

15275-B1
TESIS DI'

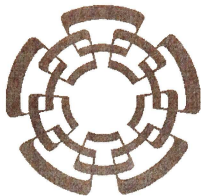


CINVESTAV-IPN
Biblioteca de Ingeniería Eléctrica



FB0000014026

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA



**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL**

**UNIDAD ZACATENCO
DEPARTAMENTO DE INGENIERIA ELECTRICA SECCION DE
COMPUTACION**

**SISTEMA DIGITAL PARA MEDIR EL NIVEL DE ESTRÉS
EN UN SER HUMANO**

Tesis que presenta
ING. MARIA ELENA ACEVEDO MOSQUEDA

Para obtener el Grado de
MAESTRO EN CIENCIAS

En la Especialidad de **Ingeniería Eléctrica**

Directores de Tesis

**DR. ADRIANO DE LUCA PENNACCHIA
M. EN C. MARCO ANTONIO ACEVEDO MOSQUEDA**

MEXICO, D.F.



DICIEMBRE DEL 2000

X M

SI. NO.	57.23
NO.	B1-15375
DATE	26. April. 2001
TR. NO.	1510.2001
\$	

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

DEDICATORIA

A MIS ADORADOS PADRES :

MARIA ELENA MOSQUEDA TREJO
Y
CRESCENCIO ACEVEDO RUEDA

A MIS HERMANOS :

JUAN Y MARCO ANTONIO

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

AGRADECIMIENTOS

Quiero agradecer muy especialmente a

Mi mamá por todo su amor y toda su paciencia al procurarme día con día.

Mi papá porque sé que siempre podré contar con él.

Mi hermano Juan por el gran cariño que siempre me ha demostrado.

Mi hermano Marco Antonio por su apoyo incondicional y por ser mi guía a lo largo de toda mi carrera profesional. Antonio, nunca terminaré de agradecértelo.

Dr. Adriano De Luca por la confianza depositada en mí para la realización de este trabajo.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

RESUMEN

En este trabajo se presenta el diseño de un sistema digital capaz de determinar el nivel de estrés en una persona implementando la escala directa e integral llamada "Estrés-Bienestar"^[1]. Esta escala se compone de siete parámetros cuyos valores parciales se suman para obtener el nivel de estrés en un ser humano. Los siete parámetros son: Cuestionario de la Medida del Estrés Psicológico (MSP), Tensión Muscular (EMG), Conductancia de la Piel (GSR), Respiración, Postura, Cortisol y Prolactina.

Los parámetros EMG, GSR y Respiración se miden directamente del cuerpo del paciente a través de sensores. Las señales que arrojan estos sensores se transforman primero, en lecturas de voltaje analógico y después, este voltaje analógico se convierte en una señal digital para que una computadora personal (PC) sea capaz de capturar estas señales, procesarlas y posteriormente, emitir un resultado parcial de cada uno de estos tres parámetros.

El parámetro de la Respiración consiste en detectar el movimiento que se presenta en el tórax y el abdomen cuando la persona respira. Existen sensores de respiración elaborados con bandas elásticas que con el tiempo tienden a aflojarse. Aquí, se propone un circuito electrónico que utiliza dispositivos infrarrojos para sensar el movimiento respiratorio torácico y abdominal sin necesidad de colocar nada alrededor del cuerpo de la persona.

Se presenta un software programado en Visual C++, desarrollado por la autora de este trabajo, que realiza el cálculo del nivel de estrés. El cálculo se hace obteniendo un resultado parcial normalizado de cada parámetro y, posteriormente, se realiza la suma de estos resultados que es el nivel total de estrés en una persona. Los datos necesarios para emitir un resultado parcial correspondiente a un parámetro se obtienen al recibir datos desde el teclado, en el caso de los parámetros MSP, Postura y los análisis de sangre (Cortisol y Prolactina), y desde el puerto paralelo, las lecturas entregadas por los sensores de EMG, GSR y Respiración. El programa, además, permite guardar, en un archivo, los datos personales de cada paciente así como los resultados parciales de cada parámetro de la escala. Los resultados parciales y el resultado final se pueden observar tanto de manera numérica como en una gráfica de barras y además se pueden imprimir.

Los resultados de este sistema son relativos y van validados forzosamente por un médico. Este sistema es el primero que implementa la escala "Estrés-Bienestar", en realidad no existe un punto de referencia que nos indique si los resultados obtenidos son verdaderos o no. De hecho, los rangos de algunos parámetros no son iguales a los que se indican en la escala aunque tampoco presentan grandes variaciones. El sistema debe utilizarse en un gran número de personas para verificar que los valores obtenidos son correctos. La escala "Estrés-Bienestar" fue diseñada por médicos^[1], por tanto, deben ser éstos los que evalúen el sistema y sus resultados.

ABSTRACT

This work presents the design of a digital system to be able to determinate the level of stress in a person by implementing the direct and integral "Stress-Comfort" scale. This scale have seven parameters whose partial values are added to obtain the level of stress in a human being. The seven parameters are: Test of the Measure of Psychological Stress (MSP), Muscle Tension (EMG), Skin Conductivity (GSR), Respiration, Posture, Cortisol and Prolactin.

The EMG, GSR and Respiration parameters are measured directly from the patient body through sensors. The signals of these sensors are converted first, in analogical voltage and then, this voltage is converted into digital signals by means of a personal computer (PC), that can capture and process them for giving a partial result for each of these three parameters.

The Respiration parameter consists on detecting the thorax and abdomen movements of a person's breath. There are respiration sensors elaborated with elastic bands that become slack with the use. This work proposes an electronic circuit that use infrared devices to sense the thorax and abdominal respiratory movement without putting anything around the human body.

Furthermore, it presents a software programmed in Visual C++, developed by the author, that calculates the level of stress. This calculation is obtained by the addition of normalized partial results of each parameter. The data, necessary to emit a partial result corresponding to each parameter, are obtained from the keyboard, the case of MSP, Posture and blood analysis (Cortisol and Prolactin), and from the parallel port, signals from EMG, GSR and Respiration sensors. Besides, the program saves, in a file, the personal data and the partial results of the parameters for each patient. These results are shown both numerically and graphically and they could be printed.

The results of this system must be validated by a doctor. This system is the first that implement the "Stress Comfort" scale, as a matter of fact it doesn't exist a reference that indicates if the results are true or not. The ranks of some parameters changed with respect to the ranks of the scale, although there's not a big difference between them. The system must be used with a big number of persons to verify that the results are correct. The "Stress Comfort" scale was designed by doctors, so they have to evaluate the system and the results.

INDICE DE FIGURAS Y TABLAS

Figuras del Capítulo 2

2.1	La unión P-N se describe como un cambio abrupto en el material que contiene una mayoría de portadores cargados positivamente o huecos a un material con exceso de portadores cargados negativamente o electrones	12
2.2	Esquema del circuito básico para un IRED	14
2.3	Gráfica de Voltaje directo V_F contra Corriente directa I_F	14
2.4	Relejo directo y difuso. El ángulo del componente de reflejo directo es igual al ángulo de incidencia, mientras que el componente difuso puede ser dirigido virtualmente hacia cualquier parte por arriba de la superficie	15
2.5	Voltaje colector-emisor contra Corriente de Colector. Esta familia de curvas es análoga a la familia de curvas de un transistor tradicional (I_C vs V_{CE}) excepto porque la radiación incidente sustituye a la corriente de base. Cada paso en el nivel del fotón es análogo a un incremento en la corriente de base, resultando en una familia de curvas	17
2.6	Esquema del circuito básico para un fototransistor	17
2.7	Gráfica de Voltaje de Colector-Emisor contra Corriente de Colector	18
2.8	El Amplificador Operacional	19
2.9	Amplificador no inversor lineal	20
2.10	El Amplificador Operacional TL082	21
2.11	El Decodificador de Tono LM567/LM567C	23
2.12	Gráfica de ancho de banda (BW) contra $f_O C_2$ y $f_O C_3$	24
2.13	Empleo del temporizador 555 como multivibrador astable	25
2.14	Diagrama general de una clase de convertidores A/D	27
2.15	ADC de aproximaciones sucesivas	28
2.16	Ilustración del funcionamiento de un CAS de cuatro bits utilizando un DAC con tamaño de paso igual a $1 V$, y $V_A = 10.4 V$	28
2.17	El ADC0804 es un ADC de aproximaciones sucesivas de ocho bits con salidas de tres estados. Los números entre paréntesis corresponden al número de las terminales del circuito integrado	30
2.18	Diagrama funcional de un multiplexor digital (MUX)	31
2.19	Multiplexor de dos entradas	32
2.20	Multiplexor de cuatro entradas	33
2.21	El Multiplexor 74LS157	33
2.22	Ubicación del puerto paralelo	34
2.23	Asignaciones de pines	37
2.24	Asignaciones de puertos	38

Figuras del Capítulo 3

3.1	Diagrama a bloques del sistema a diseñar	39
3.2	Circuito que convierte una señal analógica a digital para ser leída por el puerto paralelo	40
3.3	Conexiones entre la interfaz y el puerto paralelo	41
3.4	Programa en lenguaje C++ para enviar señales a la interfaz, leer los datos del puerto e interpretarlos	42
3.5	Pulso enviado al ADC para el inicio de la conversión	43
3.6	Diagrama a bloques del sensor del movimiento respiratorio utilizando un par infrarrojo	44
3.7	Movimiento respiratorio del abdomen observado desde un punto fijo frontal	45
3.8	Periodo para una onda cuadrada	45
3.9	Diseño del temporizador LM555 para una frecuencia de 1000 Hz y un ciclo de trabajo del 55%	46
3.10	Gráficas del TSUS520. tp vs I_F y V_F vs I_F	47
3.11	Gráfica de Corriente de Colector contra Voltaje Colector-Emisor	48
3.12	Configuración del sensor de movimiento respiratorio	49
3.13	El MyoTrac y sus entradas y salidas	50
3.14	Accesorios del MyoTrac	50
3.15	Configuración del AO TL082 para una ganancia de 2.5	51
3.16	Descripción del GSR2	52
3.17	Gráfica de la frecuencia del tono, audible y agudo, emitido por el GSR2	52
3.18	Gráfica de ancho de banda (BW) contra $f_0 C_2$ y $f_0 C_3$	53
3.19	Diagrama del circuito para la medición del parámetro GSR	54
3.20	Diagrama de la interfaz entre la PC y los cuatros sensores	55
3.21	Diagrama general de la interfaz entre los cuatro sensores y la PC utilizando el puerto paralelo en modo estándar	56

Figuras del Capítulo 4

4.1	Pantalla Inicial	58
4.2	Pantalla del Menú Principal	59
4.3	Vista actual cuando se elige el botón de Paciente Nuevo	59
4.4	Selección de datos de las listas desplegables	60
4.5	Forma de introducir el dato de la fecha	60
4.6	Pantalla de Búsqueda de Pacientes	61
4.7	Opciones para la Búsqueda de Pacientes	61
4.8	Búsqueda de Pacientes por Nombre	62
4.9	Búsqueda de pacientes por la Fecha en que se realizó el examen	62
4.10	Vista de la pantalla del Cuestionario MSP	63
4.11	Recuadro que indica el resultado final del Cuestionario MSP	64
4.12	Pantalla de la prueba Comportamiento Corporal	65
4.13	Vista de la pantalla de los exámenes de sangre para ver el nivel de Cortisol y Prolactina	65
4.14	Vista de la pantalla de la prueba de Respiración	66
4.15	Pantalla de la prueba de la Tensión Muscular (EMG)	67

4.16	Vista de la pantalla de prueba de la Conductancia de la Piel (GSR)	67
4.17	Vista de la pantalla que aparece al oprimir Repetir Evaluación	68
4.18	Vista de la pantalla de Resultados de Pruebas	68
4.19	Pantalla para la impresión de los Resultados de las Pruebas	69
4.20	Pantalla para poder guardar los registros de la base de datos actual en otro espacio	70
4.21	Vista de la pantalla que permite abrir un archivo con extensión <i>mie</i> .	70
4.22	Vista de la pantalla de Ayuda	71

Figuras del Capítulo 5

5.1	Circuito utilizado para medir el error del ADC0804	72
5.2	Señal del oscilador de onda cuadrada para ambos sensores	73
5.3	Carátula del sensor de tensión muscular (EMG) MyoTrac	74

Tablas del Capítulo 1

1.1	Valores mínimos y máximos de los parámetros que son afectados por el estrés	6
-----	---	---

Tablas del Capítulo 2

2.1	Algunos de los materiales utilizados para dispositivos LED	12
2.2	Eficiencia relativa de algunos LEDs	13
2.3	Diferentes rangos de entrada analógica y resoluciones para $V_{ref}/2$	31

Tablas del Capítulo 5

5.1	Voltajes medidos con multímetro y las lecturas del ADC interpretadas por la PC	72
5.2	Resultados de las pruebas realizadas a los sensores de Respiración	74
5.3.1	Resultados para un umbral (Threshold, THR.) = 0.5	74
5.3.2	Resultados para un umbral (THR.) = 0.6	75
5.3.3	Resultados para un umbral (THR.) = 0.7	75
5.3.4	Resultados para un umbral (THR.) = 1	75
5.3.5	Resultados para un umbral (THR.) = 2	75
5.3.6	Resultados para un umbral (THR.) = 3	75
5.3.7	Resultados para un umbral (THR.) = 5	76
5.3.8	Resultados para un umbral (THR.) = 10	76

ÍNDICE

RESUMEN	I
ABSTRACT	II
ÍNDICE DE TABLAS Y FIGURAS	III
PLANEACIÓN DEL SISTEMA	XI
PROBLEMÁTICA	XI
JUSTIFICACIÓN	XI
MODELO PROPUESTO	XII
OBJETIVOS	XIII
CAPÍTULO 1. SITUACIÓN ACTUAL	I
1.1 INTRODUCCIÓN	1
1.2 EL ESTRÉS Y SUS ETAPAS	1
1.3 ESTADO DEL CAMPO DEL ARTE	2
1.4 ESCALA ESTRÉS – BIENESTAR	4
1.5 ORGANIZACIÓN DE LA TESIS	9
CAPÍTULO 2. CIRCUITOS ANALÓGICOS Y DIGITALES	11
2.1 INTRODUCCIÓN	11
2.2 CIRCUITOS ANALÓGICOS	11
2.2.1 EMISORES Y RECEPTORES INFRARROJOS	11
2.2.1.1 Diodos emisores de luz infrarroja	11
2.2.1.2 El IRED TSUS520	13
2.2.1.3 El interruptor óptico reflejante	14
2.2.1.4 Fotodetección	15
2.2.1.5 Tipos de fotodetectores	15
2.2.1.6 El fototransistor	15
2.2.1.7 El fototransistor de silicio NPN BPW96	17
2.2.2 AMPLIFICADORES OPERACIONALES	19
2.2.2.1 El amplificador no inversor lineal	20
2.2.2.2 Ganancia del amplificador no inversor	20
2.2.2.3 El Amplificador Operacional JFET TL082	21
2.2.3 DECODIFICADOR DE TONO	22
2.2.3.1 El decodificador de tono LM567/LM567C	22
2.2.3.2 El decodificador de tono LM567/LM567C básico	22
2.3 CIRCUITOS DIGITALES	24
2.3.1 CIRCUITOS DE RELOJ	24

2.3.2 CONVERTIDORES ANALÓGICO – DIGITALES	27
2.3.2.1 Conversión Analógica – Digital	27
2.3.2.2 ADC de Aproximaciones Sucesivas	27
2.3.2.3 Convertidor Analógico – Digital de aproximaciones sucesivas ADC0804	29
2.3.3 MULTIPLEXORES (SELECTORES DE DATOS)	31
2.3.3.1 Multiplexor básico de dos entradas	31
2.3.3.2 Multiplexor de cuatro entradas	32
2.3.3.3 Multiplexor Cuádruple de dos entradas 74LS157	32
2.3.4 EL PUERTO PARALELO DE LA PC	34
2.3.4.1 El estándar IEEE 1284	35
2.3.4.2 Uso del Puerto de Impresión de la PC para el control y la adquisición de datos	36
CAPÍTULO 3. DISEÑO DEL SISTEMA	39
3.1 INTRODUCCIÓN	39
3.2 INTERFAZ ENTRE LA PC Y UN CIRCUITO CON SALIDA DE VOLTAJE ANALÓGICO	40
3.3 SENSOR DE MOVIMIENTO DE RESPIRACIÓN ABDOMINAL Y TORÁXICA	44
3.3.1 DISEÑO DEL OSCILADOR DE ONDA CUADRADA	46
3.3.2 DISEÑO DEL EMISOR DE LUZ INFRARROJA	47
3.3.3 DISEÑO DEL RECEPTOR INFRARROJO	48
3.4 SENSOR DE TENSIÓN MUSCULAR	49
3.5 SENSOR DE CONDUCTANCIA DE LA PIEL	51
3.6 INTERFAZ ENTRE LOS CUATRO SENSORES Y LA PC	54
3.7 DIAGRAMA GENERAL DEL SISTEMA	55
CAPÍTULO 4. DESCRIPCIÓN DEL SOFTWARE Y PRUEBA DE CIRCUITOS	57
4.1 PROGRAMA DE CONTROL	57
CAPÍTULO 5. PRUEBA DE CIRCUITOS	72
5.1 INTERFAZ ENTRE UN SENSOR Y LA PC	72
5.2 SENSOR DE RESPIRACIÓN	73
5.3 SENSOR DE TENSIÓN MUSCULAR (EMG)	74
5.4 SENSOR DE CONDUCTANCIA DE LA PIEL (GSR)	76
CAPÍTULO 6. CONCLUSIONES	79

6.1 APORTACIONES	79
6.2 CONCLUSIONES	79
6.3 RECOMENDACIONES Y TRABAJOS FUTUROS	81
APÉNDICE: CÓDIGO DEL PROGRAMA EN VISUAL C++ 5.0.....	83
GLOSARIO	119
BIBLIOGRAFÍA	121

PLANEACIÓN DEL SISTEMA

PROBLEMÁTICA

El estrés es una respuesta de nuestro cuerpo ante una situación que amenaza nuestro bienestar físico y mental. La forma de reaccionar es diferente en cada individuo.

Un estado de estrés se presenta a cualquier edad, incluso en bebés que se encuentran en el vientre de la madre. No depende de un estatus económico ni del nivel social, los problemas personales siempre estarán presentes y, por consiguiente, las situaciones estresantes también. Entre hombre y la mujer existen muchas diferencias, algunas insignificantes, otras abismales y todo debido a la naturaleza de cada uno; sin embargo, ninguno se salva de padecer, al menos una vez en la vida, un momento difícil que derive en nerviosismo, tensión o miedo.

JUSTIFICACIÓN

El estrés se ha convertido en la moda de los estados emocional y físico de las personas. Hay personas que con el tiempo llegan a controlarlo para que no les afecte demasiado. Sin embargo, no todos tenemos esa fuerza para manejar nuestros problemas cotidianos. Con frecuencia, cuando ya nos sentimos muy cansados, deprimidos y nuestro cuerpo ya no puede resistir tanta presión, recurrimos a tomar medicamentos relajantes que nos proporcionan un estado momentáneo de tranquilidad. Pero si no somos capaces de enfrentar nuestras dificultades podemos caer en la dependencia de los relajantes o llegar a desarrollar una enfermedad que sólo nos provoque más estrés.

Sería recomendable que una persona que presente los primeros síntomas del estrés acuda con un especialista que le proporcione el tratamiento adecuado a su caso en particular.

Se ha puntualizado que la escala Estrés-Bienestar es una escala directa e integral porque valora el estado físico actual de la persona, así como toma en cuenta los dos tipos de estrés que componen al fenómeno: estrés psicológico y fisiológico. Por esta razón, esta escala es una forma muy completa de poder detectar los parámetros del cuerpo que el estrés ha afectado, para de esta forma poder darle a la persona un tratamiento que sea acorde a los malestares que presenta. Además, el tratamiento físico se ve complementado con un tratamiento psicológico que cura la parte emocional del individuo.

Así, podemos reafirmar lo que se dice: "Mente sana, en cuerpo sano"

MODELO PROPUESTO

El estrés no sólo afecta al estado psicológico de una persona sino también a su organismo, es decir, el estrés está compuesto por dos elementos inseparables, el estrés psicológico y el estrés fisiológico.

El estrés afecta al cuerpo humano en todos sus sistemas vitales: **sistema neurovegetativo, sistema cardiocirculatorio y sistema endocrino.**

La afección del sistema neurovegetativo provocado por el estrés se ve reflejada en la respiración^[3], la tensión de los músculos, actividad del cerebro, la temperatura del cuerpo, la conductancia de la piel.

Del sistema cardiocirculatorio se afecta el ritmo cardiaco y la presión arterial.

En el caso del sistema endocrino, aumenta la cantidad de hormonas que son segregadas por las glándulas endocrinas^[1,3] como la prolactina, el cortisol, la progesterona, el nivel de estrógenos, la adrenalina entre otras.

Después de realizar estudios exhaustivos, comparar el cambio sufrido por los componentes de los sistemas neurovegetativo, cardiocirculatorio y endocrino, observar cual de ellos se afectaba más al exponer a una persona a momentos estresantes, se concluyó que los parámetros fisiológicos más significativos en la medición del estrés son^[4]: Respiración, Tensión Muscular, Conductancia de la Piel, Postura, **Cortisol y Prolactina.**

De igual manera, para el caso de la medición del estrés psicológico, se determinó que el cuestionario de la Medida del Estrés Psicológico (MSP) era la mejor opción dentro de varios cuestionarios^[2,4]

Son, por tanto, siete parámetros (seis fisiológicos y uno psicológico) los que conforman la escala “Estrés-Bienestar” El valor de cada uno de estos parámetros se utiliza para obtener un nivel de estrés existente en un ser humano.

El sistema digital que se propone y que mide el nivel de estrés en una persona, debe ser capaz de obtener los valores de cada uno de los parámetros, procesarlos y arrojar un valor correspondiente al nivel de estrés.

En la siguiente figura se presenta el diagrama a bloques del sistema propuesto.

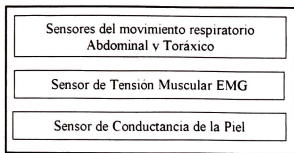
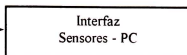
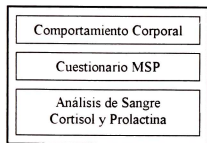
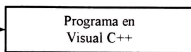
En el bloque No. 1 se encuentran los sensores de la respiración, la tensión muscular y la conductancia de la piel. El sensor del movimiento respiratorio del abdomen y del tórax no existe, por lo que es necesario construirlo. Los otros dos sensores están disponibles para su venta. Se debe observar la salida de cada uno de los sensores. Si la salida no tiene, al menos, una salida de voltaje analógico, entonces es necesario diseñar un circuito que convierta la salida del sensor a voltaje analógico.

En el bloque No.2 está la interfaz entre los sensores y la PC, la tarea de esta interfaz consiste en, primero, convertir las señales analógicas de los sensores (si es necesario) en señales digitales para que la PC las pueda leer, en segundo lugar, multiplexar estas señales digitales de manera que la PC tome las lecturas de los sensores concurrentemente (al mismo tiempo).

En el bloque No. 3 se encuentran los parámetros restantes: el Cuestionario MSP, el comportamiento corporal o postura y los análisis de sangre. El sistema debe permitir al usuario introducir los datos obtenidos de las mediciones de estos parámetros para, posteriormente, unirlos con los resultados de las otras mediciones (sensores).

En el bloque No. 4 está el programa, que se desarrollará en Visual C++ y que tiene varios problemas que resolver.

- 1) En el caso de la respiración, es necesario que el software sea capaz de graficar la respuesta de la respiración y de obtener los parámetros parciales de la respiración: Congruencia, Pausa de la respiración torácica, Pausa de la respiración abdominal, Relación entre la subida y la bajada en la respiración, Relación entre el tórax y el diafragma y la Frecuencia de la respiración.

Bloque No. 1. SENSORES**Bloque No. 2. INTERFAZ****Bloque No. 3. OTROS PARÁMETROS****Bloque No. 4. SOFTWARE****NIVEL DE ESTRÉS**

- 2) También se graficará la respuesta de la tensión muscular y la conductancia de la piel durante toda la sesión (15 minutos).
- 3) Una vez terminada la sesión de los 15 minutos, el programa debe recibir los valores de los parámetros restantes (MSP, postura y análisis de sangre). Se puede obtener el nivel de estrés sin tomar en cuenta los análisis de sangre, por lo que la escala se reduciría a 50. Si se van a tomar en cuenta los análisis de sangre, entonces el programa deberá guardar, en una sencilla base de datos, los resultados parciales obtenidos, para que una vez que se tengan los análisis se adicionen a los resultados anteriores y, de esta forma, tener la escala completa (70).
- 4) Obtener una gráfica de barras de todos los parámetros, indicando los valores máximo y mínimo, así como el valor absoluto.

Los resultados que se obtendrán serán el nivel de estrés y una gráfica de barras de los valores

OBJETIVOS

Objetivo General: Diseñar un sistema digital que mida el nivel de estrés en un ser humano.

Objetivos Específicos:

- Diseñar y construir un sensor de movimiento respiratorio abdominal y torácico.
- Diseñar y construir un circuito que convierta a voltaje analógico la salida de cada sensor (de movimiento respiratorio abdominal y torácico, de conductancia de la piel y de tensión muscular).
- Diseñar y construir una interfaz entre los cuatro sensores y la PC.
- Diseñar un software que controle y analice el sistema completo.

SITUACIÓN ACTUAL

1.1 INTRODUCCIÓN

La vida agitada de las grandes ciudades, las presiones laborales y familiares o simplemente el esfuerzo cotidiano que implican las actividades diarias de la vida, son factores que afectan el desarrollo y desenvolvimiento del ser humano.

Cualquier estímulo del medio ambiente que percibe una persona produce una respuesta en su cuerpo. La respuesta inducida por ese estímulo se llama “estrés”^[5]. Esta respuesta depende mucho de la propia percepción de cada individuo, por ejemplo, el sonido ensordecedor en una discoteca, puede parecer una fuente de energía, mientras que para otro individuo, la música escuchada a todo volumen, represente un estímulo totalmente desagradable para sus oídos y prefiera ritmos más tenues y melodiosos. Por otro lado, la forma en cómo se maneja o enfrenta el estímulo también variará de persona a persona, es decir, es posible que a un individuo le disguste esa clase de música, sin embargo, puede soportar toda una velada escuchándola, pero para otro sea insoportable permanecer un segundo más en ese lugar y decida salir huyendo.

En la mayoría de las personas, cuando enfrentamos estos “problemas” y no podemos llevarlos a buen fin, nuestro cuerpo se ve afectado por las constantes alteraciones que se producen como respuesta a todos esos estímulos ambientales o factores estresantes. Los primeros síntomas se presentan con depresiones, decaimientos físicos, dolores musculares, pérdida del apetito, pérdida del interés y la concentración, así como la disminución del **libido**^[5]. Toda esta anarquía corporal va mermando el sistema inmunitario de la persona, que por consecuencia, se encuentra en un estado latente de contraer cualquier tipo de enfermedades. Antes de que eso suceda, es necesario conocer el nivel de estrés que presenta esa persona y los daños causados en su cuerpo y en su mente para poder suministrarle un tratamiento de tipo psicológico para aliviar sus problemas emocionales y un tratamiento de tipo fisiológico de manera que sus males físicos desaparezcan. Para lograr esto es necesario contar con una escala que nos permita valorar ese nivel de estrés que la persona padece.

1.2 EL ESTRÉS Y SUS ETAPAS

Como ya se había mencionado, el estrés es una respuesta del cuerpo a un estímulo del medio ambiente. Nuestro cuerpo responde de manera física (fisiológica) y de manera emocional (psicológica). Esto lo concluyó el psicoanalista alemán Wilhelm Reich^[4] en 1930, después de realizar varias investigaciones.

El estado psicológico es un estado subjetivo^[4] en el cual el individuo tiene una percepción propia de sus emociones, sus sensaciones físicas y del mundo que le rodea. La manera de conocer el estado psicológico de una persona es a través de cuestionarios cuyas preguntas revelan los eventos estresantes a los que el paciente se ha enfrentado (o se sigue enfrentando), la manera en que los ha manejado y la percepción propia que tiene de estos eventos.

El estado objetivo o fisiológico^[4] está constituido por la condición de movilidad de todas las funciones de todas las áreas de los sistemas fisiológicos, de los movimientos y de las posturas. Estas condiciones de movilidad muchas veces no son percibidas por el propio individuo pero son evidentes para un observador externo. De este bienestar objetivo también toman parte el color de la piel, la brillantez de la mirada, la habilidad de completar movimientos fuertes o suaves acordes con las circunstancias, pasos armoniosos, un balance de operación del sistema neurovegetativo, cardiocirculatorio y endocrino, los tonos de voz. Lo que se ve afectado del sistema neurovegetativo es: el ritmo cardíaco, la respiración, la tensión de los músculos, actividad del cerebro, la temperatura del cuerpo, la conductancia de la piel^[1,3]. En el caso del sistema endocrino, son las hormonas como la adrenalina, el cortisol y la prolactina las que sufren alteraciones^[1].

La regulación de las respuestas fisiológicas a demandas estresantes es manejada principalmente por la glándula adrenal^[5]. La respuesta inicial prepara al cuerpo para lo que se ha llamado la reacción *fight-or-flight*, que comprende:

- ✓ El incremento de la actividad del sistema nervioso.
- ✓ El incremento de la producción de ciertas hormonas, como la adrenalina y el cortisol. La respuesta del cuerpo a estos químicos incluye:
 - El incremento del ritmo cardíaco y la presión de la sangre.
 - La compresión de los vasos sanguíneos, esto provoca que la sangre deje la piel para dar a los músculos más glucosa y más oxígeno (por eso nos ponemos blancos cuando sufrimos un *shock*).
 - La movilización de la glucosa almacenada para proveer de más energías al cuerpo.

En términos generales, la reacción psicológica para enfrentar el estrés toma el siguiente camino:

- ✓ La reacción inicial del *fight-or-flight* se acompaña de emociones como la ansiedad o el miedo.
- ✓ Comenzamos a pensar en la forma de enfrentar la situación estresante, formulamos estrategias.
- ✓ Si la estrategia tiene éxito, la reacción de *fight-or-flight* y el estado de ansiedad terminan.
- ✓ Si la estrategia falla y la situación de estrés continua, todo el proceso se vuelve a repetir.

La continua repetición de la respuesta, tanto fisiológica como psicológica, a una situación estresante puede llevar a la persona a un estado de total depresión. Esta depresión, que afecta indirectamente al sistema inmunitario del cuerpo puede llevarnos a un estado de *shock* irreversible y exponernos a enfermedades como la diabetes o el cáncer mismo.

1.3 ESTADO DEL CAMPO DEL ARTE

A través de los años el hombre se ha dado a la tarea de cuantificar los distintos fenómenos que rodean su existencia. Inventó el termómetro para medir la temperatura, inventó la balanza para poder contar con una lectura de medida de peso. Después inventó aparatos que midieran algunos parámetros de nuestro cuerpo para saber que tan enfermos estábamos. Por ejemplo, se puede medir la presión arterial, el nivel de glucosa en la sangre, el ritmo cardíaco, entre otros.

El tener una cuantificación de las cosas nos da un panorama más amplio para ofrecer una opinión más exacta o un dictamen más confiable de una situación o de una enfermedad.

El estrés también se ha tratado de cuantificar para tener una idea más clara de qué tanto una persona se ha visto afectada por un evento estresante que hasta ese momento no ha podido superar o manejar de una manera satisfactoria.

Al principio, se comenzaron a elaborar cuestionarios muy sencillos para determinar la forma en que el estrés había afectado al individuo y así poder brindarle la ayuda necesaria para minimizar sus males. Posteriormente y con estudios más profundos, los cuestionarios se volvieron más formales, más veraces y confiables.

