

## Capítulo 2. Obtención de las señales cardiacas.

El electrocardiógrafo diseñado registra las doce derivaciones de manera similar a los electrocardiógrafos con impresión en cinta de papel, es decir, la adquisición de cada derivación es una detrás de otra durante cierto tiempo (el registro de la señales es no simultáneo).

Para realizar un electrocardiograma a un paciente se le conectan electrodos superficiales que están en contacto con su piel; para obtener cada señal se multiplexan los arreglos de electrodos correspondientes a cada derivación empleando un PIC, multiplexores e interruptores analógicos, todos ellos incluidos en una misma tarjeta electrónica.

### 2.1 Transductores (Electrodos).

Propiamente un *transductor* es un dispositivo que es capaz de transformar energía de cierta naturaleza en energía de distinta naturaleza, por lo tanto cabe aclarar que la palabra transductor no aplica para este proyecto de tesis y fue elegida erróneamente al momento de estructurar el temario original, la palabra correcta referente a este subcapítulo es *electrodos*.

Existen en el mercado una variedad de electrodos diseñados para registrar electrocardiogramas. Los más comunes son de tres tipos; electrodos de copa de succión, electrodos de pulsera, y los electrodos desechables (**Figura 2.1**).

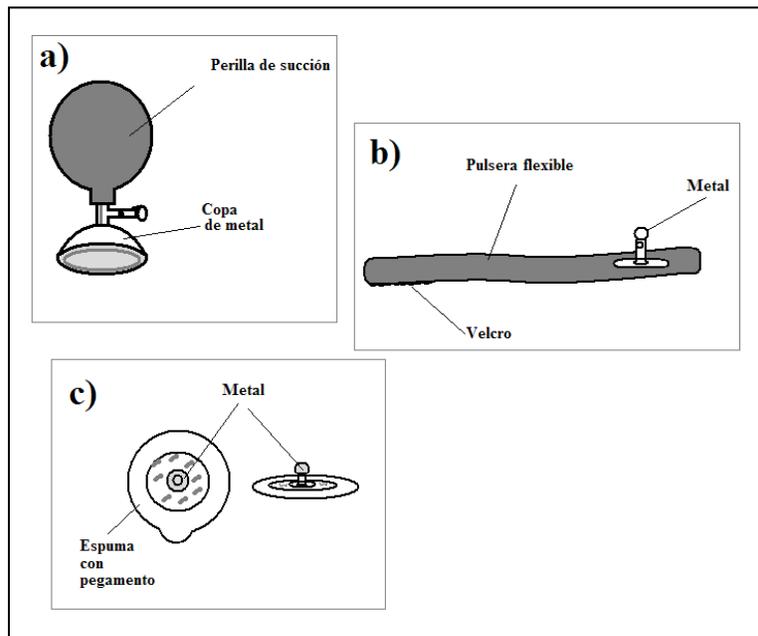


Figura 2.1 Distintos tipos de electrodos. a) de copa de succión, b) de pulsera, c) desechables.

Tanto los electrodos de pulsera como los de copa de succión son reutilizables (por lo tanto requieren esterilización), están hechos con metales como la plata, aleaciones de zinc, cobre y nickel, y para usarlos se necesita aplicar una pasta electrolítica a la piel del paciente que mejora la conducción.

Los electrodos desechables contienen a su alrededor una esponja adhesiva (para su colocación rápida y fácil) que rodea a un trozo moldeado de plata (Ag) recubierto en un lado con una delgada capa de cloruro de plata (AgCl), ambos metales se ponen en contacto con un electrolito gelatinoso. Debido a su practicidad los electrodos desechables son lo que se usaron en este proyecto de tesis.

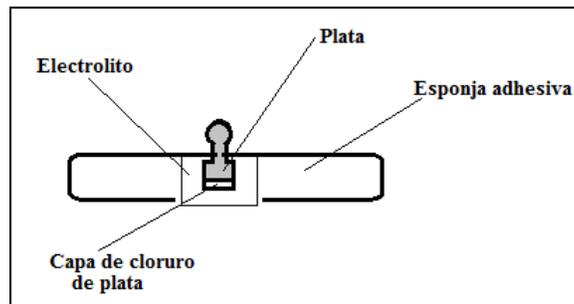


Figura 2.2 Esquema de materiales de un electrodo desechable Ag-AgCl.

Este tipo de electrodos (Ag-AgCl) presenta una variación en su impedancia dependiendo de la frecuencia de la señal aplicada debido a las capacitancias formadas por las interfaces de los metales y el electrolito, dicha variación está comprendida aproximadamente entre 100 [ $\Omega$ ] y 2 [k $\Omega$ ] (para las frecuencias entre 10 [kHz] y DC respectivamente).

Asimismo en estos electrodos desechables existe una diferencia de potencial eléctrico entre ambos metales que lo conforman y equivale aproximadamente a 0.57 [V]\*. Teóricamente todos los electrodos deberían presentar el mismo voltaje y contrarrestarse entre sí, sin embargo en la práctica esto no sucede y por lo tanto se tiene adicionado a nuestra señal de ECG un nivel de voltaje en DC del orden de milivolts que para eliminarse debe de ser filtrado con un filtro paso altas (el capítulo 5 aborda el tema de filtrado de señales).

\* Obtenido de los valores de potenciales correspondientes a electrodos de Ag (+0.7991 [V]) y AgCl (+0.2225 [V]) referidos a un electrodo de hidrógeno (0 [V]), contenidos en la tabla "Electrode potentials for commonly used materials in electrodes" presentada por Geddes L. A. y L.E. Baker, 1989. *Principles of applied biomedical instrumentation*, p. 317.

## 2.2 Multiplexado de señales.

### 2.2.1 Localización de derivaciones y multiplexado analógico.

Para la obtención de cada una de las doce derivaciones se debe contar con un circuito que sea capaz de realizar físicamente el arreglo o disposición eléctrica para cada derivación, es decir un interruptor que conecte o desconecte los electrodos correspondientes a la derivación deseada previo a la etapa de amplificación de la señal (**Figura 2.3**). Para realizar esta tarea se hizo uso de un PIC 16F877A, de dos circuitos integrados modelo DG442 (interruptor analógico) y tres circuitos integrados modelo HCF4051 (multiplexor analógico), todos ellos incluidos en una misma tarjeta.

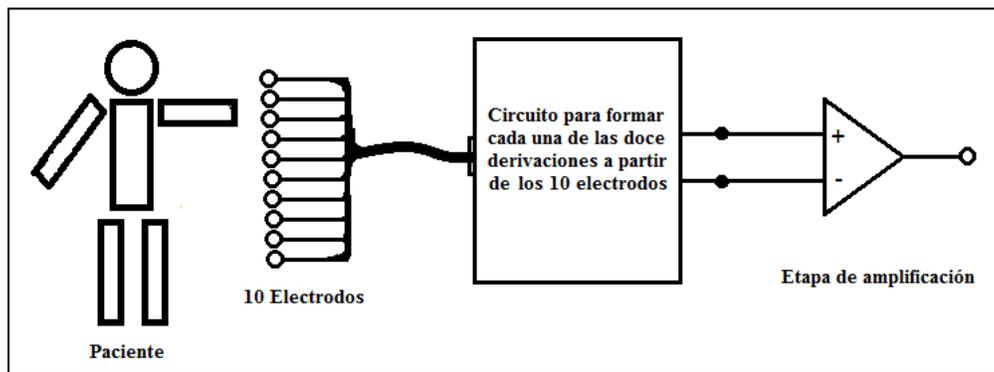


Figura 2.3 Diagrama a bloques del multiplexado de señal de ECG antes de la etapa de amplificación.

