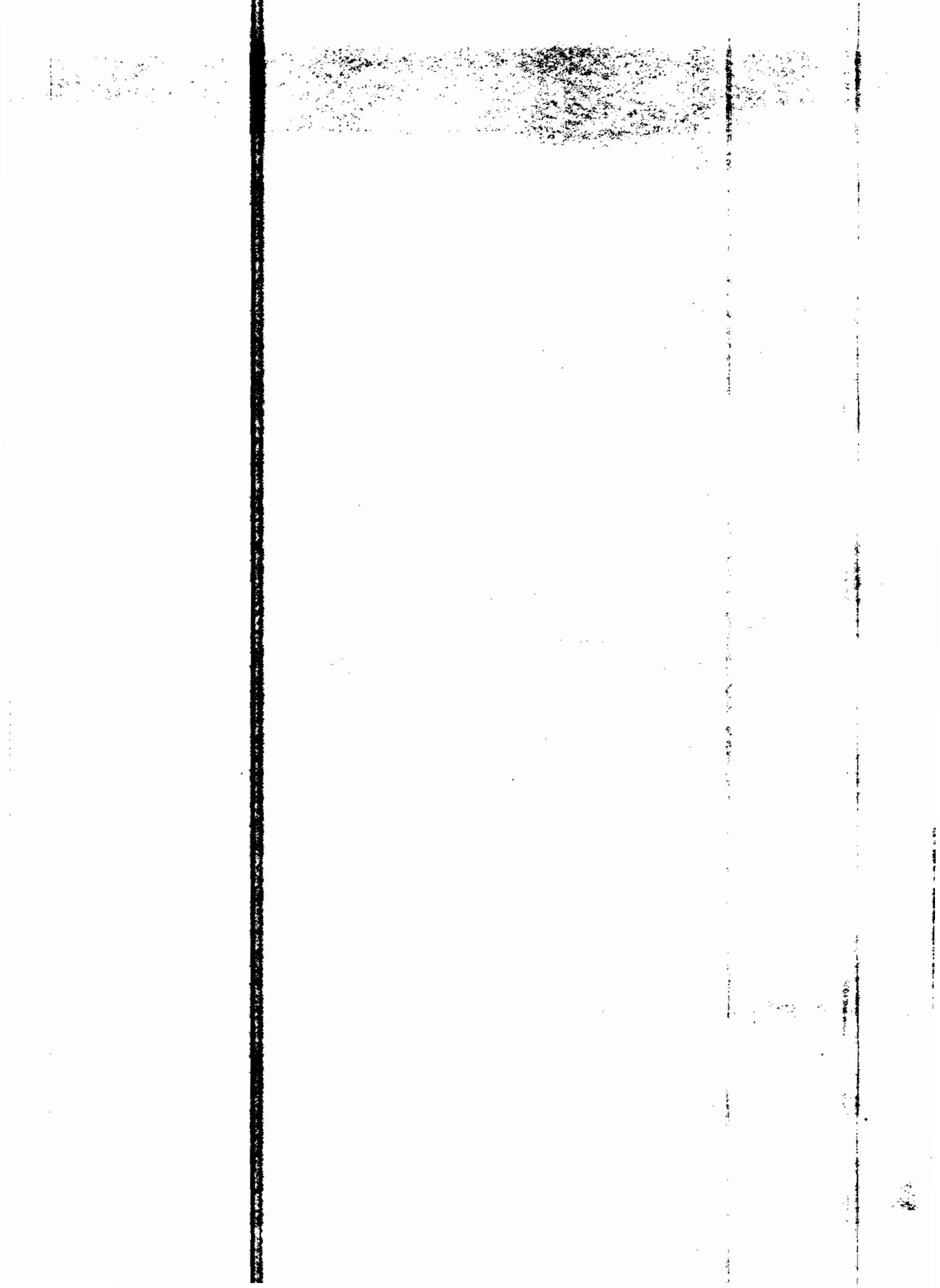
$$\theta = \theta_0 \text{ sen}(\omega t + \alpha) \quad [\text{Ec. Movimiento Pendular}]$$

Temas Relacionados:

Contenido  
Péndulo Simple  
Formulas y leyes del péndulo Simple



## **G. DATOS TECNICOS DE LOS SEMICONDUCTORES.**



**NTE960**  
**Integrated Circuit**  
**3 Terminal Positive Voltage Regulator,**  
**5V, 1A**

**Description:**

The NTE960 is a positive voltage regulator in a TO220 type package that employs current limiting, thermal shutdown, and safe-area compensation which makes it remarkably rugged under most operating conditions. With adequate heat sinking the NTE960 can deliver output currents in excess of 1 amp.

**Features:**

- No External Components Required
- Internal Thermal Overload Protection
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

**Absolute Maximum Ratings:**

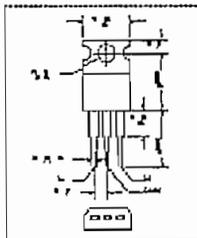
Input Voltage, $V_{IN}$	35V
Input-Output Differential	25V
Internal Power Dissipation (Note 1), $P_D$	Internally Limited
Operating Junction Temperature Range, $T_J$	0°C to +125°C
Storage Temperature Range, $T_{stg}$	-65°C to +150°C
Lead Temperature (During soldering, 10sec., $T_L$ )	+230°C

Note 1. For calculations of junction temperature rise due to power dissipation, thermal resistance junction to ambient ( $R_{thJA}$ ) is 50°C/W (no heat sink) and 5°C/W (infinite heat sink).

Electrical Characteristics: ( $V_{OUT} = 5V$ ,  $V_{IN} = 10V$ ,  $I_O = 500mA$ ,  $C_{IN} = 2.2\mu F$ ,  $C_{OUT} = 1\mu F$ ,  $P_D = 1.5W$  unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Output Voltage	$V_O$	$T_J = +25^\circ C$	4.8	5.0	5.2	V
		5mA $\leq I_O \leq$ 1A, $P_O \leq$ 15W, 7V $\leq V_{IN} \leq$ 20V	4.75	5.00	5.25	V
Line Regulation	$Reg_{line}$	$T_J = +25^\circ C$ , Note 2	-	8	50	mV
		7V $\leq V_{IN} \leq$ 25V	-	2	15	mV
Load Regulation	$Reg_{load}$	$T_J = +25^\circ C$ , Note 2	-	15	100	mV
		5mA $\leq I_O \leq$ 1.5A	-	5	50	mV
Quiescent Current	$I_B$	$T_J = +25^\circ C$	-	1.0	2.0	mA
Quiescent Current Change	$I_B$	With line, 7V $\leq V_{IN} \leq$ 25V	-	-	0.5	mA
		With load, 5mA $\leq I_O \leq$ 1A	-	-	0.5	mA
Ripple Rejection	RR	8V $\leq V_{IN} \leq$ 18V, $f = 120Hz$	54	66	-	dB
Dropout Voltage	$V_{IN} - V_{OUT}$	$T_J = +25^\circ C$ , $I_O = 1A$	-	1.1	-	V
Peak Output Current	$I_{max}$	$T_J = +25^\circ C$	-	2.2	-	A
Average Temperature Coefficient of Output Voltage	$TCV_O$	$I_O = 5mA$ , $0^\circ \leq T_J \leq +100^\circ C$	-	-0.4	-	mV/°C

Note 2. Regulation is measured at a constant junction temperature by pulse testing with a low duty cycle. Changes in output voltage due to heating effects must be taken into account.