Las escalas existentes utilizadas para la medición del estrés se dividen en cuatro diferentes grupos^[4,2]:

- 1) Las escalas conectadas a aquellas investigaciones que evalúan de forma preponderante los eventos estresantes. Ejemplos de este tipo de escalas son: la escala de reajuste social de Holmes y Rahe, la escala de Paykel. Para el diseño de dichas escalas se toma una muestra de “x” número de personas, se obtienen los eventos estresantes más comunes y se escriben en un cuestionario y a cada evento se le da un peso determinado.
- 2) Las escalas que centran su atención en las patologías derivadas del estrés (enfermedades psíquicas y somáticas, alteraciones del comportamiento). Ejemplos de estas escalas son: Be-Trait anxiety inventory de Spielberger, Maslach Burnout Inventory (MBI). Esta escala está basada en los estímulos que se presentan en el trabajo, todos los eventos estresantes que se suscitan por las presiones derivadas del trabajo.
- 3) La escala que, dando una gran atención a los eventos o a las patologías, toma también en consideración la manera en la cual el individuo reacciona a ciertos eventos y patologías. Ejemplos de estas escalas son: la prueba de modos de Coping, la escala de Hardiness, el cuestionario del Barrier, la escala Helath Orientation (HOS). Estas escalas se basan en el hecho de que la reacción de cada individuo ante un evento estresante es diferente en todos los casos, por lo tanto, primero se define si el evento estresante tiene consecuencias benéficas, neutras o dañinas. Posteriormente, se enfrenta al individuo al evento estresante y se le pregunta la manera cómo solucionaría el problema. De esta forma, se tienen dos mediciones.
- 4) En esta cuarta escala se limita al círculo psicológico y al área cognoscitiva-emocional. También se le llama medida del estado de estrés. Ejemplos de esta escala son: la medida del estrés psicológico (MSP) y el Perfil de los Estados del Humor (POMS). Estas escalas se basan en la respuesta emocional y de comportamiento del individuo al enfrentarse a una situación estresante. Estas escalas miden directamente el estado de estrés de una persona a partir de sus emociones afectivas, legales, en resumen, del humor que presentan.

Estas escalas sólo tratan de obtener el nivel de estrés psicológico. Sin embargo, como ya se ha visto, el estrés afecta física y emocionalmente a una persona. Además, estas escalas se consideran indirectas porque analizan un estado anterior para determinar el estado presente de una persona o analizan el estado presente para determinar un estado futuro.

BIOFEEDBACK

A finales de los años 60 se realizaron investigaciones científicas, en muchos laboratorios de psicofisiología en Estados Unidos, que condujeron a una terapia multifacética, llamada Biofeedback^[6]. Esta terapia permite a los pacientes modificar y controlar voluntariamente las funciones de su cuerpo. Esto se logra detectando algunos parámetros del cuerpo como son: la temperatura, el ritmo cardiaco, la

respuesta eléctrica de la piel, las ondas del cerebro, la tensión muscular, la presión de la sangre y la respiración. Estos parámetros del cuerpo se convierten en señales acústicas y/o visuales y se le muestran al paciente. Se le pide que observe detenidamente estas señales, que de tener un alto grado de estrés, son señales que muestran una anomalía en el cuerpo del paciente. Posteriormente, se le enseña al paciente a controlar sus respuestas fisiológicas. Durante la terapia, el paciente puede observar como van cambiando sus respuestas corporales. Este conocimiento de su cuerpo retroalimenta al paciente, que observa el cambio de las señales conforme cambian sus actitudes, es decir, conforme entra en un estado de relajación.

El *biofeedback* es una terapia, más no es una escala que nos proporcione un nivel de estrés. Además, no toma en cuenta el estado psicológico de la persona, es decir, se le enseña al paciente a controlar sus emociones, pero no se analiza el estado psicológico anterior, no se toman en cuenta las causas que derivaron ese estado.

Por lo tanto, no existe una escala que mida el estado actual del paciente y que además evalúe el estrés fisiológico y psicológico y los conjunte para obtener una medida total del estrés.

Una escala que ubica a una persona en un estado de estrés o bienestar permitiendo determinar el nivel presente en una persona y al mismo tiempo poder cuantificar los daños o el bienestar producidos en los sistemas más afectados por el estrés, se llama “Estrés-Bienestar”

1.4 ESCALA ESTRÉS – BIENESTAR

Esta escala es una escala directa e integral^[4]. Se dice que es directa porque es capaz de valorar el estado físico actual de una persona y se dice que es integral porque el nivel de estrés se obtiene de la unión de los dos tipos de estrés: el psicológico y el fisiológico.

Con esta escala es posible ubicar a una persona en un estado de bienestar o de estrés, es decir, se puede definir una persona como en total bienestar, un poco estresada o en completo estado de estrés.

El estrés psicológico se obtiene utilizando la escala que mide el estrés psicológico, ésta es la escala MSP^[4,2]

El cuestionario contiene 49 preguntas que se muestran a continuación.

- 1) Me siento tensa
- 2) Siento la garganta cerrada o la boca seca
- 3) Me siento presionado por el tiempo, me falta tiempo
- 4) Tiendo a saltarme alimentos u olvido comer
- 5) Examino la misma idea muchas veces, pienso continuamente en ello, tengo pensamientos que se repiten, siento la cabeza llena de pensamientos
- 6) Me siento solo, aislado, incomprendido
- 7) Me siento abrumado, amenazado, sobrecargado
- 8) Me preocupa lo que puede suceder el siguiente día
- 9) Tengo la cara (frente, cejas, labios) tensa, fruncida
- 10) Estoy continuamente atento al horario, veo la hora o pregunto por la hora
- 11) Me siento irritable, siento los nervios a flor de piel, pierdo la paciencia con las personas o las cosas
- 12) Siento dificultad en la digestión, dolor de estómago, siento un nudo en el estómago
- 13) Estoy desalentado, deprimido, abatido
- 14) Siento dolor físico: dolor de espalda, dolor de cabeza, dolor de cuello dolor de estómago
- 15) Me siento preocupado, atormentado, molesto
- 16) Tengo repentinas variaciones de temperatura (mucho calor o mucho frío)
- 17) Me como las uñas, la piel alrededor de las uñas, me muerdo los labios y la parte interna de las mejillas.

- 18) Olvido citas, objetos o actividades a realizar
- 19) Lloro fácilmente
- 20) Me siento cansado
- 21) Tengo la mandíbula cerrada
- 22) Estoy calmado
- 23) Tengo las manos sudorosas o sudo mucho (las axilas, los pies, etc)
- 24) Veo la vida simple y fácil
- 25) Siento que el corazón palpita rápida o irregularmente
- 26) Camino rápidamente
- 27) Tomo un largo suspiro o reanudo de golpe la respiración
- 28) Tengo diarrea, calambres intestinales o estreñimiento
- 29) Estoy ansioso, inquieto o angustiado
- 30) Me sobresalto por situaciones inesperadas o ruidos repentinos
- 31) Empleo más tiempo para conciliar el sueño
- 32) Soy brusco en mi comportamiento, me muevo rápida y estúpidamente
- 33) Soy ineficiente e inadecuado
- 34) Tengo los músculos tensos y me tiemblan o siento rigidez muscular
- 35) Tengo la impresión de que pierdo el control
- 36) Soy agresivo
- 37) Estoy confundido, no tengo ideas claras, me falta atención o concentración
- 38) Tengo las facciones tensas
- 39) Evito los contactos sociales, no tengo actividades culturales, no tengo pasatiempos, no salgo, me encierro
- 40) Tengo la respiración corta, escasa, limitada, rápida
- 41) Siento un gran peso sobre los hombros
- 42) Tengo la impresión de que realizo un gran esfuerzo en cada cosa
- 43) Me siento lleno de energía, en forma
- 44) Me siento cansado, tengo flojera de moverme
- 45) Como rápidamente, termino de comer en menos de 15 minutos
- 46) No controlo mis reacciones, ni mi humor, ni mi comportamiento
- 47) Me siento estresado
- 48) Me siento desmañado, tengo desaciertos, pierdo las cosas, tengo incidentes de varios tipos
- 49) Estoy relajado

En el caso del estrés fisiológico, se realizaron investigaciones^[4] sobre los sistemas que sufren alteraciones provocadas por el estrés y concluyó que los parámetros más significativos que reflejan las afecciones del estrés, son:

- La respiración
- La tensión muscular (*Electromiograma*, EMG)
- La conductancia de la piel (*Galvanic Skin Response*, GSR)
- La postura
- El cortisol
- La prolactina (hombre o mujer)

Para obtener un número concreto para representar el nivel de estrés, es necesario tener una referencia numérica de cada uno de los parámetros anteriormente mencionados. O sea que se debe conocer un valor mínimo y máximo de cada uno de ellos.

La tabla 1.1 muestra los valores mínimos y máximos para cada uno de los parámetros.

Tabla 1.1. Valores mínimos y máximos de los parámetros que son afectados por el estrés

<i>Parámetro</i>	<i>Valor mínimo</i>	<i>Valor máximo</i>
MSP	49	196
EMG	0.4	4
GSR	0	40
Respiración	6	24
Postura	3	9
Cortisol	4.1	22.4
Prolactina (Mujer)	1.3	19.9
Prolactina (Hombre)	1.9	26.1

La forma en que se va a medir cada uno de los parámetros es la siguiente.

Los parámetros como el EMG, el GSR y la respiración se medirán durante 15 minutos, descartando los primeros 5 minutos. Esto se debe a que en los primeros 5 minutos el paciente está nervioso por el desconocimiento de los aparatos que se le conectan y la rutina que se sigue para las mediciones.

En el caso del EMG se toma la lectura mínima obtenida en el lapso de los últimos diez minutos.

Para el GSR se toma el valor medio de todos los valores obtenidos durante diez minutos.

La medición de la respiración es más compleja. Aquí se toman en cuenta seis elementos como:

- **Congruencia (C)**

- ✓ *Congruencia Regular (puntuación 1)*. Cuando el abdomen y el torax se elevan y bajan con un mismo ritmo.
- ✓ *Congruencia Irregular (puntuación 2)*. El tórax no sigue el mismo ritmo que el abdomen (el tórax se eleva primero que el abdomen, o bien baja antes que el abdomen).
- ✓ *Incongruencia (puntuación 3)*. Si el tórax se eleva y baja de manera defasada respecto al abdomen (el tórax baja cuando el abdomen se eleva).
- ✓ *Incongruencia forzada (puntuación 4)*. El tórax y el abdomen no siguen el mismo ritmo y además, el movimiento es forzado y en un momento, el ritmo y el tiempo pueden cambiar (a veces el tórax baja lentamente, otras velozmente; una vez primero que el abdomen, otras después del abdomen).

- **Pausa de la respiración torácica (P₁)**

- ✓ *Mayor después de la expiración (puntuación 1)*. La pausa después de la expiración tiene una duración de tiempo mayor que la pausa después de la aspiración.
- ✓ *Pausa igual (puntuación 2)*. Las pausas después de la expiración y la aspiración son iguales.
- ✓ *Incoherente (puntuación 3)*. La pausa no tiene una marcha regular, no es siempre la misma.

- ✓ *Mayor después de la aspiración (puntuación 4).* La pausa de la aspiración tiene una duración de tiempo mayor que la de la expiración.
- **Pausa de la respiración abdominal (P₂)**
 - ✓ *Mayor después de la expiración (puntuación 1).* La pausa de la expiración tiene una duración de tiempo mayor que la de la aspiración.
 - ✓ *Pausa igual (puntuación 2).* Las pausas después de la expiración y la aspiración son iguales.
 - ✓ *Incoherente (puntuación 3).* La marcha no tiene una marcha regular, no es siempre la misma.
 - ✓ *Mayor después de la aspiración (puntuación 4).* La pausa después de la aspiración tiene un tiempo mayor que la pausa después de la expiración.
- **Relación entre la subida y bajada en la respiración (S/D)**
 - ✓ *Expiración más rápida (puntuación 1).* El tiempo de la expiración es menor que el de la aspiración.
 - ✓ *Tiempos iguales (puntuación 2).* El tiempo de aspiración y expiración son iguales.
 - ✓ *Aspiración más rápida (puntuación 3).* El tiempo de la aspiración es menor que el de la expiración.
 - ✓ *Al mínimo (puntuación 4).* El tiempo es tan corto que no se puede medir.
- **Relación entre el tórax y el diafragma (T/D)**
 - ✓ *Totalmente abdominal (puntuación 1).* La respiración, de hecho, es totalmente abdominal, el tórax no se mueve verdaderamente (no está en posición elevada, hinchado, expandido).
 - ✓ *Prevalentemente abdominal (puntuación 2).* El tórax se eleva, pero su medida es netamente menor.
 - ✓ *Prevalentemente torácica (puntuación 3).* El tórax se hincha en mayor medida (o igual) que el abdomen. Sin embargo el tórax no se mueve mucho pero siempre está hinchado.
 - ✓ *Totalmente torácica (puntuación 4).* La respiración es mucho más alta (o sea, en la parte alta del cuerpo), totalmente en la subida sólo del tórax.
- **Frecuencia de la respiración (F)**
 - ✓ *Lenta (puntuación 1).* De 4 a 5 respiraciones por minuto.
 - ✓ *Lentísima (puntuación 2).* Menos de 4 respiraciones por minuto.
 - ✓ *Incoherente (puntuación 3).* El ritmo respiratorio no es regular y cambia continuamente.
 - ✓ *Veloz (puntuación 4).* Seis o más respiraciones por minuto.

La puntuación para la variable de la respiración se obtiene sumando los seis puntos parciales. Se puede observar que lo que se está midiendo, en realidad, es el movimiento físico del abdomen y del tórax cuando la persona respira.

La evaluación del parámetro de la postura está dada sobre la base de la recolección de indicadores relativos a la postura, al movimiento y a la voz. El doctor observa a la persona durante la sesión y emite una evaluación tomando en cuenta los siguientes valores.

- **Postura (P)**

- ✓ *Móvil (puntuación 1)*. La persona asume una postura suave, no rígida, variable, no repetitiva.
- ✓ *Poco móvil (puntuación 2)*. La postura es poco móvil, tensa, no muy variable y un poco estereotipada.
- ✓ *Rígida (puntuación 3)*. El aspecto es decididamente impuesta, rígida, que no cambia.

- **Movimiento (M)**

- ✓ *Suave (puntuación 1)*. El movimiento se efectúa de un modo suave, dulce, pero no moderado, no rígido ni duro.
- ✓ *Moderado (puntuación 2)*. El movimiento es más moderado, dando el aspecto general de escasa movilidad, de falta de energía.
- ✓ *Veloz (puntuación 3)*. El movimiento es más rápido por la situación real externa, arrebatado, un poco brusco, un poco agitado.

- **Voz (V)**

- ✓ *Diafragmática (puntuación 1)*. La voz es profunda, sonora, no estrangulada o aguda, como si proviniese del tórax, con vibración.
- ✓ *De garganta (puntuación 2)*. La voz es sofocada o estrangulada, un poco ronca, un poco afónica.
- ✓ *De cabeza (puntuación 3)*. La voz es más sutil, aguda, metálica.

La puntuación total para la variable de la postura se obtiene sumando los tres parámetros parciales.

En el caso del cortisol y la prolactina, se toma una muestra de sangre y se analizan dichos parámetros.

Para cada uno de los parámetros se obtiene un valor normalizado, de la siguiente manera

$$\text{valor normalizado} = \frac{\text{valor absoluto} - \text{valor mínimo}}{\text{valor máximo} - \text{valor mínimo}} * 10$$

Con este valor normalizado se obtiene una escala más fácil de comprender. Esto es, supongamos que el valor absoluto de un parámetro fue igual al valor mínimo, entonces el valor normalizado es igual a cero, lo que significa que no existe ninguna anomalía del paciente respecto a ese parámetro. Por otra parte, si el valor absoluto es igual al valor máximo, entonces la división es igual a 1 y multiplicada por 10 resulta 10, esto quiere decir que el paciente está en el máximo nivel de anomalía con respecto a ese parámetro.

El nivel total de estrés se obtiene sumando los valores relativos correspondientes a los siete parámetros medidos. Si el resultado de la suma es cero, entonces la persona está en un nivel de bienestar, pero si el resultado es 70, la persona está en el nivel máximo de estrés.

El orden en que se aplican las pruebas es indiferente. Se puede realizar el Cuestionario MSP primero y después las pruebas de sangre o viceversa. O tal vez primero se realicen las pruebas con los sensores y después el cuestionario. En realidad, no existe un orden específico para llevar a cabo las mediciones de los siete parámetros. Lo que sí es importante recalcar es que las pruebas de la Respiración, EMG y GSR se deben realizar juntas. Para esto es necesario acostar al paciente y colocarle los sensores. La prueba de la postura se lleva a cabo mientras el paciente constata el cuestionario, ya que de esta forma, el doctor puede observar la postura, los movimientos y la voz del paciente.

Los resultados de las pruebas de sangre no se obtienen en el momento, como en el caso de los valores de los demás parámetros. De aquí, que se pueden omitir esos resultados y sólo tomar en cuenta los cinco parámetros restantes. De esta forma, la escala sería de 0 a 50, indicando con el 0 que la persona está en completo bienestar y con 50 que la persona está totalmente estresada. Lo anterior es válido, ya que se están realizando más del 70% de las pruebas. Si una persona obtuvo por arriba del 45% de nivel de estrés, no sería lógico que los valores de las pruebas de sangre correspondientes a los parámetros Cortisol y Prolactina fueran pequeños. Es decir, si una persona presenta un alto nivel de estrés en las pruebas de Respiración, EMG, GSR, Postura y MSP sería ilógico que en las otras pruebas no presentará un nivel muy parecido. El estrés afecta todo nuestro cuerpo, no sólo una parte de él.

Como se puede observar, esta escala es muy completa porque toma en cuenta los aspectos psicológico y físico de la persona para evaluar el nivel de estrés. Es por esto, que se pretende implementar la escala Estrés-Bienestar con un sistema digital que sea sencillo de manejar y facilite la evaluación del nivel de estrés en cualquier persona.

1.5 ORGANIZACIÓN DE LA TESIS

Este trabajo está dividido en seis capítulos.

En el primer capítulo se define el concepto de estrés y sus etapas. Se mencionan algunos métodos de medir de estrés en una persona. Por último, se habla de un modo, en particular, de obtener el nivel de estrés en un ser humano considerando el estado físico y emocional de la persona, éste es la escala "Estrés-Bienestar"

El segundo capítulo está compuesto de la teoría utilizada para diseñar los circuitos que servirán de interfaz entre los sensores de Respiración, Conductancia de la Piel (GSR) y Tensión Muscular (EMG) y la computadora. Este capítulo se subdivide en tres partes. La primera parte trata los circuitos analógicos como son: fotodiodos, fototransistores, amplificadores operacionales y decodificadores de tono. La segunda parte habla de los circuitos digitales: circuitos de reloj, convertidores analógico-digitales y multiplexores. En la tercera y última parte, se da una breve explicación de qué es puerto paralelo, sus aplicaciones, cómo trabaja en su forma estándar y la manera de leer y mandar datos desde y hacia el puerto paralelo utilizando un programa en lenguaje C.

En el tercer capítulo se presenta el diseño del sensor de movimiento respiratorio abdominal y torácico y los circuitos que componen la interfaz entre los sensores y la PC. En primer lugar se muestra cómo utilizar un convertidor analógico-digital para construir una interfaz entre un circuito que arroja lecturas

de voltaje analógico en un rango de 0 V a 5 V y una computadora personal, utilizando el puerto paralelo en su modo estándar para recibir los resultados digitales que se tienen a la salida del convertidor. Después, se mencionan todas consideraciones que se deben de tener en cuenta para diseñar el sensor de movimiento respiratorio. Posteriormente, se analizan las señales de salida de los sensores de EMG y GSR para construir los circuitos correspondientes que conviertan esas señales a voltajes analógicos con el rango ya especificado. Por último, se diseña el circuito que permite seleccionar, mediante un programa, las lecturas digitales de cada uno de los sensores.

El cuarto capítulo muestra y explica el funcionamiento de las pantallas del software desarrollado en Visual C 5.0 para obtener el nivel de estrés en una persona.

En el quinto capítulo se describen las pruebas realizadas a los circuitos propuestos y se muestran sus resultados.

En el sexto y último capítulo, se mencionan las aportaciones realizadas con este trabajo, se concluyen los resultados obtenidos con el sistema final y se habla de algunas recomendaciones y de trabajos futuros que se pueden llevar a cabo con base en este proyecto.

Como complemento a este trabajo se adjunta un apéndice. En este apéndice se incluye el código fuente del software desarrollado.

CIRCUITOS ANALÓGICOS Y DIGITALES

2.1 INTRODUCCIÓN

En este capítulo se va a tratar la teoría de los circuitos utilizados para armar la interfaz entre los sensores y la PC. Los sensores pueden entregar una salida analógica que no sea de voltaje, por lo que es necesario convertirla a una lectura de volts. Posteriormente, esa salida analógica de voltaje se convierte a una señal que pueda ser leída por la computadora, es decir, una señal digital. Por lo anterior, este capítulo se divide en dos secciones: Circuitos Analógicos y Circuitos Digitales.

2.2 CIRCUITOS ANALÓGICOS

2.2.1 EMISORES Y RECEPTORES INFRARROJOS^[7]

2.2.1.1 Diodos emisores de luz infrarroja

Una unión P-N se forma en un material semiconductor estimulando una región con átomos donadores y una región adyacente con átomos receptores. Esto produce una distribución no uniforme de impurezas con un abrupto cambio de un tipo de material estimulado a otro. Cuando los átomos donadores dominan, el material se conoce como tipo N y hay un exceso de átomos de tipo N o mayoría de portadores. Para un material tipo N, la mayoría de los portadores están cargados negativamente y son electrones. Cuando los átomos receptores dominan, el material se conoce como tipo P y la mayoría de los portadores están cargados positivamente y son huecos^[8]. La figura 2.1 muestra una unión P-N con un exceso de electrones cargados negativamente y de huecos cargados positivamente.

Cuando la unión se forma, estos electrones y huecos fluyen en direcciones opuestas a lo largo de la unión (sin alguna polarización aplicada) hasta que se logra un equilibrio. Esto crea una barrera de potencial. Si se aplica una polarización externa en la unión que neutraliza esta barrera de potencial, electrones y huecos adicionales serán inyectados o fluirán a través de la unión P-N. Estos portadores se recombinarán por un proceso radioactivo o no radioactivo. En el caso de un proceso radioactivo, la recombinación requiere de que el nivel de energía del electrón caiga para facilitar la recombinación con el hueco. El electrón desprende este exceso de energía en forma de una discreta cantidad de energía infrarroja conocida como fotón. En un proceso no radioactivo, el exceso de energía también se libera pero toma la forma de calor y se cuantifica como *phonons*. En ambos casos, la recombinación ocurre en la proximidad de la unión P-N.

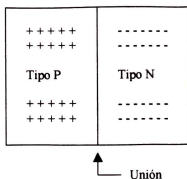


Figura 2.1. La unión P-N se describe como un cambio abrupto en el material que contiene una mayoría de portadores cargados positivamente o huecos a un material con exceso de portadores cargados negativamente o electrones.

La energía máxima posible de los fotones emitidos se determina por la energía de la banda *gap* del sólido en el cual se formó la unión P-N. Existen numerosos elementos y compuestos elementales que tienen energías band-gap que yacen en la región del ultravioleta al infrarrojo. Sin embargo, muchos de estos materiales son candidatos viables para LEDs prácticos. La tabla 2.1 lista algunos materiales utilizados para dispositivos comerciales.

La energía de band gap en la tabla 2.1 se calcula con la siguiente fórmula:

$$E = \frac{1240}{\lambda} \text{ electrón volts}$$

donde λ es la longitud de onda de la emisión y E es la energía de transición en electrón volts.

Tabla 2.1 Algunos de los materiales utilizados para dispositivos LED

Material	Energía de Banda <i>gap</i>	Longitud de onda de la emisión
	Electrón volts (eV)	Nanometros (nm)
Gallium arsenide (GaAs)	1.33	930
Gallium aluminum arsenide (GaAlAs)	1.41	880
Gallium arsenide phosphide (GaAsP)	1.91	650
Gallium Phosphide (GaP)	2.21	560

El porcentaje de la corriente que resulta en la recombinación que da lugar a los fotones de la longitud de onda deseada es una medida de la eficiencia de conversión interna del diodo P-N. Un material con baja eficiencia de conversión interna ofrece un bajo interés práctico como dispositivo de electroluminiscencia. Sin embargo, aún un material con alta eficiencia de conversión interna puede no ser útil si los fotones emitidos no pueden ser eficientemente emitidos desde la estructura del diodo o acoplamiento hasta el ambiente externo. Dos factores importantes controlan el coeficiente de acoplamiento interno-externo. Un factor es la opacidad del material del diodo. El segundo factor es la reflexión interna entre el cristal semiconductor y el material de encapsulación. Este puede causar que el

fotón sea reflejado hacia el cristal y posteriormente reabsorbido. La tabla 2.2 muestra la eficiencia relativa de algunos LEDs como un porcentaje de la eficiencia *gap*.

Tabla 2.2 Eficiencia relativa de algunos LEDs

Material	Eficiencia relativa de salida (% eficiencia de <i>gap</i>)	Longitud de onda de emisión (nm)
GaP	100	560
GaAs	400	930
GaAlAs	800	880

Esta eficiencia se incrementa si existe un decremento en la probabilidad de reabsorción o reflexión del cristal de los fotones emitidos en la unión. Si más fotones escapan del material del diodo, la eficiencia de salida relativa se incrementa.

Se observa que es obvio que los IREDs de GaAlAs y GaAs son preferidos como transmisores por la eficiencia con la cual emiten energía infrarroja. Sin embargo, para ser usados comercialmente, es necesario tener detectores disponibles que respondan a la misma longitud de onda del infrarrojo. En otras palabras, la respuesta espectral del detector debe acoplarse a la longitud de onda de emisión del material escogido para el IRED. El silicio es un material normalmente utilizado ya que su máxima absorción cae en el rango de 750 a 950 nm.

Un ejemplo de diodos emisores de luz infrarroja de GaAs es el IRED TSUS520 de Telefunken. La descripción y características de este dispositivo infrarrojo se comentan a continuación.

2.2.1.2 El IRED TSUS520^[17]

Descripción

Las series TSUS520 son diodos emisores de luz infrarroja con tecnologías estándar de GaAs, moldeados en empaque de plástico claro o con tinte azul-gris. Estos dispositivos están acoplados espectralmente para trabajar con fotodiodos y fototransistores. Sus principales características son:

- Emisor de bajo costo
- Voltaje directo bajo
- Alta potencia e intensidad de radiación
- Adecuado para DC y para operar con pulsos de corriente grande
- Alta confiabilidad
- Buen acoplamiento espectral para fotodetectores de silicio.

Tiene aplicaciones como controles remoto infrarrojos y sistemas de transmisión por aire con un voltaje directo bajo y con requerimientos de bajo costo en combinación con fotodetectores.

La configuración básica para un IRED se muestra en la figura 2.2.

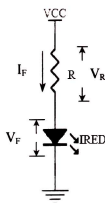


Figura 2.2. Esquema del circuito básico para un IRED.

La corriente continua máxima que puede pasar por el IRED, I_F , es de 150 mA (a temperatura ambiente). La figura 2.3 nos muestra la gráfica de voltaje directo (V_F) contra corriente directa (I_F). En esta gráfica podemos observar que para una I_F máxima, el voltaje que debe caer en el IRED o V_F corresponde a, aproximadamente, 1.4 V.

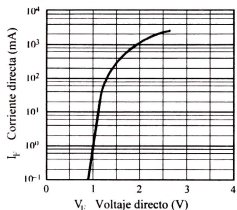


Figura 2.3. Gráfica de Voltaje directo V_F contra Corriente directa I_F

Para calcular el valor de la resistencia R supongamos que $V_{CC} = 5$ V. Por tanto, el valor de R es,

$$R = \frac{V_{CC} - V_F}{I_F} = \frac{5 - 1.4}{150 \times 10^{-3}} = \frac{3.6}{150 \times 10^{-3}} = 24 \Omega$$

Ahora, valor comercial de la resistencia se debe elegir de manera que el voltaje V_F no sea mayor a 1.4 V, si esto sucede, la corriente I_F excedería los 150 mA y el dispositivo ya no trabajaría correctamente.

2.2.1.3 El interruptor óptico reflejante

Los interruptores reflejantes pueden dividirse en dos categorías: detectores de superficie y detectores de línea.

El interruptor reflejante básico consiste de un emisor y un receptor montados juntos en un mismo lado de la superficie o línea que va a ser detectada.

Muchas unidades operan en una combinación de reflejo difuso y reflejo directo. La figura 2.4 muestra estos dos principios.

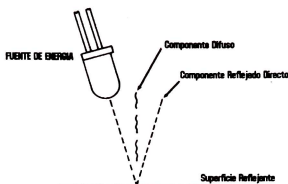


Figura 2.4. Reflejo directo y difuso. El ángulo del componente de reflejo directo es igual al ángulo de incidencia, mientras que el componente difuso puede ser dirigido virtualmente hacia cualquier parte por arriba de la superficie.

Una superficie altamente pulida (como un espejo) producirá un gran reflejo de energía, mientras que una superficie con reflejo difuso (como un papel bond blanco) tendrá un componente de reflejo difuso de energía muy grande. Para tomar ventaja de estos componentes, el diseño de interruptores reflejantes es diferente para cada caso. La unidad óptima para un reflejo directo, usualmente, tiene a los elementos emisor y receptor a lo largo de un cateto de un triángulo isósceles con la superficie reflejante localizada en la intersección de los catetos. La unidad óptima para un reflejo difuso montaría los elementos emisor y sensor paralelos con la superficie reflejante perpendicular a ellos.

2.2.1.4 FOTODETECCIÓN

El ojo humano detecta radiaciones visibles de longitud de onda de entre 3900 y 7500 Angstroms. Otros detectores son capaces de detectar radiaciones electromagnéticas desde ondas de radio hasta rayos X. La detección de longitud de onda que se está considerando incluye longitudes de onda ultravioleta, visibles e infrarrojas, esto es, longitudes de onda desde 0.005 a 4000 microns. Existen tres formas básicas de fotodetección. Estas comprenden las acciones de fotoemisión, fotoconducción y fotovoltaica. Todos los detectores cuánticos responden directamente a la acción de ondas de luz incidentes. La primera, la fotoemisión, comprende luz incidente que libera electrones de una superficie de detección. Esto, ocurre usualmente en un tubo al vacío. Con la segunda acción, la fotoconducción, la luz incidente sobre un material fotosensible causa que el fotodetector altere su conducción. La tercera, la acción fotovoltaica, genera un voltaje cuando la luz golpea el material sensible en el fotodetector.

2.2.1.5 Tipos de fotodetectores

La forma más básica de detección es el fotorresistor. El fotorresistor es una pequeña rebanada de material fotoconductor cuya resistencia se incrementa o decrece según se aplique luz. Los electrones son liberados por la luz y fluyen hacia una fuente de poder positiva. La suposición básica en el

fotorresistor es controlar la energía eléctrica de la fuente de poder. El material fotorresistivo es antirreflejante.

Un segundo detector es un fotodiodo de simple unión. Un fotodiodo es la versión óptica de un diodo estándar. Está construido de una unión PN. Los fotones de luz son absorbidos adentro del dispositivo. Se generan pares hueco-electrón. Los pares se combinan a diferentes profundidades dentro del diodo dependiendo del nivel de energía del fotón. Se utiliza una superficie ancha y delgada para asegurar una máxima absorción. El flujo de la corriente depende de la cantidad de radiación absorbida. Los fotodiodos operan en el modo fotoconductor con polarización de inversa. Operan en el modo fotovoltaico sin polarización.

La celda solar es un fotodiodo que es altamente absorbente. El área de vaciamiento es extremadamente delgada. La celda es cubierta para evitar la reflexión. Los pares hueco-electrón se difunden hacia el área de vaciamiento del diodo donde son convertidos a corriente útil. La corriente de salida depende de la radiación de entrada. Las celdas solares no están polarizadas y su operación es fotovoltaica.

