

56

207



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

Adquisición y Análisis Automatizado de Corrientes de Despolarización Termoestimuladas (ITC): el Caso de Rb Ca F₃; Eu: Mn

T E S I S
QUE PARA OBTENER EL GRADO DE:
LICENCIADO EN FISICA
P R E S E N T A :
ANA LEONOR RIVERA LOPEZ

FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

9. INTRODUCCION

9.1. Objetivos	4-1
9.2. Contenidos	4-1

I. EL PROCESO DE DESPOLARIZACION TERMICAMENTE ESTIMULADA (ITC).

I.1. La técnica de ITC.	I-1
I.1.1. Descripción de la técnica.	I-2
I.1.2. Procedimiento experimental.	I-4
I.1.3. Borrado de picos.	I-6
I.1.4. Corrientes de despolarización debidas a carga espacial iónica.	I-7
I.2. Análisis de los datos de un proceso de ITC.	
I.2.1. Las ecuaciones de ITC.	I-8

II. ADQUISICION Y ANALISIS DE DATOS.

II.1. La adquisición de datos mediante la tarjeta PCL-712.	
II.1.1. Descripción de la tarjeta de adquisición.	II-1
II.1.2. Conversión Analógico/Digital.	II-2
II.1.2.1. Programando la conversión A/D con disparos directos.	II-4
II.1.2.2. Programando la conversión A/D empleando los contadores.	II-8
II.2. Un programa de adquisición y análisis: ITC.EXE	
II.2.1. Descripción del programa ITC.EXE	II-16
II.2.2. Diagrama de flujo.	II-23
II.2.3. Listado del programa.	II-30

III. RESULTADOS.

III.1) Una prueba del programa: el sistema KCl:Eu. III-1

III.2) Una aplicación: el sistema RbCaF₂:Eu:Mn. III-4

IV. CONCLUSIONES.

IV-1

APENDICE I. INCERTIDUMBRES ASOCIADAS A E Y τ_0 .

A-1

APENDICE II. PROCESOS TERMOESTIMULADOS.

A.II.1. ¿Qué son los procesos termoeestimulados? A-2

A.II.2. Ecuaciones generales de los procesos termoeestimulados. A-3

BIBLIOGRAFIA

B-1

Introducción

3.1. OBJETIVOS

Este trabajo surgió por la necesidad de automatizar técnicas del Laboratorio de Propiedades Ópticas y Eléctricas de Sólidos del IFUNAM. Para cumplir con tal objetivo se puso en operación una tarjeta de adquisición de datos PCL-712 conectada a una computadora PC compatible. Además se elaboró un programa que hace la adquisición y el análisis de los datos de la técnica de *corrientes de despolarización termoestimuladas* (ITC). Este programa puede extenderse fácilmente al análisis de otros procesos termoestimulados, en particular, la *Termoluminiscencia* que es otra de las técnicas que se usan en el laboratorio. El uso de la tarjeta de adquisición de datos permitirá la automatización de otros experimentos.

En este trabajo se comprueba la eficiencia del programa desarrollado comparando los resultados obtenidos de $KCl:Eu$ con los de otros autores. Además, en él se inician los estudios de una fluoroperovskita: $RbCaF_3:Eu:Mn$.

3.2. CONTENIDOS

En el capítulo I se describe el proceso de despolarización térmicamente estimulada (ITC). En la sección I.1 se presenta la técnica y la sección I.2 las ecuaciones que describen este proceso. Se hace énfasis en el método de áreas que permite calcular los dos parámetros que describen el pico de ITC: la energía de activación E y el factor preexponencial τ_0 .

El capítulo II trata de la adquisición y análisis de datos de ITC. En la sección II.1 se mencionan las características generales de la tarjeta PCL-712 y en la sección II.1.2 se da una explicación de como programar la conversión A/D de las dos formas que esta tarjeta permite: con disparos directos o periódicos. Para ver una aplicación directa de estas dos maneras de disparo de la conversión A/D se listan programas elaborados en BASIC, C y TURBO-PASCAL. En la sección II.2 se describe el programa de adquisición y análisis de datos de ITC desarrollado en este trabajo (llamado ITC.EXE), se da un diagrama de flujo del mismo y se lista el programa fuente en Turbo-Pascal.