[Home](#) [Index](#) [Back](#)

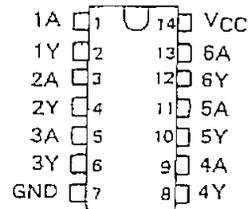


**SN5414, SN54LS14,  
SN7414, SN74LS14**  
**HEX SCHMITT-TRIGGER INVERTERS**  
DECEMBER 1983—REVISED MARCH 1988

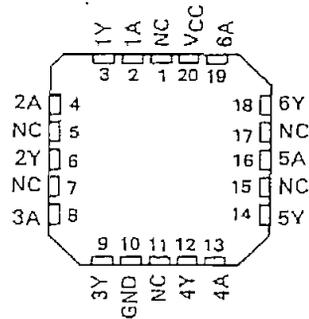
- Operation from Very Slow Edges
- Improved Line-Receiving Characteristics
- High Noise Immunity

SN5414, SN54LS14 . . . J OR W PACKAGE  
SN7414 . . . N PACKAGE  
SN74LS14 . . . D OR N PACKAGE

(TOP VIEW)



SN54LS14 . . . FK PACKAGE  
(TOP VIEW)



NC—No Internal connection

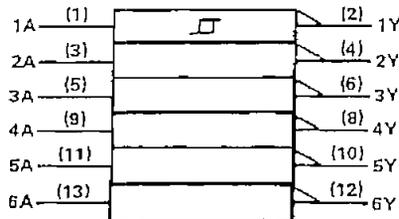
**description**

Each circuit functions as an inverter, but because of the Schmitt action, it has different input threshold levels for positive ( $V_{T+}$ ) and for negative going ( $V_{T-}$ ) signals.

These circuits are temperature-compensated and can be triggered from the slowest of input ramps and still give clean, jitter-free output signals.

The SN5414 and SN54LS14 are characterized for operation over the full military temperature range of  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ . The SN7414 and the SN74LS14 are characterized for operation from  $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ .

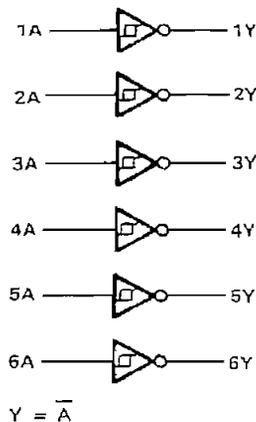
**logic symbol†**



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

Pin numbers shown are for D, J, N, and W packages.

**logic diagram (positive logic)**



PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

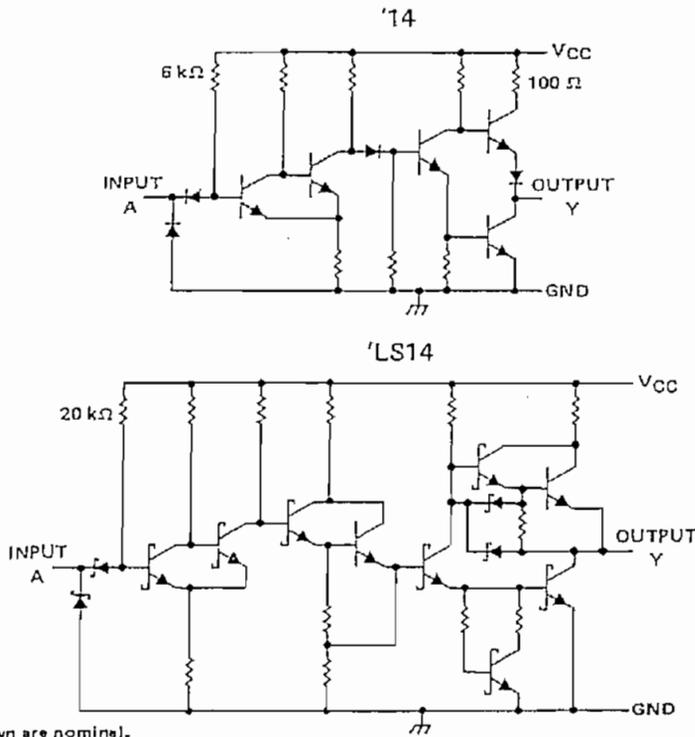


POST OFFICE BOX 655012 • DALLAS, TEXAS 75265



SN5414, SN54LS14, SN7414, SN74LS14  
 HEX SCHMITT-TRIGGER INVERTERS

schematics



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, $V_{CC}$ (see Note 1) .....	7 V
Input voltage: '14 .....	5.5 V
'LS14 .....	7 V
Operating free-air temperature: SN54' .....	-55°C to 125°C
SN74' .....	0°C to 70°C
Storage temperature range .....	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.



# SN5414, SN7414 HEX SCHMITT-TRIGGER INVERTERS

## recommended operating conditions

	SN5414			SN7414			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{CC}$ Supply voltage	4.5	5	5.5	4.75	5	5.25	V
$I_{OH}$ High-level output current			-0.8			-0.8	mA
$I_{OL}$ Low-level output current			16			16	mA
$T_A$ Operating free-air temperature	-55		125	0		70	°C

## electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
$V_{T+}$	$V_{CC} = 5\text{ V}$	1.5	1.7	2	V
$V_{T-}$	$V_{CC} = 5\text{ V}$	0.6	0.9	1.1	V
Hysteresis ( $V_{T+} - V_{T-}$ )	$V_{CC} = 5\text{ V}$	0.4	0.8		V
$V_{IK}$	$V_{CC} = \text{MIN}, I_I = -12\text{ mA}$			-1.5	V
$V_{OH}$	$V_{CC} = \text{MIN}, V_I = 0.6\text{ V}, I_{OH} = -0.8\text{ mA}$	2.4	3.4		V
$V_{OL}$	$V_{CC} = \text{MIN}, V_I = 2\text{ V}, I_{OL} = 16\text{ mA}$		0.2	0.4	V
$I_{T+}$	$V_{CC} = 5\text{ V}, V_I = V_{T+}$		-0.43		mA
$I_{T-}$	$V_{CC} = 5\text{ V}, V_I = V_{T-}$		-0.56		mA
$I_I$	$V_{CC} = \text{MAX}, V_I = 5.5\text{ V}$			1	mA
$I_{IH}$	$V_{CC} = \text{MAX}, V_{IH} = 2.4\text{ V}$			40	$\mu\text{A}$
$I_{IL}$	$V_{CC} = \text{MAX}, V_{IL} = 0.4\text{ V}$		-0.8	-1.2	mA
$I_{OS}\S$	$V_{CC} = \text{MAX}$	-18		-55	mA
$I_{CCH}$	$V_{CC} = \text{MAX}$		22	36	mA
$I_{CCL}$	$V_{CC} = \text{MAX}$		39	60	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$ .

§ Not more than one output should be shorted at a time.