Los interruptores y multiplexores analógicos a diferencia de los digitales tienen la característica de que la señal de entrada y de salida pueden ser analógicas y las líneas de selección son digitales como usualmente los conocemos, esto permite que las señales analógicas del corazón tomadas a partir de la piel por medio de los electrodos puedan formar cada derivación para su posterior amplificación.

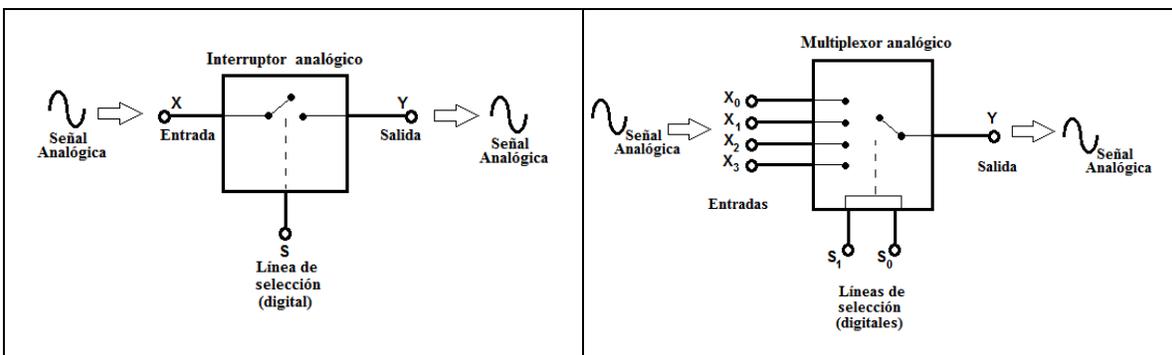


Figura 2.4 Esquema de un multiplexor y un interruptor analógico.

Para diseñar la tarjeta de multiplexado que incluye a los interruptores y multiplexores analógicos, se tuvo que conocer la localización de los electrodos en el cuerpo y su conexión hacia la etapa de amplificación para obtener cada derivación. Las doce derivaciones de electrocardiografía utilizadas por convención comprenden tres derivaciones bipolares (DI, DII, DIII), tres derivaciones unipolares aumentadas (aVR, aVL, aVF) y seis derivaciones unipolares precordiales (V1 a V6):

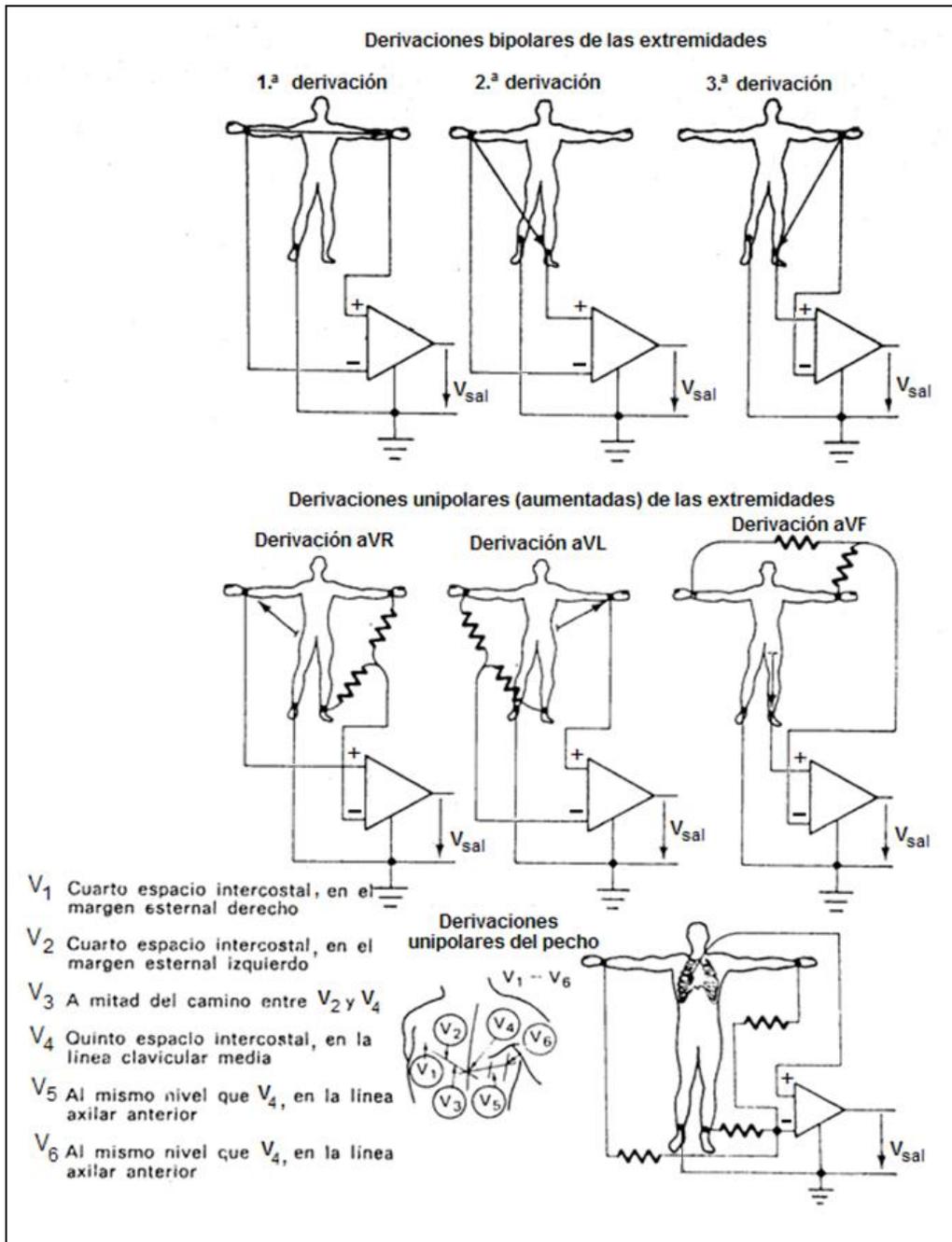


Figura 2.5 Diagrama de conexión de las doce derivaciones para electrocardiografía.  
Fuente: Cromwell, Leslie, et.al.,1980, *Instrumentación y medidas biomédicas*, p. 88.

En el diagrama anterior se observan cuatro electrodos para las extremidades del paciente y seis electrodos para la zona del pecho (llamados precordiales), es decir se tienen diez electrodos conectados de los cuales se obtienen las doce derivaciones.

A partir de la localización de las conexiones para formar las doce derivaciones se realizó la siguiente tabla que muestra qué electrodos se utilizan en cada derivación con su respectiva polaridad en las entradas del amplificador de señales (por convención escritos por su abreviatura en inglés correspondiente a cada extremidad).