Los fototransistores son dispositivos de dos uniones que tienen un área base muy larga. La región de base del fototransistor absorbe los fotones de energía y genera pares hueco-electrón en una gran región base-colector. El colector, al polarizarse inversamente, drena los huecos hacia la base y los electrones hacia el colector. La unión base emisor polarizada directamente causa que los huecos fluyan de la base al emisor y los electrones del emisor a la base. La polarización directa causa que el fototransistor opere como un transistor convencional. La función básica, entonces, es que la luz induzca al transistor a operar. La polarización del colector es proporcionada por una fuente de poder. La polarización de la base es proporcionada por la luz incidente. La luz también controla la corriente de colector.

2.2.1.6 El fototransistor

Un fototransistor opera de manera similar a un transistor de señal pequeña excepto porque la corriente de base se genera de los fotones que chocan en la región de vaciamiento o de la tradicional polarización directa en la unión base-emisor. Los fotones utilizan la región de vaciamiento formada por la unión colector-base polarizada inversamente para crear una foto corriente, que actúa como una corriente de base o un control de corriente. Muchos fototransistores son del tipo NPN ya que este material ofrece parámetros eléctricos útiles y es fácil de construir. Los pares electrón-hueco creados en la región de vaciamiento base colector causan una foto corriente. Los electrones se mueven hacia el colector o la región tipo N mientras que los huecos se mueven hacia la base o región tipo P.

La figura 2.5 muestra la corriente de colector contra los voltajes colector-emisor para varios niveles de irradiación (o fotones que chocan) en la base del transistor NPN.

Sin cero fotones brincando en la región de base, el voltaje colector-a-emisor se incrementa gradualmente, y sólo la corriente de fuga (o corriente oscura) fluirá hasta que el punto de quiebra es alcanzado. Si la cantidad de fotones se incrementa a un cierto nivel y se estabiliza, y de nuevo, el nivel de fotones se incrementa y se estabiliza con un incremento en el voltaje colector-emisor, se forma una familia de curvas características.

El BPW96 es un fototransistor de silicio NPN y es un ejemplo de lo anteriormente analizado. A continuación se describe este fototransistor.

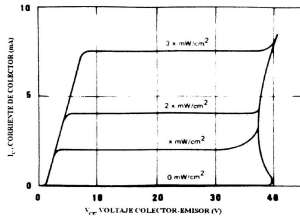


Figura 2.5. Voltaje colector-emisor contra Corriente de Colector. Esta familia de curvas es análoga a la familia de curvas de un transistor tradicional (I_C vs V_{CE}) excepto porque la radiación incidente sustituye a la corriente de base. Cada paso en el nivel del fotón es análogo a un incremento en la corriente de base, resultando en una familia de curvas.

2.2.1.7 El fototransistor de silicio NPN BPW96^[18]

El BPW96 es un fototransistor de alta velocidad y sensibilidad fabricado de silicio de tipo NPN en un empaque estándar de 5 mm. Debido a que su cubierta es de color claro como el agua, el dispositivo es sensible a la radiación visible e infrarroja. El ángulo de vista de $\pm 20^\circ$ lo hace insensible a la luz ambiental. Sus principales características son:

- Tiempos de respuesta rápidos
- Alta fotosensibilidad
- Ángulo de sensibilidad media $\phi = \pm 20^\circ$
- Adecuado para la radiación visible e infrarroja

Tiene aplicaciones en detectores electrónicos y como circuitos manejadores. La configuración básica del fototransistor se muestra en la figura 2.6.

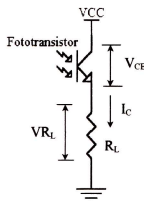


Figura 2.6. Esquema del circuito básico para un fototransistor.

Este circuito se podría utilizar como un interruptor. El voltaje que cae en la resistencia de carga R_L podría alimentar a otro circuito y en un momento dado tendría la posibilidad de encenderlo o apagarlo dependiendo de la luz que alimente al fototransistor.

Para el cálculo de VR_L supongamos que $VCC = 5$ V. Necesitamos definir un valor de irradiación (corriente de base en un transistor normal) para el cual el fototransistor bloqueará el funcionamiento del circuito al que alimente. Veamos la gráfica característica del fototransistor que se muestra en la figura 2.7

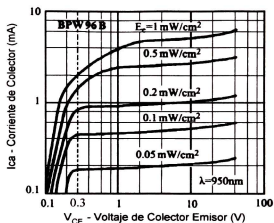


Figura 2.7. Gráfica de Voltaje de Colector-Emisor contra Corriente de Colector.

En esta gráfica tenemos la corriente de colector dependiente del voltaje de colector – emisor para diferentes valores de irradiación. En la gráfica, también se marca el voltaje de saturación de colector emisor V_{CESAT} cuyo valor es de 0.3 V (véanse las hojas de especificaciones del BPW96).

Para nuestro ejemplo se elige una irradiación de 1 mW/cm^2 . Cuando el fototransistor reciba mucha luz entrará en estado de saturación y entonces el $V_{CE} = 0.3$ V, en ese momento el voltaje en R_L será (según el circuito de la figura 2.6)

$$VR_L = VCC - V_{CE} = 5 - 0.3 = 4.7 \text{ V}$$

La corriente de colector para una irradiación de 1 mW/cm^2 y cuando el fototransistor está saturado es de aproximadamente 2 mA (véase la gráfica 2.7). Ahora, ya podemos calcular el valor de la resistencia R_L ,

$$R_L = \frac{VR_L}{I_C} = \frac{4.7}{2 \times 10^{-3}} = 2.35 \times 10^3 \Omega$$

Por tanto, cuando el fototransistor se sature en la resistencia se tendrán 4.7 V y con este voltaje se podría activar alguna alarma para apagar una máquina.

2.2.2 AMPLIFICADORES OPERACIONALES

El nombre de amplificador operacional (AO) se aplica cuando se usa un amplificador muy estable para implementar una amplia variedad de operaciones lineales o no lineales cambiando solamente algunos elementos externos tales como resistencias, capacitores, diodos, etc^[11].

En la figura 2.8a se muestra el modelo del circuito equivalente de un AO que consiste de una impedancia de entrada R_i conectada entre las dos terminales de entrada v_1 y v_2 . El circuito de salida consiste de una fuente de voltaje controlado $A_d v_d$ en serie con una resistencia de salida R_o conectada entre la terminal de salida y tierra (una de las terminales de salida siempre está conectada a tierra). También, ya que se asume que la fuente de voltaje controlado debe ser $A_d v_d$, se tiene que asumir que la ganancia en modo común es cero. Entonces, se diseñará el circuito y se seleccionará el circuito AO integrado de modo que esta suposición sea válida par todas las aplicaciones con AO. El símbolo usual del circuito para el AO es un triángulo que se muestra en la figura 2.8b. No muestra explícitamente ningún componente del circuito equivalente, pero las terminales de entrada invertida y no invertida siempre están marcadas comúnmente por los signos más y menos.

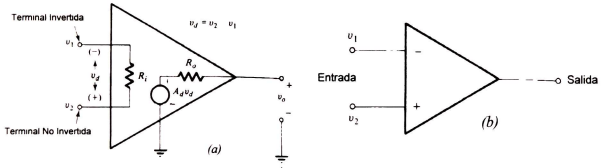


Figura 2.8 El Amplificador Operacional: (a) Circuito equivalente; (b) Símbolo del circuito.

El voltaje de ganancia A_d del AO es usualmente muy grande (típicamente de 100,000) en comparación con la ganancia completa del sistema en cual se está empleando. De hecho, es conveniente asumir que la ganancia es infinita. Similarmente, la impedancia de entrada R_i es mucho más grande (típicamente de 100 k Ω) que las resistencias externas en el sistema y, regularmente, también se asume que deben ser infinitas. Por otra parte, la impedancia de salida R_o es típicamente de 100 Ω y para muchas aplicaciones debe ser despreciada. Cuando se hacen estas aproximaciones, se dice que el AO resultante es ideal.

Una observación importante que se debe hacer acerca del AO ideal es que el voltaje de entrada diferencial $v_d = v_2 - v_1 \approx 0$. La razón para esto es que $v_d = v_o / A_d$ (véase figura 2.7a), y si v_o es finito y A_d es infinita, v_d debe ser cero. En la práctica, el voltaje de salida de un amplificador típico es menos de 10 V. Si se asume que $A_d = 100,000$, el voltaje de entrada diferencial, que produce 10 V a la salida, es 100 μV , una cantidad tan pequeña que puede ser despreciada. Por tanto, regularmente $v_d \approx 0$, aún en un AO real, y se dice que la entrada de un AO es un corto circuito virtual. Esto implica que $v_1 = v_2$, y ya que R_i , la impedancia entre v_1 y v_2 , es muy grande, la corriente en R_i puede ser despreciada, siendo del orden de $(100 \mu\text{V} / 100 \text{ k}\Omega) = 1 \text{ nA}$.

2.2.2.1 El amplificador no inversor lineal

El AO también puede ser usado como un amplificador no inversor. Tal configuración se muestra en la figura 2.9a, y resulta en un amplificador con ganancia de voltaje total mayor o igual que la unidad y con una impedancia de entrada casi infinita.

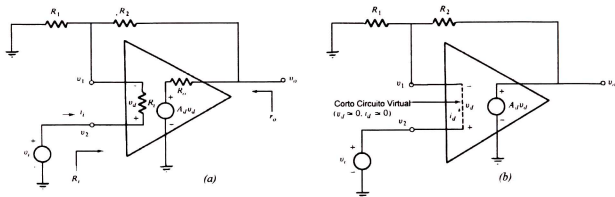


Figura 2.9. Amplificador no inversor lineal. (a) Circuito; (b) Circuito con un AO ideal.

2.2.2.2 Ganancia del amplificador no inversor

La ganancia total del amplificador no inversor se determina más fácilmente si se supone un AO ideal. Entonces $R_o = 0$, R_i es infinita y A_d es infinita, por tanto $v_d \approx 0$. El resultado del circuito equivalente se muestra en la figura 2.9b. Usando este circuito, tenemos

$$v_i = v_2 = v_f \quad (2.1)$$

$$y \quad v_1 = \frac{R_1}{R_1 + R_2} v_o \quad (2.2)$$

$$\text{Entonces } A_v = \frac{v_o}{v_i} = \frac{v_o}{v_1} = \frac{R_1 + R_2}{R_1} = 1 + \frac{R_2}{R_1} \quad (2.3)$$

Esto es, la ganancia total del amplificador no inversor debe ser siempre mayor o igual a la unidad. La ganancia medida A_v para un AO real es muy cercana a la obtenida en (2.3) ya que A_d y R_i son muy grandes y $R_o \ll R_1 + R_2$.

Los valores de R_2 y R_1 son típicamente tales que la magnitud de la ganancia total es menos de 50 y $R_2 < 100 \text{ k}\Omega$. En la práctica, R_1 tiene un valor típico de $1 \text{ k}\Omega$ pero nunca es menor de 100Ω .

La ganancia puede hacerse igual a la unidad reemplazando la resistencia R_2 por un corto circuito y quitando a R_1 . Bajo estas condiciones el circuito se llama *seguidor de voltaje*.

El TL082 es amplificador operacional con un amplio ancho de banda, está empacado con dos AO de tipo JFET. A continuación veremos su descripción.

2.2.2.3 El Amplificador Operacional JFET TL082^[19]

Estos dispositivos son amplificadores operacionales de bajo costo, alta velocidad, cuentan con dos AO de tipo JFET y un voltaje de entrada compensado que se ajusta internamente. Requieren de una corriente pequeña para mantener un amplio ancho de banda de ganancia.

Estos amplificadores pueden ser usados en aplicaciones tales como integradores de alta velocidad, convertidores digitales-analógicos veloces, circuitos de muestreo y retención y en muchos otros circuitos un voltaje de entrada compensado bajo, corrientes de entrada pequeñas, alta impedancia de entrada y un ancho de banda amplio.

En la figura 2.10a se muestra el diagrama lógico del AO TL082 y en la figura 2.10b tenemos la configuración del AO de no inversor.

Veamos un ejemplo de cómo utilizar este amplificador.

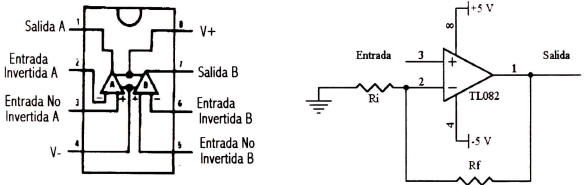


Figura 2.10. (a) Diagrama lógico del amplificador operacional TL082; (b) Configuración básica del AO utilizado como no inversor.

Supongamos que tenemos un circuito que entrega un voltaje de CD de 1 V y queremos obtener a la salida un voltaje de 2 V, por tanto, el voltaje de entrada $v_i = 1$ V y el voltaje de salida $v_o = 2$ V. Además, supongamos también que alimentamos al amplificador con +5V y -5 V. Utilizando la ecuación 2.3 tenemos que,

$$\frac{v_o}{v_i} = 1 + \frac{R_f}{R_i} = \frac{2}{1} = 2$$

despejando R_f nos queda,

$$R_f = (2 - 1)R_i = R_i$$

De la ecuación anterior se observa que ambas resistencias son iguales. Se había comentado (sección 2.2.2.2) que el valor típico de R_i es de 1 k Ω pero no menor de 100 Ω , además R_f debe ser menor de 100 k Ω . De lo anterior se puede concluir que el valor de las resistencias, ya que son iguales, puede ir desde 1 k Ω hasta un valor de resistencia menor de los 100 k Ω .

2.2.3 DECODIFICADOR DE TONO^[20]

2.2.3.1 El decodificador de tono LM567/LM567C

Los decodificadores de tono LM567 y LM567C de propósito general están diseñados para ofrecer un interruptor a tierra con transistor saturado cuando se tiene a la entrada una señal dentro del circuito pasabanda. El circuito consiste de un detector de I y Q manejado por oscilador de voltaje controlado que determina la frecuencia central del decodificador. Se utilizan componentes externos para obtener una frecuencia central y un ancho de banda. En la figura 2.11a se muestra el diagrama lógico del decodificador de tono LM567/LM567C.

Las características del LM567/LM567C son: ancho de banda ajustable de 0 a 14%, inmunidad a falsas señales, frecuencia central muy estable y una frecuencia central ajustable desde 0.01 Hz hasta 500 KHz.

Algunas de sus aplicaciones son: decodificador de tono de toque, oscilador de precisión, control y monitoreo de frecuencias^[9]

2.2.3.2 El decodificador de tono LM567/LM567C básico

En la figura 2.11b se muestra el circuito básico del decodificador de tono LM567/LM567C. La frecuencia central del decodificador de tono está dada por la ecuación 2.16.

$$f_o = \frac{1}{1.1 R_1 C_1} \quad (2.16)$$

El ancho de banda del filtro, BW , se calcula con la ecuación 2.17.

$$BW = \% f_o \quad (2.17)$$

La ecuación anterior nos indica que el ancho de banda está en función de la frecuencia central. Según las especificaciones del decodificador de tono, el ancho de banda mayor es el 14% de f_o . El significado de BW es un rango de error alrededor de f_o , es decir, por ejemplo, si $BW = 14\% f_o$ entonces la frecuencia central variará entre $\pm 7\% f_o$.

Se propone un ejemplo para entender mejor el uso de este decodificador de tono. Supongamos que se quiere detectar una frecuencia de 5 KHz

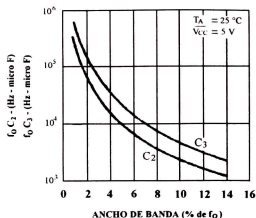


Figura 2.12. Gráfica de ancho de banda (BW) contra $f_0 C_2$ y $f_0 C_3$.

2.3 CIRCUITOS DIGITALES

2.3.1 CIRCUITOS DE RELOJ^[21]

Muchos circuitos digitales necesitan un tren de pulsos de reloj para sincronizar las operaciones de un sistema completo. El tren de pulsos es generado por circuitos de reloj llamados *multivibradores astables*^[12,13]

2.3.1.1 El temporizador LM555 como multivibrador astable

El LM555 es un dispositivo altamente estable para generar tiempos exactos de retardo u oscilación. Se tienen terminales adicionales para el disparo o inicialización si se desea. En el modo de operación de tiempo de retardo, el tiempo se controla por una resistencia y un capacitor externos. Para la operación astable, es decir, como oscilador, la frecuencia de corrida libre y el ciclo de trabajo se controlan con exactitud con dos resistencias externas y un capacitor. Tiene aplicaciones como generador de pulsos, generador de tiempos de retardo, regulador de tiempos precisos y secuenciales.

El temporizador LM555 es un dispositivo compatible con TTL. La figura 2.13 muestra la forma en que deben conectarse los componentes externos para que el LM555 opere como un oscilador. Su salida es un pulso rectangular que oscila entre dos niveles lógicos; el tiempo que el oscilador dura en cada estado lógico depende de los valores de R y C . En la figura aparecen las fórmulas para el cálculo de los intervalos t_1 y t_2 y del periodo completo de la oscilación, T . La frecuencia de oscilación es el recíproco de T .

Como lo indican las fórmulas, los intervalos t_1 y t_2 no pueden ser iguales a menos que R_A sea cero. Esto no puede hacerse porque circularía una corriente excesiva por el dispositivo. Lo anterior significa que es imposible producir como salida una onda cuadrada perfecta con 50% de ciclo de trabajo. Sin embargo es posible obtener un ciclo de trabajo muy cercano al 50% al hacer $R_B \gg R_A$ (mientras que, al mismo tiempo, se mantiene R_A mayor que $1\text{ K}\Omega$), de forma que $t_1 \approx t_2$.

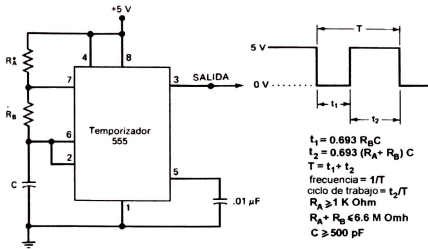


Figura 2.13. Empleo del temporizador 555 como multivibrador astable.

EJEMPLO: Se quiere una frecuencia de oscilación de 5 kHz y un ciclo de trabajo del 55%, encuentre los valores de los dispositivos externos del 555.

Solución: Para una frecuencia de 5 kHz tenemos un periodo de,

$$T = \frac{1}{F} = \frac{1}{5 \text{ kHz}} = 200 \mu\text{s}$$

Para un ciclo de trabajo del 55%, tenemos que t_2 es,

$$\text{ciclo de trabajo} = \frac{t_2}{T} \quad \therefore \quad t_2 = \text{ciclo de trabajo} * T = 0.55 * 200 \mu\text{s} = 110 \mu\text{s}$$

y para t_1 tenemos que,

$$T = t_1 + t_2 \quad \therefore \quad t_1 = T - t_2 = 200 \times 10^{-6} - 110 \times 10^{-6} = 90 \times 10^{-6} = 90 \mu\text{s}$$

Se propone $R_B = 10 \text{ K}\Omega$, por lo tanto,

$$t_1 = 0.693 R_B C \quad \therefore \quad C = \frac{t_1}{0.693 * R_B} = \frac{90 \times 10^{-6}}{0.693 * 10 \times 10^3} = 1.298 \times 10^{-8} = 12.98 \text{ nF}$$

Ahora encontramos el valor de R_A ,

$$t_2 = 0.693 (R_A + R_B) C \quad \therefore \quad R_A = \frac{t_2}{0.693 * C} - R_B = \frac{110 \times 10^{-6}}{0.693 * 12.98 \times 10^{-9}} - 10 \times 10^3 = 2.22 \text{ k}\Omega$$

2.3.2 CONVERTIDORES ANALÓGICO – DIGITALES

2.3.2.1 Conversión Analógica – Digital

Un convertidor A/D (*Analógico / Digital*) toma un voltaje de entrada analógico y después de cierto tiempo produce un código de salida digital que representa la entrada analógica^[12]. El proceso de conversión A/D es generalmente más complejo y largo que el proceso D/A (*Digital / Analógico*), y se han creado y utilizado muchos métodos. Se examinarán varios de estos métodos con detalle aunque nunca se llegue a necesitar diseñar o construir convertidores A/D (se tienen a disposición como unidades completamente empaquetadas). Sin embargo, las técnicas que se utilizan ofrecen una perspectiva de qué factores determinan el rendimiento de un convertidor A/D.

Varios tipos importantes de ADC (*Analog to Digital Converter*) utilizan un convertidor D/A como parte de sus circuitos. La figura 2.14 es un diagrama de bloque general para esta clase de ADC. La temporización para realizar la operación la proporciona la señal de reloj de entrada. La unidad de control contiene los circuitos lógicos para generar la secuencia de operaciones adecuada en respuesta al comando de INICIO, el cual comienza el proceso de conversión. El comparador con amplificador operacional (cuyo funcionamiento se describe más adelante) tiene dos entradas analógicas y una salida digital que intercambia estados, según qué entrada analógica sea mayor.

La operación básica de los convertidores A/D de este tipo consta de los siguientes pasos:

1. El comando de INICIO pasa a ALTO, dando inicio a la operación.
2. A una frecuencia determinada por el reloj, la unidad de control modifica continuamente el número binario que está almacenado en el registro.
3. El número binario del registro es convertido a un voltaje analógico, V_{AX} , por el convertidor D/A.
4. El comparador compara V_{AX} con la entrada analógica V_A . En tanto que $V_{AX} < V_A$, la salida del comparador permanece en ALTO. Cuando V_{AX} excede a V_A por lo menos en una cantidad igual a V_T (Voltaje de umbral), la salida del comparador pasa a BAJO y suspende el proceso de modificación del número del registro. En este punto, V_{AX} es un valor muy aproximado de V_A , y el número digital del registro, que es el equivalente digital de V_{AX} es asimismo el equivalente digital de V_A , en los límites de la resolución y exactitud del sistema.
5. La lógica de control activa la señal de fin de conversión, FDC, cuando se completa el proceso de conversión.

Las diversas variaciones de este esquema de conversión A/D difieren principalmente en la forma en que la sección de control modifica continuamente los números contenidos en el registro. De lo contrario, la idea básica es la misma, con el registro que contiene la salida digital requerida cuando se completa el proceso de conversión.

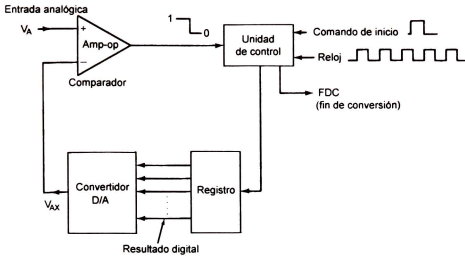


Figura 2.14. Diagrama general de una clase de convertidores A/D

2.3.2.2 ADC de Aproximaciones Sucesivas

El convertidor de aproximaciones sucesivas (CAS) es uno de los tipos de convertidor más utilizado. Tienen un tiempo fijo de conversión que no depende del valor de la señal analógica.

La configuración básica del ADC de aproximaciones sucesivas se muestra en la figura 2.15a. La lógica de control modifica bit por bit el contenido del registro hasta que el contenido de éste se convierte en el equivalente digital de la entrada analógica V_A , claro está, dentro de la resolución del convertidor. La secuencia básica de operación está dada por el diagrama de flujo de la figura 2.15b. Se seguirá este diagrama de flujo para el ejemplo de la figura 2.16.

Para este ejemplo se ha seleccionado un convertidor sencillo de cuatro bits con un tamaño de paso igual a un volt. Aunque en la práctica muchos CAS tienen más bits, la operación es exactamente la misma. Los cuatro bits del registro que sirven como entradas al DAC tienen como factores de ponderación 8, 4, 2 y 1 volt, respectivamente.

Supongamos que la entrada analógica es $V_A = 10.4$ V. La operación del convertidor comienza cuando la lógica de control borra todos los bits del registro y, con ello, los pone en 0; así es como $Q_3 = Q_2 = Q_1 = Q_0 = 0$. Se describirá esta condición como $[Q] = 0000$. Lo anterior hace que la salida del DAC sea $V_{AX} = 0$ V, como se indica en el tiempo t_0 del diagrama de temporización de la figura 2.16. Con $V_{AX} < V_A$, la salida del comparador es ALTO.

En el siguiente paso (tiempo t_1), la lógica de control pone el MSB del registro en 1, de forma que $[Q] = 1000$. Esto hace que $V_{AX} = 8$ V. Dado que $V_{AX} < V_A$, la salida COMP permanece en ALTO. Este nivel le indica a la lógica de control que la parte del MSB en 1 no ha hecho que V_{AX} sea mayor que V_A ; por tanto el MSB se mantiene en 1.

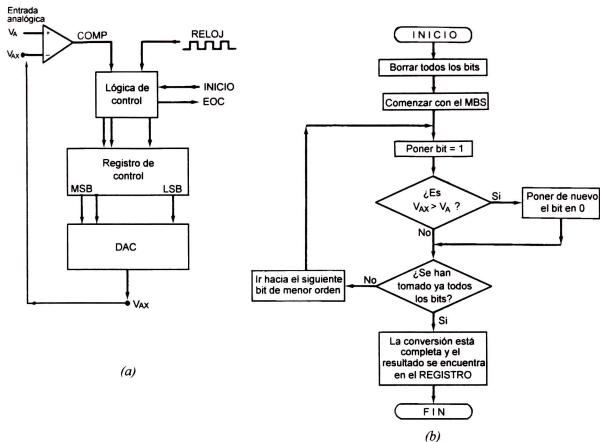


Figura 2.15. ADC de aproximaciones sucesivas: (a) Diagrama de bloques simplificado; (b) Diagrama de flujo para la operación.

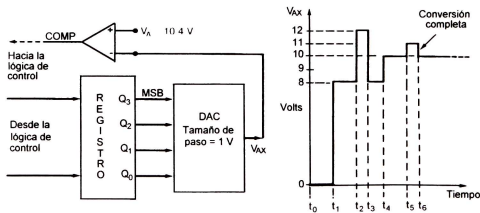


Figura 2.16. Ilustración del funcionamiento de un CAS de cuatro bits utilizando un DAC con tamaño de paso igual a 1 V, y $V_A = 10.4$ V.

Ahora la lógica de control avanza hacia el siguiente bit, Q_2 . Lo pone en 1 para producir $[Q] = 1100$ y con ello $V_{AX} = 12$ V en t_2 . Dado que $V_{AX} > V_A$, la salida COMP cambia hacia el estado BAJO. Esto indica a la lógica de control que el valor de V_{AX} es muy grande; por tanto, la lógica de control pone a Q_2 en cero en t_3 . De este modo el contenido del registro regresa a 1000 y V_{AX} es de nuevo igual a 8 V.

El siguiente paso ocurre en t_4 , instante en que la lógica de control pone en 1 el bit correspondiente a Q_1 de modo que $[Q] = 1010$ y $V_{AX} = 10$ V. Con $V_{AX} < V_A$, COMP es ALTO y esto le indica a la lógica de control que mantenga Q_1 en 1.

El paso final ocurre en t_5 , que es donde la lógica de control pone el bit Q_0 en 1 con lo que $[Q] = 1011$ y $V_{AX} = 11$ V. Dado que $V_{AX} > V_A$, COMP cambia hacia el estado BAJO para indicar que V_{AX} es muy grande y entonces la lógica de control pone a Q_0 en 0 de nuevo en 0.

En este momento se han procesado todos los bits del registro, la conversión está completa y la lógica de control activa la señal de salida \overline{FDC} para indicar que en el registro se encuentra el equivalente digital de V_A . Para este ejemplo, la salida digital correspondiente a $V_A = 10.4$ es $[Q] = 1010$. Nótese que en realidad es equivalente a un voltaje de 10 V, el cual es menor que la entrada analógica; esto es una característica del método de aproximaciones sucesivas.

EJEMPLO. Un CAS de 8 bits tiene una resolución de 20 mV. ¿Cuál será la salida digital para una entrada analógica de 2.17 V?

Solución.

$$2.17 \text{ V} / 20 \text{ mV} = 108.5$$

de modo que el paso 108 produce un $V_{AX} = 2.16$ V, mientras que el 109 genera un $V_{AX} = 2.18$ V. El CAS siempre produce un V_{AX} final que corresponde al voltaje generado por un paso anterior a V_A . Por consiguiente, para el caso donde $V_A = 2.17$ V la salida digital es $108_{10} = 01101100_2$

2.3.2.3 Convertidor Analógico – Digital de aproximaciones sucesivas ADC0804^[12,22]

Los diversos fabricantes de circuitos integrados proporcionan convertidores analógico digitales con un amplio rango de características de operación^[12]. La figura 2.17 presenta la distribución de terminales para el ADC0804, que es un circuito integrado de 20 terminales fabricado con tecnología CMOS y que lleva a cabo la conversión A/D utilizando el método de aproximaciones sucesivas. Algunas de sus características importantes son:

- Tiene dos entradas analógicas: $V_{IN}(+)$ y $V_{IN}(-)$ que permiten tener entradas diferenciales. En otras palabras, el voltaje real de entrada V_{IN} es la diferencia entre los voltajes aplicados en dichas terminales ($V_{IN} \text{ analógico} = V_{IN}(+) - V_{IN}(-)$). En mediciones hechas en un solo punto, la entrada analógica se aplica en $V_{IN}(+)$ mientras que $V_{IN}(-)$ se conecta a tierra analógica. Durante la operación normal, el convertidor utiliza $V_{CC} = +5$ V como voltaje de referencia y la entrada analógica puede variar desde 0 hasta 5 V, que es el valor de escala completa.
- Convierte el voltaje analógico de entrada en una salida digital de ocho bits. Con 8 bits la resolución es de $5\text{V}/256 = 19.53$ mV. Tiene un circuito de reloj interno que produce una frecuencia igual a $1/(1.1 \text{ RC})$, donde R y C son los valores de los componentes conectados al convertidor de manera externa. Una frecuencia típica de reloj es de 606 KHz y se obtiene con $R=10 \text{ K}\Omega$ y $C=150 \text{ pF}$.

- Al utilizar una frecuencia de 606 KHz, el tiempo de conversión es, aproximadamente, igual a 100 μ s.

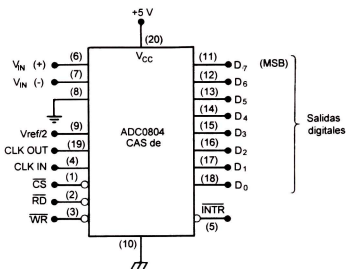


Figura 2.17. El ADC0804 es un ADC de aproximaciones sucesivas de ocho bits con salidas de tres estados. Los números entre paréntesis corresponden al número de las terminales del circuito integrado.

Este circuito integrado está diseñado para ser conectado con facilidad al canal de datos de un microprocesador. Por esta razón, los nombres de algunas entradas y salidas del ADC0804 tienen origen en las funciones que son comunes en los sistemas basados en un microprocesador. Las funciones de estas entradas y salidas son las siguientes:

\overline{CS} Habilitación del microcircuito (Chip Select). Esta entrada debe conectarse en su estado activo en BAJO para que las entradas \overline{RD} o \overline{WR} tengan efecto. Con \overline{CS} en el estado ALTO, las salidas digitales se encuentran en su estado de alta impedancia y no se lleva a cabo ninguna conversión.

\overline{RD} Habilitación de la salida (Output Enable). Esta entrada se emplea para habilitar los buffers de las salidas digitales.

\overline{WR} Inicio de conversión (Start Conversion). En esta entrada se aplica un pulso BAJO para dar inicio a un nuevo proceso de conversión.

\overline{INTR} Fin de conversión (End of Conversion). Esta señal cambia hacia el estado BAJO al inicio de la conversión cuando la conversión termina, regresa al estado ALTO.

$V_{ref/2}$. Esta es una entrada opcional que puede emplearse para disminuir el voltaje de referencia interno y con ello cambiar el rango analógico de entrada que el convertidor puede manejar, Cuando esta entrada no está conectada, permanece en 2.5 V ($V_{CC}/2$) debido a que en este caso V_{CC} se emplea como referencia. Si se conecta un voltaje interno en esta terminal, la referencia interna cambia y su valor es el doble del correspondiente al voltaje externo. La tabla 2.3 ilustra este hecho.

Entrada de reloj (CLK IN). Se utiliza como entrada del reloj externo o para conectar un capacitor cuando se hace uso del reloj interno.

Tabla 2.3. Diferentes rangos de entrada analógica y resoluciones para $V_{ref}/2$.