## switching characteristics, $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH}$	A	Y	$R_L = 400\ \Omega, C_L = 15\ \text{pF}$		15	22	ns
$t_{PHL}$					15	22	ns

  
**TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265



# SN54LS14, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

recommended operating conditions

	SN54LS14			SN74LS14			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{CC}$ Supply voltage	4.5	5	5.5	4.75	5	5.25	V
$I_{OH}$ High-level output current			-0.4			-0.4	mA
$I_{OL}$ Low-level output current			4			8	mA
$T_A$ Operating free-air temperature	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54LS14			SN74LS14			UNIT
		MIN	TYP‡	MAX	MIN	TYP‡	MAX	
$V_{T+}$	$V_{CC} = 5\text{ V}$	1.4	1.6	1.9	1.4	1.6	1.9	V
$V_{T-}$	$V_{CC} = 5\text{ V}$	0.5	0.8	1	0.5	0.8	1	V
Hysteresis ( $V_{T+} - V_{T-}$ )	$V_{CC} = 5\text{ V}$	0.4	0.8		0.4	0.8		V
$V_{IK}$	$V_{CC} = \text{MIN}$ , $I_I = -18\text{ mA}$			-1.5			-1.5	V
$V_{OH}$	$V_{CC} = \text{MIN}$ , $V_I = 0.5\text{ V}$ , $I_{OH} = -0.4\text{ mA}$	2.5	3.4		2.7	3.4		V
$V_{OL}$	$V_{CC} = \text{MIN}$ , $V_I = 1.9\text{ V}$ $I_{OL} = 4\text{ mA}$ $I_{OL} = 8\text{ mA}$		0.25	0.4		0.25	0.4	V
$I_{T+}$	$V_{CC} = 5\text{ V}$ , $V_I = V_{T+}$		-0.14			-0.14		mA
$I_{T-}$	$V_{CC} = 5\text{ V}$ , $V_I = V_{T-}$		-0.18			-0.18		mA
$I_I$	$V_{CC} = \text{MAX}$ , $V_I = 7\text{ V}$		0.1			0.1		mA
$I_{IH}$	$V_{CC} = \text{MAX}$ , $V_{IH} = 2.7\text{ V}$		20			20		μA
$I_{IL}$	$V_{CC} = \text{MAX}$ , $V_{IL} = 0.4\text{ V}$		-0.4			-0.4		mA
$I_{OS}§$	$V_{CC} = \text{MAX}$	-20		-100	-20		-100	mA
$I_{CCH}$	$V_{CC} = \text{MAX}$		8.6	16		8.6	16	mA
$I_{CCL}$	$V_{CC} = \text{MAX}$		12	21		12	21	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

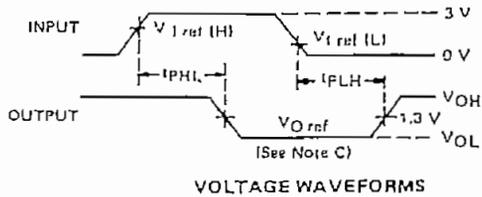
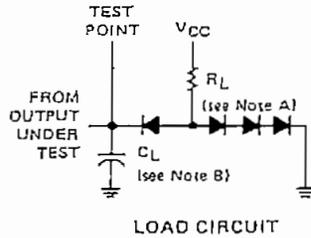
§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics,  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH}$	A	Y	$R_L = 2\text{ k}\Omega$ , $C_L = 15\text{ pF}$		15	22	ns
$t_{PHL}$					15	22	ns

# SN5414, SN54LS14, SN7414, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

## PARAMETER MEASUREMENT INFORMATION

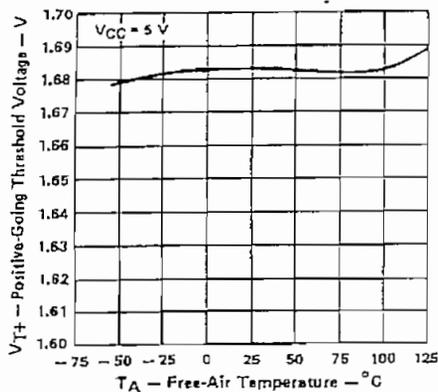


- NOTES: A. All diodes are 1N3064 or equivalent.  
B.  $C_L$  includes probe and jig capacitance.  
C. Generator characteristics and reference voltage are:

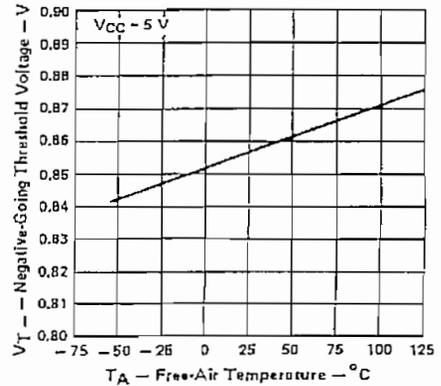
	Generator Characteristics				Reference Voltages		
	$Z_{out}$	PRR	$t_r$	$t_f$	$V_{I\ ref\ (H)}$	$V_{I\ ref\ (L)}$	$V_{O\ ref}$
SN54*/SN74*	50 $\Omega$	1 MHz	10 ns	10 ns	1.7 V	0.9 V	1.5 V
SN54LS*/SN74LS*	50 $\Omega$	1 MHz	15 ns	6 ns	1.6 V	0.8 V	1.3 V

## TYPICAL CHARACTERISTICS OF '14 CIRCUITS

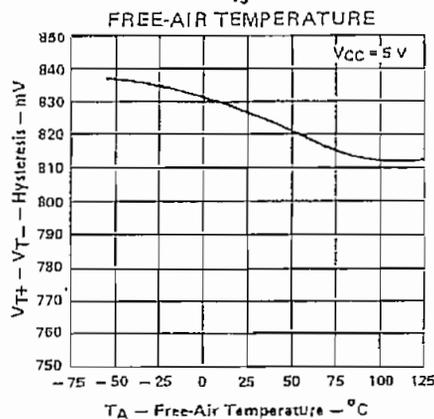
POSITIVE-GOING THRESHOLD VOLTAGE  
vs  
FREE-AIR TEMPERATURE



NEGATIVE-GOING THRESHOLD VOLTAGE  
vs  
FREE-AIR TEMPERATURE



HYSTERESIS  
vs  
FREE-AIR TEMPERATURE



Data for temperatures below  $0^{\circ}C$  and  $70^{\circ}C$  and supply voltages below 4.75V and above 5.25 V are applicable for SN5414 only.

TEXAS  
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN5414, SN7414  
 HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF '14 CIRCUITS