		(+)	(-)
D	DI	LA	RA
	DII	LL	RA
	DIII	LL	LA
R	aVR	RA	LA, LL
I	aVL	LA	RA, LL
V	aVF	F	R, LL
A	V1	V1	RA, LA, LL
C	V2	V2	RA, LA, LL
I	V3	V3	RA, LA, LL
Ó	V4	V4	RA, LA, LL
N	V5	V5	RA, LA, LL
	V6	V6	RA, LA, LL

Tabla 2.1 Electrodos utilizados para formar cada derivación

Con base en la **tabla 2.1** surgió la **figura 2.6** que muestra la manera en que se conectaron los diez electrodos a los interruptores, multiplexores y hacia las entradas del amplificador de instrumentación. En ella se observa que:

- La tarjeta de multiplexado tiene como función obtener únicamente dos salidas (positiva y negativa) para cada derivación formando un arreglo en particular a partir de los diez electrodos conectados al paciente. Estas salidas provienen de los multiplexores “A” y “B” e irán conectadas a las dos entradas de la etapa de amplificación.
- Las doce combinaciones se hacen con los diez electrodos: LA (brazo izquierdo), RA (brazo derecho), LL (pierna izquierda), RL (pierna derecha; es el electrodo “común” o de referencia respecto a todos los demás y es necesario para registrar todas las señales), y V1 a V6 (precordiales).
- Los primeros tres electrodos mencionados van conectados de manera duplicada a las entradas de los interruptores analógicos (el mismo electrodo a dos entradas distintas) para que las salidas repetidas se conecten a tres resistores unidos por un nodo común, el cual es llamado *terminal central de Wilson* y que sirve para obtener todas las derivaciones unipolares (aVR a V6). Tanto la *terminal central de Wilson*, como las salidas no duplicadas de los interruptores se conectaron a las entradas de los MUX “A” y “B”.

- Para la obtención de las derivaciones precordiales se conectaron los electrodos V1 a V6 a las entradas del MUX "P" y su salida se conectó a la entrada indicada en el multiplexor A. Para formar cada arreglo posible se habilita o deshabilita (según el caso) cada terminal de selección tanto de los interruptores como de los multiplexores mediante señales digitales provenientes del PIC 16F877A.

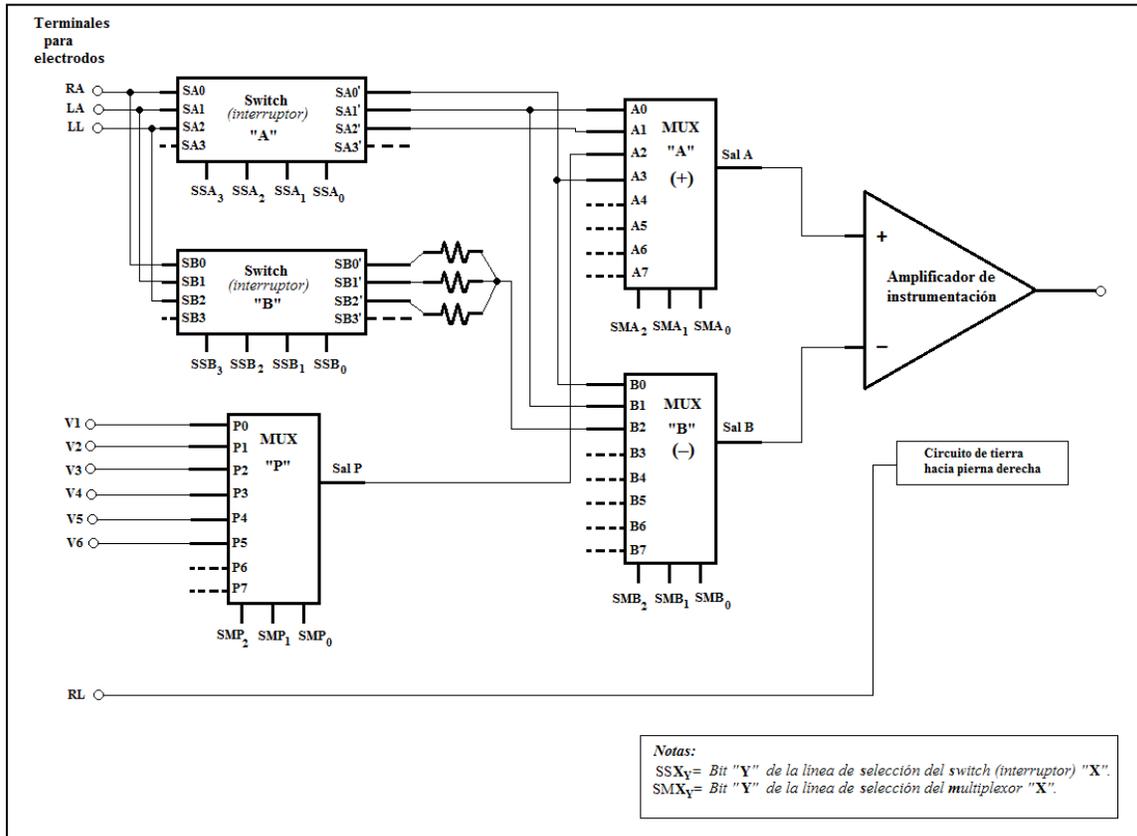


Figura 2.6 Diagrama eléctrico "guía" para la conexión de los electrodos con los interruptores y multiplexores.

A partir de la **figura 2.6** se diseñó la tarjeta de multiplexado mostrada a continuación que incluye al PIC 16F877A, interruptores DG442 y multiplexores HCF4051 (la pantalla LCD alfanumérica y el teclado se explican en la siguiente sección de este capítulo):