$V_{ref}/2$	Rango de entrada analógico (V)	Resolución (mV)
Abierto	0 - 5	19.6
2.25	0 - 4.5	17.6
2.0	0 - 4	15.7
1.5	0 - 3	11.8

2.3.3 MULTIPLEXORES (SELECTORES DE DATOS)

Un *multiplexor o selector de datos* es un circuito lógico que acepta varias entradas de datos y permite sólo a una de ellas alcanzar la salida. La dirección deseada de los datos de entrada hacia la salida es controlada por entradas de SELECCIÓN (que algunas veces se conocen como entradas de DIRECCIÓN)^[12] La figura 2.18 muestra el diagrama funcional de un multiplexor general (MUX). En este diagrama las entradas y salidas se trazan como flechas grandes para indicar que pueden ser una o más líneas de señales.

El multiplexor actúa como un interruptor de posiciones múltiples controlado digitalmente, donde el código digital que se aplica a las entradas de SELECCIÓN controla qué entradas de datos serán trasladadas hacia la salida. Por ejemplo, la salida Z será igual a la entrada de datos I_0 de algún código de entrada de SELECCIÓN determinado; Z será igual a I_1 para otro código de entrada de SELECCIÓN específico, y así sucesivamente. Dicho de otra manera, un multiplexor selecciona una de N fuentes de datos de entrada y transmite los datos seleccionados a un solo canal de salida. A esto se le llama multiplexación.

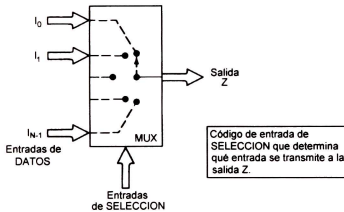


Figura 2.18 Diagrama funcional de un multiplexor digital (MUX).

2.3.3.1 Multiplexor básico de dos entradas

La figura 2.19 muestra la circuitería lógica de un multiplexor de dos entradas, I_0 e I_1 y entrada de SELECCIÓN S . El nivel lógico que se aplica a la entrada S determina qué compuerta AND se habilita de manera que su entrada de datos atraviese la compuerta OR hacia la salida Z. La expresión booleana de la salida Z es

$$Z = I_0 \bar{S} + I_1 S$$

Con $S = 0$, esta expresión se convierte en

$$Z = I_0 \cdot 1 + I_1 \cdot 0 = I_0$$

lo cual indica que Z será idéntica a la señal de entrada I_0 , que puede ser un nivel lógico fijo o bien, una señal lógica que varía con el tiempo. Con $S = 1$, la expresión se transforma en

$$Z = I_0 \cdot 0 + I_1 \cdot 1$$

lo cual muestra que la salida Z será idéntica a la señal de entrada I_1 .

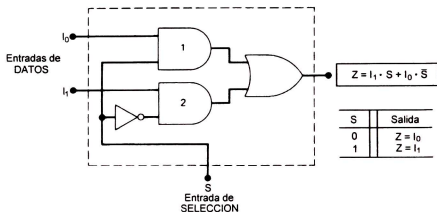


Figura 2.19. Multiplexor de dos entradas.

2.3.3.2 Multiplexor de cuatro entradas

Se puede aplicar la misma idea básica para formar el multiplexor de cuatro entradas que se muestra en la figura 2.20. Aquí se tienen cuatro entradas, que se transmiten en forma selectiva a la salida con base en las cuatro combinaciones posibles de las entradas de selección $S_1 S_0$. A cada entrada de datos se accede con una combinación diferente de niveles de entrada de selección. I_0 se captura con $\bar{S}_1 \bar{S}_0$ de manera que I_0 pase a través de su compuerta AND hacia la salida Z sólo cuando $S_1 = 0$ y $S_0 = 0$. La tabla de la figura da las salidas de otros tres códigos de selección de entrada.

2.3.3.3 Multiplexor Cuádruple de dos entradas 74LS157

Este es un circuito integrado que contiene cuatro multiplexores de dos entradas como el de la figura 2.19. El diagrama lógico del 74LS157 se muestra en la figura 2.21a^[16]

Resulta ventajoso pensar en este multiplexor como un simple multiplexor de dos entradas, pero donde cada entrada tiene cuatro líneas y la salida también. Las cuatro líneas de salida conmutan entre los dos

conjuntos de cuatro líneas de entrada bajo el control de la entrada de selección. Esta operación se representa por el símbolo lógico del 74LS157 de la figura 2.21b.

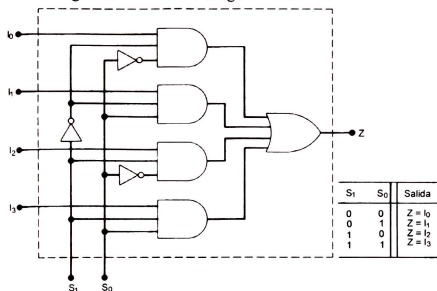


Figura 2.20. Multiplexor de cuatro entradas.

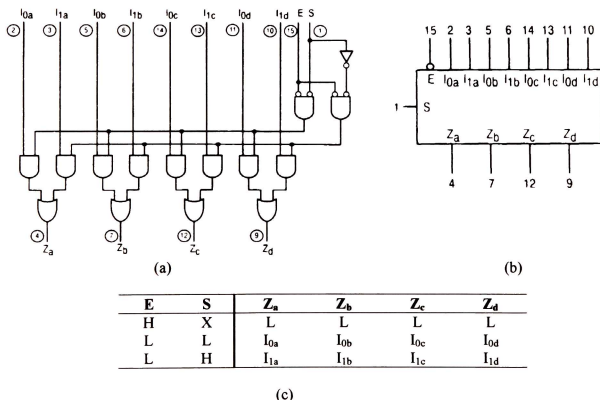


Figura 2.21. (a) Diagrama lógico del multiplexor 74LS157; (b) símbolo lógico; (c) tabla de verdad.

2.3.4 EL PUERTO PARALELO DE LA PC

Cuando IBM introdujo la PC, en 1981, incorporó el puerto paralelo como una alternativa al lento puerto serial como un medio para manejar lo último en tecnología, la impresora de matriz de puntos^[14]. El puerto paralelo tiene la capacidad de transmitir 8 bits de datos al mismo tiempo, mientras el puerto serie transmite un solo bit. Cuando la PC se introdujo, las impresoras de matriz de puntos fue el periférico principal que utilizó el puerto paralelo. La tecnología progresaba y la necesidad de una mayor conectividad externa se incrementaba, se convirtió en el medio por el cual se podían conectar periféricos de alto rendimiento. Estos periféricos van desde dispositivos que comparten impresoras, manejadores de discos portables, escáner y reproductores de CD. En la figura 2.22 se muestra la ubicación del puerto paralelo en la PC.

Los diseñadores de periféricos siempre se enfrentaban con las limitaciones del puerto paralelo, entre ellas se encuentran la falta de rendimiento dada la baja velocidad de transferencia de información (máximo 150 kbytes por segundo), problemas de compatibilidad en la interfaz eléctrica entre las diferentes plataformas y la limitación de la longitud máxima del cable (1.5 m).

En 1991, los fabricantes de impresoras entre los que se incluían Lexmark, IBM y Texas entre otros, decidieron reunirse y formar una alianza para buscar la estandarización en todos los aspectos que tienen que ver con el control de las impresoras a través del puerto paralelo; dicha reunión recibió el nombre de NPA (Network Printing Alliance). Durante dichas reuniones se llegó a la conclusión de que el objetivo buscado por todos no se podría conseguir con las limitaciones que presentaba el puerto paralelo tradicional, sino, con una nueva interfaz paralela de alto rendimiento.

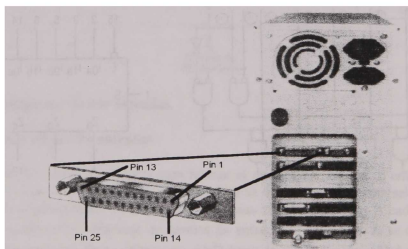


Figura 2.22. Ubicación del puerto paralelo.

La IEEE hizo un llamado a la NPA para conformar un grupo encargado de desarrollar el nuevo estándar para el puerto paralelo bidireccional de alta velocidad, las exigencias eran que éste debía ser totalmente compatible, tanto en hardware como en software, con la interfaz original y además, debería transferir información hasta 1 Mbyte por segundo. El resultado de este proceso fue el estándar IEEE 1284, el cual lleva el nombre de Método de señalización estándar para una interfaz periférica

bidireccional para computadoras personales (“IEEE Std. 1284 – 1994, Estándar Signalling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers”).

2.3.4.1 El estándar IEEE 1284

En lo que se refiere a la parte física el estándar IEEE 1284 define tres tipos diferentes de conectores, llamados A B y C. El tipo A es el conector tipo DB de 25 pines que encontramos tradicionalmente en la parte trasera de las computadoras. El tipo B corresponde al conector Centronics de 36 pines, el cual se encuentra generalmente en las impresoras. El tipo C es un conector miniatura de 36 pines con clips de seguridad, el cual es recomendado para los nuevos diseños dadas sus ventajas en cuanto a la seguridad en la conexión y el ahorro de espacio en el circuito.

En la parte de operación o comunicaciones el IEEE 1284 define 5 modos diferentes que son:

- Modo de compatibilidad (SPP)
- Modo nibble (SPP)
- Modo byte (SPP)
- Modo EPP (*Enhanced Parallel Port*: Puerto Paralelo Mejorado)
- Modo ECP (*Extended Capabilities Mode*: Modo de Capacidades Extendidas)

Los dos modos que nos interesa conocer son los modos nibble y byte, a continuación se describen.

Los modos compatibilidad, nibble y byte trabajan utilizando el mismo hardware que se tenía para el SPP; mientras que los modos EPP y ECP requieren hardware externo que les permita tener alta velocidad. Cuando se desea configurar una computadora personal para que trabaje en uno de estos tres modos, se debe ir al Setup de la BIOS (*Basic Input Output System*: Sistema básico de entrada/salida) en donde se encuentra una opción que permite seleccionar uno de ellos.

Modo nibble

En el modo nibble se pueden leer 4 bits de información al mismo tiempo, es la forma más comúnmente usada para llevar datos del mundo exterior hacia la PC. Para realizar esta lectura de datos se recurre a los pines que corresponden al registro status, el cual tiene 5 líneas que sirven para dicho propósito. En este caso el procedimiento consiste en hacer dos lecturas de 4 bits cada una y luego el software de la computadora se encarga de reorganizar la información tomada esto último es importante ya que los bits no entran en orden lógico y algunos de ellos se leen invertidos (un nivel de 5 voltios se lee como un "0" lógico y viceversa).

El promedio de transferencia de información en este modo no supera los 50 kbytes por segundo dado que se tiene que hacer doble lectura (dos lecturas de 4 bits), por lo tanto no es muy usada por periféricos externos como CD-ROM, manejadores o similares. La ventaja es que es muy simple y se utiliza ampliamente en los aparatos diseñados por los hobystas además, trabaja en forma similar en cualquier PC (vieja o nueva).

Modo byte

El modo byte permite hacer lecturas y escrituras de 8 bits de datos al mismo tiempo, para esto se requiere que el puerto paralelo sea bidireccional. La velocidad con que se pueden leer datos del puerto

en este modo es similar a la velocidad con que se pueden escribir en el modo compatibilidad, es, decir 150 kbytes por segundo.

Esta característica de puerto bidireccional es propia de las computadoras modernas, para habilitarlo o deshabilitarlo se utiliza el bit 5 del puerto de control. Cuando el bit se pone en "1", los pines 2 a 9 del conector (correspondientes a los datos) se colocan en estado de alta impedancia. Una vez que esto se ha hecho, se puede hacer la lectura de la información a través del puerto de datos. A partir de ese momento cualquier byte que se envíe hacia el puerto será almacenado, mas no estará disponible en los pines del conector. Para deshabilitar el modo bidireccional solo basta con poner en "0" el bit 5 del puerto de control.

En algunas computadoras la forma para habilitar el modo bidireccional es poniendo en "1" el bit 6 del puerto de control y para deshabilitarlo se pone en "1" el bit 5 del puerto de control. El método para saber si nuestra computadora tiene puerto de datos bidireccional sería realizando el procedimiento anterior y después revisar con una punta lógica el estado de los pines para comprobar si se encuentra en alta impedancia.

Como se puede observar, el modo byte no es muy común en todas las computadoras, por lo tanto no se utilizará ese modo. En cambio, se usará el modo nibble, es más problemático al programar, pero con esto se asegura que el programa funcionará tanto en computadoras nuevas como en equipos no compatibles con el modo byte.

2.3.4.2 Uso del Puerto de Impresión de la PC para el control y la adquisición de datos^[15]

Asignaciones del Puerto

Cada puerto de impresión consiste de tres direcciones de puerto: de datos, de estado y de control. Estas direcciones están en orden secuencial. Esto es, si el puerto de datos está en la dirección 0x0378, el puerto correspondiente al estado está en 0x0379 y el puerto de control está en 0x037a.

Lo siguiente es típico,

Puerto de la Impresora	Datos	Estado	Control
LPT1	0x03bc	0x03bd	0x03be
LPT2	0x0378	0x0379	0x037a
LPT3	0x0278	0x0279	0x027a

Por lo general, la dirección base para el LPT1 es 0x0378 o 0x03bc.

Para identificar las asignaciones de direcciones para una máquina en particular, se puede usar el programa de debug del DOS para desplegar las localidades de memoria 0040:0008. Por ejemplo:

```
>debug
-d 0040:0008 L8
0040:0008    78 03 78 02 00 00 00 00
```

Nótese que en el ejemplo LPT1 está en 0x0378, LPT2 está en 0x0278 y LPT3 y LPT4 no están asignados.

Una técnica alternativa es ejecutar el Diagnostics de Microsoft (MSD.EXE) y revisar las asignaciones del LPT.

Una tercera opción es incluir el siguiente segmento de programa en C (sólo disponible para DOS) para detectar automáticamente los puertos LPT disponibles en una PC

```

#include <stdio.h>
#include <dos.h>
#define MAXLPT 3
void main(void){
int lptno;
unsigned lptadir[MAXLPT];
for(lptno=0; lptno<MAXLPT; lptno++) {
lptdir[lptno]=*(unsigned far *) MK_FP(0x40, 0x008+(lptno*2));
if(lptdir[lptno]) printf("LPT%d encontrado en 0x%04x\n", lptno+1, lptdir[lptno]);
else printf("LPT%d no encontrado\n", lptno+1); } }

```

Salidas del Puerto

La figura 2.23 muestra las asignaciones de los pines del conector DB-25 y la figura 2.24 muestra las asignaciones de los bits en los tres puertos

Pin	Descripción	Dirección
1	Strobe	PC Output
2	Data 0	PC Output
3	Data 1	PC Output
4	Data 2	PC Output
5	Data 3	PC Output
6	Data 4	PC Output
7	Data 5	PC Output
8	Data 6	PC Output
9	Data 7	PC Output
10	ACK	PC Input
11	Busy	PC Input
12	Paper Empty	PC Input
13	Select	PC Input
14	Auto Feed	PC Output
15	Error	PC Input
16	Initialize Printer	PC Output
17	Select Input	PC Output

Vista del lado del conector DB-25 macho



Nota: Los pines 18-25 van conectados a tierra.

Figura 2.23 Asignaciones de pines.

Nótese que hay ocho salidas en el Puerto de Datos (Data 7(msb) – Data 0) y cuatro salidas adicionales en el nibble bajo del Puerto de Control: Select_In, INIT, Autofeed y Strobe.

Todas las salidas en el Puerto de Datos son valores lógicos. Esto es, escribir un 1 lógico en un bit causa que la salida correspondiente tengo un nivel ALTO. Sin embargo, las salidas Select_In, Autofeed y Strobe en el Puerto de Control tienen una lógica invertida. Esto es, enviar un 1 lógico a un bit causa un cero en la salida correspondiente, o lo que es lo mismo, la salida toma un nivel BAJO. Esto, suma un poco de complejidad al uso del puerto de impresión, pero esto se arregla invirtiendo esos bits utilizando la función OR exclusiva antes de enviar los datos a la salida.



Figura 2.24. Asignaciones de puertos.

A continuación se presenta un segmento de programa en C como ejemplo para invertir una salida utilizando la función OR exclusiva:

```
#define DATOS 0x03bc
#define ESTADO DATOS + 1
#define CONTROL DATOS + 2
...
int val1, val2;
...
val1 = 0x81; outportb(DATOS, val1); /* 1000 0001 */
val2 = 0x08; /* 0000 1000 */
outportb(CONTROL, val2 ^ 0x0b); /* 0000 1000 ⊕ 0000 1011 = 0000 0011 */
```

Se sabe que en el Puerto de Control tenemos tres salidas que están invertidas, si se enviarán los bits directamente a este puerto, estas tres salidas invertirían los bits y la lectura no sería la correcta. Lo que se hizo en el ejemplo anterior, en el caso de val2, fue negar los bits correspondientes a Select_In, Autofeed y Strobe para que cuando se vuelvan a negar los bits, la lectura de la salida sea igual a los bits enviados originalmente, esto es, en el nibble más bajo se envía originalmente 1000, al aplicar la función OR exclusiva con 1011, se obtiene 0011. Al ser enviados al Puerto de Control, los bits correspondientes a las salidas negadas se vuelven a invertir teniendo como resultado 1000, que es el valor que se desea a la salida.

Entradas del Puerto

De la figura 2.24 se puede observar que en el Puerto de Estado hay 5 pines que llegan desde la impresora: BSY, ACK, PE (paper empty), SELECT y ERROR.

El intento original del nombre de los pines es intuitivo. Un nivel ALTO en SELECT indica que la impresora está en línea. Un nivel ALTO en BSY o PE le indica a la PC que la impresora está ocupada o que no tiene papel. Un pulso bajo en ACK indica que la impresora recibió algo. Un nivel BAJO en ERROR indica que la impresora presenta una condición de error.

DISEÑO DEL SISTEMA

3.1 INTRODUCCIÓN

El sistema que se va a diseñar se muestra a bloques en la figura 3.1.

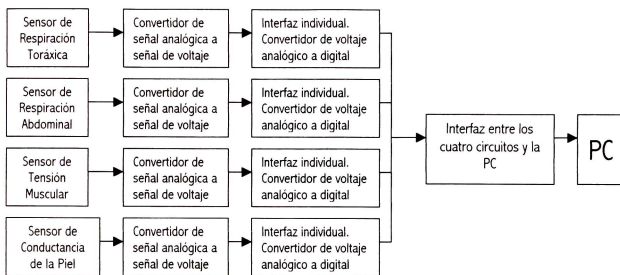


Figura 3.1. Diagrama a bloques del sistema a diseñar.

En la parte izquierda de la figura 3.1 tenemos los cuatro sensores que se utilizarán para medir los parámetros de Respiración (torácica y abdominal), Tensión Muscular y Conductancia de la Piel. Puede ser que cada uno de los sensores necesite un convertidor de una señal analógica a una señal de voltaje analógico, primero debe observarse la salida de cada sensor. Después de este convertidor está una interfaz individual que convierte un voltaje analógico a voltaje digital. El objetivo de esta interfaz individual es que cada sensor pueda conectarse a la PC para analizar sus señales sin necesidad de tener los otros sensores. De esta manera, cada sensor se puede utilizar de manera individual en otras aplicaciones. Por ejemplo, en el capítulo 1 se habló de la terapia de *Biofeedback* en donde se individualizan algunas señales del cuerpo y se permite que el paciente las vea para que observe los cambios en las señales mientras se relaja o se estresa más. Por tanto, puede usarse cada uno de los sensores para llevar a cabo la terapia.

En el penúltimo bloque de la figura 3.1 aparece una interfaz que conecta a los cuatro circuitos con la PC, esta interfaz permite que las señales puedan obtenerse en forma concurrente o, también, de forma individual dependiendo de un software.

3.2 INTERFAZ ENTRE LA PC Y UN CIRCUITO CON SALIDA DE VOLTAJE ANALÓGICA

Debido a que las salidas de los cuatro sensores, de movimiento respiratorio abdominal y torácico, de tensión muscular y conductancia de la piel van a ser analizadas por un software, es necesario diseñar un dispositivo que convierta la señal analógica de cada uno de los sensores a una señal que pueda ser leída por la computadora, una señal digital. Además, para el diseño del dispositivo, se debe tener en cuenta que se va a utilizar en puerto paralelo en forma estándar, es decir, sólo se pueden leer 4 bits o un nibble, a la vez. También es necesario contar con un software que lea y ensamble los nibbles para convertirlos a bytes, posteriormente debe interpretar el byte para arrojar una lectura de voltaje.

El circuito que convierte la señal analógica a digital para que pueda ser leída desde el puerto paralelo estándar se muestra en la figura 3.2.

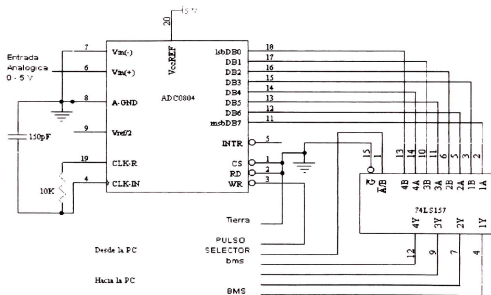


Figura 3.2. Circuito que convierte una señal analógica a digital para ser leída por el puerto paralelo.

En la sección 2.4.5 del capítulo anterior se comentaron las características del convertidor analógico digital ADC0804, ahí se observó que la frecuencia óptima necesaria para que el dispositivo funcionara era de 606 KHz y esta frecuencia se obtenía con los valores de resistencia y capacitor de 10 K Ω y 150pF respectivamente. La resistencia se conecta del pin 19 al pin 4, mientras que el capacitor va conectado del pin 4 a tierra. En el pin 6 con la leyenda $V_{(+)}$ se conecta la señal de voltaje analógica, en el rango de 0 a 5 V, del circuito a analizar, el circuito se aterriza con la tierra del dispositivo convertidor. El pin \overline{CS} , habilitación del microcircuito, se conecta al estado activo BAJO para permitir el funcionamiento de los pines \overline{RD} (habilitación de salida) y \overline{WR} (inicio de conversión). \overline{RD} también se alimenta con un estado BAJO para que las salidas digitales se habiliten. Con esta configuración se logra que el ADC siempre se encuentre habilitado y que las salidas digitales siempre estén presentes de manera que puedan ser leídas en cualquier momento.

El pin \overline{WR} se utiliza para que el ADC comience a convertir, para que esto suceda es necesario alimentarlo con un pulso de bajada de al menos 100 ns; el pulso se crea desde la PC con software y se le envía al pin 3.

Del pin 11 al 18 tenemos las salidas digitales, siendo el bit más significativo la salida 11 y el menos significativo la salida 18. Ahora, al manejar el puerto paralelo en modo estándar sólo es posible que entren 4 bits simultáneamente, por lo que es necesario separar en dos nibles al byte de salida del ADC. El primer nibble, el más significativo, está compuesto por las salidas 11, 12, 13 y 14. El nibble menos significativo lo componen los pines 15, 16, 17 y 18.

El multiplexor, 74LS157, nos va a permitir habilitar un nibble a la vez. Esto se logra manipulando la entrada A/B que es el selector. Cuando esta entrada tiene un nivel BAJO, se selecciona el nibble más significativo y cuando se tiene un nivel ALTO, el nibble menos significativo se habilita. El manejo de esta entrada también se realiza por medio de software. El pin 15 del se aterriza para habilitar al multiplexor. En las salidas 4, 7, 9 y 12 tenemos el nibble seleccionado, siendo el pin 4 el bit más significativo y el pin 12 el bit menos significativo.

La tierra de la PC se conecta con la tierra del dispositivo convertidor.

Estas entradas y salidas del circuito, que de aquí en adelante llamaremos interfaz, se conectan al puerto paralelo como se muestra en el figura 3.3.

Pin	Descripción	Dirección
8	Data 6	Salida
9	Data 7	Salida
10	ACK	Entrada
11	Busy	Entrada
12	Paper Empty	Entrada
13	Select	Entrada
18	Tierra	

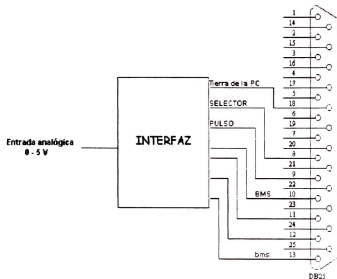


Figura 3.3. Conexiones entre la interfaz y el puerto paralelo.

El pin 9 se utiliza para enviar la señal del pulso para el ADC (\overline{WR}), obsérvese que el byte completo (figura 2.28) se tiene desde el pin 2 (bit menos significativo) hasta el pin 9 (bit más significativo). Para la selección de los nibles ($\overline{A/B}$) se utiliza en pin 8. En la entrada \overline{ACK} se tiene el bit más significativo del nibble y en la entrada \overline{Select} está el bit menos significativo del nibble. Se observa que el pin 10 del puerto está negado, por lo tanto esta entrada debe negarse por medio de software, porque de otro modo lectura sería errónea.

En la figura 3.4 se muestra el programa en Borland C++ que lee e interpreta las señales provenientes de la interfaz.

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
#define DATA 0x378 // Nota 1
#define STATUS DATA+1 // Nota 2

void pulso(void);
void pone_a_uno(void);
void pone_a_cero(void);
unsigned char bls,bms;

void main(void)
{
    int i;
    clrscr();
    pulso(); // Nota 3
    pone_a_uno(); // Nota 4
    bls=(inportb(STATUS) ^ 0x80) & 0xF0; // Nota 5
    bls = bls / 16; // Nota 6
    pone_a_cero(); // Nota 7
    bms=(inportb(STATUS) ^ 0x80) & 0xF0;
    printf("%X %X = %f\n",bms,bls,(bms+bls)*19.53e-3); // Nota 8
    getch();
}