En el capítulo III se presentan los resultados de ITC obtenidos para $KCl:Eu$ y $RbCaF_2:Eu:Mn$ haciendo uso del programa ITC.EXE para su adquisición y análisis. En la sección III.1 se comparan los resultados de $KCl:Eu$ con los obtenidos por otros autores con la finalidad de checar el programa. La sección 3.2 contiene los resultados de $RbCaF_2:Eu:Mn$ obtenidos hasta el momento.

En el capítulo IV se presentan las conclusiones de este trabajo.

El trabajo tiene dos apéndices. En el Apéndice I se citan las posibles fuentes de error para el ajuste y se hace un análisis de las incertidumbres asociadas a la energía de activación y el factor pre-exponencial. En el Apéndice II se discute una posible extensión de este trabajo al análisis de otro proceso termoestimulado: la termoluminiscencia.

I

El Proceso De Despolarización Termicamente Estimulada

1.1. LA TECNICA DE ITC

En este método una muestra polarizada eléctricamente a una temperatura T_0 es despolarizada mediante un calentamiento dando lugar a una corriente de desplazamiento. Para polarizar la muestra se aplica un campo eléctrico externo a una temperatura "alta" T_p y luego sin quitar el campo se enfría la muestra hasta una temperatura "baja" T_0 . A T_0 , se quita el campo eléctrico externo y los electrodos conectados a la muestra son cortocircuitados a través de un detector de corrientes sensible. Parte de la polarización no se anula cuando se quita el campo eléctrico externo; esta polarización "congelada" es debida al alineamiento de defectos dipolares o a una distribución no uniforme de portadores de carga electrónicos o iónicos. Durante el calentamiento se da energía al sistema, por lo que a alguna temperatura superior a T_0 los portadores de carga atrapados se vuelven móviles o bien los dipolos pueden empezar a rotar, con lo que en su movimiento hacia el equilibrio se produce la despolarización de la muestra originando un pico de corrientes en el circuito externo.

Los primeros en realizar una investigación sistemática de corrientes de despolarización térmicamente estimuladas en dieléctricos fueron Bucci y Fieschi^(1,2), los cuales concentraron su atención en cristales que contenían defectos iónicos dipolares. Ellos llamaron al fenómeno corrientes termo-iónicas (ionic ThermoCurrents: ITC) y este término es el más usual para referirse a corrientes debidas al movimiento de dipolos.

I.1.1 DESCRIPCION DE LA TECNICA

El método de ITC ha sido usado en muchos cristales que contienen defectos dipolares que pueden ser orientados por un campo eléctrico externo. En cristales iónicos tales dipolos pueden formarse cuando los iones del cristal son sustituidos por iones impureza de carga diferente. La compensación de carga se alcanza generalmente por vacancias de la red, iones intersticiales o iones impureza de otros tipos. Si el compensador de carga está cerca del ión impureza se forma un complejo dipolar. Tales defectos dipolares son el principal origen de ITC en cristales iónicos.

La figura 1.1 ilustra el procedimiento de ITC y su descripción es la siguiente:

- (1): En ausencia de campo eléctrico (α), los dipolos están orientados al azar (d).
- (2): Si se aplica un campo eléctrico E_p (α) a la temperatura T_p (b) durante un intervalo de tiempo t_p , los dipolos serán polarizados a saturación, es decir, se ordenaran (d) y se observará una corriente que decae exponencialmente (c).
- (3): Al alcanzar la polarización de saturación (para cristales iónicos usualmente en menos de un minuto, a temperatura ambiente), el dieléctrico es enfriado hasta una temperatura baja T_0 (b) sin quitar el campo (α). A $T = T_0$ se quita el campo eléctrico E_p y se conecta el electrómetro. Es común observar entonces una corriente (c) que decrece en el tiempo debido a la descarga del condensador y a relajación de la carga espacial capturada en los electrodos. Esta corriente se vuelve finalmente muy pequeña (del orden de las fluctuaciones del electrómetro) en tanto que los dipolos permanecen orientados con la misma configuración obtenida en T_p (d).
- (4): Se inicia el calentamiento controlado del dieléctrico con una rapidez constante (b), conforme los dipolos pierdan su orientación preferencial (d) se detectará una corriente de