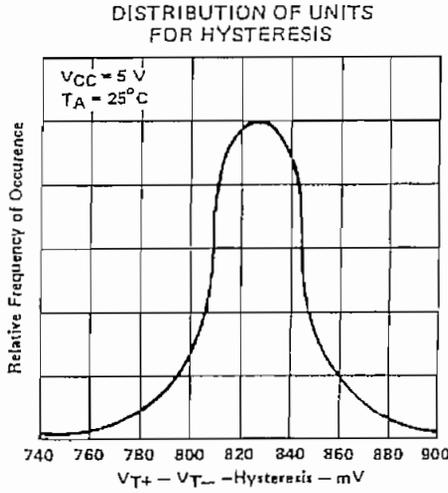


FIGURE 4

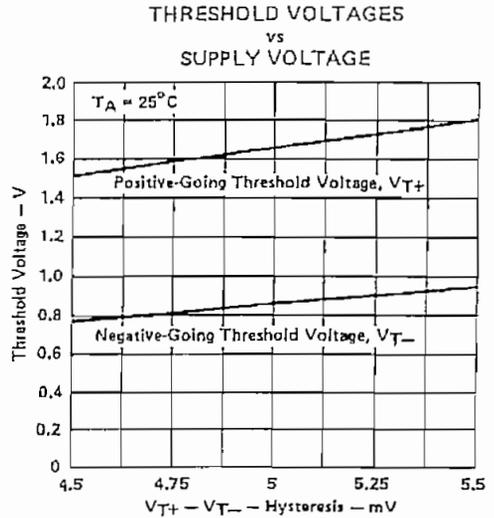


FIGURE 5

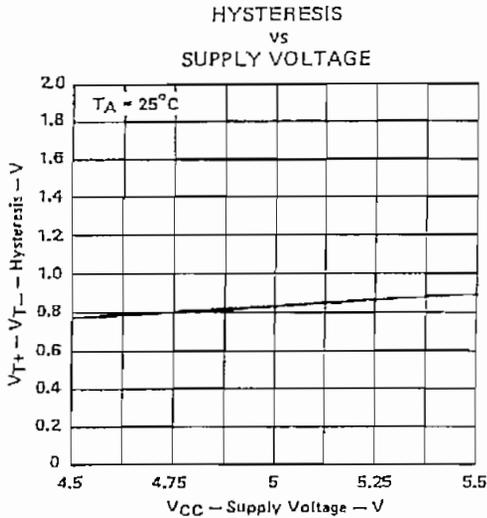


FIGURE 6

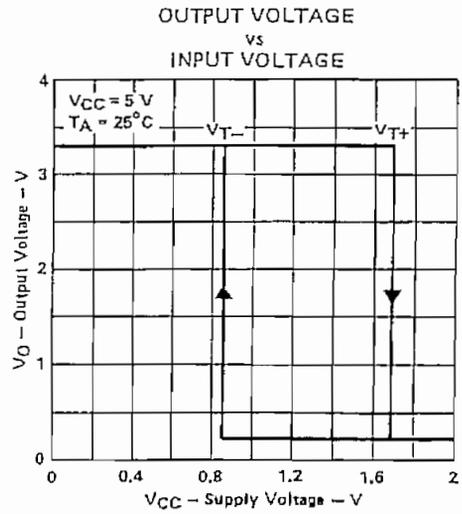


FIGURE 7

Data for temperatures below  $0^\circ\text{C}$  and  $70^\circ\text{C}$  and supply voltages below 4.75 V and above 5.25 V are applicable for SN5414 only.

TYPICAL CHARACTERISTICS OF 'LS14 CIRCUITS

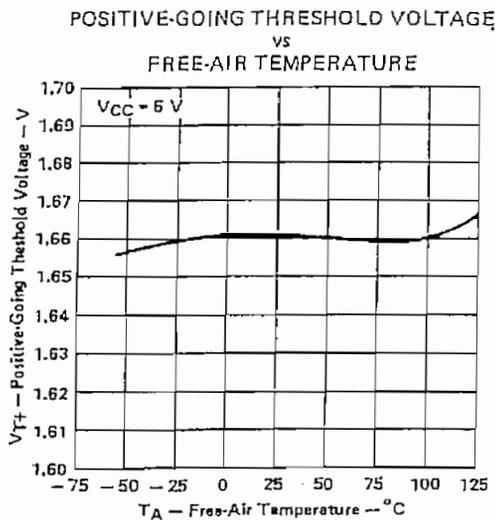


FIGURE 8

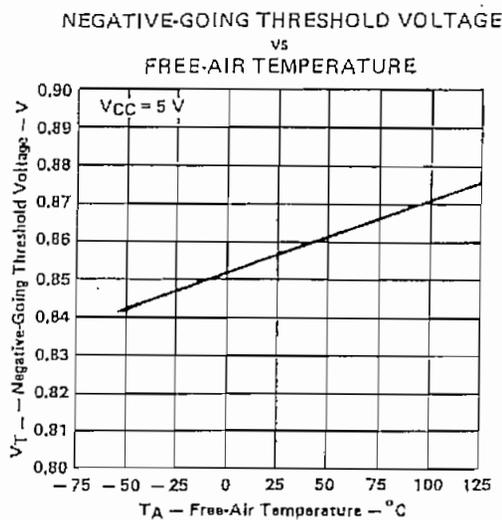


FIGURE 9

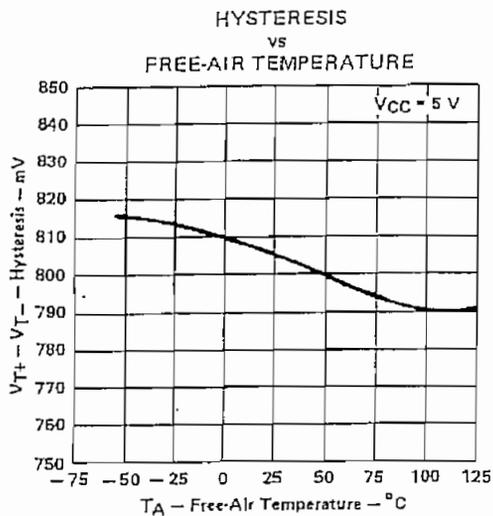


FIGURE 10

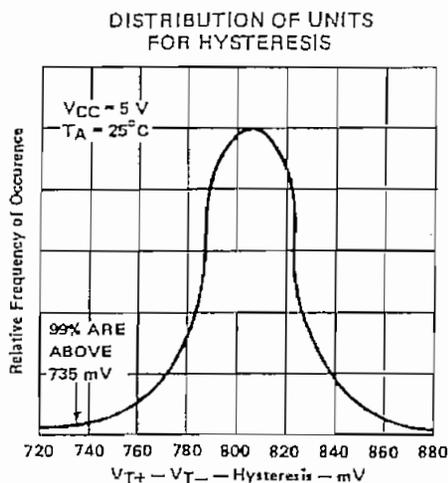


FIGURE 11

Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS14 only.

SN54LS14, SN74LS14  
HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF 'LS14 CIRCUITS