void pulso(void) void pone_a_cero(void)
{
    outportb(DATA, 0x80); { outportb(DATA, 0x00); }
    delay(1);
    outportb(DATA, 0x00); void pone_a_uno(void)
    delay(1); { outportb(DATA, 0x40); }
    outportb(DATA, 0x80);
}

```

Figura 3.4. Programa en lenguaje C++ para enviar señales a la interfaz, leer los datos del puerto e interpretarlos.

En la nota 1 tenemos que, como ya se vio en el capítulo anterior sección 2.6.3, el puerto de datos está en la dirección $378H$. El puerto de estado que controla los pines 10, 11, 12 y 13 del puerto, está en la dirección $DATA - 1$ (Nota 2).

Lo primero que se debe hacer para que el ADC convierta es mandar el pulso a \overline{WR} para que se inicie la conversión. Veamos lo que hace la función *pulso()*: con la función *outportb()* se manda al puerto de datos el dato $80H$ o en código binario el $1000\ 0000$. Lo que está sucediendo es que mandamos un 1 al pin 9 (bit más significativo). Después con la función *delay()*, se da un espacio de tiempo de 1 milisegundo. A continuación se manda el dato $00H$ o $0000\ 0000$, lo que provoca el pulso de bajada

El valor en decimal de b/s es $128 + 64 + 32 = 224$, pero b/s es el nibble de menor peso en la salida original del ADC, por lo tanto, debemos dividir 224 por 16 para obtener el resultado correcto que sería 14 o sea $8 + 4 + 2 = 14$.

Una vez que se leyó el nibble más bajo, ahora es necesario cambiar el valor del selector del multiplexor para leer el nibble más significativo. Esto se logra con la función *pone_a_cero()* (nota 7), que manda al puerto de estado el dato 00H, de esta forma el pin 8 esta en 0. El valor leído se almacena en bms . También es necesario aplicar la operación OR exclusiva y AND para tener el resultado correcto. Supongamos que la lectura fue 0001 xxxx, el resultado final de bms sería 0001 0000 = 16. Aquí no es necesario dividir este resultado por 16 ya que el nibble leído corresponde al nibble de mayor peso en la salida original del ADC.

En la nota 8 lo que se hace es imprimir el resultado total como un valor de voltaje. Esto se hace de la manera siguiente: sumamos bms con b/s (tomando los valores de los ejemplos anteriores), esto es $16 + 14 = 30$, este resultado se multiplica por 19.53×10^{-3} que es la resolución para 8 bits ($5 / 256 = 0.01953$) del ADC. La lectura en volts sería $30 \times 19.53 \times 10^{-3} = 585.9 \text{ mV}$.

3.3 SENSOR DE MOVIMIENTO DE RESPIRACIÓN ABDOMINAL Y TORÁCICA

El movimiento que realizamos, tanto en el tórax como en el abdomen, cuando respiramos, es uno de los parámetros importantes que deben analizarse cuando se trata de medir el nivel de estrés en una persona.

El movimiento de respiración se presenta con un desplazamiento, del tórax y del abdomen, hacia arriba y hacia abajo, en un plano perpendicular a ellos.

Si fijáramos un punto de observación frontal, por ejemplo, al abdomen (figura 3.6), se vería que cuando una persona inhala aire el abdomen se inflama y se desplaza cierta distancia quedando más cerca del punto de observación. Cuando ocurre la exhalación el abdomen se contrae y lo que se ve desde el punto de observación, es que el abdomen está más lejos de él.

Por lo anterior, para poder sensar el movimiento respiratorio es necesario contar con un detector de distancias. Para nuestro caso es indispensable que el sensor arroje lecturas de variaciones de voltaje dependiendo de la distancia entre el punto de observación y el tórax o el abdomen.

Un par infrarrojo que se compone de un diodo emisor de luz infrarroja, emisor, y un fototransistor, receptor, (descritos en el capítulo 2, sección 2.2.1) que permite sensar distancias de la siguiente manera.

Siguiendo el principio básico de un interruptor óptico reflejante (Capítulo 2, sección 2.2.1) se monta el emisor y el receptor en un mismo lado y ambos dispositivos deben formar un ángulo isósceles con la superficie reflejante (en nuestro caso el abdomen o el tórax) para que se tenga un reflejo óptico.

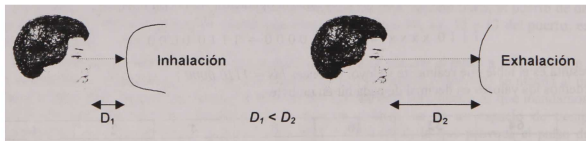


Figura 3.6. Movimiento respiratorio del abdomen observado desde un punto fijo frontal.

Un espejo es colocado sobre el abdomen de la persona como superficie reflejante.

Cuando la persona inhale el espejo reflejará más luz del emisor (la distancia entre el espejo y el par infrarrojo es más corta) y el receptor producirá más corriente. En el tiempo de la exhalación, el receptor recibirá menos luz reflejada del espejo (hay mayor distancia entre la superficie reflejante y el par infrarrojo) y producirá menos corriente. Esta corriente se hace pasar a través de una resistencia en donde se presentarán caídas de voltaje que variarán de forma dependiente con el movimiento respiratorio, véase figura 3.7.

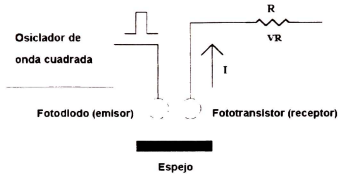


Figura 3.7. Diagrama a bloques del sensor del movimiento respiratorio utilizando un par infrarrojo.

El diseño de ambos sensores (abdomen y tórax) será el mismo.

Si se utiliza una alimentación de corriente continua para alimentar al fotodiodo, existe una potencia constante sobre él, esto puede reducir la vida útil del dispositivo. En este caso, el sensor debe estar funcionando alrededor de 15 minutos, por lo que es mejor aplicarle una señal alterna. Una señal de onda cuadrada permite que el dispositivo en un momento opere y en otro no. Por lo tanto, este hace que el fotodiodo no siempre esté prendido. Ahora, se necesita definir cuánto tiempo estará prendido el fotodiodo y cuánto estará apagado. Veamos la figura 3.8.

La figura 3.8 muestra el periodo de la onda cuadrada, es decir, el tiempo que dura la onda. El tiempo t_1 es el tiempo en que la onda produce un voltaje y en t_2 no hay voltaje. Un periodo $T = 1$ ms, nos dice que cada un milisegundo el fotodiodo prenderá y apagará, es decir, 0.5 ms estará prendido y 0.5 ms estará apagado. Un periodo $T = 1$ ms es adecuado para nuestro sensor porque, aproximadamente, el tiempo que tomamos para aspirar y exhalar es de 2 s. Por tanto, en 2 s el sensor realizará 2000 mediciones.

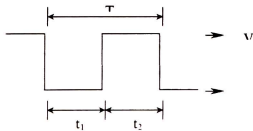


Figura 3.8. Periodo para una onda cuadrada.

A partir de estas consideraciones se va a diseñar el oscilador de onda cuadrada

3.3.1 DISEÑO DEL OSCILADOR DE ONDA CUADRADA

Para implementar el oscilador de onda cuadrada se va a utilizar el temporizador LM555 descrito en la sección 2.3.1.1 y, a partir del ejemplo mostrado en la misma sección, se procederá al diseño.

El ciclo de trabajo no puede ser del 50% porque eso significaría que la resistencia R_A fuera cero y, por lo tanto, se estaría haciendo un corto en las entradas 6 y 7 del LM555. Un ciclo de trabajo del 55% es una buena aproximación y permite que se tengan valores de capacitor y resistencias dentro de los límites permitidos en el diseño del temporizador.

Para un ciclo de trabajo del 55%, tenemos que t_2 es,

$$\text{ciclo de trabajo} = \frac{t_2}{T} \quad \therefore \quad t_2 = \text{ciclo de trabajo} * T = 0.55 * 1 \times 10^{-3} = 550 \mu\text{s}$$

y para t_1 tenemos que,

$$T = t_1 + t_2 \quad \therefore \quad t_1 = T - t_2 = 1000 \times 10^{-6} - 550 \times 10^{-6} = 450 \times 10^{-6} = 450 \mu\text{s}$$

Se propone $R_B = 10 \text{ k}\Omega$, por lo tanto,

$$t_1 = 0.693 R_B C \quad \therefore \quad C = \frac{t_1}{0.693 * R_B} = \frac{450 \times 10^{-6}}{0.693 * 10 \times 10^3} = 6.493 \times 10^{-8} = 64.93 \text{ nF}$$

Ahora encontramos el valor de R_A ,

$$t_2 = 0.693(R_A + R_B)C \quad \therefore \quad R_A = \frac{t_2}{0.693 * C} - R_B = \frac{550 \times 10^{-6}}{0.693 * 64.93 \times 10^{-9}} - 10 \times 10^3 = 2.22 \text{ k}\Omega$$

El circuito final se muestra en la figura 3.9.

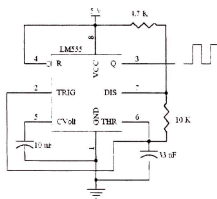


Figura 3.9. Diseño del temporizador LM555 para una frecuencia de 1000 Hz y un ciclo de trabajo del 55%.

Una vez diseñado el oscilador de onda cuadrada se procede al diseño de la parte emisora.

3.3.2 DISEÑO DEL EMISOR DE LUZ INFRARROJA

Para el diseño del emisor se utilizará el fotodiodo descrito en la sección 2.2.1.2, el TSUS520. También se utiliza la configuración básica del emisor mostrada en la figura 2.2. En este caso, para el cálculo de la resistencia se tiene que tomar en cuenta que el fotodiodo se alimentará con una señal alterna. En la figura 3.10a se muestra una gráfica que relaciona la duración del pulso de la señal y la corriente I_F .

La duración del pulso cuando existe un voltaje (t_2) es de 0.55 ms. El ciclo de trabajo fue del 55%. En la gráfica, tp equivaldría a t_2 , es decir, $tp \approx 0.5$ ms y T sería 1 ms, entonces, el ciclo de trabajo es $tp/T \approx 0.5$. Por tanto, para un $tp = 0.5$ ms y un ciclo de trabajo de 0.5 la corriente I_F es de 0.3 A. Ahora, pasemos a la gráfica 3.10b. Ahí se puede observar que para una $I_F = 0.3$ A o 300 mA corresponde un voltaje directo $V_F = 1.5$ V. En este caso, la polarización del fotodiodo se hace a través del LM555 ya que es éste el que alimenta al fotodiodo. El voltaje de salida en el pin 3 que entrega el LM555 es de 5 V.

De lo anterior tenemos que, el voltaje en la resistencia es:

$$V_R = V_{\text{salida del LM555}} - V_F = 5 - 1.5 = 3.5 \text{ V}$$

El valor de la resistencia se obtiene utilizando la ley de Ohm ,

$$R = \frac{V_R}{I_F} = \frac{3.5}{300 \times 10^{-3}} = 11.66 \Omega$$

Ahora es necesario diseñar la parte receptora.

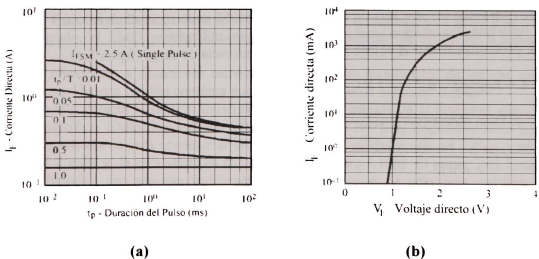


Figura 3.10. (a) Gráfica de la Duración de Pulso tp contra Corriente directa I_F ; (b) Gráfica de Voltaje Directo V_F contra Corriente Directa I_F .

3.3.3 DISEÑO DEL RECEPTOR INFRARROJO

El espejo utilizado para reflejar la luz del emisor se colocará a 4 cm del sensor. Esta distancia es el máximo desplazamiento que se registra cuando una persona realiza la respiración.

El fototransistor utilizado como receptor fue el Fototransistor NPN de Silicio BPW96 descrito en la sección 2.2.1.7.

Se desea que el fototransistor reciba una cantidad grande de luz para que los cambios en la distancia, que son de milímetros, sean significativos. En la figura 3.11 se muestra (igual que la figura 2.7) la gráfica de corriente de colector contra voltaje colector emisor.

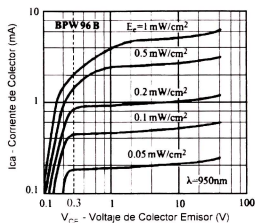


Figura 3.11. Gráfica de Corriente de Colector contra Voltaje Colector-Emisor.

Deseamos una irradiación de 2 mW/cm^2 . La curva para esa irradiación no se muestra en la figura. Ahora, el voltaje de saturación del fototransistor es de 0.3 V . Aproximadamente, para la irradiación deseada y cuando el fototransistor esté saturado se tiene una corriente de colector $I_{ca} = 10 \text{ mA}$. La polarización del fototransistor será de 5 V porque el LM555 también se alimenta con 5 V , de esta forma podemos tener una alimentación igual para los dispositivos utilizados.

De aquí, podemos calcular el voltaje que cae en la resistencia de carga (figura 2.6)

$$I R_L = V_{CC} - V_{CE} = 5 - 0.3 = 4.7 \text{ V}$$

Por la ley de Ohm obtenemos el valor de la resistencia R_L ,

$$R = \frac{I R_L}{I_{ca}} = \frac{4.7}{10^{-3}} = 470 \Omega$$

El circuito final se muestra en la figura 3.12. La salida del sensor de movimiento respiratorio se puede conectar a la interfaz, descrita en la sección 3.2, y se puede probar su funcionamiento con el programa analizado en la misma sección.

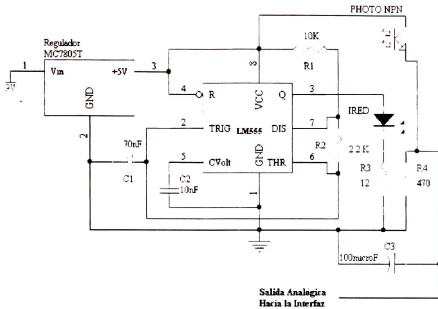


Figura 3.12. Configuración del sensor de movimiento respiratorio.

3.4 SENSOR DE TENSIÓN MUSCULAR

Otro parámetro esencial para la medición del estrés es la tensión muscular.

Los músculos realizan un amplio rango de actividades. Normalmente, se conoce la actividad muscular asociada con el movimiento, tal como ocurre al balancear una raqueta de tenis. Sin embargo, la actividad muscular produce movimientos no visibles y como resultado, pasan inadvertidos. La actividad EMG (electromyographic) es una medida de la actividad eléctrica en los músculos. Debido a que la actividad eléctrica de los músculos se incrementa cuando el músculo está tenso y se decrementa cuando está relajado, el EMG provee información acerca del estado de relajación o tensión de los músculos.

Los músculos responden a situaciones estresantes con grandes incrementos de tensión, que a veces no se pueden disipar. Si esta tensión continua puede resultar en fatiga muscular o dolor.

Un dispositivo que detecta la tensión muscular y ofrece una medida de la actividad eléctrica de los músculos es el MyoTrac EMG Biofeedback Systems (figura 3.13a). Este dispositivo tiene una entrada y dos salidas. La entrada es para conectar el sensor, una de las salidas arroja un voltaje analógico que va desde 0 hasta 2 volts y la otra salida es para audifonos (figura 3.13b). El sensor está compuesto de tres electrodos (triado), dos son electrodos activos y el tercero es de referencia, también tiene una cabeza en donde se insertan los electrodos (figura 3.14a). El sensor se coloca en un músculo en donde los electrodos activos deben estar en posición paralela a las fibras de los músculos (figura 3.14b).

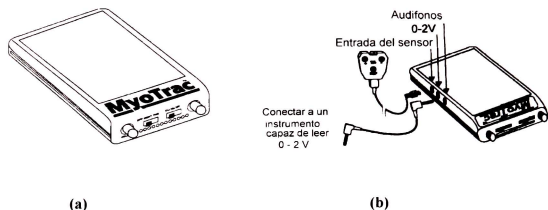


Figura 3.13. (a) Gráfica del MyoTrac EMG Biofeedback Systems; (b) Entradas y salidas del MyoTrac.

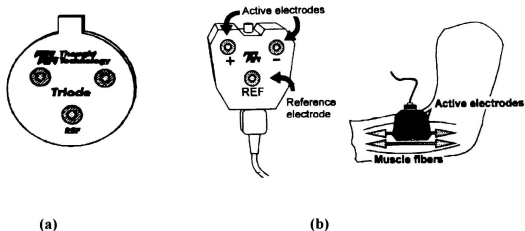


Figura 3.14. (a) Triodo; (b) Cabeza del sensor y posición del sensor en el músculo.

Para analizar la información que nos da el MyoTrac a través de la PC, era necesario tomar el voltaje analógico y convertirlo a digital. Pero con un voltaje máximo de 2 V se tenía una resolución muy pobre, por tanto, se debía amplificar ese voltaje para tener, aproximadamente, 5 V como salida máxima. Los 5 V propuestos es porque este voltaje analógico se va a aplicar al ADC. El ADC si se polariza con 5 V, entonces, como máximo, puede recibir en su entrada 5 V.

Se utilizó el amplificador operacional TL082 descrito en la sección 2.2.2.3.

Si la salida del MyoTrac es de máximo 2 V y la interfaz puede recibir hasta 5 V, entonces se propone una ganancia del amplificador de 2.5. Para este valor de ganancia y utilizando la ecuación 2.3, tenemos que,

$$Ganancia = 1 + \frac{R_f}{R_i} = 2.5$$

despejando a R_f nos queda,

$$R_f = (2.5 - 1)R_i = 1.5R_i$$

Se ve claramente que si $R_i = 1\text{ K}\Omega$, entonces $R_f = 1.5\text{ K}\Omega$.

La configuración del circuito amplificador para una ganancia de 2.5 se muestra en la figura 3.15

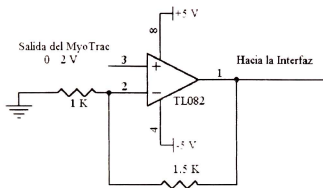


Figura 3.15. Configuración del AO TL082 para una ganancia de 2.5.

3.5 SENSOR DE CONDUCTANCIA DE LA PIEL

Un cuarto parámetro importante en la medición del estrés es la Conductancia de la Piel.

Una de los medios que tiene el cuerpo para demostrar su reacción ante la tensión o el estrés es la piel. Los científicos la llaman resistencia *Galvanic Skin Response*, **GSR** (conductancia de la piel).

El GSR es un reflejo de las variaciones de la actividad de la glándula del sudor y del tamaño del poro, ambos se controlan por el sistema nervioso simpático. Cuando una persona se excita, tiene miedo o se altera en cualquier grado, el sistema activa cambios físicos y químicos a través de todo el cuerpo. El nivel de GSR también cambia.

La resistencia de la piel se incrementa cuando la persona se encuentra en calma y relajada. Cuando la tensión aumenta, aún ligeramente, la resistencia de la piel decrece.

Un dispositivo que detecta estos cambios de resistencia en la piel es el GSR2. La descripción del dispositivo se muestra en la figura 3.16.

El GSR2 está compuesto de dos placas sensoras de acero inoxidable en donde se colocan los dedos índice y medio. Tiene una entrada para conectar un termistor o dos electrodos. Además, cuenta con una segunda entrada en donde se conectan los audífonos y un medidor de sensibilidad dual.

El sensor tiene una bocina que emite un tono. La frecuencia del tono emitido depende del estado en que se encuentra la persona. Si el tono es grave (frecuencia baja), la persona está estresada, conforme ésta se relaja el tono se va agudizando (frecuencia alta). La frecuencia se puede regular con el volumen del GSR2, este volumen tiene grabados números que van desde 1 hasta 10. El número en el cual se alcanza la frecuencia mayor es diferente para cada persona (según el manual). El manual no especifica la frecuencia en la cual se debe tomar una medición, sólo indica que el tono debe escucharse y debe ser agudo. Debido a lo anterior, se probó el GSR2 y se escogió un tono audible y agudo y, con un osciloscopio, se midió la frecuencia de ese tono. El resultado de esa prueba se muestra en la figura 3.17

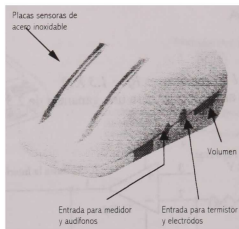


Figura 3.16. Descripción del GSR2.

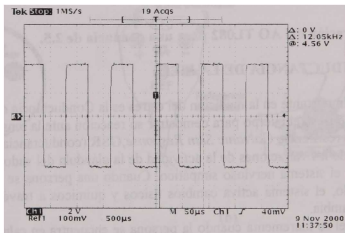


Figura 3.17. Gráfica de la frecuencia del tono, audible y agudo, emitido por el GSR2.

El osciloscopio utilizado fue un Tektronix TDS 360, con dos canales y digital.

En la parte superior derecha se puede observar que la frecuencia resultante fue de 12.05 KHz. Esta lectura se toma de la entrada para los audífonos y el medidor de sensibilidad dual. Las señales que se leen de esta entrada varían en frecuencia y tienen una amplitud (voltaje) constante.

Se propone el diseño de un circuito que detecte la frecuencia de 12 KHz (redondeando la cifra) y que avise, por medio de un led, cuando se alcanza esa frecuencia. Para el diseño de este circuito se utilizará el decodificador de tono, el LM567 cuyo funcionamiento se describió en la sección 2.2.3 del capítulo anterior. Primero se encontrará el valor de la resistencia R_1 , para esto, despejamos, como en el ejemplo de la sección 2.2.3.2, a R_1 de la ecuación 2.16. Entonces, para una frecuencia de 12 KHz y un valor de $0.1 \mu\text{F}$ para C_1 , tenemos que,

$$R_1 = \frac{1}{1.1 f_0 C_1} = \frac{1}{1.1(12 \times 10^3)(0.1 \times 10^{-6})} = 757.57 \Omega$$

El ancho de banda debe escogerse de modo que la frecuencia central no varíe mucho, por lo tanto, se propone un BW del 1% de f_0 ,

$$BW = (0.01) (12 \times 10^3) = 120 \text{ Hz}$$

De lo anterior se observa que la frecuencia central variará entre 11.94 KHz y 12.06 KHz.

Se utilizará la gráfica de la figura 3.18 para obtener los valores de los capacitores C_2 y C_3 .

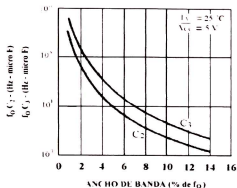


Figura 3.18. Gráfica de ancho de banda (BW) contra $f_0 C_2$ y $f_0 C_3$.

Para un BW igual al 1% de f_0 tenemos que $f_0 C_2 = 4 \times 10^5 \text{ Hz}\mu\text{F}$ y $f_0 C_3 = 7 \times 10^5 \text{ Hz}\mu\text{F}$, por lo tanto,

$$C_2 = \frac{4 \times 10^5}{12 \times 10^3} = 33.33 \mu\text{F} \quad \text{y} \quad C_3 = \frac{7 \times 10^5}{12 \times 10^3} = 58.33 \mu\text{F}$$

Para calcular la resistencia de carga, R_L , se debe tomar en cuenta que se debe adicionar el diodo (led) que permite visualizar la frecuencia central. Este led va conectado en serie con R_L . El voltaje a la salida del pin 8 es de 5 V esos 5 V deben repartirse entre R_L y el led. El diodo, para prender, necesita 1.5 V y una corriente mínima de 5 mA, entonces, en la resistencia deben caer 3.5 V y por ella debe pasar una corriente de 5 mA. El valor de R_L , por la ley de Ohm es,

$$R_L = \frac{3.5}{5 \times 10^{-3}} = 700 \Omega$$

Cuando se alcanza la frecuencia central el voltaje de salida es 0 V, por lo tanto, el led se mantendrá apagado mientras no se tenga la frecuencia central y se prenderá cuando se registre esta frecuencia.

Ahora, como la señal anteriormente estudiada varía en frecuencia pero tiene un voltaje constante se debe buscar una señal que, para la frecuencia propuesta, muestre variaciones. Dicha señal se tomó de una resistencia del sensor de GSR. El voltaje en esta resistencia varía conforme aumenta o disminuye el volumen del sensor. El voltaje mínimo es de 0 V cuando se tiene el límite inferior del volumen (1) y el voltaje es de 4 V cuando se alcanza el límite superior del volumen (10). Estas variaciones de voltaje se enviarán a la PC para que las registre y arroje el nivel parcial de estrés del parámetro GSR. Por tanto, este voltaje alimentará a la interfaz descrita en la sección 3.2 de este capítulo.

En la figura 3.19 se muestra el circuito completo para obtener el valor del parámetro GSR.

En resumen, podemos decir (según la figura 3.19) que la salida de audifonos se lleva la decodificador de tono para detectar cuando se tienen 12 KHz de frecuencia, lo que se puede visualizar a través del led. En ese momento se toma la lectura de voltaje obtenida de la resistencia del sensor de GSR y ese voltaje corresponderá al nivel parcial de estrés correspondiente al parámetro GSR.

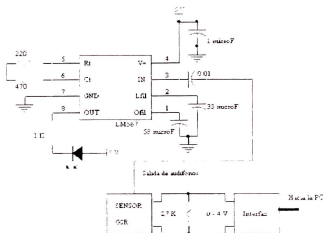


Figura 3.19. Diagrama del circuito para la medición del parámetro GSR.

3.6 INTERFAZ ENTRE LOS CUATRO SENSORES Y LA PC

La interfaz descrita en la primera sección de este capítulo funciona para poder analizar, a través de la PC, un sólo circuito a la vez. El sistema completo está compuesto de cuatro sensores, por lo que es necesario diseñar un dispositivo que permita la entrada de las cuatro señales de manera “concurrente”. En la realidad, no entran las cuatro señales al mismo tiempo, lo hacen en forma secuencial, pero al momento de visualizar las lecturas de voltaje en la pantalla, da la impresión de que la PC lee, en un mismo instante, todas las señales.

Como ya se había visto, un multiplexor permite la selección de una señal entre varias señales. También se había comentado que, si el puerto paralelo de la PC se configura en modo estándar, solamente se puede un nibble a la vez. Entonces, se requiere diseñar un multiplexor que tenga 16 entradas (4 por cada sensor) y cuatro salidas. Este tipo de multiplexores no se encuentra en forma comercial, por lo que se procede a su diseño.

En la sección 3.2 se utilizó el multiplexor 74LS157, este dispositivo tiene 8 entradas y cuatro salidas. Si juntamos dos 74LS157, entonces ya tenemos 16 entradas y 8 salidas. A través de un arreglo de compuertas se pueden reducir esas ocho salidas a sólo cuatro. Y con diferentes combinaciones en las dos entradas de Selección y Habilitación de ambos multiplexores, se logra la selección de un circuito a la vez. En la figura 3.20 se muestra el diagrama del circuito completo.

En la figura 3.20 se observa que al multiplexor de la parte de arriba (MUX2) se conectan los sensores: GSR (conductancia de la piel) y RESP1 (movimiento de respiración abdominal) y el multiplexor de abajo (MUX1) tiene como entradas a EMG (tensión muscular) y RESP2 (movimiento respiratorio torácico). Las cuatro salidas de MUX1 se unen con las salidas de MUX2 a través de compuertas OR. Cuando uno de los MUX está deshabilitado en sus salidas existe un nivel BAJO (ver

figura 2.19c), cualquier nivel que se tenga en las salidas del otro MUX aparecerá en las salidas de las compuertas OR, D1, D2, D3 y D4.

La forma de realizar la selección de cada uno de los sensores se describe a continuación. Primero se debe elegir cuál de los MUX estará habilitado. Si $E1 = 0$ (los MUX se habilitan con niveles bajos) y $E2 = 1$ se habilitará al MUX1 y MUX2 estará deshabilitado, entonces se manipula la entrada del Selector1 para elegir entre EMG y RESP2. Si hacemos $E1 = 1$ y $E2 = 0$, entonces se elige a EMG o RESP1 a través de Selector2. De hecho, se pueden unir las entradas de los selectores de ambos MUX, es decir,

$$\text{Selector1} = \text{Selector2}$$

Se pueden tener las mismas entradas en los selectores porque solamente uno de los MUX estará habilitado a la vez.

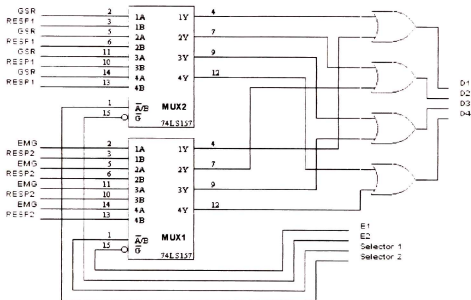


Figura 3.20. Diagrama de la interfaz entre la PC y los cuatro sensores.

3.7 DIAGRAMA GENERAL DEL SISTEMA

En la figura 3.21 se muestra el diagrama general del sistema. Se pueden observar los circuitos que convierten a voltaje analógico la señal de cada uno de los aparatos sensores. Después se tiene la interfaz con entradas analógicas y salidas digitales. Posteriormente, se observa la segunda interfaz, que conecta a los cuatro sensores con la PC. Y por último, están las conexiones hacia el puerto paralelo. Tenemos cuatro datos (D0 – D3), el pulso que permite que los cuatro ADC comiencen a convertir, S1 que es el selector para el multiplexor de la primera interfaz, S2 (se hizo Selector 1 = Selector 2) es el selector de la segunda interfaz y E1 y E2 las habilitaciones de los multiplexores de la segunda interfaz.

DESCRIPCIÓN DEL SOFTWARE

4.1 PROGRAMA DE CONTROL

En general, el programa permite almacenar datos personales de los pacientes, navegar a través de los datos, modificarlos, realizar búsquedas, salvar los datos en disco y recuperarlos. También, permite la realización de las pruebas para medir el nivel de estrés en una persona, éstas pruebas son: la tensión muscular, la conductancia de la piel, la respiración, el cuestionario MSP (medida del estrés psicológico), la postura y los exámenes de sangre que son el cortisol y la prolactina. Las tres primeras pruebas se hacen a través de sensores cuya señal de salida se introduce por el puerto paralelo y es analizada por el programa. Las preguntas del cuestionario y de la prueba de la postura están dentro del programa. En el caso de los exámenes de sangre, los valores resultantes se insertan en el programa para que éste los compute.

Microsoft Visual C++ es un entorno de desarrollo diseñado especialmente para crear aplicaciones gráficas orientadas a objetos. Para crear una aplicación se crean ventanas y sobre ellas se dibujan controles (etiquetas, botones, cajas de texto, listas desplegables, etc) y a continuación se escribe el código fuente relacionado con cada objeto. Esto es, cada objeto está ligado a un código que permanece inactivo hasta que se dé el suceso que lo activa (por ejemplo, un clic del ratón).

Se utilizó Visual C++ para desarrollar el programa de control debido a las bondades que brinda la programación orientada a objetos, además de la facilidad para obtener un programa con ambiente Windows.

A continuación se muestran cada una de las pantallas del programa y se proporciona una explicación detallada de las opciones que se presentan.

PANTALLA INICIAL

En esta primera pantalla, que se muestra en la figura 4.1, se tiene el nombre del programa, el nombre de los autores y dos botones, uno con la opción de entrar al programa y el otro para salir.

PANTALLA DEL MENÚ PRINCIPAL

Una vez que en la pantalla Inicial se eligió la opción de entrar al programa, aparece esta pantalla que es el menú principal del programa. Esta pantalla se muestra en la figura 4.2.

Existe un archivo llamado *Archivo.mie* que es donde se guardan los datos de los pacientes. Este archivo se carga automáticamente cuando aparece la pantalla del **Menú Principal** y se guarda cuando se sale del programa. De esta manera, el usuario no tiene buscar el archivo ni tampoco debe recordar el nombre de éste.

La pantalla está organizada por bloques y cada bloque tiene un nombre. Veamos en detalle cada uno de los bloques.

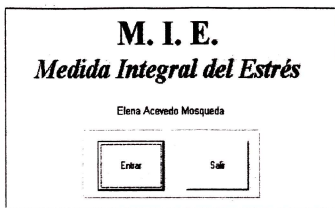


Figura 4.1. Pantalla Inicial.

Bloque Acciones con Datos Personales

Si un control (botón, caja de texto o lista desplegable) está deshabilitado no se puede realizar ninguna acción sobre él. Por ejemplo, si un botón está deshabilitado, se puede hacer clic con el ratón sobre el botón, pero éste no hará nada. Si una caja de texto está deshabilitada, entonces no se puede escribir ningún texto en ella. Si una lista desplegable se encuentra en este mismo estado no será posible desplegar su contenido.

En este bloque, el único botón que está habilitado es el de *Paciente Nuevo*. Como aún no existen datos guardados entonces no se puede modificar algún campo del registro actual. Cuando se hace clic en el botón de *Paciente Nuevo*, la pantalla tiene la vista que se muestra en la figura 4.3.

En esta figura, se observa que las cajas de texto de los datos se habilitan y entonces es posible introducir los datos. Además, todos los botones se deshabilitan a excepción de dos que son *Registra Datos* y *Cancelar*, más adelante se detallará la acción de estos botones. Los botones son deshabilitados para evitar que el usuario, por error, oprima cualquier otro botón y entorpezca el buen funcionamiento del programa.

Los datos de las cajas de texto con título *Nombre*, *Apellido Paterno*, *Edad*, *Ocupación* y *Motivo* se introducen desde el teclado. Las listas desplegables y sus opciones correspondientes a *Sexo*, *Estado Civil* y *Título de Estudio* se muestran en la figura 4.4. Para seleccionar alguna de las opciones de la lista se utiliza el ratón para hacer clic en la flecha hacia abajo que tiene la caja y entonces se despliegan las opciones de datos, a continuación se ilumina el dato requerido manteniendo oprimido el botón izquierdo del ratón y una vez seleccionada la opción se deja de oprimir el botón, entonces, el dato aparecerá en la caja.

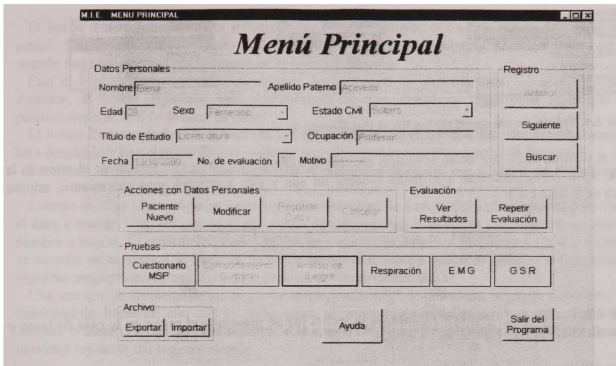


Figura 4.2. Pantalla del Menú Principal.

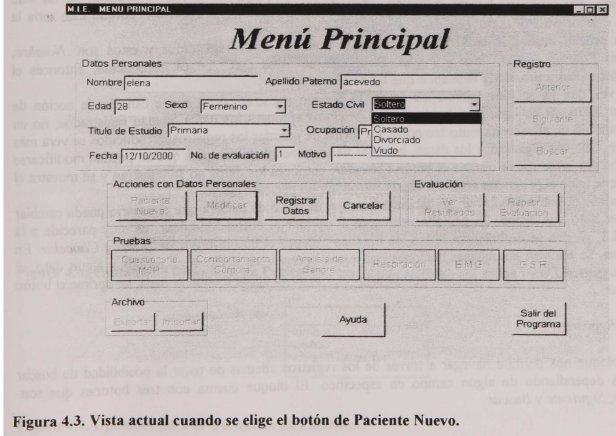


Figura 4.3. Vista actual cuando se elige el botón de Paciente Nuevo.

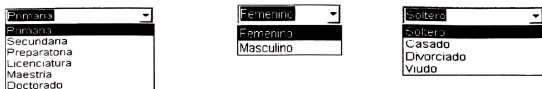


Figura 4.4. Selección de datos de las listas desplegables

Para el dato de la *Fecha*, no es necesario escribir la fecha actual, basta con posicionar el ratón en la caja de texto correspondiente a la fecha y hacer un clic y ésta aparecerá automáticamente, esto se muestra en la figura 4.5.

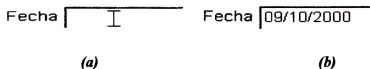


Figura 4.5. Forma de introducir el dato de la fecha. (a) Se posiciona el ratón en la caja de texto y se hace un clic. (b) Aparece la fecha actual.

Y por último, para el dato de *No. de Evaluación*, el programa busca, a partir del *Nombre, Apellido Paterno y Edad*, en todos los registros si ese paciente ya existe y cuantas veces aparece. Si el paciente nunca ha sido registrado, entonces el *No. de Evaluación* es igual a 1, si el paciente ya ha sido registrado, por ejemplo, dos veces, entonces el *No. de Evaluación* es igual a 3 porque ésta sería la tercera evaluación. Por tanto, este dato se registra automáticamente.

Cabe señalar, que existen 5 datos importantes que deben ser registrados y estos son: *Nombre, Apellido Paterno, Edad, Sexo y Fecha*. Si alguno de estos datos no es introducido, entonces el programa muestra una alerta indicando que falta un dato.

Una vez que ya se introdujeron los datos, se pueden registrar o se puede cancelar la acción de registrar datos. Si se oprime el botón *Registrar Datos* entonces los datos quedan registrados, no en disco, sino en un almacenamiento temporal. La forma de guardar los registros introducidos se verá más adelante. Una vez registrados los datos, las cajas de texto se inhabilitan para que no puedan modificarse los datos por error. Si se oprime el botón *Cancelar*, entonces los datos no se registran y se muestra el estado que tenía el programa antes de querer introducir un paciente nuevo.

Al oprimir el botón *Modificar*, se habilitan las cajas de texto de manera que el usuario pueda cambiar cualquier dato que desee. La pantalla resultante al oprimir el botón *Modificar* es muy parecida a la figura 4.3, ya que se deshabilitan todos los botones excepto el botón *Registrar Datos* y *Cancelar*. En este caso, se realizan las mismas acciones con estos botones, es decir, si se modifica algún dato, el botón *Registrar Datos* aplica los cambios realizados y si no se desea modificar nada, se oprime el botón *Cancelar*.

Bloque Registro

Este bloque nos permite navegar a través de los registros además de tener la posibilidad de buscar registros dependiendo de algún campo en específico. El bloque cuenta con tres botones que son: *Anterior, Siguiente y Buscar*.

El botón *Anterior* nos permite movernos hacia atrás pasando por todos los registros anteriores al actual. Si sólo existe un registro, este botón se encuentra deshabilitado. También toma este estado cuando nos encontramos posicionados en el primer registro.

Con el botón *Siguiente* podemos navegar hacia delante de los registros. Al igual que el botón *Anterior*, el botón *Siguiente* está deshabilitado si sólo existe un registro al igual que si nos posicionamos al final de la base de datos.

El botón *Buscar* nos lleva a la pantalla que se muestra en la figura 4.6. En esta pantalla tenemos una lista desplegable con el título *Buscar paciente por* y las opciones se muestran en la figura 4.7.

Entonces, es posible buscar pacientes por nombre, por apellido, por número de evaluación y por fecha.

Cuando se elige alguna opción aparecen diferentes cajas de texto o listas desplegables para introducir el dato a buscar. Una vez que se eligió alguna opción, por ejemplo buscar por Nombre, se introduce el nombre a buscar. El nombre se puede escribir en mayúsculas, minúsculas o ambas a la vez. El resultado se muestra en la figura 4.8. Después de haber escrito el nombre a buscar se pulsa el botón *Buscar* y los registros encontrados se muestran en la pantalla de fondo blanco (lista).

Una vez que se desplegaron los registros correspondientes a la búsqueda, se puede elegir uno de ellos iluminándolo haciendo clic con el ratón sobre el registro, a continuación se puede pulsar el botón *Aceptar* o hacer dos clic sobre el registro, esta acción nos regresará a la pantalla del **Menú Principal** y mostrará los datos del registro elegido.

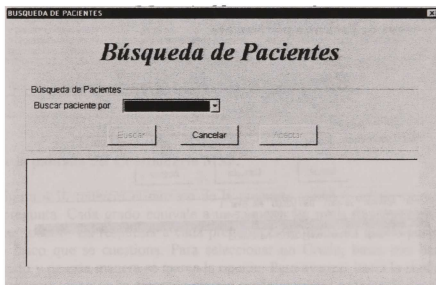


Figura 4.6. Pantalla de Búsqueda de Pacientes.

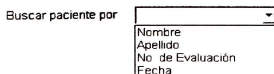


Figura 4.7. Opciones para la Búsqueda de Pacientes.

Para la búsqueda por *Fecha*, la pantalla tiene la vista que se muestra en la figura 4.9. En este caso, hay tres listas desplegables correspondientes al día, mes y año. Una vez más, ya teniendo la fecha, se pulsa el botón *Buscar* y se despliegan los resultados.

Búsqueda de Pacientes

Búsqueda de Pacientes

Buscar paciente por