despolarización $i(T)$ (c). Durante el tiempo en que este proceso ocurre, la corriente aumentará primero exponencialmente, alcanzará un máximo y entonces caerá a cero, esta es la corriente termo-iónica (ITC).

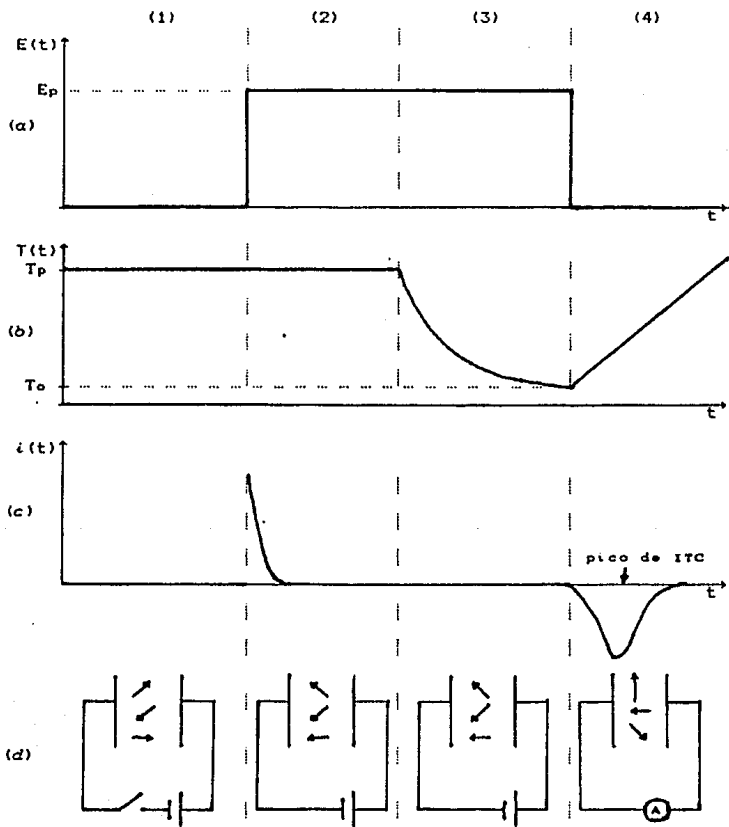


Figura 1.1
El proceso de ITC

1.12. PROCEDIMIENTO EXPERIMENTAL.

El dispositivo experimental empleado se ilustra en la figura 1.2.

1. Se coloca la muestra en el criostato, el cual permite variaciones de temperatura controladas, cubriéndola con la placa de cobre, que esta conectada a un atravesador, la cual hace las veces de electrodo, el fondo del dedo frío actúa como el otro electrodo. Se cierra perfectamente el criostato y se conecta al sistema de vacío.
2. Cuando se alcanza un vacío de aproximadamente 10^{-5} Torr, se gotea nitrógeno líquido en el dedo frío para mantener la muestra a una temperatura de polarización $T_p \approx 0^\circ\text{C}$ (la temperatura es medida por un termopar de Kromel-Alumel). A T_p se aplica un campo eléctrico $E_p \approx 1200$ V durante 5 minutos.
3. Con el campo aún presente se vacía nitrógeno líquido en el dedo frío hasta llegar a una temperatura $T_0 \approx 80$ K. Alcanzada T_0 , se quita el campo. Debido a la temperatura baja, los dipolos son "congelados" en sus posiciones ordenadas y el cristal permanece polarizado aún en ausencia de campo.
4. Se conecta el electrómetro a la muestra, cortocircuitando a los electrodos. Cuando se estabiliza la lectura de fondo del electrómetro se introduce el cautín en el dedo frío para producir un calentamiento controlado de la muestra mientras se registran la corriente a través del electrómetro y la temperatura de la muestra con el termopar. Ambos dispositivos están conectados al adquisidor de datos de la computadora con la cual se registran la corriente y la temperatura como funciones del tiempo.