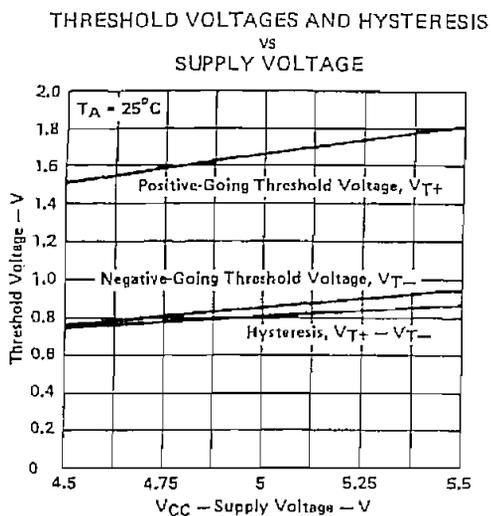


FIGURE 12

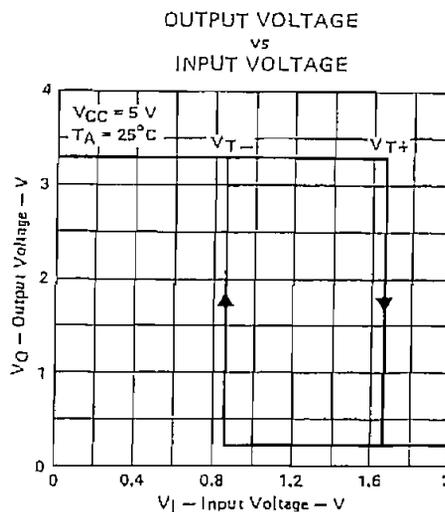
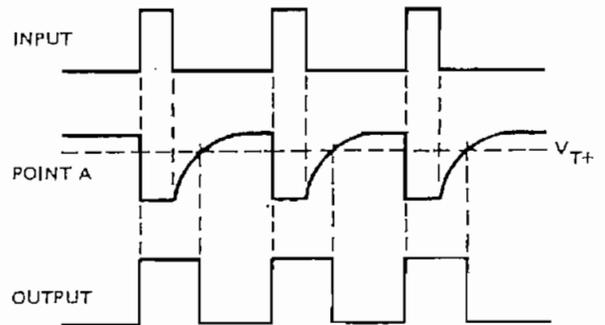
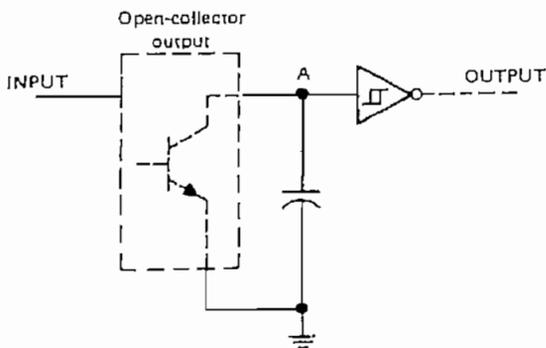
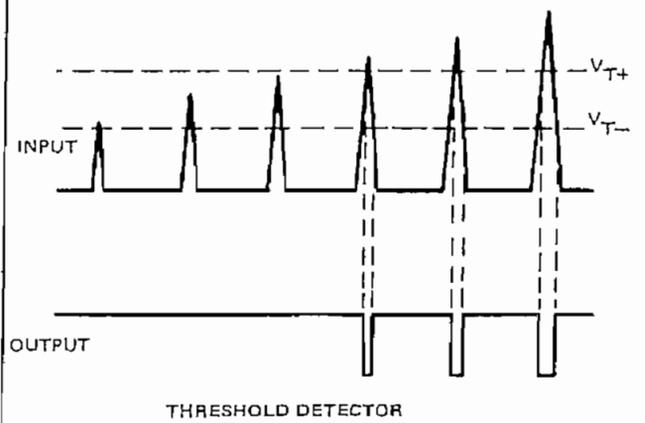
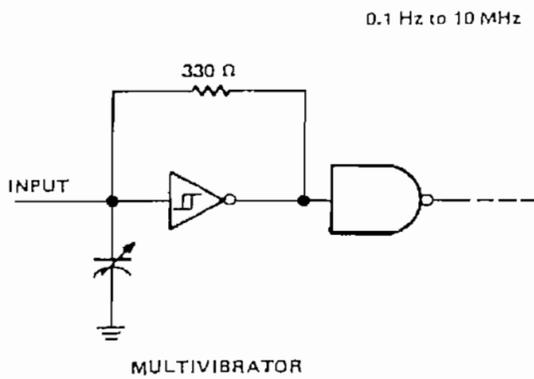
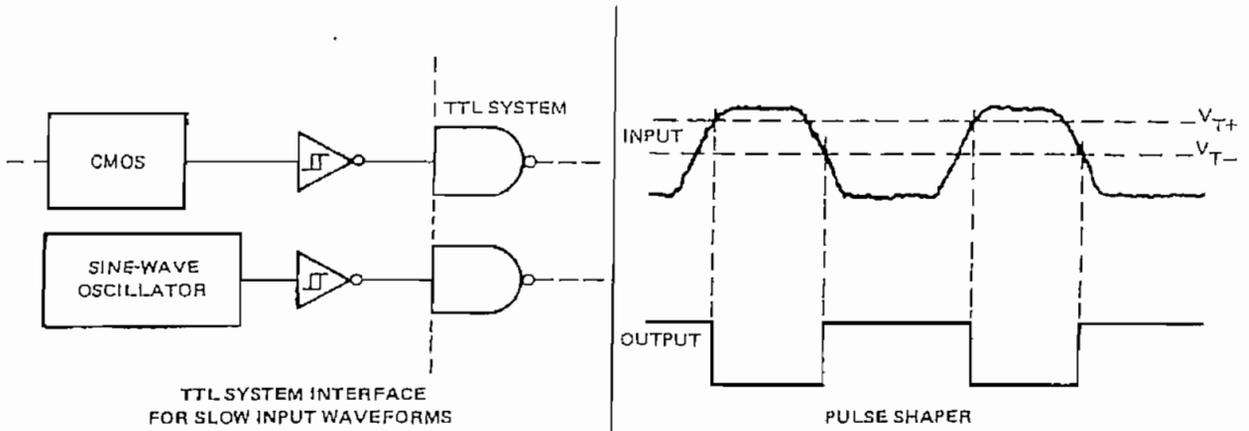


FIGURE 13

Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS14 only.

TYPICAL APPLICATION DATA



PULSE STRETCHER

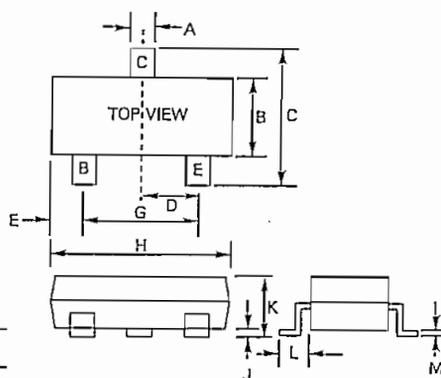
### Features

- Epitaxial Planar Die Construction
- Available in both Through-Hole and Surface Mount Packages
- Ideal for Switching and Amplifier Applications
- Complementary PNP Types Available (2N3906/MMBT3906)

### Mechanical Data

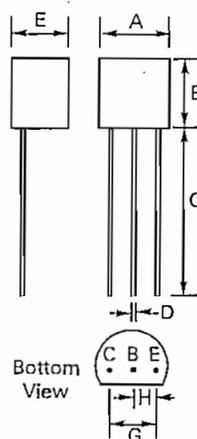
- Case: TO-92/SOT-23, Plastic
- Leads/Terminals: Solderable per MIL-STD-202, Method 208
- Terminal Connections: See Diagram
- Marking: TO-92 Type Number  
SOT-23 1A, 1N, R1A, K1N
- Approx Weight: TO-92 0.18 grams  
SOT-23 0.008 grams

MMBT3904



SOT-23		
Dim	Min	Max
A	0.37	0.51
B	1.19	1.40
C	2.10	2.50
D	0.89	1.05
E	0.45	0.61
G	1.78	2.05
H	2.65	3.05
J	0.013	0.15
K	0.89	1.10
L	0.45	0.61
M	0.076	0.178
All Dimensions in mm		

2N3904



TO-92		
Dim	Min	Max
A	4.32	4.83
B	4.32	4.78
C	12.50	15.62
D	0.36	0.56
E	3.15	3.94
G	2.29	2.79
H	1.14	1.40
All Dimensions in mm		

### Maximum Ratings @ T<sub>A</sub> = 25°C unless otherwise specified

Characteristic	Symbol	Value	Unit
Collector-Base Voltage	V <sub>CB0</sub>	60	V
Collector-Emitter Voltage	V <sub>CE0</sub>	40	V
Emitter-Base Voltage	V <sub>EB0</sub>	5.0	V
Collector Current - Continuous	I <sub>C</sub>	100	mA
Collector Current - Peak	I <sub>CM</sub>	200	mA
Power Dissipation TO-92 (2N3904) (Note 1) SOT-23 (MMBT3904) (Note 2) T <sub>SB</sub> = 50°C	P <sub>d</sub>	500 300	mW
Operating and Storage Temperature Range	T <sub>J</sub> , T <sub>STG</sub>	-55 to +150	°C