Elena	Acevedo	28 años	03/11/2000	No Eval.	1
-------	---------	---------	------------	----------	---

Figura 4.8. Búsqueda de Pacientes por Nombre.

Búsqueda de Pacientes

Búsqueda de Pacientes

Buscar paciente por

Elena	Acevedo	28 años	03/11/2000	No Eval.	1
Sucara	Mosqueda	3 años	03/11/2000	No Eval.	1
Lasky	Lascurain	14 años	03/11/2000	No Eval.	1
Arizona	Perez	6 años	03/11/2000	No Eval.	1

Figura 4.9. Búsqueda de pacientes por la Fecha en que se realizó el examen.

Si no se desea buscar ningún registro se oprime el botón *Cancelar* y automáticamente se regresa a la pantalla del **Menú Principal**.

Bloque Pruebas

En este bloque se tienen seis botones que son: *Cuestionario MSP*, *Comportamiento Corporal*, *Análisis de Sangre*, *Respiración*, *EMG* y *GSR*. Cada uno de estos botones representa la prueba que se puede realizar. Comencemos observando la pantalla que aparece al pulsar el botón *Cuestionario MSP* que se muestra en la figura 4.10

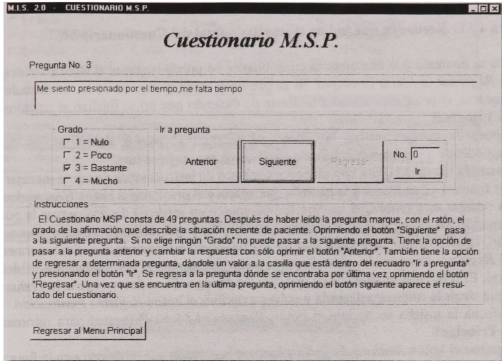


Figura 4.10. Vista de la pantalla del Cuestionario MSP.

La pantalla de la figura 4.10 muestra el número de la pregunta actual y debajo de este número se encuentra escrita la pregunta. Cada grado equivale a un número. La suma de estas cantidades será el resultado final del cuestionario. La respuesta a cada pregunta es el grado en que la persona padece el estado psicológico o físico que se cuestiona. Para seleccionar un Grado, basta con hacer clic en el cuadro del grado elegido y de esta manera se marca la opción. Para avanzar hacia la siguiente pregunta se oprime el botón *Siguiente*, si no se ha marcado ningún grado entonces aparecerá una alerta diciendo que no se ha contestado la pregunta. Si se desea cambiar la respuesta de alguna de las preguntas anteriores se puede oprimir el botón *Anterior* o si, por ejemplo, estamos en la penúltima pregunta y se quiere ir a la pregunta número 2, tenemos que escribir el número 2 en el cuadro de texto con título *No.* y después oprimir el botón *Ir* y esto nos lleva a la segunda pregunta. Cuando se quiera regresar a la pregunta actual es posible hacerlo con el botón *Siguiente* oprimiéndolo hasta llegar a dicha pregunta, la otra manera de hacerlo es presionar el botón *Regresar* y esto nos llevará a la pregunta actual.

Al alcanzar la última pregunta, después de contestarla, aparece un recuadro o caja de diálogo con la calificación final, esto se muestra en la figura 4.11.

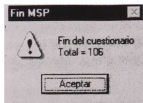


Figura 4.11. Recuadro que indica el resultado final del Cuestionario MSP.

Si ya se contestó por completo el cuestionario, se puede regresar al **Menú Principal** presionando el botón *Regresar al Menú Principal*. Si se presionó este botón y no se ha contestado completamente el cuestionario, el programa emitirá una alerta de diciendo que no se finalizó el cuestionario y retornará al **Menú Principal**.

Después de que se realizó la prueba, al regresar al **Menú Principal**, se deshabilita el botón correspondiente a esa prueba indicando que la evaluación ya se hizo.

La pantalla también cuenta con un breve conjunto de instrucciones de cómo manejar el cuestionario.

En la figura 4.12 se muestra la pantalla que aparece al presionar el botón *Comportamiento Corporal*.

Para realizar esta prueba, el usuario (comúnmente un médico) debe observar la postura del paciente, sus movimientos corporales y la forma de su voz. Una vez realizado esto, se marca, haciendo clic con el ratón, la calificación para cada uno de los parámetros mencionados. Posteriormente, se presiona el botón *Registra Puntuación* para llevar a cabo la suma de las calificaciones del comportamiento corporal. El resultado se muestra en una caja de diálogo. Si se desea repetir la evaluación, solamente se tiene que elegir la respuesta deseada y volver a oprimir el botón *Registra Puntuación*.

Satisfecha la prueba se oprime el botón *Regresa el Menú Principal* para retornar a la pantalla del **Menú Principal**.

Al oprimir el botón *Análisis de Sangre* aparece la pantalla que se muestra en la figura 4.13.

Después de realizar (en un laboratorio) las pruebas de sangre, se introducen los niveles de Cortisol y Prolactina resultantes de los estudios. Al presionar el botón *Registrar Datos* se guardan los datos y se regresa al **Menú Principal**.

La pantalla de la prueba de la respiración torácica y abdominal aparece cuando se presiona el botón *Respiración* y se muestra en la figura 4.14. En esta figura aparecen dos gráficas correspondientes a los movimientos de respiración torácico y abdominal. Las gráficas tienen los límites mínimo de 0 V y máximo de 5 V, ambas gráficas se dividen en 5 partes por lo que cada fracción representa 1 V. En esta pantalla, también se muestra una breve ayuda de cómo comenzar la prueba. Cuenta con dos botones que nos permiten iniciar la prueba y en un momento dado, sin finalizar correctamente la prueba, terminarla. Con el botón localizado en la parte baja del lado izquierdo, retornamos al **Menú Principal**.

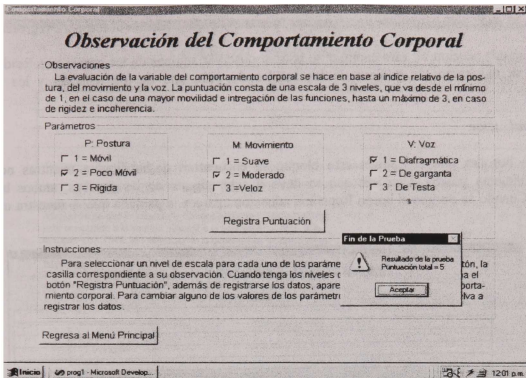


Figura 4.12. Pantalla de la prueba Comportamiento Corporal.

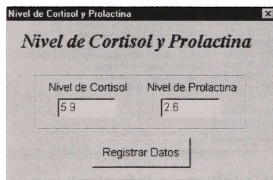


Figura 4.13. Vista de la pantalla de los exámenes de sangre para ver el nivel de Cortisol y Prolactina.

Al oprimir el botón EMG podemos acceder a la pantalla de la prueba de tensión muscular (EMG), esta pantalla se muestra en la figura 4.15. La gráfica que se muestra en esta pantalla también está dividida en 5 partes cada una con un valor de 1 V, el voltaje menor que se puede graficar es 0 V y el mayor es de 5 V. Cuenta con una breve explicación del modo de uso para llevar a cabo la prueba de EMG. Tiene la opción de cancelar la prueba en cualquier momento.

La figura 4.16 muestra la pantalla que aparece cuando se oprime el botón GSR. Esta pantalla corresponde a la prueba de conductancia de la piel (GSR). Al igual que en las dos pantallas anteriores, esta pantalla cuenta con una gráfica dividida en 5 partes con un valor de 1 V cada una. Permite cancelar

la prueba en cualquier momento, tiene una ayuda breve y cuenta con un botón para regresar al **Menú Principal**.

En la segunda sección de este capítulo se podrán observar algunas de las pruebas realizadas con los sensores, se mostrarán las gráficas de las señales de cada uno de los sensores y los resultados obtenidos.

Bloque Evaluación

Los dos botones que componen este bloque se encuentran deshabilitados mientras no se haya realizado ninguna prueba. Una vez que se llevó a cabo alguna de las pruebas, ambos botones se habilitan. Cuando se oprime el botón Repetir Evaluación aparece la pantalla que se muestra en la figura 4.17.

Prueba de Respiración

Respiración Torácica y Abdominal

Torácica

Abdominal

Instrucciones
Energice los dos circuitos sensores de respiración y revise que estén conectados a la interfaz. Coloque un espejo en el abdomen del paciente y otro en el pecho. Posicione los circuitos de modo que el sensor de luz, esté perpendicular al espejo. La distancia óptima entre el sensor y el espejo se alcanza cuando, al aspirar el paciente, se registra la máxima lectura de voltaje (4V aproximadamente). Inicie la prueba, asegure la distancia óptima y reinicie la prueba.

Iniciar Prueba

Cancelar Prueba

Regresar al Menú Principal

Figura 4.14. Vista de la pantalla de la prueba de Respiración.

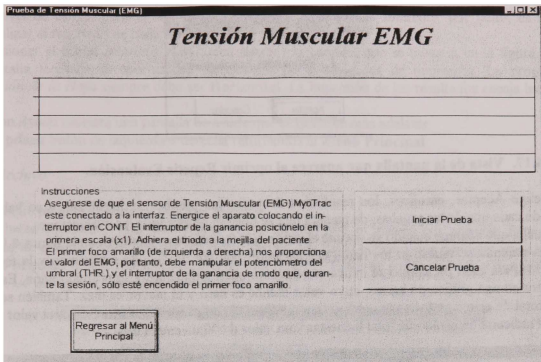


Figura 4.15. Pantalla de la prueba de la Tensión Muscular (EMG).

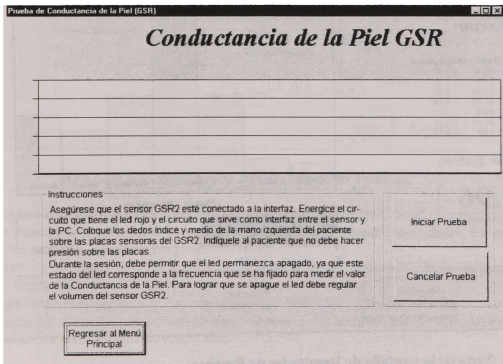


Figura 4.16. Vista de la pantalla de prueba de la Conductancia de la Piel (GSR).

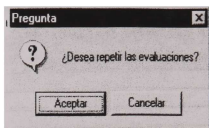


Figura 4.17. Vista de la pantalla que aparece al oprimir Repetir Evaluación.

Si se elige Aceptar, entonces, los resultados de todas las pruebas se inicializan y se habilitan los botones de cada una de las pruebas, de manera que se puedan repetir toda las evaluaciones.

La pantalla que aparece cuando se oprime el botón Ver Resultados se muestra en la figura 4.18.

En esta pantalla se muestran los datos del paciente: nombre, el No. de evaluación y la fecha de la evaluación. Para cada parámetro se indica su resultado correspondiente y su normalización. Esta última se presenta en una gráfica de barras, cuyo valor menor es cero y el mayor es diez. También se muestra el valor total de estrés en forma numérica y en una barra de color rojo. Para este caso, el valor menor es 0, lo cual indicaría un estado de total bienestar y un valor de 70 un estrés total.

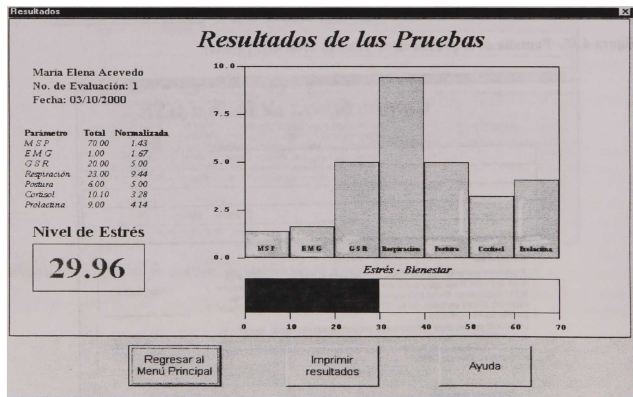


Figura 4.18. Vista de la pantalla de Resultados de Pruebas.

El tener un nivel de estrés individual para cada parámetro auxilia al médico para brindar el mejor tratamiento que corresponda al grado de cada una de las pruebas, ya que dos pacientes pueden tener un

mismo nivel de estrés, pero el resultado de las pruebas puede ser diferente, por tanto, es necesario individualizar el resultado de cada parámetro.

Al presionar el botón *Imprimir Resultados* aparece la pantalla que se muestra en la figura 4.19. En esta pantalla tenemos la oportunidad de cambiar las propiedades de impresión. La propiedad de *Orientación de la Hoja* siempre debe ser **Horizontal**. La impresión de los resultados encaja bien en una sola hoja.

El botón *Ayuda* muestra una pantalla de ayuda que se describe más adelante.

Con el primer botón de izquierda a derecha retornamos al **Menú Principal**.

Bloque Archivo

Este bloque permite importar y exportar archivos.

Ya se había comentado que *Archivo.mie* es el archivo que se maneja por omisión. Pero existe la posibilidad de que se deseen guardar, en disco flexible, los datos actuales que contiene este archivo, ya sea como respaldo o para transportar estos datos a otra PC. Estos se puede realizar al oprimir el botón *Exportar*. Lo que aparece se muestra en la figura 4.20.

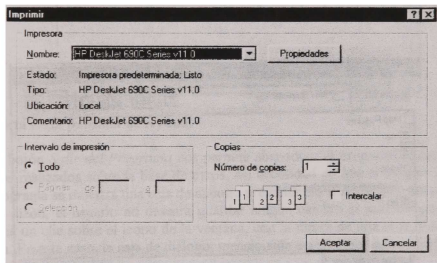


Figura 4.19. Pantalla para la impresión de los Resultados de las Pruebas.

Se puede observar que la figura es igual a la caja de diálogo “Guardar Como” que presenta Windows. Por tanto, se pueden salvar los datos en cualquier disco o subdirectorio que se desee. La extensión del archivo siempre será *mie*.

Si se quieren cargar datos de otro archivo se oprime el botón *Importar* y la pantalla resultante se muestra en la figura 4.21.

Esta pantalla también es igual a la caja de diálogo “Abrir” que presenta Windows. Así, es posible navegar a través del disco duro o buscar en algún otro disco el archivo con la información requerida. Sólo se muestran los archivos con extensión *mie*.

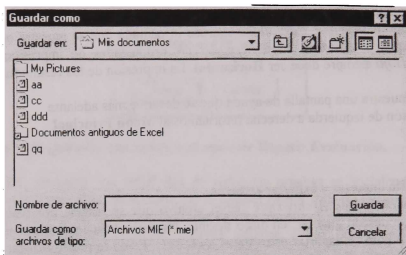


Figura 4.20. Pantalla para guardar los registros de la base de datos actual en otro espacio.

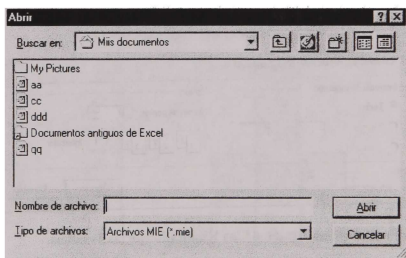


Figura 4.21. Vista de la pantalla que permite abrir un archivo con extensión *mie*.

Al presionar el botón Ayuda aparece la pantalla que se muestra en la figura 4.22.

Al hacer doble clic sobre la cabecera Ayuda de las Pantallas se despliegan las diferentes opciones de la ayuda, que consisten en la explicación de cada una de las pantallas. Haciendo doble clic en cualquiera de las opciones se muestra el texto de auxilio correspondiente a la opción pantalla elegida.

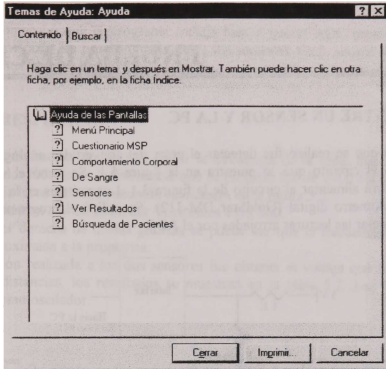


Figura 4.22. Vista de la pantalla de Ayuda.

Por último, el botón *Salir del Programa* nos permite abandonar el programa. Además, antes de salir, salva los cambios realizados sobre la base de datos, sin necesidad de que el usuario los salve. Antes de abandonar el programa se muestra una caja de diálogo preguntando si desea salir del programa.

Si por alguna razón, el usuario no deseara guardar ningún cambio al salir del programa, es posible hacerlo si se hace un clic sobre el icono de la ventana, con la figura de una cruz, localizado en la parte superior derecha. En este caso, la caja de diálogo, mencionada anteriormente, no aparece.

PRUEBA DE CIRCUITOS

5.1 INTERFAZ ENTRE UN SENSOR Y LA PC

La primera prueba que se realizó fue detectar el error del convertidor analógico-digital ADC0804. Para esto, se utilizó el circuito que se muestra en la figura 5.1. Se tomó el voltaje que caía en la resistencia variable para alimentar al circuito de la figura 3.1. Los voltajes en la resistencia variable se midieron con un multímetro digital (GoldStar DM-312). Se utilizó el programa en Borland C de la figura 3.3 para interpretar las lecturas arrojadas por el ADC.

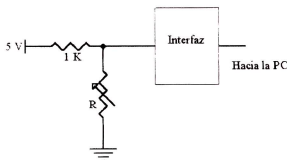


Figura 5.1. Circuito utilizado para medir el error del ADC0804.

En la tabla 5.1 se muestran los resultados de las mediciones realizadas.

Tabla 5.1. Voltajes medidos con multímetro y las lecturas del ADC interpretadas por la PC.

Valor de Resistencia Ohms	Voltaje Medido (multímetro) Volts	Valor Interpretado (PC) Volts	Error Volts
10 K	4.54	4.557	0.017
1.5 K	2.965	2.979	0.014
1 K	2.483	2.489	0.006
470	1.554	1.568	0.014
330	1.204	1.215	0.011
220	0.886	0.882	0.004
100	0.435	0.431	0.004
47	0.216	0.216	0.0

El error del ADC0804 es de $\pm 20\text{mV}$ sin ajustar, es decir con el pin V_{REF2} sin conexión^[22]. De la tabla podemos observar que el error mayor medido fue de 17mV. Además, también podemos notar que para

voltajes pequeños el error es menor. Esta prueba no sólo se hizo para detectar el error del ADC sino, también, para ver el buen funcionamiento del programa.

Una vez que se comprobó que el programa trabaja bien y que el ADC presenta un muy aceptable error, entonces se mostrarán las pruebas realizadas a los sensores. Cada sensor se probó por separado utilizando la misma interfaz para todos.

5.2 SENSOR DE RESPIRACIÓN

Para el sensor de respiración primero se midió la frecuencia de la señal que arroja el oscilador de onda cuadrada. El valor teórico de la frecuencia que se propuso fue de 1400 Hz. La señal resultante de los dos circuitos osciladores de onda cuadrada se muestra en la figura 5.2.

Las señales se obtuvieron con un oscilador Tektronics TDS 360.

En la parte superior derecha de ambas gráficas se puede ver que la frecuencia de los osciladores es de 1493 Hz, muy aproximada a la propuesta.

La segunda medición realizada a los dos sensores fue obtener el voltaje que resulta al posicionar el espejo a diferentes distancias, los resultados se muestran en la tabla 5.2. Las lecturas de voltaje se obtuvieron con el mismo oscilador.

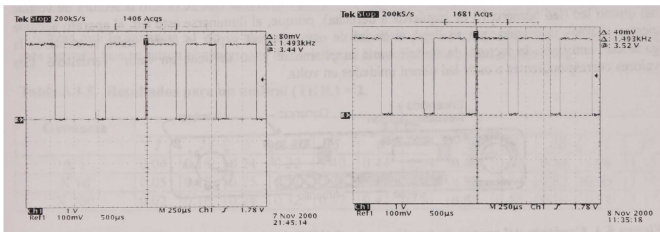


Figura 5.2. Señal del oscilador de onda cuadrada para ambos sensores.

Además, las lecturas de voltaje se obtuvieron para poder posicionar el par infrarrojo de manera que el fototransistor (receptor) recibiera el máximo reflejo de la luz emitida por el fotodiodo. Para realizar esta prueba, se trató de colocar el espejo perpendicular al plano de emisión de la luz y al plano de una mesa. El par infrarrojo quedó bien posicionado una vez que se registró un voltaje máximo. El valor de este voltaje es el que se muestra en la primera fila de la tabla 4.2.

Tabla 5.2. Resultados de las pruebas realizadas a los sensores de Respiración.

SENSOR 1	
Distancia (cm)	Voltaje Medido (V)
0.5	3.58
1.0	3.24
1.5	2.64
2.0	1.68
2.5	1.08
3.0	0.840
3.5	0.640
4.0	0.360

SENSOR 2		
Distancia (cm)	Voltaje (V)	Medido
0.5		4.12
1.0		3.96
1.5		3.02
2.0		2.68
2.5		1.84
3.0		1.4
3.5		1.2
4.0		0.880

5.3 SENSOR DE TENSIÓN MUSCULAR (EMG)

La prueba realizada sobre este sensor fue medir, a la salida del amplificador operacional TL082, las variaciones de voltaje correspondientes a cada uno de los leds que aparecen en la carátula del sensor que se muestra en la figura 5.3. Las mediciones se hicieron con el multímetro GoldStar DM-312. Los resultados se muestran en el conjunto de tablas 5.3. Cabe mencionar que se omitió la lectura de voltaje del último led (led número 13 de izquierda a derecha), porque, al iluminarse este led, el aparato acciona ciertas alarmas indicando que se llegó al límite de esa ganancia y por lo tanto, debe aumentarse la ganancia; entonces, la lectura de voltaje varía ampliamente y no se tiene un valor verdadero. Los valores correspondientes a cada led tienen unidades en volts.

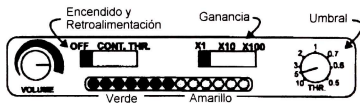


Figura 5.3. Carátula del sensor de tensión muscular (EMG) MyoTrac.

Tabla 5.3.1. Resultados para un umbral (*threshold*, THR.) = 0.5.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 10	0.04	0.06	0.08	0.10	0.11	0.13	0.15	0.17	0.21	0.24	0.28	0.33
X 100	0.03	0.06	0.08	0.09	0.12	0.14	0.14	0.18	0.20	0.25	0.29	0.33

Tabla 5.3.2. Resultados para un umbral (THR.) = 0.6.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 10	0.04	0.07	0.09	0.10	0.12	0.14	0.16	0.20	0.23	0.25	0.31	0.35
X 100	0.02	0.07	0.08	0.10	0.13	0.14	0.16	0.19	0.22	0.26	0.31	0.34

Tabla 5.3.3. Resultados para un umbral (THR.) = 0.7.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 10	0.04	0.08	0.10	0.12	0.15	0.18	0.20	0.23	0.26	0.32	0.38	0.44
X 100	0.03	0.08	0.10	0.12	0.15	0.18	0.19	0.24	0.25	0.34	0.38	0.46

Tabla 5.3.4. Resultados para un umbral (THR.) = 1.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 10	0.05	0.11	0.15	0.18	0.22	0.24	0.28	0.33	0.38	0.45	0.55	0.66
X 100	0.03	0.11	0.14	0.17	0.22	0.23	0.26	0.30	0.41	0.47	0.56	0.69

Nótese que la ganancia x1 no se muestra, esto es debido a que, es tanta la sensibilidad del sensor en esta entrada que se satura el aparato, es decir, la única lectura posible es la del último led.

Tabla 5.3.5. Resultados para un umbral (THR.) = 2.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 1	0.04	0.18	0.24	0.32	0.40	0.44	0.54	0.60	0.76	0.83	1.08	1.15
X 10	0.05	0.18	0.25	0.31	0.38	0.44	0.50	0.58	0.64	0.82	0.95	1.18
X 100	0.02	0.16	0.24	0.32	0.37	0.42	0.53	0.57	0.65	0.84	0.98	1.17

Tabla 5.3.6. Resultados para un umbral (THR.) = 3.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 1	0.04	0.35	0.44	0.59	0.66	0.82	0.90	1.07	1.23	1.39	1.64	2.04
X 10	0.05	0.34	0.45	0.55	0.65	0.80	0.91	1.09	1.20	1.40	1.62	2.08
X 100	0.02	0.34	0.44	0.57	0.64	0.79	0.89	1.07	1.20	1.38	1.63	2.07

Tabla 5.3.7. Resultados para un umbral (THR.) = 5.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 1	0.44	0.93	1.20	1.48	1.83	2.12	2.42	2.78	3.26	3.97	4.23	4.27
X 10	0.44	0.90	1.19	1.46	1.87	2.10	2.41	2.89	3.22	3.83	4.22	4.25

Tabla 5.3.8. Resultados para un umbral (THR.) = 10.

Ganancia	LEDS											
	1	2	3	4	5	6	7	8	9	10	11	12
X 1	0.5	1.11	1.50	1.73	2.14	2.51	3.00	3.62	3.91	4.21	4.23	4.27
X 10	0.5	1.06	1.43	1.75	2.10	2.55	3.01	3.60	3.92	4.21	4.25	4.27

En las dos últimas tablas la ganancia $\times 100$ se omite porque, en este caso, la sensibilidad ya es muy poca y ya no es perceptible ningún cambio en los leds. El único led que prende es el primero.

De las tablas podemos observar que el voltaje obtenido no depende de la ganancia sino del valor del umbral. El voltaje es el mismo para un mismo umbral pero diferente ganancia.

Este aparato se utiliza para sesiones de Biofeedback. En estas sesiones es importante ver el cambio en los leds y en el umbral conforme el paciente se va relajando. Para nuestro caso, el voltaje de interés es el correspondiente al led número 8 (primer led amarillo de izquierda a derecha). Este voltaje es la valuación de la tensión muscular, es decir, es el valor del EMG. Por tanto, cuando se vaya a realizar la prueba es necesario manejar los controles de la escala y el umbral de modo que sea el led número 8 el que siempre esté encendido. No importa que cambie el valor del umbral al tratar de mantener prendido el octavo led, el resultado final es la lectura mínima de voltaje obtenida.

Recordemos que los valores límite para el EMG (Tabla A del capítulo 1) son 0.4 V el mínimo y 4 V el valor máximo. En la tabla 5.3.1 el voltaje del octavo led es de 0.17 V para el umbral menor y en la tabla 5.3.8 el voltaje del led 8 es de 3.62 V para el mayor umbral. Se puede observar que existen diferencias (no muy grandes) entre los valores límite de la escala Estrés-Bienestar y los voltajes arrojados por el circuito propuesto. Esto se debe a que, para obtener los valores de la escala, se utilizaron otros aparatos y el modo de obtener el valor del EMG también fue diferente, éste se obtenía por la observación de una persona y realizando cálculos matemáticos. En este trabajo, es un circuito el que recoge la lectura de voltaje y un programa el que la interpreta. También, cabe señalar que este es un circuito experimental y que las personas adecuadas (médicos) son las indicadas para realizarle todas las pruebas necesarias a todo el sistema para respaldar la veracidad de los resultados y hacer la calibración de cada uno de los circuitos.

Una vez explicado lo anterior, se proponen como límites del EMG el voltaje de 0.17 V y 3.62 V

5.4 SENSOR DE CONDUCTANCIA DE LA PIEL (GSR)

Para las pruebas al circuito de medición de GSR se midió con el osciloscopio la frecuencia observada al prenderse el led (circuito con el decodificador de tono LM567, figura 3.18) y se midió el voltaje en la resistencia del GSR2. Las lecturas les fueron tomadas a 15 personas. Los resultados se pueden observar en la tabla 5.4.

La lectura de frecuencia se obtuvo con el osciloscopio Tektronics TDS 360. Los voltajes se midieron con el multimetro GoldStar DM-312.

En la tabla 5.4 se puede observar que las frecuencias leídas no se alejan mucho de los 12 KHz. Además, en esta misma tabla, se resaltan los valores de voltaje mayores a 2.5 V. Estos voltajes corresponden a personas relajadas. A los 15 voluntarios se les hizo subir unas escaleras para llegar al cuarto en donde se realizó la medición. A unos se les permitió descansar antes de realizar la prueba, mientras que a otros se le midió la conductancia de la piel inmediatamente.

Tabla 5.4. Resultados de las pruebas realizadas al circuito de medición de conductancia de la piel (GSR).

Persona No.	Frecuencia (KHz)	Voltaje (V)
1	11.82	2.45
2	12.14	1.22
3	12.29	2.73
4	12.10	0.92
5	11.79	3.20
6	12.40	3.77
7	12.55	2.23
8	12.70	0.44
9	12.34	1.67
10	12.13	1.45
11	11.97	0.33
12	12.35	1.86
13	12.40	1.67
14	12.15	3.12
15	11.88	3.56

Los resultados de estas pruebas demuestran que el voltaje es mayor cuando la persona está relajada, es decir, la piel tiene más resistencia. Y cuando la persona presenta un nivel de estrés mayor, la resistencia en la piel es menor y por lo tanto, el voltaje también es menor.

Cabe mencionar, que la conductancia es el inverso de la resistencia, esto es, cuando la piel ofrece más resistencia hay menos conductancia y viceversa, si la piel tiene menos resistencia se presenta mayor conductancia. Por lo tanto, entre más relajada esté una persona menos conductancia tendrá y entre más estresada se encuentre una persona mayor conductancia presentará.

La conclusión anterior es útil para poder adecuar los límites propuestos en la escala Estrés-Bienestar (0 – 40) con los obtenidos en las pruebas realizadas al circuito diseñado para medir el parámetro GSR.

La lectura mayor de voltaje es 4 V (con el volumen en su límite máximo que es 10). Si a mayor voltaje menor conductancia, entonces cuando la lectura sea de 4 V, se tendrá que la persona está muy relajada y esto debe corresponder al límite inferior del rango de la escala Estrés-Bienestar, el cual es cero. Y, si la persona está muy estresada, el mínimo valor de voltaje que se puede obtener es 0, que debe corresponder al límite superior de la escala Estrés-Bienestar, que es 40. Entonces,

$$4V \rightarrow 0 \quad \text{y} \quad 0V \rightarrow 40$$

Para ajustar las escalas, se propone multiplicar el promedio de las lecturas de voltaje, obtenidas durante una sesión, por 10 y 40 restarle esta cantidad. Por ejemplo, si el voltaje leído es 4 V,

$40 - 4 * 10 = 40 - 40 = 0$ esto equivale a que la persona está totalmente relajada.

Ahora, si el voltaje es 0 V,

$40 - 0 * 10 = 40 - 0 = 40$ esto equivale a que la persona se encuentra en total estado de estrés.

Debe tenerse en cuenta que estas escalas que se proponen son experimentales y que son los médicos las personas adecuadas que deben realizar pruebas exhaustivas para poder validar los resultados obtenidos.

CONCLUSIONES

6.1 APORTACIONES

El estrés es un estado físico y psicológico que afecta el bienestar de todas las personas en el mundo. A pesar de que cada día y con mayor frecuencia nos enfrentamos a situaciones estresantes, en muchos países no se le presta la atención debida a este problema. En México, por ejemplo, el cuerpo médico no toma en cuenta el estrés del paciente para completar un diagnóstico o para prevenir enfermedades. En algunos casos, se realizan chequeos generales del paciente, muy costosos y que toman mucho tiempo para realizarse, en donde se analizan los estados de los órganos vitales y normalmente se concluye con la prescripción de algún fármaco que de ninguna manera elimina las causas de la enfermedad o el mal funcionamiento del órgano.

El sistema digital desarrollado permite obtener un nivel de estrés que involucra el resultado de siete parámetros, cuya base teórica esta sustentada por numerosos estudios. Estos siete parámetros son los indicadores más significativos para la medición del estrés. El resultado final integra las partes física y emocional de cada persona.

El sistema no sólo arroja un número que indica que tan estresada está una persona, sino también, muestra los resultados individuales de cada prueba, lo que ayuda a dar un tratamiento específico para cada caso.

6.2 CONCLUSIONES

Este trabajo es el primer sistema digital desarrollado para medir el nivel de estrés en un ser humano de manera integral, ya que se mide el estrés físico y el emocional. Esto permite al médico evaluar mejor al paciente para identificar con más precisión las molestias de la persona.

En la realización de este trabajo se pudo comprobar que el sensor de movimiento respiratorio y torácico propuesto es muy adecuado, ya que además de que no se afloja (como los sensores construidos con cintas elásticas), no tiene contacto con el cuerpo de la persona evitando así causar incomodidad en ésta. Además, se realizaron varias pruebas con los sensores sin necesidad de calibrarse cada vez que se usan. Al igual que en toda medición sólo se necesita colocar de manera adecuada el dispositivo para no obtener mediciones erróneas.

Para medir la conductancia de la piel, en el manual del sensor de GSR se especifica que se debe mantener constante un tono audible y agudo, nunca especifica alguna frecuencia en particular, por tanto, el tono elegido puede variar para cada persona y para cada momento, es decir, tal vez en una sesión el doctor elige un tono (el que le parece adecuado) y para una sesión posterior, el tono que eligió puede ser diferente al anterior. Además, cada doctor puede elegir el tono que crea es el más adecuado para realizar la prueba. Cabe agregar, que no sólo debía asegurarse de que el tono fuera el mismo, sino

que se debía estar atento al cambio de escala en el volumen del sensor e ir registrando los valores para posteriormente, se obtuviera el valor más pequeño de la escala. Ahora, realizar la prueba es más sencillo. El circuito propuesto para medir la conductancia de la piel cuenta con un led el cual sirve para la calibración del aparato. Al encenderse el led se esta indicando que se ha alcanzado la frecuencia del tono adecuado para llevar a cabo la medición. El valor de la frecuencia se estableció en 12 KHz, esta frecuencia corresponde al tono audible y agudo mencionado en el manual del sensor. De esta manera, la calibración del aparato ya no se realiza mediante la percepción, muy particular, de cada persona. La calibración resulta, ahora, más confiable ya que el led siempre prenderá cuando se alcance la frecuencia fija de 12 KHz. Otra ventaja que ofrece el circuito propuesto, es que el doctor ya no necesita estar pendiente del cambio de escala ya que el circuito toma el voltaje de una resistencia (incluida en el sensor de GSR) en donde se registran esos cambios de escala. Las variaciones de voltaje las toma un programa que procesa esas señales y obtiene el valor del parámetro de conductancia de la piel. El resultado de todo esto es que el doctor sólo debe estar atento, regulando el volumen del sensor, para que el led permanezca encendido y se realice una prueba adecuada.

En el caso de la medición de la tensión muscular (EMG), el manual del sensor establece que el valor de EMG se obtiene cuando se enciende el primer led amarillo u octavo led (de izquierda a derecha). El doctor debe manipular el umbral del sensor de manera que permanezca encendido dicho led durante toda la sesión. Una vez más, el doctor debe estar pendiente del cambio de escala del umbral para que al final, obtenga el promedio de los diferentes valores del umbral. A este promedio se le realizan otras acciones matemáticas que comprenden a la escala y a la relación que guardan el número de led con el umbral. El circuito aquí propuesto libera al doctor de realizar todas esas actividades, es decir, ahora sólo necesita mantener encendido el octavo led del sensor variando el umbral sin preocuparse por ir registrando los diferentes valores del umbral y sin llevar a cabo ninguna acción matemática. La medición de este parámetro ahora es automática. El circuito toma los valores de voltaje que entrega el sensor y los envía a la PC donde un programa procesa las señales y obtiene el resultado final que corresponde al valor del parámetro del EMG.

En ambos casos, se tuvieron que adecuar los resultados arrojados por los circuitos con los límites de la escala "Estrés-Bienestar". Esto debido a que, para obtener la escala, las mediciones se realizaron de forma diferente y con sensores distintos.

El uso de convertidores analógico-digital de 8 bits en la interfaz entre los circuitos y la PC proporcionan una conversión binaria adecuada ya que los errores entre la lectura medida (con un voltímetro) y la lectura interpretada (por el software), son muy pequeños por lo que se pueden despreciar.

El sistema de medición desarrollado cumple con la norma de que los aparatos médicos que tienen contacto con el cuerpo del paciente no deben estar energizados directamente con la corriente proveniente de los enchufes. Esto se logra al trabajar con baterías de 9 V

El uso de este sistema proporciona la ventaja que las mediciones las realizan los circuitos evitando errores humanos en las lecturas. Por ejemplo anteriormente, los resultados de las mediciones de los siete parámetros se realizaban mediante la observación del doctor. En particular, para la prueba de la respiración, el doctor tenía que observar durante todo el examen el movimiento respiratorio del tórax y del abdomen, esto resultaba fatigoso y poco confiable, porque el doctor debía de mantener una posición fija de observación lo cual resulta difícil de lograr, además, tenía que ir tomando el tiempo que dura la inhalación y el tiempo que dura la exhalación, al mismo tiempo, tenía que observar si el tórax se levanta antes que el abdomen y debía ir registrando la frecuencia de la respiración. Todo esto ya lo realiza el software. Ahora, lo único que tiene que hacer el doctor para realizar la prueba es colocar los sensores del movimiento respiratorio perpendiculares al tórax y al abdomen y esperar que el programa obtenga el valor correspondiente al parámetro de la respiración.

Lo anterior resulta muy ventajoso porque, de esta manera, los resultados son más precisos (al realizarlos una máquina) y es menos laborioso para el doctor.

El software desarrollado es muy amigable, muy sencillo de utilizar, bastante claro y es compatible con el sistema operativo Windows 95 y 98. Además, no hay necesidad de configurar el puerto paralelo porque se utiliza en su forma estándar.

Al unir las bondades ofrecidas por el software y el hardware se obtuvo un sistema digital muy completo y fácil de manejar.

6.3 RECOMENDACIONES Y TRABAJOS FUTUROS

Una primera recomendación, es adquirir los electrodos adicionales para medir la conductancia de la piel que se sujetan directamente en los dedos sin necesidad de sostener el sensor con la mano. De esta manera, se evitan lecturas diferentes cuando el paciente, al principio, aprieta las placas sensoras y, al final, cuando la presión que ejerce sobre las placas es menor. Además, resulta cansado y molesto sostener algo en las manos durante varios minutos.

Al utilizar el sistema propuesto, no se elimina la intervención humana. Para las pruebas de EMG y GSR es necesario que el doctor regule las mediciones para mantener una calibración correspondiente a un valor ya establecido. Por lo que se propone la construcción de circuitos sensores (como los que se adquirieron, MyoTrac y GSR2) para realizar de manera automática las pruebas de conductancia y tensión muscular. En general, una mejora al sistema es la automatización completa de las pruebas para evitar por completo la intervención humana en la toma de lecturas, dejando únicamente la parte de interpretación a las personas calificadas.

La validación de los datos del sistema debe hacerse por doctores, ya que lo que se mide son parámetros físicos. Sólo las personas que conocen el funcionamiento del cuerpo humano pueden ser capaces de interpretar los resultados obtenidos por el sistema. Solamente los doctores, que tienen experiencia en el campo de la medicina, pueden afirmar que lo que se midió corresponde a los valores arrojados por el sistema.

APÉNDICE : CÓDIGO DEL PROGRAMA EN VISUAL C++ 5.0

En este apéndice se muestra el programa fuente del software desarrollado en el lenguaje Visual C++ versión 5.0.

Este apéndice está estructurado de manera que se menciona el nombre de la pantalla y los botones que la conforman. Se presenta el código fuente correspondiente a cada botón.

PANTALLA INICIAL

Botón *Entrar*