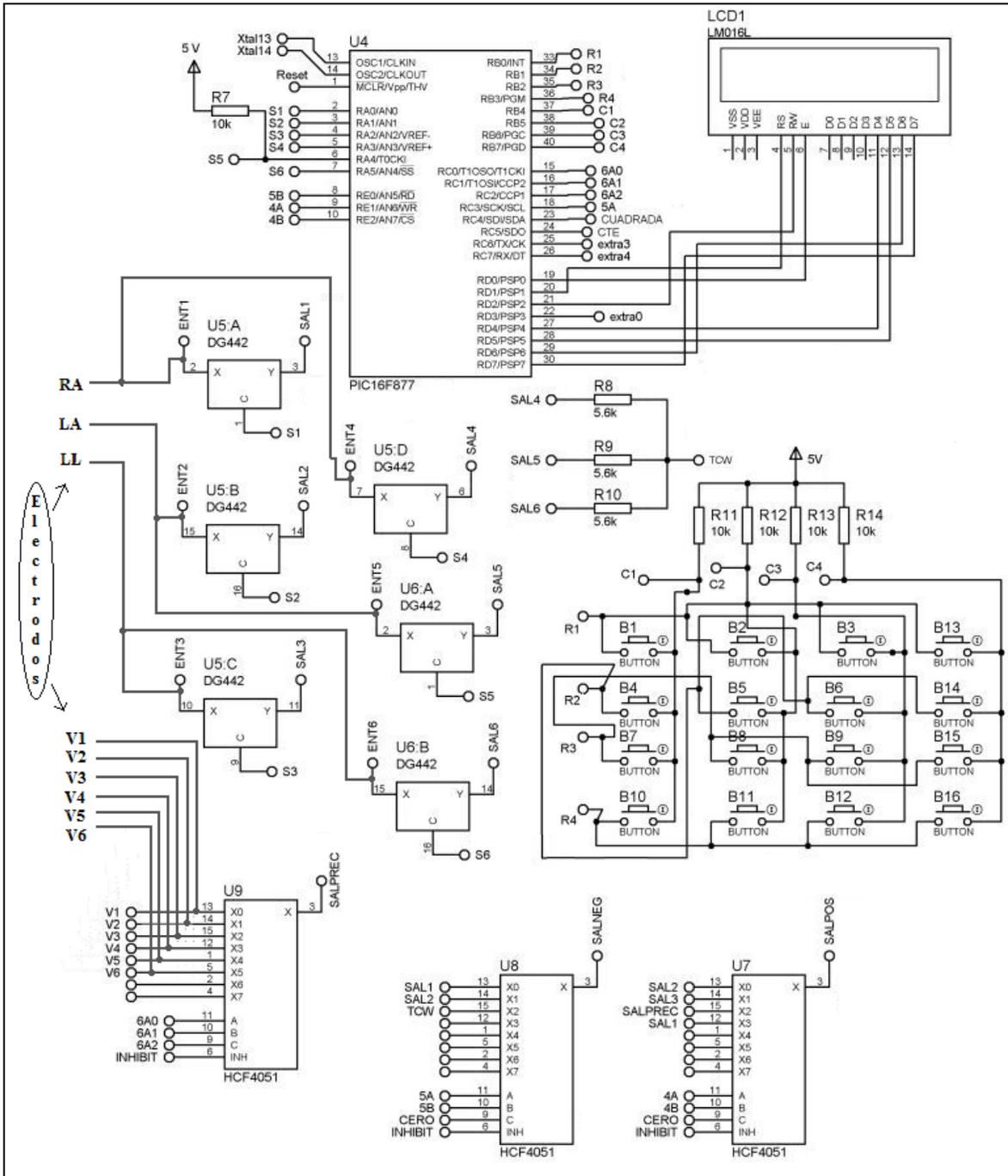


Figura 2.7 Diagrama eléctrico de la tarjeta de multiplexado.

y su respectivo circuito impreso:

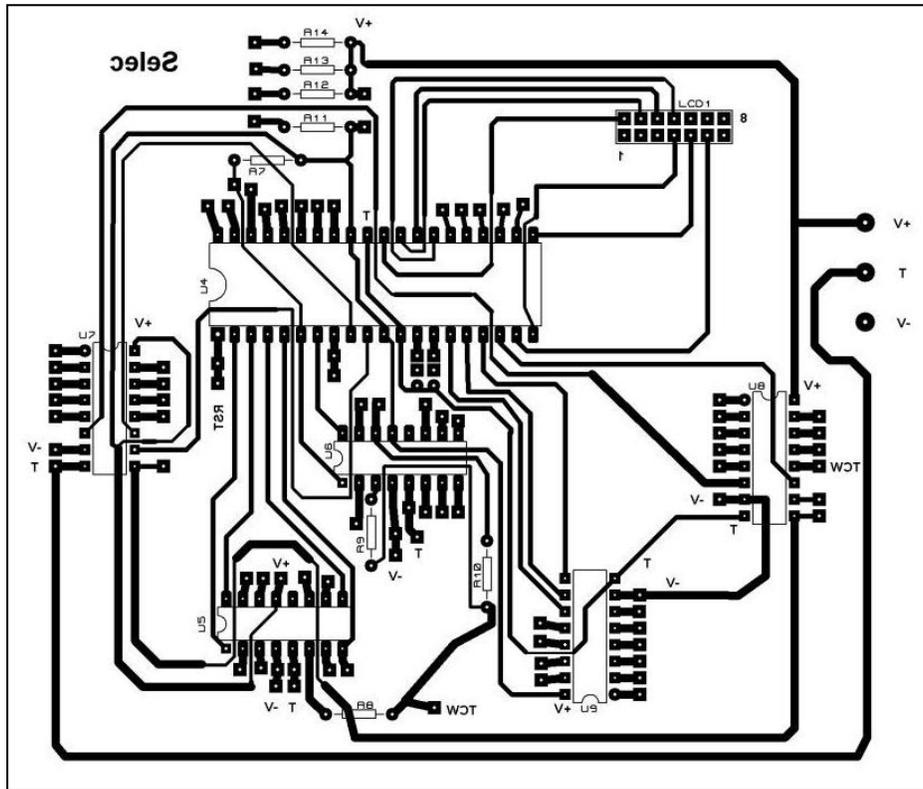


Figura 2.8 Circuito impreso de la tarjeta de multiplexado.

## 2.2.2 Selección de derivaciones.

La selección de derivaciones en el electrocardiógrafo se realiza mediante líneas de selección digitales (señales de control) provenientes del PIC16F877A, el código fuente para programarlo fue realizado en lenguaje C en el compilador "CCS Compiler". Dichas señales salen por el puerto "A" (bit 0 a bit 5), el puerto "E" (bit 0 a bit 2) y el puerto "C" (bit 0 a bit 3) y a su vez están conectadas a las líneas de selección de los interruptores y multiplexores analógicos. La siguiente tabla de verdad sirvió para la programación del PIC y está basada en la **tabla 2.1** y en la **figura 2.6** de la sección anterior.

		Líneas de selección para interruptores y multiplexores/ Bits de salida del PIC												
		SSA3	SSA2	SSA1	SSA0	SSB1	SSB0	SMA1	SMA0	SMB1	SMB0	SMP2	SMP1	SMP0
		A5	A4	A3	A2	A1	A0	E2	E1	E0	C3	C2	C1	C0
D	DI	0	0	0	0	1	1	0	0	0	0	*	*	*
	DII	0	0	0	1	0	1	0	1	0	0	*	*	*
E	DIII	0	0	0	1	1	0	0	1	0	1	*	*	*
	aVR	1	1	0	0	0	1	1	1	1	0	*	*	*
I	aVL	1	0	1	0	1	0	0	0	1	0	*	*	*
	aVF	0	1	1	1	0	0	0	1	1	0	*	*	*
V	V1	1	1	1	0	0	0	1	0	1	0	0	0	0
	V2	1	1	1	0	0	0	1	0	1	0	0	0	1
A	V3	1	1	1	0	0	0	1	0	1	0	0	1	0
	V4	1	1	1	0	0	0	1	0	1	0	0	1	1
C	V5	1	1	1	0	0	0	1	0	1	0	1	0	0
	V6	1	1	1	0	0	0	1	0	1	0	1	0	1

**Notas sobre :**  
 SSXY= Bit "Y" de la línea de Selección del Switch (interruptor) "X".  
 SMXY= Bit "Y" de la línea de Selección del Multiplexor "X".