**Electrical Characteristics** @  $T_A = 25^\circ\text{C}$  unless otherwise specified

Characteristic	Symbol	Min	Max	Unit	Test Condition
DC Current Gain	$h_{FE}$	50 70 100 60 30	— — 300 — —	—	$-V_{CE} = 1.0\text{V}, -I_C = 0.1\text{mA}$ $-V_{CE} = 1.0\text{V}, -I_C = 1.0\text{mA}$ $-V_{CE} = 1.0\text{V}, -I_C = 10\text{mA}$ $-V_{CE} = 1.0\text{V}, -I_C = 50\text{mA}$ $-V_{CE} = 1.0\text{V}, -I_C = 100\text{mA}$
Collector Saturation Voltage	$V_{CE(SAT)}$	—	0.25 0.40	V	(Note 3) $-I_C = 10\text{mA}, -I_B = 1.0\text{mA}$ $-I_C = 50\text{mA}, -I_B = 5.0\text{mA}$
Base Saturation Voltage	$V_{BE(SAT)}$	—	0.85 0.95	V	(Note 3) $-I_C = 10\text{mA}, -I_B = 1.0\text{mA}$ $-I_C = 50\text{mA}, -I_B = 5.0\text{mA}$
Collector Cutoff Current	$I_{CEX}$	—	50	nA	$-V_{EB} = 3.0\text{V}, -V_{CE} = 30\text{V}$
Emitter Cutoff Current	$I_{BL}$	—	50	nA	$-V_{EB} = 3.0\text{V}, -V_{CE} = 30\text{V}$
Collector-Base Breakdown Voltage	$V_{(BR)CBO}$	60	—	V	$-I_C = 10\mu\text{A}, -I_B = 0$
Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	40	—	V	$-I_C = 1.0\text{mA}, -I_E = 0$ (Note 3)
Emitter-base Breakdown voltage	$V_{(BR)EBO}$	5.0	—	V	$-I_E = 10\mu\text{A}, -I_C = 0$
Gain Bandwidth Product	$f_T$	250	—	MHz	$V_{CE} = 20\text{V}, -I_C = 10\text{mA},$ $-f = 100\text{MHz}$
Collector-Base Capacitance	$C_{CBO}$	—	4.5	pF	$-V_{CB} = 5.0\text{V}, -I_E = 0, f = 100\text{kHz}$
Emitter-Base Capacitance	$C_{EBO}$	—	10	pF	$-V_{EB} = 0.5\text{V}, -I_C = 0, f = 100\text{kHz}$
Noise Figure	—	—	5.0	dB	$-V_{CE} = 5.0\text{V}, -I_C = 100\mu\text{A},$ $R_G = 1.0\text{k}\Omega, -f = 10$ to $15000\text{Hz}$
Delay Time	$t_d$	—	35	ns	$-I_{B1} = 1.0\text{mA}, -I_C = 10\text{mA},$ $V_{CC} = 3.0\text{V}, V_{BE(off)} = 0.5\text{V}$
Rise Time	$t_r$	—	35	ns	$-I_{B1} = 1.0\text{mA}, -I_C = 10\text{mA},$ $-V_{CC} = 3.0\text{V}, -V_{BE(off)} = 0.5\text{V}$
Storage Time	$t_s$	—	225	ns	$-I_{B1} = -I_{B2} = 1.0\text{mA},$ $-I_C = 10\text{mA}, -V_{CC} = 3.0\text{V}$
Fall Time	$t_f$	—	75	ns	$-I_{B1} = -I_{B2} = 1.0\text{mA},$ $-I_C = 10\text{mA}, -V_{CC} = 3.0\text{V}$
Thermal Resistance, Junction to Ambient TO-92, 2N3904 SOT-23, MMBT3904	$R_{\theta JA}$	—	250 625	K/W	Note 1 Note 2

- Notes:
1. Leads maintained at a distance of 2.0mm from body at specified ambient temperature (TO-92).
  2. Device mounted on ceramic substrate 0.7mm x 2.5 cm<sup>2</sup> area (SOT-23).
  3. Pulse test: Pulse width  $\leq 300\mu\text{s}$ , duty cycle  $\leq 2\%$ .

# NTE3034A

## Phototransistor Detector

### Description:

The NTE3034A is designed for industrial processing and control applications such as light modulators, shaft or position encoders, and end tape detectors. The NTE3034A is designed to be used with the NTE3029A infrared emitter in optical slotted coupler/interrupter applications.

### Features:

- Economical, Miniature Plastic Package
- Package Designed for Accurate Positioning
- Lens Molded into Package

#### Absolute Maximum Ratings:

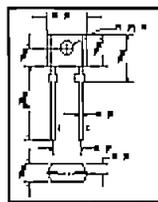
Collector-Emitter Voltage, $V_{CEO}$	30V
Total Device Dissipation ( $T_A = +25^\circ\text{C}$ ), $P_D$	150mW
Derate Above $25^\circ\text{C}$ (Note 1)	2mW/ $^\circ\text{C}$
Operating Junction Temperature Range, $T_J$	-40° to +100°C
Storage Temperature Range, $T_{stg}$	-40° to +100°C
Lead Temperature (During Soldering, 1/16" from case, 5sec max, Note 2), $T_L$	+260°C

Note 1. Measured with device soldered into a typical PC board.

Note 2. Heat sink should be applied to leads during soldering to prevent case temperature from exceeding +100°C.

#### Electro-Optical Characteristics: ( $T_A = +25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Collector Dark Current	$I_D$	$V_{CE} = 10\text{V}, H = 0$	-	-	100	nA
Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	$I_C = 10\text{mA}, H = 0$	30	-	-	V
Collector Light Current	$I_L$	$V_{CE} = 5\text{V}, H = 500\mu\text{W}/\text{cm}^2$	100	500	-	$\mu\text{A}$
Turn-On Time	$t_{on}$	$V_{CC} = 5\text{V}, R_L = 2400\ \Omega, H = 500\mu\text{W}/\text{cm}^2$	-	60	-	$\mu\text{s}$
Turn-Off Time	$t_{off}$		-	0.25	0.4	$\mu\text{s}$
Saturation Voltage	$V_{CE(sat)}$	$V_{CC} = 5\text{V}, I_C = 2\text{mA}, H = 500\mu\text{W}/\text{cm}^2$	-	0.25	0.4	V
Wavelength of Maximum Sensitivity			-	0.8	-	$\mu\text{m}$



[Home](#) [Index](#) [Back](#)

# NTE3017

## Infrared Emitting Diode

### High Speed for Remote Control

#### Description:

The NTE3017 is an infrared emitting diode in standard GaAs on GaAs technology, molded in a clear, blue-grey tinted plastic package.

#### Features:

- Low Forward Voltage
- High Radiant Power and Radiant Intensity
- Suitable for DC and High Pulse Current Operation
- High Reliability
- Standard T-1 3/4 (5mm) Package

#### Applications:

Infrared remote control and free air transmission systems with low forward voltage and comfortable radiation and angle requirements in combination with PIN photodiodes or phototransistors.

Absolute Maximum Ratings: ( $T_A = +25^\circ\text{C}$  unless otherwise specified)

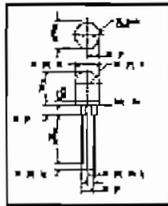
Reverse Voltage, $V_R$	5V
Forward Current, $I_F$	
Continuous	150mA
Peak (Note 1)	300mA
Surge Forward Current (Note 2), $I_{FSM}$	2.5A
Power Dissipation, $P_D$	210mW
Junction Temperature, $T_J$	+100°C
Operating Temperature Range, $T_{opr}$	-55° to +100°C
Storage Temperature Range, $T_{stg}$	-55° to +100°C
Lead Temperature (During Soldering, $t \leq 5\text{sec}$ , 2mm from case), $T_L$	+260°C
Thermal Resistance, Junction-to-Ambient, $R_{thJA}$	375K/W

Note 1.  $t_p = 100\mu\text{s}$ ,  $t_p / T = 0.5$ .