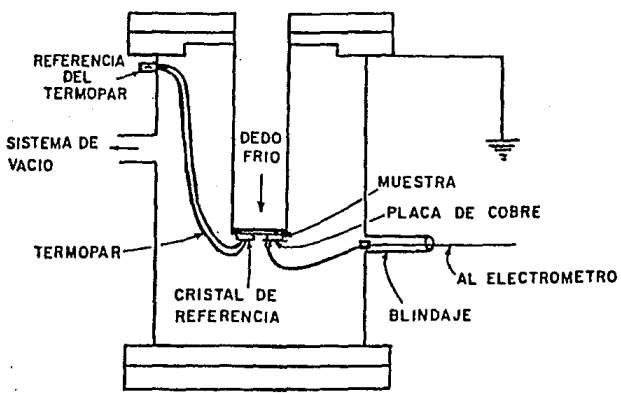
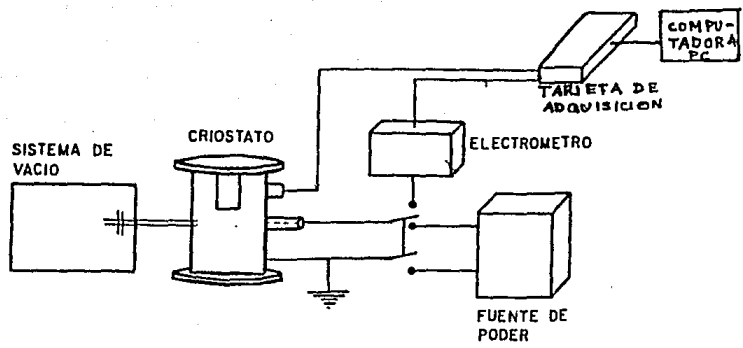


Figura 1.2
Dispositivo Experimental

I.13. BORRADO DE PICOS

Si en un cristal existe más de un tipo de iones o de dipolos que den lugar a corrientes de despolarización termicamente estimuladas se producirá un pico de corriente por cada proceso de relajación, por lo que, en general no es frecuente la ocurrencia de un solo pico. En la mayoría de los casos se observan varios picos durante el calentamiento de la muestra en un cierto intervalo de temperatura. En los casos más favorables, el traslape entre picos vecinos es mínimo, lo cual permite tratar a cada uno de ellos por separado.

Si ocurre un traslape parcial entre dos picos, la separación del pico de alta temperatura se logra con la técnica llamada borrado térmico que consiste en lo siguiente: Si las temperaturas del máximo de cada uno de los dos picos son T_1 y T_2 , el calentamiento del cristal a una temperatura intermedia apropiada T' ($T_1 < T' < T_2$) y su enfriamiento subsecuente, borra el pico de baja temperatura; si la muestra es calentada nuevamente se observará el segundo pico limpio o casi limpio. Para obtener el primer pico, la muestra es polarizada a T_p con $T_1 < T_p < T_2$ durante un período de tiempo $t_p \approx \tau_1 \ll \tau_2$; con lo cual el primero de los dos procesos de polarización llega a su saturación mientras el pico conectado al segundo proceso aparecerá muy débil.

Un ejemplo de la obtención de los picos componentes mediante el proceso descrito se observa en la figura 1.3⁽⁸⁾.