Tabla 2.2 Tabla de verdad programada para los puertos de salida del PIC y conectados a los interruptores y multiplexores para formar cada derivación.

Para facilitar la programación en cada puerto, los valores anteriores se convirtieron a código hexadecimal:

		Puerto A	Puerto E	Puerto C
D	DI	03	00	00
	DII	05	02	00
E	DIII	06	02	08
	aVR	31	07	00
I	aVL	2A	01	00
	aVF	1C	03	00
V	V1	38	05	00
	V2	38	05	01
A	V3	38	05	02
	V4	38	05	03
C	V5	38	05	04
	V6	38	05	05

Tabla 2.3 Valores lógicos en código hexadecimal para la programación de cada puerto del PIC 16F877A para formar cada derivación.

La manera en que el PIC se “comunica” con el usuario para que este último elija cada derivación es por medio de un teclado matricial de cuatro renglones por cuatro columnas (4x4) conectado a los ocho bits del puerto “B” del PIC. Para visualizar las instrucciones leídas en el teclado se incluyó una pantalla de cristal líquido (LCD) monocromática de 2 renglones por 16 columnas (2x16) conectada al puerto “D” del PIC18F877A.

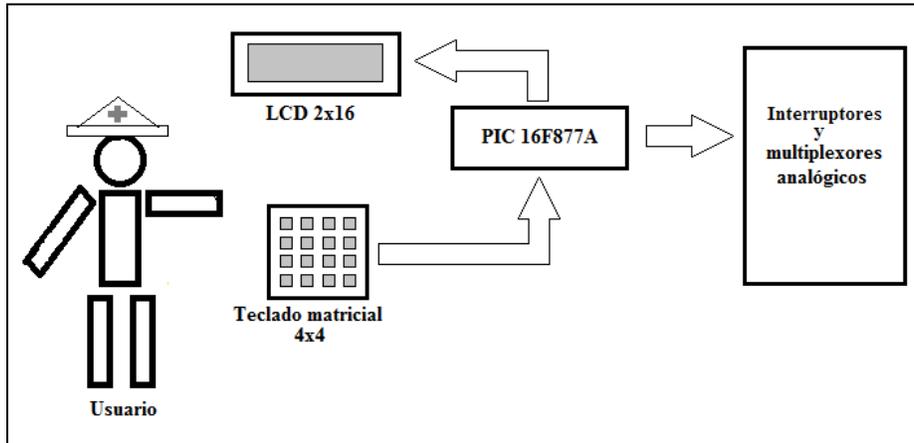


Figura 2.9 Comunicación entre el usuario y el PIC 16F877A por medio de un teclado matricial y una pantalla LCD.

El teclado matricial se realizó en circuito impreso con el fin de tener un espacio adecuado para la descripción de cada tecla en la carátula del teclado:

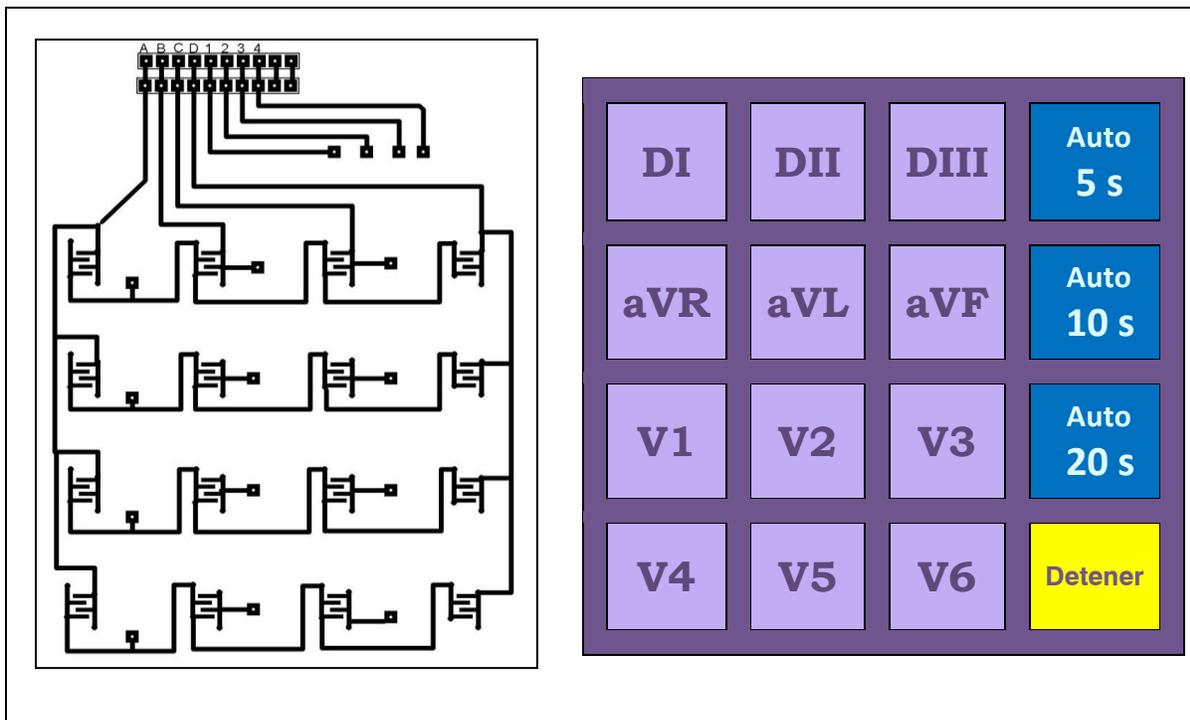


Figura 2.10 Circuito impreso y carátula del teclado matricial implementado.

Para finalizar este capítulo tenemos el diagrama de flujo realizado para el programa grabado en el PIC16F877A:

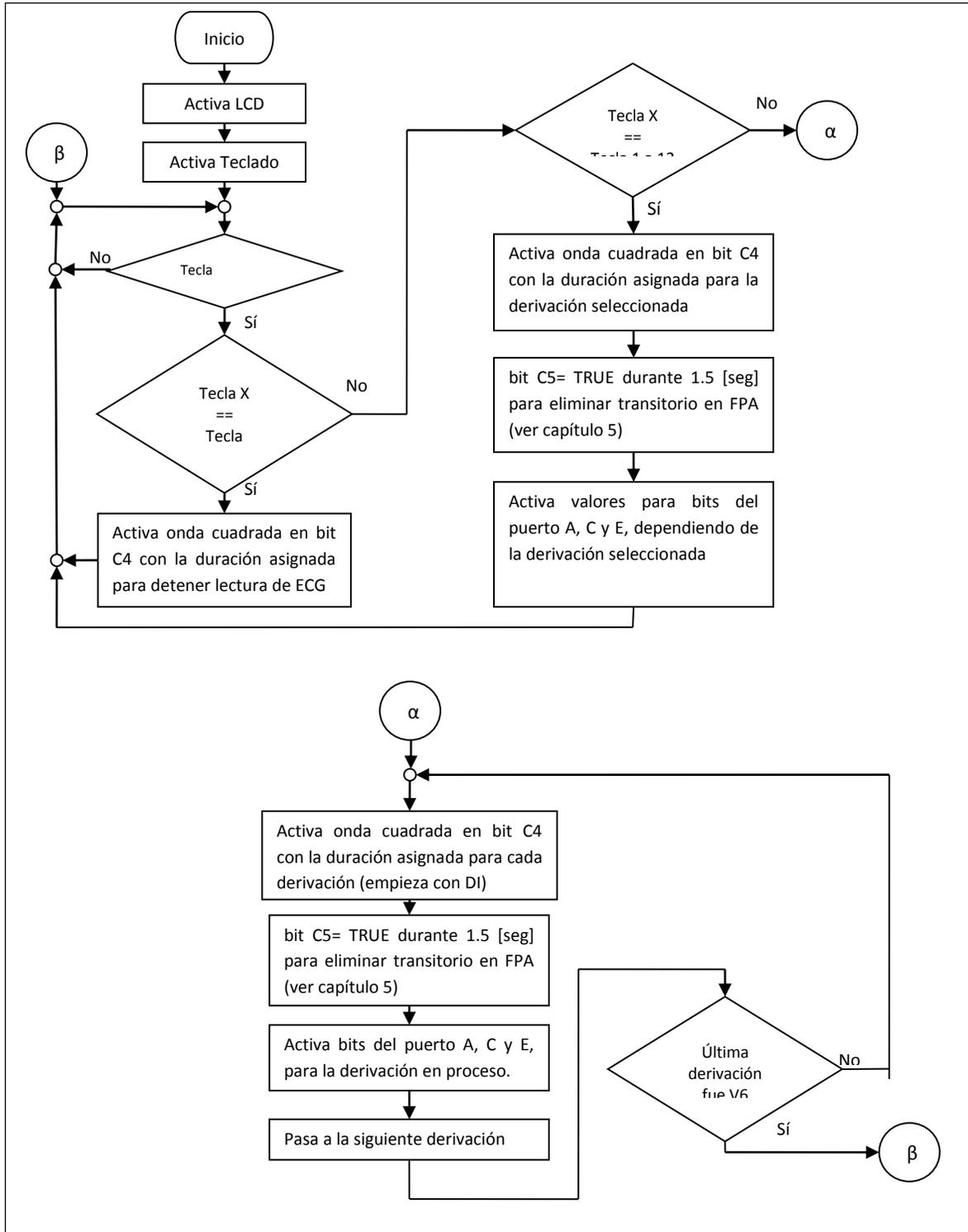


Figura 2.11 Diagrama de flujo del programa grabado en el PIC18F877A para la selección de derivaciones.

Y el código del programa en lenguaje C (se lee de arriba a abajo comenzando con la columna izquierda y luego la derecha):

<pre>//Programa para el PIC de la tarjeta de multiplexado #include &lt;16F877A.h&gt;  #FUSES NOWDT #FUSES XT #FUSES PUT #FUSES NOPROTECT #FUSES NODEBUG #FUSES BROWNOUT #FUSES NOLVP #FUSES NOCPD #FUSES NOWRT  #use delay(clock=4096000)  #define use_portd_lcd TRUE #define use_portb_kbd TRUE  #define use_porta_salid6 TRUE #define use_porte_salid3 TRUE #define use_portc_salid8 TRUE  #include &lt;Auxi_Tecla.c&gt; #include &lt;lcd.c&gt;  int manda;  void solito(){     char k;     int j,dispti,x,p5,lug,sup,i;     int de1,de2,de3,de4,de5,de6,de7;     int de8,de9,de10,de11,de12,de13,tiemp;     long disi;      disi=1500;      de1=10;     de2=20;     de3=30;     de4=40;     de5=50;     de6=60;     de7=70;     de8=80;     de9=90;     de10=100;     de11=110;     de12=120;     de13=130;      switch(manda){         case 5: p5=5;                 sup=55;                 dispti=5;                 break;          case 10: p5=5;                 sup=116;                 dispti=10;                 break;     } </pre>	<pre>case 20: p5=5;         sup=250;         dispti=20;         break;         default:             break;     }      for(j=1;j&lt;=1;++j){         lcd_putc('\f');         printf(lcd_putc,"ECG AUTOM %d Seg",dispti);          /***** Cuadrada *****/         for(tiemp=1;tiemp&lt;=de1;++tiemp){             OUTPUT_C(0x10); //indica             delay_ms(50);             OUTPUT_C(0x00); //indica             delay_ms(50);         }         /***** FIN Cuadrada *****/         OUTPUT_A(0x03);         OUTPUT_E(0x00);         OUTPUT_C(0x60);         delay_ms(disi); //disipa         OUTPUT_C(0x40);          for(i=0;i&lt;=sup;++i){ //5 Seg Deriv I             delay_ms(p5);             k=kbd_getc();             if((k!=false) &amp;&amp; (k=='F'))                 return;             lcd_putc("\n Deriv: DI");         }          for(j=1;j&lt;=1;++j){             /***** Cuadrada *****/             for(tiemp=1;tiemp&lt;=de2;++tiemp){                 OUTPUT_C(0x10); //indica                 delay_ms(50);                 OUTPUT_C(0x00); //indica                 delay_ms(50);             }             /***** FIN Cuadrada *****/             OUTPUT_A(0x05);             OUTPUT_E(0x02);             OUTPUT_C(0x60);             delay_ms(disi); //disipa             OUTPUT_C(0x40);              for(i=0;i&lt;=sup;++i){ //5 Seg Deriv II                 delay_ms(p5);                 k=kbd_getc();                 if((k!=false) &amp;&amp; (k=='F'))                      return;                 lcd_putc("\n Deriv: DII");             }         }     } </pre>
---	---

```

for(j=1;j<=1;++j){
  //**** Cuadrada ****
  for(tiemp=1;tiemp<=de3;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  }
  //**** FIN Cuadrada ****
  OUTPUT_A(0x06);
  OUTPUT_E(0x02);
  OUTPUT_C(0x68);
  delay_ms(disi); //disipa
  OUTPUT_C(0x48);

  for(i=0;i<=sup;++i){ //5 Seg Deriv III
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
    lcd_putc("\n Deriv: DIII");
  }
}

for(j=1;j<=1;++j){
  //**** Cuadrada ****
  for(tiemp=1;tiemp<=de4;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //**** FIN Cuadrada ****
  OUTPUT_A(0x31);
  OUTPUT_E(0x07);
  OUTPUT_C(0x60);
  delay_ms(disi); //disipa
  OUTPUT_C(0x40);
  lcd_putc('\f');
  printf(lcd_putc,"ECG AUTOM %d Seg",dispti);
  lcd_putc("\n Deriv: avR");

  for(i=0;i<=sup;++i){ //5 Seg Deriv avR
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
  }
}

for(j=1;j<=1;++j){
  //**** Cuadrada ****
  for(tiemp=1;tiemp<=de5;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //**** FIN Cuadrada ****
  OUTPUT_A(0x2A);
  OUTPUT_E(0x01);
  OUTPUT_C(0x60);
  delay_ms(disi); //disipa
  OUTPUT_C(0x40);
}

for(i=0;i<=sup;++i){ //5 Seg Deriv avL
  delay_ms(p5);
  k=kbd_getc();
  if((k!=false) && (k=='F'))
    return;
  lcd_putc("\n Deriv: avL");
}

for(j=1;j<=1;++j){
  //**** Cuadrada ****
  for(tiemp=1;tiemp<=de6;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //**** FIN Cuadrada ****
  OUTPUT_A(0x1C);
  OUTPUT_E(0x03);
  OUTPUT_C(0x60);
  delay_ms(disi); //disipa
  OUTPUT_C(0x40);

  for(i=0;i<=sup;++i){ //5 Seg Deriv avF
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
    lcd_putc("\n Deriv: avF");
  }

  for(j=1;j<=1;++j){
    //**** Cuadrada ****
    for(tiemp=1;tiemp<=de7;++tiemp){
      OUTPUT_C(0x10); //indica
      delay_ms(50);
      OUTPUT_C(0x00); //indica
      delay_ms(50);
    } //**** FIN Cuadrada ****
    OUTPUT_A(0x38);
    OUTPUT_E(0x05);
    OUTPUT_C(0x60);
    delay_ms(disi); //disipa
    OUTPUT_C(0x40);
    lcd_putc('\f');
    printf(lcd_putc,"ECG AUTOM %d Seg",dispti);
    lcd_putc("\n Deriv: V1");

    for(i=0;i<=sup;++i){ //5 Seg Deriv V1
      delay_ms(p5);
      k=kbd_getc();
      if((k!=false) && (k=='F'))
        return;
    }

    for(j=1;j<=1;++j){
      //**** Cuadrada ****
      for(tiemp=1;tiemp<=de8;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
      } //**** FIN Cuadrada ****