```
void CProg1Dlg::OnBoton1()
{
    // TODO: Add your control notification handler code here
    CArchivoDlg dlg;
    OnOK();
    dlg.DoModal();
}
```

Botón *Salir*

```
void CProg1Dlg::OnSalir()
{
    // TODO: Add your control notification handler code here
    OnOK();
}
```

PANTALLA DEL MENÚ PRINCIPAL

Botón *Anterior*

```
void CArchivoDlg::OnAnterior()
{
    // TODO: Add your control notification handler code here
    if(n == ultimo)
        m_sig.EnableWindow(TRUE);
    if(n == 1)
        m_anterior.EnableWindow(FALSE);
    n--;
    escribe_datos_cajas();
}
```

```
        checa_estado_prueba());  
    }
```

Botón *Siguiente*

```
void CArchivoDlg::OnSig()  
{  
    // TODO: Add your control notification handler code here  
    if(n == (ultimo - 1))  
        m_sig.EnableWindow(FALSE);  
    if(n == 0)  
        m_anterior.EnableWindow(TRUE);  
    n++;  
    escribe_datos_cajas();  
    checa_estado_prueba();  
}
```

Botón *Buscar*

```
void CArchivoDlg::OnBuscar()  
{  
    // TODO: Add your control notification handler code here  
    CBuscar Dlg;  
    Dlg.DoModal();  
    escribe_datos_cajas();  
    checa_estado_prueba();  
    if(n==0)  
    {  
        m_anterior.EnableWindow(FALSE);  
        m_sig.EnableWindow(TRUE);  
    }  
    else  
        if(n==ultimo)  
        {  
            m_anterior.EnableWindow(TRUE);  
            m_sig.EnableWindow(FALSE);  
        }  
        else  
        {  
            m_anterior.EnableWindow(TRUE);  
            m_sig.EnableWindow(TRUE);  
        }  
    }  
}
```

Botón *Paciente Nuevo*

```
void CArchivoDlg::OnNew()
{
    // TODO: Add your control notification handler code here

    // Se habilitan las cajas
    guarda_estado();
    estado_cajas(TRUE);
    m_registrar.EnableWindow(TRUE);
    m_cancelar.EnableWindow(TRUE);
    m_apat = _T("");
    m_edad = 0;
    m_edocivil = _T("");
    m_neval = 0;
    m_motivo = _T("");
    m_nom = _T("");
    m_ocupacion = _T("");
    m_titulo = _T("");
    m_fecha = _T("");
    m_sexo = _T("");
    UpdateData(FALSE);
    m_e1.SetFocus();
    // Se deshabilitan todos los botones
    deshabilita_botones();
    nuevo_modifica = nuevo;
}
}
```

Botón *Modificar*

```
void CArchivoDlg::OnModif()
{
    // TODO: Add your control notification handler code here
    guarda_estado();
    estado_cajas(TRUE);
    m_e1.SetFocus();
    m_registrar.EnableWindow(TRUE);
    m_cancelar.EnableWindow(TRUE);
    // Se deshabilitan todos los botones
    deshabilita_botones();
    nuevo_modifica = modifica;
}
}
```

Botón Registrar Datos

```

void CArchivoDlg::OnRegdat()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    if(m_nom == "" || m_apat == "" || m_edad == 0 || m_sexo == "" || m_fecha == "")
    {
        MessageBox("Falta un dato importante", "Error", MB_OK);
        return;
    }
    if(nuevo_modifica == nuevo)
    {
        int veces=0,i;
        for(i=0;i<=ultimo,i++)
            if(m_nom == B[i].Nom && m_apat == B[i].Apat && m_edad == B[i].Edad)
                veces++;
        UpdateData(TRUE);
        if(veces != 0)
            m_neval = veces + 1;
        else
            m_neval = 1;
        UpdateData(FALSE);
        ultimo++;
        n = ultimo;
        B[n].Msp = 0; B[n].Emg = 0; B[n].Gsr = 0; B[n].Postura = 0;
        B[n].Resp = 0; B[n].Cortisol = 0; B[n].Prolac = 0;
        B[n].numero = ultimo;
        if(ultimo > 0)
            m_anterior.EnableWindow(TRUE);
        m_msp.EnableWindow(TRUE);
        m_postura.EnableWindow(TRUE);
        m_indice.EnableWindow(TRUE);
        m_sensor.EnableWindow(TRUE);
        m_emg.EnableWindow(TRUE);
        m_gsr.EnableWindow(TRUE);
    }
    else
    {
        checa_estado_prueba();
        if(n != ultimo)
            m_sig.EnableWindow(TRUE);
        if(n != 0)
            m_anterior.EnableWindow(TRUE);
    }
    escribe_datos_estructura();
    estado_cajas(FALSE);
}

```

```

m_registrar.EnableWindow(FALSE),
m_cancelar.EnableWindow(FALSE);
// Habilitar botones
m_buscar.EnableWindow(TRUE);
m_new.EnableWindow(TRUE);
m_modif.EnableWindow(TRUE);
m_exportar.EnableWindow(TRUE);
m_importar.EnableWindow(TRUE);

```

```

}

```

Botón *Cancelar*

```

void CArchivoDlg::OnCancelar()

```

```

{
    // TODO: Add your control notification handler code here
    estado_cajas(FALSE);
    restaura_estado();
}

```

Botón *Ver Resultados*

```

void CArchivoDlg::OnVerres()

```

```

{
    // TODO: Add your control notification handler code here
    CRes resul;
    char buffer[10];
    CString S1,S2;
    resul.val_p[0] = B[n].Msp;           // MSP
    resul.val_p[1] = B[n].Emg;         // EMG
    resul.val_p[2] = B[n].Gsr;         // GSR
    resul.val_p[3] = B[n].Resp;        // Respiración
    resul.val_p[4] = B[n].Postura;     // Postura
    resul.val_p[5] = B[n].Cortisol;    // Cortisol
    resul.val_p[6] = B[n].Prolac;      // Prolactina
    resul.sexo = B[n].Sexo;
    S1 = B[n].Nom; S2 = B[n].Apat;
    resul.Nombre = S1 + " " + S2;
    resul.Fecha = B[n].Fecha;
    sprintf(buffer,"%d",B[n].Neval);
    S1 = buffer;
    S2 = "No. de evaluación: " + S1;
    resul.Eval = S2;
    resul.DoModal();}

```


Botón Repetir Evaluación

```

void CArchivoDlg::OnRepetir()
{
    // TODO: Add your control notification handler code here
    int Resp;
    Resp = MessageBox("¿Desea repetir las evaluaciones?", "Pregunta", MB_OKCANCEL +
MB_ICONQUESTION);
    if(Resp == IDOK)
    {
        B[n].Msp = 0; B[n].Emg = 0; B[n].Gsr = 0; B[n].Postura = 0;
        B[n].Resp = 0; B[n].Cortisol = 0; B[n].Prolac = 0;
        m_msp.EnableWindow(TRUE);
        m_postura.EnableWindow(TRUE);
        m_indice.EnableWindow(TRUE);
        m_sensor.EnableWindow(TRUE);
        m_emg.EnableWindow(TRUE);
        m_gsr.EnableWindow(TRUE);
        m_ver.EnableWindow(FALSE);
        m_repetir.EnableWindow(FALSE);
    }
}

```

Botón Cuestionario MSP

```

void CArchivoDlg::OnMsp()
{
    // TODO: Add your control notification handler code here
    MSPDlg dialogo;
    dialogo.DoModal();
    if(dialogo.total_msp > 0)
    {
        m_msp.EnableWindow(FALSE);
        m_repetir.EnableWindow(TRUE);
        m_ver.EnableWindow(TRUE);
        B[n].Msp = dialogo.total_msp;
    }
}

```

Botón Comportamiento Corporal

```

void CArchivoDlg::OnPostura()
{
    // TODO: Add your control notification handler code here
    CuerpoDlg dialogo;
    m_modif.EnableWindow(TRUE);
}

```

```

dialogo.DoModal();
if(dialogo.Total > 0)
{
    m_postura.EnableWindow(FALSE);
    m_repetir.EnableWindow(TRUE);
    m_ver.EnableWindow(TRUE);
    B[n].Postura = dialogo.Total;
}
}

```

Botón *Análisis de Sangre*

```

void CArchivoDlg::OnIndice()
{
    // TODO: Add your control notification handler code here
    CDialogoDlg Di;
    Di.DoModal();
    if(Di.m_cortisol > 0 && Di.m_prolactina >= 0)
    {
        B[n].Cortisol = Di.m_cortisol;
        B[n].Prolac = Di.m_prolactina;
        m_indice.EnableWindow(FALSE);
        m_repetir.EnableWindow(TRUE);
    }
}

```

Botón *Respiración*

```

void CArchivoDlg::OnSensor()
{
    // TODO: Add your control notification handler code here
    CSensorDlg Dialogo;
    Dialogo.DoModal();
    if(Dialogo.resultado > 0)
    {
        m_sensor.EnableWindow(FALSE);
        m_repetir.EnableWindow(TRUE);
        m_ver.EnableWindow(TRUE);
        B[n].Resp = Dialogo.resultado;
    }
}

```

Botón *EMG*

```

void CArchivoDlg::OnEmg()
{
    // TODO: Add your control notification handler code here

```

```

CEMG Dialogo;
Dialogo.DoModal();
if(Dialogo.resultado > 0)
{
    m_emg.EnableWindow(FALSE);
    m_repetir.EnableWindow(TRUE);
    m_ver.EnableWindow(TRUE);
    B[n].Emg = Dialogo.resultado;
}
}

```

Botón GSR

```

void CArchivoDlg::OnGsr()
{
    // TODO: Add your control notification handler code here
    CGSR Dialogo;
    Dialogo.DoModal();
    if(Dialogo.resultado > 0)
    {
        m_gsr.EnableWindow(FALSE);
        m_repetir.EnableWindow(TRUE);
        m_ver.EnableWindow(TRUE);
        B[n].Gsr = Dialogo.resultado;
    }
}

```

Botón Exportar

```

void CArchivoDlg::OnExporta()
{
    // TODO: Add your control notification handler code here

    if(n < 0)
        MessageBox("No existen datos para exportar", "Error", MB_OK);
    else
    {
        CFileDialog guarda(FALSE);
        guarda.m_ofn.lpstrFilter="Archivos MIE (*.mie)0*.mie\0";
        guarda.m_ofn.lpstrDefExt="*.mie";
        if(guarda.DoModal() == IDOK)
        {
            Nom_Archivo = guarda.GetPathName();
            salva_archivo();
        }
    }
}

```

```

void CArchivoDlg::salva_archivo()
{
    int i;
    UINT modo_Cfile = CFile::modeCreate|CFile::modeWrite;
    UINT TamReg = sizeof(struct lista);
    CFile Archivo;
    if(!Archivo.Open(Nom_Archivo,modo_Cfile))
    {
        AfxMessageBox("No se puede abrir el archivo");
        return;
    }
    for(i=0;i<=ultimo,i++)
    {
        Archivo.SeekToEnd();
        Archivo.Write(&B[i],TamReg);
    }
    Archivo.Close();
}

```

Botón *Importar*

```

void CArchivoDlg::OnImporta()
{
    // TODO: Add your control notification handler code here

    if(ultimo > -1)
        salva_archivo();
    CFileDialog guarda(TRUE);
    guarda.m_ofn.lpstrFilter="Archivos MIE (*.mie)|0*.mie|0";
    guarda.m_ofn.lpstrDefExt="*.mie";
    if(guarda.DoModal() == IDOK)
    {
        Nom_Archivo = guarda.GetPathName();
        lee_archivo();
        escribe_datos_cajas();
        checa_estado_prueba();
        m_anterior.EnableWindow(FALSE);
        if(ultimo==0)
        {
            m_sig.EnableWindow(FALSE);
            m_buscar.EnableWindow(FALSE);
        }
        else
        {
            m_sig.EnableWindow(TRUE);
            m_buscar.EnableWindow(TRUE);
        }
    }
}

```

```

    }
}

void CArchivoDlg::lee_archivo()
{
    UINT modo_Cfile = CFile::modeRead;
    CFile Archivo;
    int i;
    UINT TamReg = sizeof(struct lista);
    if(!Archivo.Open(Nom_Archivo,modo_Cfile))
    {
        AfxMessageBox("No se puede abrir el archivo");
        return;
    }
    long pos = (long)Archivo.GetLength();
    long nRegs = pos / (long)TamReg;
    for(i=0;i<(int)nRegs;i++)
        Archivo.Read(&B[i],TamReg);
    Archivo.Close();
    ultimo = (int) nRegs - 1;
    n = 0;
}

```

Botón Ayuda

```

void CArchivoDlg::OnButton4()
{
    // TODO: Add your control notification handler code here
    ::WinHelp(m_hWnd,"c:/tesis/programa/sucara1/release/prog1.hlp",HELP_FINDER,NULL);
}

```

Botón Salir del Programa

```

void CArchivoDlg::OnS()
{
    // TODO: Add your control notification handler code here
    int Res;
    Res = MessageBox("Está seguro de Salir?","Pregunta",MB_OKCANCEL
    MB_ICONQUESTION);
    if(Res == IDOK)
    {
        if(n > -1)
            salva_archivo();
        OnOK();
    }
}

```

}

PANTALLA BÚSQUEDA DE PACIENTES**Botón *Buscar***

void CBuscar::OnBbusca()

```

{
    // TODO: Add your control notification handler code here
    int i,num=0,I;
    char buffer[5];
    CString EDAD,EVAL,
    UpdateData(TRUE);
    CString Cad,Low1,Low2,nombre,apellido,fecha;
    switch(naef)
    {
        case 1:Low1 = m_NomAp;
                Low1.MakeLower();
                for(i=0,i<=ultimo;i++)
                {
                    Low2 = B[i].Nom;
                    Low2.MakeLower();
                    if(Low1 == Low2)
                    {
                        find[num] = i;
                        num++;
                    }
                }
                break;
        case 2:Low1 = m_NomAp;
                Low1.MakeLower();
                num = 0;
                for(i=0,i<=ultimo;i++)
                {
                    Low2 = B[i].Apat;
                    Low2.MakeLower();
                    if(Low1 == Low2)
                    {
                        find[num] = i;
                        num++;
                    }
                }
                break;
        case 3: for(i=0,i<=ultimo;i++)
                if(m_eval == B[i].Neval)
                {

```

```

        find[num] = i;
        num++;
    }
    break;
case 4: Cad = m_dia + "/" + m_mes + "/" + m_ano;
    for(i=0; i<=ultimo; i++)
        if(Cad == B[i].Fecha)
        {
            find[num] = i;
            num++;
        }
    break;
}
}
if(num == 0)
{
    MessageBox("Registro NO Encontrado", "Afirmación", MB_OK +
    MB_ICONEXCLAMATION);
    m_ver.EnableWindow(FALSE);
}
else
{
    m_lista.ResetContent( );
    for(i=0; i<num; i++)
    {
        I = find[i];
        Cad = "";
        sprintf(buffer, "%d", B[I].Edad);
        EDAD = buffer;
        sprintf(buffer, "%d", B[I].Neval);
        EVAL = buffer;
        nombre = B[I].Nom;
        apellido = B[I].Apat;
        fecha = B[I].Fecha;
        Cad = nombre + " " + apellido + " ";
        Cad += EDAD + " años " + fecha;
        Cad += " No. Eval. " + EVAL;
        m_lista.AddString(Cad);
    }
}
}
}
}

```

Botón Cancelar

```
void CBuscar::OnCancelar()
```

```

{
    // TODO: Add your control notification handler code here
    OnOK();
}

```

```

}

Botón Aceptar

void CBuscar::OnVer()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    sel=m_lista.GetCurSel();
    n=find[sel];
    OnOK();
}

```

PANTALLA CUESTIONARIO MSP

Botón Anterior

```

void MSPDlg::OnAnterior()
{
    // TODO: Add your control notification handler code here
    control = 1;
    m_back.EnableWindow(TRUE);
    if(preg_actual == 48)
        m_sig.EnableWindow(TRUE);
    borra();
    if(npreg == 1)
        m_anterior.EnableWindow(FALSE);
    npreg--;
    m_preg = msp[npreg].Preg;
    m_npreg = npreg + 1;
    UpdateData(FALSE);
    pinta();
}

```

Botón Siguiente

```

void MSPDlg::OnSig()
{
    // TODO: Add your control notification handler code here
    int i;
    char buffer[200];
    if(npreg==0)
        m_anterior.EnableWindow(TRUE);
    if(valor[npreg] == 0)
        MessageBox("Pregunta NO Contestada", "Error", MB_OK + MB_ICONSTOP);
}

```



```

else
{
    if(npreg == 48)
    {
        m_sig.EnableWindow(FALSE);
        total_msp = 0;
        for(i=0;i<49;i++)
            total_msp += valor[i];
        sprintf(buffer, "Fin del cuestionario\nTotal = %3.0f", total_msp );
        MessageBox(buffer, "Fin MSP", MB_OK + MB_ICONEXCLAMATION);
    }
    else
    {
        borra();
        if(npreg < preg_actual)
        {
            if((npreg+1) == preg_actual)
                m_back.EnableWindow(FALSE);
            npreg++;
            m_preg = msp[npreg].Preg;
            pinta();
        }
        else
        {
            if(control == 1)
                control=0;
            npreg++;
            preg_actual = npreg;
            m_preg = msp[npreg].Preg;
            UpdateData(FALSE);
        }
        m_npreg = npreg+1;
        UpdateData(FALSE);
    }
}
}
}

```

Botón Regresar

```

void MSPDlg::OnBack()
{
    // TODO: Add your control notification handler code here
    if(npreg == 0)
        m_anterior.EnableWindow(TRUE);
    m_back.EnableWindow(FALSE);
    borra();
}

```

```

    npreg = preg_actual;
    m_preg = msp[npreg].Preg;
    m_npreg = npreg + 1;
    UpdateData(FALSE);
    pinta();
}

Botón Ir

void MSPDlg::OnIr()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    npreg = m_nopreg;
    if(npreg == 0)
        MessageBox("Número de pregunta NO Válida", "Error", MB_OK +
MB_ICONSTOP);
    else
    {
        control = 1;
        m_nopreg = 0;
        UpdateData(FALSE);
        m_back.EnableWindow(TRUE);
        npreg--;
        if(npreg > preg_actual)
        {
            MessageBox("Preguntas anteriores NO Contestadas", "Error",MB_OK +
MB_ICONSTOP);
            npreg=preg_actual;
        }
        else
        {
            if(npreg == 0)
                m_anterior.EnableWindow(FALSE);
            if(npreg == 48)
                m_back.EnableWindow(FALSE);
            if(preg_actual == 48)
                m_sig.EnableWindow(TRUE);
            borra();
            m_preg = msp[npreg].Preg;
            m_npreg = npreg + 1;
            UpdateData(FALSE);
            pinta();
        }
    }
}
}

```

Botón Regresar al Menú Principal

```

void MSPDlg::OnRegresa()
{
    // TODO: Add your control notification handler code here
    int Res;
    CArchivoDlg D;
    if(npreg < 48)
    {
        Res = MessageBox("¿Abortar Cuestionario MSP?", "Alerta", MB_OKCANCEL +
MB_ICONQUESTION);
        if(Res == IDOK)
        {
            OnOK();
            npreg = 0;
            total_msp = 0;
        }
    }
    else
        OnOK();
}
}

```

PANTALLA COMPORTAMIENTO CORPORAL**Botón Registra Puntuación**

```

void CuerpoDlg::OnRegpunt()
{
    // TODO: Add your control notification handler code here
    char buffer[200];
    Total = Postura + Movimiento + Voz;
    sprintf(buffer, "Resultado de la prueba\nPuntuación total = %1.0lf", Total);
    MessageBox(buffer, "Fin de la Prueba", MB_OK + MB_ICONEXCLAMATION);
}
}

```

Botón Regresar al Menú Principal

```

void CuerpoDlg::OnRegresa()
{
    // TODO: Add your control notification handler code here
    int resp;
    if(Total == 0)
    {
        resp = MessageBox("¿Desea salir sin haber realizado la prueba?", "Pregunta",
MB_OKCANCEL + MB_ICONQUESTION);
    }
}
}

```

```

        if(resp == IDOK)
            OnOK();
    }
    else
        OnOK();
}

```

PANTALLA NIVEL DE CORTISOL Y PROLACTINA

Botón Registrar Datos

```

void CDialogoDlg::OnM()
{
    // TODO: Add your control notification handler code here
    int resp;
    UpdateData(TRUE);
    if(m_cortisol == 0 && m_prolactina == 0)
    {
        resp = MessageBox("¿Desea salir sin haber\nintroducido ningún dato?", "Pregunta", MB_OKCANCEL + MB_ICONQUESTION);
        if(resp == IDOK)
            OnOK();
    }
    else
        OnOK();
}

```

PANTALLA RESPIRACIÓN TORÁXICA Y ABDOMINAL

Botón Iniciar Prueba

```

void CSensorDlg::OnInicia()
{
    // TODO: Add your control notification handler code here
    MSG message;
    CClientDC dc(this);
    CPen pluma(PS_SOLID, 1, RGB(0,0,0));
    dc.SelectObject(&pluma);
    CRect rect(50,50,790,310);
    CTime T;
    int x,min,tiempo,durac=20,pasa = 0;
    char buffer[400];
    double byte1,byte2,a1,a2,b1,b2;
    DATA = 0x0378;
    STATUS = DATA+1;
}

```

```

m_inicia.EnableWindow(FALSE);
m_cancelar.EnableWindow(TRUE);
cont = 0; a2=0; b2=0;
T = CTime::GetCurrentTime();
tiempo = T.GetSecond( );
min=tiempo-1;
for(x=0;cont <= 0,x+=5)
{
    pulso();
    byte1=lee_byte(1);
    byte2=lee_byte(2);
    congruencia(byte1,byte2);
    pausa1(byte1);
    pausa2(byte2);
    relacion_sd(byte2);
    relacion_td(byte1,byte2);
    frecuencia(byte2);
    a1=byte1*20.0; b1=byte2*20.0;
    if(x>670)
    {
        x=-5;
        RedrawWindow(rect,NULL,RDW_INVALIDATE | RDW_UPDATENOW
RDW_ERASE );
    }
    else
    {
        dc.MoveTo(50+x,180-(int)a2);
        dc.LineTo(55+x,180-(int)a1);
        dc.MoveTo(50+x,310-(int)b2);
        dc.LineTo(55+x,310-(int)b1);
        Sleep(70);
    }
    a2=a1; b2=b1;

    if(::PeekMessage(&message,NULL,0,0,PM_REMOVE))
    {
        ::TranslateMessage(&message);
        ::DispatchMessage(&message);
    }
    T = CTime::GetCurrentTime();
    tiempo = T.GetSecond();
    if(min==tiempo)
    {
        test = 1;
        cont=1;
        resp_cong=evalua(punto_cong,1);

```