Note 2.  $t_p = 100\mu\text{s}$

Electrical Characteristics: ( $T_A = +25^\circ\text{C}$  unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Forward Voltage	$V_F$	$I_F = 100\text{mA}, t_p = 20\text{ms}$	-	1.3	1.7	V
		$I_F = 1.5\text{A}, t_p = 100\mu\text{s}$	-	2.2	3.4	V
Temperature Coefficient of Forward Voltage		$I_F = 100\text{mA}$	-	-1.3	-	mV / °C
Reverse Current	$I_R$	$V_R = 5\text{V}$	-	-	100	$\mu\text{A}$
Junction Capacitance	$C_j$	$V_R = 0, f = 1\text{MHz}, E = 0$	-	30	-	pF
Radiant Intensity	$I_e$	$I_F = 100\text{mA}, t_p = 20\text{ms}$	7	14	-	mW / sr
		$I_F = 1.5\text{A}, t_p = 100\mu\text{s}$	60	140	-	mW / sr
Angle of Half Intensity			-	$\pm 22$	-	deg
Radiant Power		$I_F = 100\text{mA}, t_p = 20\text{ms}$	-	13	-	mW
Temperature Coefficient of Radiant Power		$I_F = 20\text{mA}$	-	-0.8	-	% / °C
Peak Wavelength		$I_F = 100\text{mA}$	-	950	-	nm
Temperature Coefficient of Peak Wavelength		$I_F = 100\text{mA}$	-	0.2	-	nm / °C
Spectral Bandwidth		$I_F = 100\text{mA}$	-	50	-	nm
Rise Time	$t_r$	$I_F = 100\text{mA}$	-	800	-	ns
		$I_F = 1.5\text{A}$	-	400	-	ns
Fall Time	$t_f$	$I_F = 100\text{mA}$	-	800	-	ns
		$I_F = 1.5\text{A}$	-	400	-	ns



[Home](#) [Index](#) [Back](#)

## BIBLIOGRAFIA

1. *Sistemas Digitales, Principios y Aplicaciones*, Ronald J. Tocci, Prentice Hall Hispanoamericana, 5ta. Edición, México, 1993.
2. *Programación en Turbo Pascal, Versiones 5.5, 6.0, y 7.0*, Luis Joyanes Aguilar, Ed. Mc Graw-Hill, 2da. Edición, Madrid, 1993.
3. *How to use a PC's Parallel Port for Monitoring and Control Purposes, Part 2*, Microcomputer Journal, July/August, 1994.
4. *Guía Completa de Visual Basic para WINDOWS, "Una Introducción Práctica a la Programación con WINDOWS"*, Ross Nelson, Ed. Mc Graw-Hill, 2da. Edición, España, 1994.
5. *Manual de Visual Basic 3 para Windows*, Gary Cornell, Ed. Osborne/Mc Graw-Hill, España, 1994.
6. *Introducción a la Física I*, Alonso/Acosta, Ed. Cultural, Bogotá-Colombia, 1982.
7. *FISICA, Volumen I: Mecánica*, Marcelo Alonso, Edward J. Finn, EPN, 1989.
8. *Controlling de World with your PC*, Paul Bergsman, High Text Publications, Inc., Solanda Beach, CA., 1994.
9. *Tarjeta Experimental para el PC*, Felipe González, Electrónica & Computadores, Proyectos • Tecnología • Aplicaciones Prácticas • Teoría, Año 1, No.1, Publicaciones CEKIT S.A., Pereira-Colombia, 1994.
10. *Microcontroladores*, Tomás Delgado P., y *Automatice su hogar con un PC*, por: John Jairo Naranjo, Electrónica & Computadores, Proyectos • Tecnología • Aplicaciones Prácticas • Teoría, Año 1, No.2, Publicaciones CEKIT S.A., Pereira-Colombia, 1994.

11. *PC para Ingenieros, Capítulo IX (Puerto Paralelo)*, Zelenovsky, Ed. ESPE, Quito-Ecuador, 1994.
12. *Cómo usar el Puerto Paralelo de un PC para Monitoreo y Control*, Jan Axelson, Electrónica & Computadores, Proyectos • Tecnología • Aplicaciones Prácticas • Teoría, Publicaciones CEKIT S.A., Pereira-Colombia, 1994.
13. *1001 Trucos de Programación con Visual Basic*, Kris Jamsa, Lars Klander, Ed. ANAYA MULTIMEDIA S.A.; España, 1998.
14. *Práctica No. 8, Polarización de Transistores y Trabajo en Corte y Saturación*, EPN, Facultad de Ingeniería Eléctrica, Departamento de Electrónica y Telecomunicaciones, Laboratorio de Electrónica I, EPN, Quito-Ecuador, 1994.
15. *Circuitos de Pulsos Digitales y de Conmutación*, Jacob Millman, Ph. D, Herbert Taub, Ph. D, Ed. Mc Graw-Hill, México, 1965.
16. *IEEE 1284, Electrical Interface*, [www.fapo.com/1284elec.htm](http://www.fapo.com/1284elec.htm)
17. *IEEE 1284, Cables Assemblies*, [www.fapo.com/1284cabl.htm](http://www.fapo.com/1284cabl.htm)
18. *IEEE 1284, Connectors*, [www.fapo.com/1284conn.htm](http://www.fapo.com/1284conn.htm)
19. *Parallel Port Background*, [www.fapo.com/porthist.htm](http://www.fapo.com/porthist.htm)
20. *Sistema de Adquisición de Datos y Procesamiento Digital de Laringogramas*, Olga Lidia Rivera Rivera, Tesis de Grado, EPN, Quito, Julio, 1993.
21. *NTE 960, Integrated Circuit 3 Terminal Positive Voltage Regulator, 5V, 1A*, [www.nteinc.com](http://www.nteinc.com)
22. *SN74LS14, Hex Schmitt-Trigger Inverters*, [www.ti.com/sc/docs/products/logic/sn7414.html#Datasheets](http://www.ti.com/sc/docs/products/logic/sn7414.html#Datasheets), [www.questlink.com](http://www.questlink.com)

23. *NTE 3034A, Phototransistor Detector*, [www.nteinc.com](http://www.nteinc.com)
24. *NTE 3017, Infrared Emitting Diode High Speed for Remote Control*,  
[www.nteinc.com](http://www.nteinc.com)
25. *2N3904, Transistor BJT tipo NPN*, [www.vishay-liteon.com/cgi-bin/cgiwrap/~diodes/product.cgi?type=search&query=2N3904&by=product&B1=Search](http://www.vishay-liteon.com/cgi-bin/cgiwrap/~diodes/product.cgi?type=search&query=2N3904&by=product&B1=Search)
26. *Manual del Ingeniero Mecánico Volumen I*, Theodore Baumeister, Ed. McGraw-Hill, México DF, 1992.
27. *IOPORT.OCX*, [www.uct.on.ca](http://www.uct.on.ca), [www.download.com](http://www.download.com)
28. *PARALLEL PORT CENTRAL*, [www.uct.on.ca](http://www.uct.on.ca)
29. *Use a Parallel Port to Make Voltage Measurements*, Microcomputer Journal, EPN, March/April, 1994.
30. *Recycling Old Computers*, Raymond H. Green, COMPUTERCRAFT, EPN, January, 1993.
31. *Using Rotary Encoders*, COMPUTERCRAFT, EPN, February, 1993.
32. *ComputerCraft Magazine's PC Connector Guide*, T.J. Byers, COMPUTERCRAFT, EPN, February, 1993.
33. *Schmitt-trigger*, [www.e-online.de/sites/Schmitt-Trigger.htm](http://www.e-online.de/sites/Schmitt-Trigger.htm)
34. *Interface digital basado en Centronics*, Sebastián Pastor Romero/Abel Miguel Garijo/J. A. Felipe Fernández Hernández, Revista Española de Electrónica, EPN, Octubre, 1991.
35. *Electrónica III*, Ing. Tarquino Sánchez, EPN, Quito, 1995.