```

```

OUTPUT_A(0x38);
OUTPUT_E(0x05);
OUTPUT_C(0x61);
delay_ms(disi); //disipa
OUTPUT_C(0x41);

for(i=0;i<=sup;++i){ //5 Seg Deriv V2
  delay_ms(p5);
  k=kbd_getc();
  if((k!=false) && (k=='F'))
    return;
  lcd_putc("\n Deriv: V2");
}
}

for(j=1;j<=1;++j){
  //***** Cuadrada *****
  for(tiemp=1;tiemp<=de9;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //***** FIN Cuadrada *****
  OUTPUT_A(0x38);
  OUTPUT_E(0x05);
  OUTPUT_C(0x62);
  delay_ms(disi); //disipa
  OUTPUT_C(0x42);

  for(i=0;i<=sup;++i){ //5 Seg Deriv V3
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
    lcd_putc("\n Deriv: V3");
  }
}

for(j=1;j<=1;++j){
  //***** Cuadrada *****
  for(tiemp=1;tiemp<=de10;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //***** FIN Cuadrada *****
  OUTPUT_A(0x38);
  OUTPUT_E(0x05);
  OUTPUT_C(0x63);
  delay_ms(disi); //disipa
  OUTPUT_C(0x43);

  for(i=0;i<=sup;++i){ //5 Seg Deriv V4
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
    lcd_putc("\n Deriv: V4");
  }
}

for(j=1;j<=1;++j){
  //***** Cuadrada *****
  for(tiemp=1;tiemp<=de11;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //***** FIN Cuadrada *****
  OUTPUT_A(0x38);
  OUTPUT_E(0x05);
  OUTPUT_C(0x64);
  delay_ms(disi); //disipa
  OUTPUT_C(0x44);

  for(i=0;i<=sup;++i){ //5 Seg Deriv V5
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
    lcd_putc("\n Deriv: V5");
  }
}

for(j=1;j<=1;++j){
  //***** Cuadrada *****
  for(tiemp=1;tiemp<=de12;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //***** FIN Cuadrada *****
  OUTPUT_A(0x38);
  OUTPUT_E(0x05);
  OUTPUT_C(0x65);
  delay_ms(disi); //disipa
  OUTPUT_C(0x45);

  for(i=0;i<=sup;++i){ //5 Seg Deriv V6
    delay_ms(p5);
    k=kbd_getc();
    if((k!=false) && (k=='F'))
      return;
    lcd_putc("\n Deriv: V6");
  }
}

  //***** Cuadrada ***** FINALIZANDO
  for(tiemp=1;tiemp<=de13;++tiemp){
    OUTPUT_C(0x10); //indica
    delay_ms(50);
    OUTPUT_C(0x00); //indica
    delay_ms(50);
  } //***** FIN Cuadrada *****
  delay_ms(20);
  OUTPUT_A(0x00);
  OUTPUT_E(0x00);
  OUTPUT_C(0x00);
}
}

```

<pre> lcd_putc('\f'); lcd_gotoxy(2,1); lcd_putc("Finalizando"); delay_ms(230); lcd_putc('\f'); lcd_gotoxy(2,1); lcd_putc("Finalizando."); delay_ms(230); lcd_putc('\f'); lcd_gotoxy(2,1); lcd_putc("Finalizando.."); delay_ms(230); lcd_putc('\f'); lcd_gotoxy(2,1); lcd_putc("Finalizando..."); delay_ms(230);  delay_ms(1000); lcd_putc('\f'); delay_ms(80);  lcd_gotoxy(1,1); printf(lcd_putc,"FIN DE ESTUDIO\n"); delay_ms(100); lcd_gotoxy(15,2); delay_ms(50); printf(lcd_putc," ECG AUTO %d Seg ",dispti); for(lug=1;lug&lt;=18;++lug){     printf(lcd_putc,"b");     delay_ms(380); } delay_ms(800); lcd_putc('\f'); } //end func solito  void main(){     int i,lug,tiemp;     char k;     int der1,der2,der3,der4,der5,der6;     int der7,der8,der9,der10,der11,der12,der13;     long disip;      setup_adc_ports(NO_ANALOGS);      OUTPUT_A(0x00);     setup_psp(PSP_DISABLED);     setup_spi(FALSE);     setup_timer_0(RTCC_INTERNAL RTCC_DIV_1);     setup_timer_1(T1_DISABLED);     setup_timer_2(T2_DISABLED,0,1);     setup_comparator(NC_NC_NC_NC);     setup_vref(FALSE);      kbd_init();     lcd_init(); </pre>	<pre> lcd_putc('\f'); lcd_putc(" ECG "); delay_ms(300); lcd_putc('\f'); lcd_putc(" ECG "); delay_ms(300); lcd_putc('\f'); lcd_putc(" ECG "); delay_ms(300); lcd_putc('\f'); lcd_putc(" ECG "); delay_ms(800);  lcd_gotoxy(1,1); lcd_putc(" Seleccione\n DerivaciOn:"); delay_ms(100); lcd_gotoxy(5,1); OUTPUT_A(0x00); OUTPUT_E(0x00); OUTPUT_C(0x10); delay_ms(300); OUTPUT_C(0x20); delay_ms(1500); OUTPUT_C(0x00);  der1=10; der2=20; der3=30; der4=40; der5=50; der6=60; der7=70; der8=80; der9=90; der10=100; der11=110; der12=120; der13=130;  disip=1500;  for(;;){     k=kbd_getc();     if(k!=false){         switch (k){             case '0':{                 delay_ms(20);                 lcd_putc('\f');                 lcd_putc(" ECG\n Deriv: D!");                 lcd_gotoxy(5,1);                 //**** Cuadrada ****                 for(tiemp=1;tiemp&lt;=der1;++tiemp){                      OUTPUT_C(0x10); //indica                     delay_ms(50);                     OUTPUT_C(0x00); //indica                     delay_ms(50);                 } //**** FIN Cuadrada ****                 OUTPUT_A(0x03);                 OUTPUT_E(0x00);                 OUTPUT_C(0x60);                 delay_ms(disip); //disipa                 OUTPUT_C(0x40);             }             delay_ms(20);             break; </pre>
---	--

```

case '1': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: DII");
    lcd_gotoxy(5,1);
    /***** Cuadrada *****/
    for(tiemp=1;tiemp<=der2;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    }/***** FIN Cuadrada *****/
    OUTPUT_A(0x05);
    OUTPUT_E(0x02);
    OUTPUT_C(0x60);
    delay_ms(disip); //disipa
    OUTPUT_C(0x40);
}
delay_ms(20);
break;

case '2': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: DIII");
    lcd_gotoxy(5,1);
    /***** Cuadrada *****/
    for(tiemp=1;tiemp<=der3;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    }/***** FIN Cuadrada *****/
    OUTPUT_A(0x06);
    OUTPUT_E(0x02);
    OUTPUT_C(0x68);
    delay_ms(disip); //disipa
    OUTPUT_C(0x48);
}
delay_ms(20);
break;

/*****/
case '3': {
    delay_ms(20);
    manda=5;
    solito();
    goto deten;
}
delay_ms(20);
break;
/*****/