```

        resp_p1=evalua(punto_p1,0),
        resp_p2=evalua(punto_p2,0),
        resp_sd=evalua(punto_sd,0),
        resp_td=evalua(punto_td,2),
        resp_frec=evalua(punto_frec,0),
        resultado = resp_cong+resp_p1+resp_p2+resp_sd+resp_td+resp_frec;

        sprintf(buffer,"C=%d\nP1=%d\nP2=%d\nSD=%d\nTD=%d\nFrec=%d\nRespiracion=%3.2lf",r
esp_cong,resp_p1,resp_p2,resp_sd,resp_td,resp_frec,resultado);
        MessageBox(buffer, "Resultado", MB_OK),
        OnCancela(),
    }
}

void CSensorDlg::congruencia(double b1, double b2)
{
    static double a1,a2;
    static int prim=0,sube1=0,sube2=0,baja1=0,baja2=0;
    static int cont1=0,cont2=0,entra=0,Cont1,Cont2;
    static int subida1,bajada1,subida2,bajada2,
    if(prim)
    {
        if(a1<b1) sube1=1;
        if(a1>b1)
        {
            cont1++;
            if(cont1==2)
            {
                cont1=0;
                baja1=1;
            }
        }
        if(a2<b2) sube2=1;
        if(a2>b2)
        {
            cont2++;
            if(cont2==2)
            {
                cont2=0;
                baja2=1;
            }
        }
        if(a1==b1 && sube1)
        {
            sube1=0;
            subida1=1;

```

```

        bajada1=0;
        Cont1=0;
        prim++;
        entra=1;
    }
    if(a2==b2 && sube2)
    {
        sube2=0;
        subida2=1;
        bajada2=0;
        Cont2=0;
        prim++;
        entra=1;
    }
    if(a1==b1 && baja1)
    {
        baja1=0;
        bajada1=1;
        subida1=0;
        prim++;
        entra=1;
    }
    if(a2==b2 && baja2)
    {
        baja2=0;
        bajada2=1;
        subida2=0;
        prim++;
        entra=1;
    }
    if(prim>2 && entra)
    {
        if(subida1 && subida2) || (bajada1 && bajada2))
            punto_cong[0]++; //punto = 1
        if(subida1 && bajada2) || (bajada1 && subida2))
            punto_cong[1]++; //punto = 3
        if(subida1 && !subida2) || (!subida1 && subida2))
            punto_cong[2]++; //punto = 2
        if(bajada1 && !bajada2) || (!bajada1 && bajada2))
            punto_cong[2]++; //punto = 2
        entra = 0;
        subida1=0; subida2=0; bajada1=0; bajada2=0;
    }
}
else
    prim=1;

```

```

    a1=b1; a2=b2;
}

void CSensorDlg::pausa1(double b)
{
    static int sube=0,baja=0,pausa_inhala=0,pausa_exhala=0,Cont=0;
    static int i_seg,f_seg,i_mseg,f_mseg,time_mseg,prim=0;
    static int i2_seg,f2_seg,i2_mseg,f2_mseg,time2_mseg;
    static double a;
    struct _timeb timebuffer;
    if(prim)
    {
        if(b>a)
            sube=1;
        if(b<a)
            baja=1;
        if(a==b && sube)
        {
            _ftime( &timebuffer );
            i_seg=timebuffer.time;
            i_mseg=timebuffer.millitm;
            sube=0;
            pausa_inhala=1;
            Cont=0;
        }
        if(a>b && pausa_inhala)
        {
            Cont++;
            if(Cont == 2)
            {
                _ftime( &timebuffer );
                f_mseg=timebuffer.millitm;
                f_seg=timebuffer.time;
                if(i_mseg > f_seg)
                    time_mseg=(999-i_mseg)+f_mseg+(f_seg-1);
                else
                    time_mseg=f_seg + (f_mseg-i_mseg);
                pausa_inhala=0;
                Cont=0;
                prim++;
            }
        }
        if(a==b && baja)
        {
            _ftime( &timebuffer );
            i2_seg=timebuffer.time;
            i2_mseg=timebuffer.millitm;

```



```

        baja=0;
        pausa_exhala=1;
    }
    if(a<b && pausa_exhala)
    {
        _ftime( &timebuffer );
        f2_mseg=timebuffer.millitm;
        f2_seg=timebuffer.time;
        if(i2_mseg > f2_seg)
            time2_mseg=(999-i2_mseg)+f2_mseg+(f2_seg-1)*1000;
        else
            time2_mseg=f2_seg*1000 + (f2_mseg-i2_mseg);
        pausa_exhala=0;
        prim++;
    }
    if(prim>1 && pow(-1,prim)<0)
    {
        if(time2_mseg>time_mseg) //punto=1
            punto_p1[0]++;
        if(time2_mseg<time_mseg) //punto=4
            punto_p1[1]++;
        if(time2_mseg==time_mseg) //punto=2
            punto_p1[2]++;
    }

}
else
    prim=1;
a=b;
}

void CSensorDlg::pausa2(double b)
{
    static int sube=0,baja=0,pausa_inhala=0,pausa_exhala=0,Cont=0;
    static int i_seg,f_seg,i_mseg,f_mseg,time_mseg,prim=0;
    static int i2_seg,f2_seg,i2_mseg,f2_mseg,time2_mseg;
    static double a;
    struct _timeb timebuffer;
    if(prim)
    {
        if(b>a)
            sube=1;
        if(b<a)
            baja=1;
        if(a==b && sube)
        {
            _ftime( &timebuffer );

```

```
i_seg=timebuffer.time;
i_mseg=timebuffer.millitm;
sube=0;
pausa_inhala=1;
Cont=0;
}
if(a>b && pausa_inhala)
{
    Cont++;
    if(Cont == 2)
    {
        _ftime( &timebuffer );
        f_mseg=timebuffer.millitm;
        f_seg=timebuffer.time;
        if(i_mseg > f_seg)
            time_mseg=(999-i_mseg)+f_mseg+(f_seg-1);
        else
            time_mseg=f_seg + (f_mseg-i_mseg);
        pausa_inhala=0;
        cont=0;
        prim++;
    }
}
if(a==b && baja)
{
    _ftime( &timebuffer );
    i2_seg=timebuffer.time;
    i2_mseg=timebuffer.millitm;
    baja=0;
    pausa_exhala=1;
}
if(a<b && pausa_exhala)
{
    _ftime( &timebuffer );
    f2_mseg=timebuffer.millitm;
    f2_seg=timebuffer.time;
    if(i2_mseg > f2_seg)
        time2_mseg=(999-i2_mseg)+f2_mseg+(f2_seg-1)*1000;
    else
        time2_mseg=f2_seg*1000 + (f2_mseg-i2_mseg);
    pausa_exhala=0;
    prim++;
}
if(prim>1 && pow(-1,prim)<0)
{
    if(time2_mseg>time_mseg) //punto=1
        punto_p1[0]++;
}
```

```

        if(time2_mseg<time_mseg) //punto=4
            punto_p1[1]++;
        if(time2_mseg==time_mseg) //punto=2
            punto_p1[2]++;
    }

}
else
    prim=1;
a=b;
}

void CSensorDlg::relacion_sd(double b)
{
    static int sube=1,baja=1,Cont=0;
    static int i_seg,f_seg,i_mseg,f_mseg,time_mseg,prim=0;
    static int i2_seg,f2_seg,i2_mseg,f2_mseg,time2_mseg;
    static double a;
    struct _timeb timebuffer;
    if(prim)
    {
        if(a<b && sube)
        {
            _ftime( &timebuffer );
            i_seg=timebuffer.time;
            i_mseg=timebuffer.millitm;
            _ftime( &timebuffer );
            f2_mseg=timebuffer.millitm;
            f2_seg=timebuffer.time;
            if(i2_mseg > f2_seg)
                time2_mseg=(999-i2_mseg)+f2_mseg+(f2_seg-1)*1000;
            else
                time2_mseg=f2_seg*1000 + (f2_mseg-i2_mseg);
            sube=0;
            baja=1;
            prim++;
        }
        if(a>b && baja)
        {
            Cont++;
            if(Cont == 2)
            {
                _ftime( &timebuffer );
                i2_seg=timebuffer.time;
                i2_mseg=timebuffer.millitm;
                _ftime( &timebuffer );
                f_mseg=timebuffer.millitm;

```

```

        f_seg=timebuffer time;
        if(i_mseg > f_seg)
            time_mseg=(999-i_mseg)+f_mseg+(f_seg-1);
        else
            time_mseg=f_seg + (f_mseg-i_mseg);
        prim++;
        baja=0;
        sube=1;
        Cont=0;
    }
}
if(prim>2 && pow(-1,prim)>0)
{
    if(time2_mseg>time_mseg) //punto=1
        punto_sd[0]++;
    if(time2_mseg<time_mseg) //punto=4
        punto_sd[1]++;
    if(time2_mseg==time_mseg) //punto=2
        punto_sd[2]++;
}
}
else
    prim=1;
a=b;
}

```

```

void CSensorDlg::relacion_td(double b1, double b2)
{
    static double a1,a2,may1,may2,men1,men2,amp1,amp2;
    static int prim=0,sube1=0,sube2=0,baja1=0,baja2=0;
    static int cont1=0,cont2=0,ciclo1=0,ciclo2=0;
    if(prim)
    {
        if(a1<b1) sube1=1;
        if(a1>b1)
        {
            baja1=1;
        }
        if(a2<b2) sube2=1;
        if(a2>b2)
        {
            baja2=1;
        }
        if(a1==b1 && sube1)
        {
            sube1=0;

```

```

        prim++;
        ciclo1++;
        may1=a1;
    }
    if(a2==b2 && sube2)
    {
        sube2=0;
        prim++;
        ciclo2++;
        may2=a2;
    }
    if(a1==b1 && baja1)
    {
        baja1=0;
        prim++;
        ciclo1++;
        men1=a1;
    }
    if(a2==b2 && baja2)
    {
        baja2=0;
        prim++;
        ciclo2++;
        men2=a2;
    }
    if(ciclo2==2 && ciclo1==2)
    {
        amp1=may1-men1; // Toraxica
        amp2=may2-men2; // Abdominal
        if(amp1<=200e-3 && amp2>0.3)
            punto_td[0]++; //punto = 1
        if((amp1>=200e-3 || amp1<=1) && amp2>amp1)
            punto_td[1]++; //punto = 2
        if(amp1>=amp2)
            punto_td[2]++; //punto = 3
        if(amp2<=200e-3 && amp1>0.3)
            punto_td[3]++; //punto = 4
        ciclo1=0, ciclo2=0;
    }
}
else
    prim=1;
a1=b1, a2=b2;
}

```

void CSensorDlg::frecuencia(double b)

```

{
    static int tiempo,primera=0,veces=0,pasa2=0,seg_frec;
    static double v;
    static CTime Frec;
    Frec = CTime::GetCurrentTime();
    if(!primera)
    {
        tiempo = Frec.GetSecond();
        seg_frec = tiempo + 20;
        if(seg_frec > 59)
        {
            pasa2 = 1;
            seg_frec = seg_frec - 60;
        }
        v = b;
    }
    else
    {
        if(v == b)
            veces++;
        primera++;
        Frec = CTime::GetCurrentTime();
        tiempo = Frec.GetSecond();
        if((pasa2==1 && tiempo<=seg_frec) || (pasa2==0 && tiempo>=seg_frec))
        {
            if(veces==4 || veces==5) //punto = 1
                punto_frec[0]++;
            if(veces>=6) //punto = 4
                punto_frec[1]++;
            if(veces<4) //punto = 2
                punto_frec[2]++;
            veces=0;
            primera=0;
        }
    }
}

void CSensorDlg::pulso()
{
    _outp(DATA,0x10); Sleep(1);
    _outp(DATA,0x00); Sleep(1);
    _outp(DATA,0x10);
}

double CSensorDlg::lee_byte(int circuito)
{
    double byte;

```

```

unsigned int bls,bms;
switch(circuito)
{
    // Abdominal
    case 1: _outp(DATA,0x68);
           bls=(_inp(STATUS) ^ 0x80) & 0xF0;
           _outp(DATA,0x60);
           bms=(_inp(STATUS) ^ 0x80) & 0xF0;
           break;

    // Torácica
    case 2: _outp(DATA,0xA8);
           bls=(_inp(STATUS) ^ 0x80) & 0xF0;
           _outp(DATA,0xA0);
           bms=(_inp(STATUS) ^ 0x80) & 0xF0;
           break;
}
byte=(bms+(bls/16)) * 19.6e-3;
return byte;
}

```

Botón Cancelar Prueba

```

void CSensorDlg::OnCancela()
{
    // TODO: Add your control notification handler code here
    CRect rect(50,80,790,310);
    cont = 1;
    m_inicia.EnableWindow(TRUE);
    m_cancelar.EnableWindow(FALSE);
    RedrawWindow(rect,NULL,RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
}

```

PANTALLA TENSIÓN MUSCULAR (EMG)

Botón Iniciar Prueba

```

void CEMG::OnIniciar()
{
    // TODO: Add your control notification handler code here
    MSG message;
    CClientDC dc(this);
    CPen pluma(PS_SOLID,1,RGB(0,0,0));
    dc.SelectObject(&pluma);
    CRect rect(50,100,790,250);
    int x;
    double byte,a1,a2;
}

```

```

DATA = 0x0378;
STATUS = DATA+1;
m_inicia.EnableWindow(FALSE);
m_cancelar.EnableWindow(TRUE);
cont = 0; a2=0;
for(x=0;cont <= 0;x+=5)
{
    pulso();
    byte=lee_byte();
    a1=byte*30.0;
    if(x>670)
    {
        x=-5;
        RedrawWindow(rect,NULL,RDW_INVALIDATE | RDW_UPDATENOW |
RDW_ERASE );
    }
    else
    {
        dc.MoveTo(50+x,250-(int)a2);
        dc.LineTo(55+x,250-(int)a1);
        Sleep(70);
    }
    a2=a1;

    if(::PeekMessage(&message,NULL,0,0,PM_REMOVE))
    {
        ::TranslateMessage(&message);
        ::DispatchMessage(&message);
    }
}

void CEMG::pulso()
{
    _outp(DATA,0x10); Sleep(1);
    _outp(DATA,0x00); Sleep(1);
    _outp(DATA,0x10);
}

double CEMG::lee_byte()
{
    double byte;
    unsigned int bls,bms;
    _outp(DATA,0x48);
    bls=(_inp(STATUS) ^ 0x80) & 0xF0;

```



```

    _outp(DATA,0x40);
    bms=(_inp(STATUS) ^ 0x80) & 0xF0;
    byte=(bms+(bms/16)) * 19.6e-3;
    return byte;
}

```

Botón *Cancelar Prueba*

```

void CEMG::OnCancelar()
{
    // TODO: Add your control notification handler code here
    CRect rect(50,100,790,250);
    cont = 1;
    m_inicia.EnableWindow(TRUE);
    m_cancelar.EnableWindow(FALSE);
    RedrawWindow(rect,NULL,RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
}

```

PANTALLA CONDUCTANCIA DE LA PIEL (GSR)

Botón *Iniciar Prueba*

```

void CGSR::OnIniciar()
{
    // TODO: Add your control notification handler code here
    MSG message;
    CClientDC dc(this);
    CPen pluma(PS_SOLID,1,RGB(0,0,0));
    dc.SelectObject(&pluma);
    CRect rect(50,100,790,250);
    int x;
    double byte,a1,a2;
    DATA = 0x0378;
    STATUS = DATA+1;
    m_inicia.EnableWindow(FALSE);
    m_cancelar.EnableWindow(TRUE);
    cont = 0; a2=0;
    for(x=0;cont <= 0;x+=5)
    {
        pulso();
        byte=lee_byte();
        a1=byte*30.0;
        if(x>670)
        {
            x=-5;

```

```

        RedrawWindow(rect,NULL,RDW_INVALIDATE | RDW_UPDATENOW |
RDW_ERASE ),
    }
    else
    {
        dc.MoveTo(50+x,250-(int)a2);
        dc.LineTo(55+x,250-(int)a1);
        Sleep(70);
    }
    a2=a1;

    if (::PeekMessage(&message,NULL,0,0,PM_REMOVE))
    {
        ::TranslateMessage(&message);
        ::DispatchMessage(&message);
    }
}

void CGSR::pulso()
{
    _outp(DATA,0x10); Sleep(1);
    _outp(DATA,0x00); Sleep(1);
    _outp(DATA,0x10);
}

double CGSR::lee_byte()
{
    double byte;
    unsigned int bls,bms;
    _outp(DATA,0x88);
    bls=(_inp(STATUS) ^ 0x80) & 0xF0;
    _outp(DATA,0x80);
    bms=(_inp(STATUS) ^ 0x80) & 0xF0;
    byte=(bms+(bls/16)) * 19.6e-3;
    return byte;
}

```

Botón *Cancelar Prueba*

```

void CGSR::OnCancelar()
{
    // TODO: Add your control notification handler code here
    CRect rect(50,100,790,250);
    cont = 1;
    m_inicia.EnableWindow(TRUE);
}

```

```

m_cancelar.EnableWindow(FALSE);
RedrawWindow(rect,NULL,RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
}

```

PANTALLA RESULTADOS

Botón *Imprimir Resultados*

```

void CRes::OnImprimir()
{
    // TODO: Add your control notification handler code here
    CPrintDialog Print(FALSE,PD_ALLPAGES | PD_USEDEVMODECOPIES
PD_NOPAGENUMS | PD_HIDEPRINTTOFILE | PD_NOSELECTION,NULL);
    Print.DoModal();
    HDC hDC = Print.GetPrinterDC();
    CDC dc;
    dc.Attach(hDC);
    DOCINFO di;
    di.cbSize = sizeof(DOCINFO);
    di.lpszDocName = "Texto";
    di.lpszOutput = NULL;
    if(dc.StartDoc(&di) == -1)
    {
        MessageBox("No se puede iniciar la impresion");
        return;
    }

    CFont f;
    CBrush b; //,*bb;
    CPen pen; //,*pp;
    CString Val1,Val2;
    char buffer[10];
    int i,y,x1,inc1,inc2;
    int R=200,L=2594,T=-130,B=-1832;
    double norm,suma;
    CRect Rec1,Rec2,Rec3;
    DWORD color[8]={RGB(255,0,0), // rojo
                    RGB(0,0,0), // negro
                    RGB(0,255,0), // verde
                    RGB(0,0,255), // azul
                    RGB(255,255,0), // amarillo
                    RGB(255,0,255), // magenta
                    RGB(0,255,255), // cyan
                    RGB(255,255,255)}; // white

    dc.StartPage();

```

```
dc.SetMapMode(MM_LOMETRIC);
b.CreateSolidBrush(RGB(255,255,255));
dc.SelectObject(&b);
Rec1.right=R, Rec1.left=L, Rec1.top=T, Rec1.bottom=B;
dc.Rectangle(Rec1);
Rec2.right=R+995, Rec2.left=R+2213, Rec2.top=T-250, Rec2.bottom=T-1110;
dc.Rectangle(Rec2);
Rec3.right=Rec2.right, Rec3.left=Rec2.left, Rec3.top=T-1240, Rec3.bottom=T-1440;
b.DeleteObject();

f.CreateFont(-100,0,0,0,600,FALSE,FALSE,0,ANSI_CHARSET,
            OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,
            DRAFT_QUALITY,DEFAULT_PITCH||FF_ROMAN,
            "Times New Roman");
dc.SelectObject(&f);
dc.TextOut(R+450,T-100,"Resultados de las Pruebas");

f.CreateFont(-25,0,0,0,700,TRUE,FALSE,0,ANSI_CHARSET,
            OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
            DEFAULT_QUALITY,DEFAULT_PITCH||FF_ROMAN,
            "Abadi MT Condensed Light");
dc.SelectObject(&f);
inc1 = -(-Rec2.bottom + Rec2.top) / 4;
for(i=0;i<5;i++)
{
    sprintf(buffer,"%2.1f",i*2.5);
    Val1 = buffer;
    y = Rec2.bottom-(inc1*i);
    dc.TextOut(R+870,y+15,Val1);
    dc.MoveTo(Rec2.right,y);
    dc.LineTo(Rec2.right-50,y);
}

f.CreateFont(45,0,0,0,700,FALSE,FALSE,0,ANSI_CHARSET,
            OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
            DEFAULT_QUALITY,DEFAULT_PITCH||FF_ROMAN,
            "Times New Roman");
dc.SelectObject(&f);
dc.TextOut(R+50,T-300,Nombre);
dc.TextOut(R+50,T-350,Eval);
dc.TextOut(R+50,T-400,Fecha);

f.CreateFont(45,0,0,0,700,FALSE,FALSE,0,ANSI_CHARSET,
            OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
            DEFAULT_QUALITY,DEFAULT_PITCH||FF_ROMAN,
            "Times New Roman");
dc.SelectObject(&f);
```

```

dc.SetTextColor(color[3]);
dc.TextOut(R+50,T-550,"Parámetro");
dc.TextOut(R+300,T-550,"Total");
dc.TextOut(R+450,T-550,"Normalizada");
// Pinta nombre de parametros
f.CreateFont(40,0,0,0,700,TRUE,FALSE,0,ANSI_CHARSET,
            OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
            DEFAULT_QUALITY,DEFAULT_PITCH||FF_ROMAN,
            "Times New Roman");

dc.SelectObject(&f);
for(i=0;i<7;i++)
{
    dc.TextOut(R+50,T-610-40*i,nom_p[i]);
    if(val_p[i] == 0)
    {
        Val1 = "---";
        Val2 = "---";
    }
    else
    {
        sprintf(buffer,"%3.2f",val_p[i]);
        Val1 = buffer;
        norm = normaliza(i);
        sprintf(buffer,"%3.2f",norm);
        Val2 = buffer;
    }
    dc.TextOut(R+320,T-610-40*i,Val1);
    dc.TextOut(R+500,T-610-40*i,Val2);
}
inc1 = (Rec2.left - Rec2.right) / 7;
inc2 = -(-Rec2.bottom + Rec2.top) / 10;
x1 = Rec2.right;
dc.SetTextColor(color[1]);
dc.SetBkColor(color[6]);
f.CreateFont(-22,0,0,0,700,FALSE,FALSE,0,ANSI_CHARSET,
            OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
            DEFAULT_QUALITY,DEFAULT_PITCH||FF_SWISS,
            "Times New Roman");

dc.SelectObject(&f);
for(i=0;i<7;i++)
{
    if(val_p[i] > 0)
    {
        norm = normaliza(i);
        y = Rec2.bottom - inc2 * (int)norm;
        b.CreateSolidBrush(color[6]);
        dc.SelectObject(&b);
    }
}

```

```

        dc.Rectangle(x1,Rec2.bottom,x1+incl,y);
    }
    dc.TextOut(x1+10,T-1080,nom_p[i]);
    x1 += incl;
}

dc.SetBkColor(color[7]);
f.CreateFont(-55,0,0,0,700,FALSE,FALSE,0,ANSI_CHARSET,
    OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
    DEFAULT_QUALITY,DEFAULT_PITCH||FF_SWISS,
    "Times New Roman");

dc.SelectObject(&f);
dc.TextOut(R+70,T-1000,"Nivel de Estrés");
suma = 0;
for(i=0;i<7;i++)
    if(val_p[i] > 0)
    {
        norm = normaliza(i);
        suma += norm;
    }
sprintf(buffer,"%4.2f",suma);
b.CreateSolidBrush(color[7]);
dc.SelectObject(&b);
dc.Rectangle(R+50,T-1060,R+600,T-1320);
f.CreateFont(-150,0,0,0,700,FALSE,FALSE,0,ANSI_CHARSET,
    OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
    DEFAULT_QUALITY,DEFAULT_PITCH||FF_SWISS,
    "Times New Roman");

dc.SelectObject(&f);
dc.SetTextColor(color[0]);
dc.TextOut(R+100,T-1110,buffer);
dc.SetTextColor(color[1]);
f.CreateFont(-50,0,0,0,700,TRUE,FALSE,0,ANSI_CHARSET,
    OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
    DEFAULT_QUALITY,DEFAULT_PITCH||FF_SWISS,
    "Times New Roman");

dc.SelectObject(&f);
dc.TextOut(R+1350,T-1150,"Estrés - Bienestar");
b.CreateSolidBrush(color[7]);
dc.SelectObject(&b);
dc.Rectangle(Rec3);
f.CreateFont(-24,0,0,0,700,FALSE,FALSE,0,ANSI_CHARSET,
    OUT_CHARACTER_PRECIS,CLIP_DEFAULT_PRECIS,
    DEFAULT_QUALITY,DEFAULT_PITCH||FF_SWISS,
    "Abadi MT Condensed Light");

dc.SelectObject(&f);
incl = (1220) / 7;

```

```
for(i=0;i<8;i++)
{
    sprintf(buffer,"%d",i*10);
    Val1 = buffer;
    x1 = R+995+(int)(inc1*i);
    dc.TextOut(x1-15,Rec3.bottom-60,Val1);
    dc.MoveTo(x1,Rec3.bottom);
    dc.LineTo(x1,Rec3.bottom-50);
}
b.CreateSolidBrush(color[0]);
dc.SelectObject(&b);
inc2 = 1220/70;
dc.Rectangle(R+995,Rec3.top,R+995+(int)(inc2*suma),Rec3.bottom);

b.DeleteObject();
dc.EndPage();
dc.EndDoc();
GlobalFree(Print_m_pd.hDevMode);
GlobalFree(Print_m_pd.hDevNames);
DeleteDC(hDC);
}
```

Botón Regresar al Menú Principal

```
void CRes::OnSalir()
{
    // TODO: Add your control notification handler code here
    OnOK();
}
```

GLOSARIO

Cortisol o Cortisona. Hormona corticosuprarrenal que posee propiedades antiinflamatorias y metabólicas.

Electromiograma. Estudio de los fenómenos eléctricos que acompañan la contracción muscular.

Libido. Energía de la pulsión sexual.

Prolactina. Hormona hipofisaria responsable de la secreción láctea del postpartum e implicada en la inhibición de la fertilidad materna que se observa en dicho periodo.

Sistema Cardiocirculatorio. Este sistema se ocupa del buen funcionamiento de los órganos que aseguran la circulación de la sangre y de la linfa (corazón, arterias, capilares y venas).

Sistema Endocrino. Este sistema regula el desarrollo y las funciones de las glándulas endocrinas, como son la hipófisis, el tiroides, los ovarios, los testículos y las suprarrenales.

Sistema Neurovegetativo. Este sistema, también llamado sistema nervioso autónomo, regula la vida vegetativa, es decir, la vida puramente orgánica a excepción de las funciones reproductoras. Este sistema está compuesto por los dos subsistemas que inervan las mismas vísceras, pero tienen efectos antagónicos, estos son:

- **Simpático.** La excitación del simpático acelera el corazón, aumenta la tensión arterial, dilata los bronquios y ralentiza las contracciones del tubo digestivo. El parasimpático tiene una acción inversa; del equilibrio entre los dos resulta el funcionamiento normal de organismo.
- **Parasimpático.** Este subsistema aminora el ritmo cardíaco y acelera los movimientos del tubo digestivo.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- [1] Buckingham J. C. Gilies G. E., Cowell A. M.
Stress, stress hormones and the immune system
Wiley, Chichester 1997
- [2] Kline P
The handbook of psychological testing
Astrolabio, Roma 1993
- [3] Henry J. P
Stress, neuroendocrine patterns and emotional response
Wiley, New York 1990
- [4] Santo Di Nuovo, Luciano Rispoli, Emilia Genta
MISURARE LO STRESS. Il test M.S.P. e altri strumenti per una valutazione integrata
FrancoAngeli/Linea Test 2000
- [5] "What is stress?"
<http://www.healthy.net/library/books/hoffman/nervousconditions/STRESS.HTM>
- [6] Clinical Biofeedback Therapy
<http://www.innernet.net/doco/sjo/biofeedback.html>
- [7] William Nunley, J. Scott Bechtel
INFRARED OPTOELECTRONICS, Devices and Applications
Marcel Dekker, Inc. 1897
- [8] Ray M. Marston
Photosensitive Devices
Electronics Now, July 1992
- [9] Ray M. Marston
Tone Decoder
Electronics Now, July 1994
- [10] Marco Antonio Acevedo, Jorge Sosa Pedroza, Alberto Gutiérrez Domínguez
Tarjeta de adquisición de datos para la detección de obstáculos y niveles de luz visible
Cuarto Congreso de Proyectos de Investigación, IPN ESIME, 1999

- [11] Donald L. Schilling, Charles Belove
ELECTRONIC CIRCUITS, Discrete and Integrated
McGraw-Hill International Editions
Tercera Edición, 1989

- [12] Ronald J. Tocci
SISTEMAS DIGITALES, Principios y Aplicaciones
Prentice Hall
Quinta Edición, 1993

- [13] Thomas L. Floyd
Electronic Devices
MacMillan Publishing Company, New York
Third Edition

- [14] THE IEEE EXPERTS, Parallel Port Background
<http://www.fapo.com/porthist.htm>

- [15] Peter H. Anderson, Department of Electrical Engineering Morgan State University
USE OF A PRINTER PORT FOR CONTROL AND DATA ACQUISITION
<http://et.nmsu.edu/~etti/fall96/computer/printer/printer.html>

- [16] FAST and LS TTL Data
Motorola Inc.
Fifth Edition

- [17] GaAs Infrared Emitting Diodes TSUS520
www.vishay.com/temporary/datasheets/optoelectronics/emitters/tsus520.html

- [18] Silicon NPN Phototransistor
www.vishay.com/temporary/datasheets/optoelectronics/detectors/bpw96.html

- [19] TL082 Wide Bandwidth Dual JFET Input Operational Amplifier
www.national.com/pf/TL/TL082.html

- [20] LM567/LM567C Tone Decoder
www.national.com/pf/LM/LM567.html

- [21] LM555 Timer
www.national.com/pf/LM/LM555.html

- [22] ADC0804 8-Bit μ P Compatible A/D Converter
www.national.com/pf/AD/ADC0804.html

-
- [23] Francisco Javier Ceballos
Visual C++: Aplicaciones para Win32
ALFAOMEGA, 1998
- [24] Francisco Javier Ceballos
Visual C++ 6.0: Programación avanzada en Win32
ALFAOMEGA, 1999
- [25] David J. Kruglinski
Visual C++: Programación avanzada
McGraw-Hill / INTERAMERICANA DE ESPAÑA, S. A. U., 1997
- [26] Kate Gregory
Using Visual C++ 6: The One Source for Comprehensive Solutions
QUE Special Edition, 1998
- [27] El Pequeño Larousse Ilustrado, Diccionario Enciclopédico
1998

Los abajo firmantes, integrantes de jurado para el examen de grado que sustentará la **Srita. Ma. Elena Acevedo Mosqueda**, declaramos que hemos revisado la tesis titulada:

“SISTEMA DIGITAL PARA MEDIR EL NIVEL DE ESTRÉS EN UN SER HUMANO”.

Y consideramos que cumple con los requisitos para obtener el Grado de Maestra en Ciencias en la especialidad en Ingeniería Eléctrica opción Computación.

Atentamente,

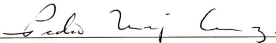
Dr. Adriano de Luca Pennacchia



Dr. José Oscar Olmedo Aguirre



Dr. Pedro Mejía Alvarez



*

ESTA

T E S I S

SE IMPRIMIO EN

CENTRO DE COPIADO

F U T U R O

AV. INSTITUTO POLITECNICO NACIONAL

NUM. 1911 COL LINDAVISTA TEL 55869295

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL

BIBLIOTECA DE INGENIERIA ELECTRICA
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro
antes del vencimiento de préstamo señalado
por el último sello.

21 NOV. 2003

11 DIC. 2003

DEVOLUCION