# INDICE

## A

Aceleración de la Gravedad .....	248
Acerca del Puerto Paralelo.....	5
Análisis de una Configuración en Colector Abierto.....	23
Antecedentes Generales.....	32
Ayuda del Programa Computacional "EFC" .....	245

## B

Bus Externo .....	12
Bus Interno .....	12
Buses en un Computador PC o Compatible .....	12

## C

Caída Libre .....	173
Caída Libre .....	204
Caída Libre .....	210
Caída Libre .....	247
Calibración del Interfaz "Caída Libre" y "Plano Inclinado" .....	194
Calibración del Interfaz "Péndulo Simple" .....	191
Calibración Electrónica del Equipo .....	191
Cambio del Sistema de Iluminación de Luz Visible a Luz Infrarroja .....	138
Casos Especiales en Caída Libre .....	250
Conclusiones y Recomendaciones .....	156

## D

Datos Técnicos de los Semiconductores .....	258
Deducción de la Fórmula de Péndulo Simple .....	257
Diagrama de Bloques del Circuito Interfaz .....	34
Diagrama de Bloques del Programa Computacional .....	74
Diagrama Electrónico de "Caída Libre" .....	237
Diagrama Electrónico de "Péndulo Simple" .....	236
Diagrama Electrónico de "Plano Inclinado" .....	238
Diagramas de los Módulos Físicos .....	215
Diagramas Electrónicos Finales .....	235

Diagramas para la Construcción del Prototipo "Caída Libre" .....	220
Diagramas para la Construcción del Prototipo "Péndulo Simple" .....	215
Diagramas para la Construcción del Prototipo "Plano Inclinado" .....	224
Direcciones posibles del Puerto Paralelo .....	13
Diseño del Circuito Impreso de "Caída Libre" .....	230
Diseño del Circuito Impreso de "Péndulo Simple" .....	229
Diseño del Circuito Impreso de "Plano Inclinado" .....	231
Diseño del Interfaz General del Proyecto .....	52
Diseño del Interfaz para el Prototipo de "Caída Libre" .....	67
Diseño del Interfaz para el Prototipo "Péndulo Simple" .....	63
Diseño del Interfaz para el Prototipo de "Plano Inclinado" .....	70
Diseño de los Circuitos Impresos .....	229
Diseño del Programa en Visual Basic .....	74

## E

Ecuaciones de la Caída Libre de los Cuerpos ....	249
Elección de los elementos para el Diseño .....	59
Electrónica del Puerto Paralelo, Características .....	19
Especificaciones de Procedimiento .....	12
Especificaciones Eléctricas .....	8
Especificaciones Funcionales .....	9
Especificaciones Mecánicas .....	8
Esquemas Gráficos .....	215
Esquemas Gráficos .....	220
Esquemas Gráficos .....	224

## F

Fórmulas y Leyes del Péndulo Simple .....	256
Fricción .....	253
Funciones Públicas Utilizadas en el Programa .....	123

<i>I</i>		Propiedades de la Herramienta IOPORT .....	136
Información adicional para el Puerto Paralelo ....	241	Pruebas de Electrónica .....	145
Interfaz CENTRONICS .....	8	Pruebas de los Interfaces Electrónicos Mediante el uso de rutinas de Programa(Software) .....	152
Introducción .....	1, 198		
<i>L</i>		<i>R</i>	
Líneas de Protocolo .....	17	Recomendaciones .....	160
Líneas de Tierra .....	18	¿Registro de Datos Bidireccional? .....	19
Lista de materiales del Prototipo "Caída Libre" .....	233	Registros de Entrada y Salida .....	14
Lista de materiales del Prototipo "Péndulo Simple" .....	232	Resultados .....	138
Lista de materiales del Prototipo "Plano Inclinado" .....	234		
Listado del Programa de pruebas en "QBASIC" .....	154	<i>S</i>	
<i>M</i>		Simulación .....	76, 202
Manejo de los Registros del Puerto Paralelo mediante el uso de Programación Computacional (Software) .....	25	Sistema de Unidades CGS (Centímetro-Gramo-Segundo) .....	212
Manejo del Paquete Computacional del Proyecto .....	198	Sistema de Unidades Inglés (Pie-Libra-Segundo) .....	213
Manejo del Puerto Paralelo del PC.....	5, 14	Sistema de Unidades MKS (Metro-Kilogramo-Segundo) .....	212
Mantenimiento de los Equipos .....	196	Sitios WEB importantes para visitar .....	239
Manual del Usuario .....	163	Sitios WEB para obtener el "IOPORT.OCX" ...	239
Materiales Utilizados .....	232		
Menú de Opciones .....	200	<i>T</i>	
<i>O</i>		Tabla de Coeficientes de Fricción .....	254
Opción "AYUDA" .....	214	Tabla de Longitudes Físicas en [cm] .....	220
Opción "EXPERIMENTO" .....	201	Tabla de Longitudes Físicas en [cm] .....	223
Opción "UNIDADES" .....	212	Tabla de Longitudes Físicas en [cm] .....	228
Opción "VENTANA" .....	213	Tabla de Resultados Experimentales del Interfaz "Caída Libre" .....	147
Operación de los Prototipos por el Usuario .....	163	Tabla de Resultados Experimentales del Interfaz "Péndulo Simple" .....	145
<i>P</i>		Tabla de Resultados Experimentales del Interfaz "Plano Inclinado" .....	150
Péndulo Simple .....	163	Teoría del Funcionamiento del Interfaz Básico .....	38
Péndulo Simple .....	206	Teoría del Funcionamiento del "Schmitt Trigger" .....	45
Péndulo Simple .....	211	Teoría y Diseño del Interfaz General del Proyecto .....	32
Péndulo Simple .....	255		
Plano Inclinado .....	182	<i>U</i>	
Plano Inclinado .....	205	Utilización de la Herramienta IOPORT.OCX ...	135
Plano Inclinado .....	211		
Plano Inclinado .....	251	<i>V</i>	
Práctica .....	76	Ventajas y Desventajas de la Comunicación Paralela respecto a la Comunicación Serial .....	30
Práctica .....	202	Ventana de Acceso Directo .....	200
Práctica .....	208	Ventanas de Formularios y Código de Programa .....	82
Procedimientos para realizar la Práctica de "Caída Libre" .....	176	Ventanas Gráficas para la Simulación .....	100
Procedimientos para realizar la Práctica de "Péndulo Simple" .....	167		
Procedimientos para realizar la Práctica de "Plano Inclinado" .....	186		