```

```

case '4': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: aVR");
    lcd_gotoxy(5,1);
    /***** Cuadrada *****/
    for(tiemp=1;tiemp<=der4;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    }/***** FIN Cuadrada *****/
    OUTPUT_A(0x31);
    OUTPUT_E(0x07);
    OUTPUT_C(0x60);
    delay_ms(disip); //disipa
    OUTPUT_C(0x40);
}
delay_ms(20);
break;

case '5': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: aVL");
    lcd_gotoxy(5,1);
    /***** Cuadrada *****/
    for(tiemp=1;tiemp<=der5;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    }/***** FIN Cuadrada *****/
    OUTPUT_A(0x2A);
    OUTPUT_E(0x01);
    OUTPUT_C(0x60);
    delay_ms(disip); //disipa
    OUTPUT_C(0x40);
}
delay_ms(20);
break;

case '6': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: aVF");
    lcd_gotoxy(5,1);
    /***** Cuadrada *****/
    for(tiemp=1;tiemp<=der6;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    }/***** FIN Cuadrada *****/
    OUTPUT_A(0x1C);
    OUTPUT_E(0x03);
    OUTPUT_C(0x60);
    delay_ms(disip); //disipa
    OUTPUT_C(0x40);
}
delay_ms(20);
break;

```

```

case '7': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_gotoxy(5,1);
    manda=10;
    solito();
    goto deten;
}

delay_ms(20);
break;
lcd_putc('\f');

case '8': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: V1");
    lcd_gotoxy(5,1);
    //**** Cuadrada ****
    for(tiemp=1;tiemp<=der7;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    } //**** FIN Cuadrada ****
    OUTPUT_A(0x38);
    OUTPUT_E(0x05);
    OUTPUT_C(0x60);
    delay_ms(disip); //disipa
    OUTPUT_C(0x40);
}
delay_ms(20);
break;

case '9': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: V2");
    lcd_gotoxy(5,1);
    //**** Cuadrada ****
    for(tiemp=1;tiemp<=der8;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    } //**** FIN Cuadrada ****
    OUTPUT_A(0x38);
    OUTPUT_E(0x05);
    OUTPUT_C(0x61);
    delay_ms(disip); //disipa
    OUTPUT_C(0x41);
}
delay_ms(20);
break;

case 'A': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: V3");
    lcd_gotoxy(5,1);

    //**** Cuadrada ****
    for(tiemp=1;tiemp<=der9;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    } //**** FIN Cuadrada ****
    OUTPUT_A(0x38);
    OUTPUT_E(0x05);
    OUTPUT_C(0x62);
    delay_ms(disip); //disipa
    OUTPUT_C(0x42);
}
delay_ms(20);
break;

case 'B': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_gotoxy(5,1);
    manda=20;
    solito();
    goto deten;
}
delay_ms(20);
break;

case 'C': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: V4");
    lcd_gotoxy(5,1);
    //**** Cuadrada ****
    for(tiemp=1;tiemp<=der10;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    } //**** FIN Cuadrada ****
    OUTPUT_A(0x38);
    OUTPUT_E(0x05);
    OUTPUT_C(0x63);
    delay_ms(disip); //disipa
    OUTPUT_C(0x43);}
    delay_ms(20);
    break;

case 'D': {
    delay_ms(20);
    lcd_putc('\f');
    lcd_putc(" ECG\n Deriv: V5");
    lcd_gotoxy(5,1);
    //**** Cuadrada ****
    for(tiemp=1;tiemp<=der11;++tiemp){
        OUTPUT_C(0x10); //indica
        delay_ms(50);
        OUTPUT_C(0x00); //indica
        delay_ms(50);
    } //**** FIN Cuadrada ****
    OUTPUT_A(0x38);
    OUTPUT_E(0x05);
    OUTPUT_C(0x64);
    delay_ms(disip); //disipa
    OUTPUT_C(0x44);
}
}

```

<pre> delay_ms(20); break;  case 'E': {   delay_ms(20);   lcd_putc('\f');   lcd_putc(" ECG\n Deriv: V6");   lcd_gotoxy(5,1);   /***** Cuadrada **** for(tiemp=1;tiemp&lt;=der12;++tiemp){   OUTPUT_C(0x10); //indica   delay_ms(50);   OUTPUT_C(0x00); //indica   delay_ms(50); }/***** FIN Cuadrada **** OUTPUT_A(0x38); OUTPUT_E(0x05); OUTPUT_C(0x65); delay_ms(disip); //disipa OUTPUT_C(0x45); } delay_ms(20); break; case 'F': { deten:   /***** Cuadrada **** for(tiemp=1;tiemp&lt;=der13;++tiemp){   OUTPUT_C(0x10); //indica   delay_ms(50);   OUTPUT_C(0x00); //indica   delay_ms(50); }/***** FIN Cuadrada **** for(i=0;i&lt;=0;++i){ //5 Seg Apaga Solo   delay_ms(20);   lcd_putc('\f');   lcd_gotoxy(2,1);   printf(lcd_putc,"ECG: Detenido");   delay_ms(100);   for(lug=36;lug&gt;=19;--lug){     lcd_gotoxy(lug-18,2);     printf(lcd_putc,"Esperando Orden ");// Finalizo "); </pre>	<pre>     delay_ms(380);   }   OUTPUT_C(0x00); } delay_ms(20); break;  default: { delay_ms(20);   lcd_putc('\f');   lcd_putc(" ERROR\n de SelecciOn");   OUTPUT_A(0x00);   OUTPUT_E(0x00);   OUTPUT_C(0x00);   delay_ms(20); lcd_gotoxy(5,1); } break;  } //end switch  } //end if  } //end for  } //end main </pre>
--	--