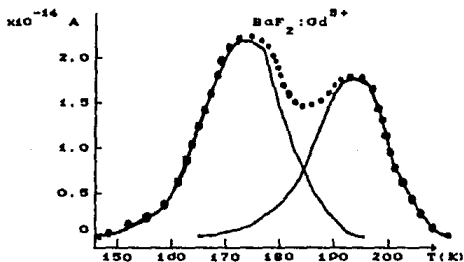


Figura 1.3
Borrado de picos

114. Corrientes de Despolarización debidas a carga espacial iónica.

Además de los picos de corriente que involucran la rotación de dipolos, hay otro tipo de picos de ITC que son debidos a la migración de portadores de carga iónicos (intersticiales o vacancias). El campo de polarización externo aplicado a una temperatura T_p , donde la movilidad de los iones intersticiales o vacancias es considerable, crea una distribución no uniforme de portadores de carga iónicos y se acumula carga espacial cerca de los electrodos. La polarización resultante es llamada polarización de carga espacial iónica. A temperatura baja T_0 , la distribución de carga espacial esta "congelada", por lo que al calentar el cristal en ausencia del campo externo se produce un pico de corrientes de desplazamiento en el circuito externo mientras los defectos regresan a la distribución uniforme. Estos picos de corriente aparecen a temperaturas más altas que los de ITC dipolares (usualmente arriba de temperatura ambiente) y son generalmente más intensos ya que los portadores se desplazan distancias macroscópicas.

De acuerdo a Bucci y su grupo⁽²⁾ los picos de ITC debidos a la relajación de carga espacial iónica tienen las siguientes propiedades:

1. La temperatura del máximo T_m no esta bien definida; a mayor T_p aumenta T_m .
2. El área delimitada por el pico no es una función lineal del campo eléctrico aplicado, particularmente para campos bajos.
3. La forma del pico no permite la medición de ninguna energía de activación.

Con esto podemos distinguir si los picos de ITC se deben a dipolos o a carga espacial.

I.2 ANALISIS DE LOS DATOS DE UN PROCESO DE ITC

I.2.1. LAS ECUACIONES DE ITC

Por simplicidad se considera un dieléctrico homogéneo que contiene un solo tipo de dipolos no interactuantes. El momento dipolar eléctrico para un dipolo se denotará por μ y la concentración de dipolos por unidad de volumen por N . El dipolo puede reorientarse por un brinco activado termicamente de uno de los componentes que forman el complejo hacia otro sitio equivalente dentro de la red cristalina. Tal brinco puede caracterizarse por el tiempo de relajación τ , el cual es una función exponencial de la temperatura⁽⁸⁾:

$$\tau(T) = \tau_0 \exp(E/KT) \quad (1.1)$$

donde τ^{-1} es la probabilidad de un solo salto por unidad de tiempo, es decir, es la probabilidad de que un dipolo cambie su orientación por segundo. El factor pre-exponencial τ_0 se supone independiente de la temperatura; E es la energía de activación para la rotación de un dipolo (expresada en eV); T es la temperatura absoluta y K la constante de Boltzmann. τ_0 y E son los parámetros más importantes por obtener con la técnica de ITC.

En ausencia de un campo eléctrico externo los dipolos están orientados al azar. Si un campo externo E_p , es aplicado al cristal a una temperatura T_p se provoca la polarización de los dipolos hasta un valor de saturación P_0 dado por la función de Langevin⁽⁹⁾:

$$P_0 = \mu^2 \left[\operatorname{cotgh} \left[\frac{\mu E_p}{KT_p} \right] - \frac{KT_p}{\mu E_p} \right] \quad (1.2)$$

que puede ser aproximada (cuando $KT_p \gg \mu E_p$) por:

$$P_0 = \frac{\mu^2 E_p N \alpha}{K T_p} \quad (1.3)$$

donde α es un factor adimensional geométrico que depende de la estructura del cristal (para dipolos que pueden rotar libremente $\alpha = 1/3$).

En particular, a temperatura ambiente, la saturación de polarización de los dipolos es alcanzada en unos pocos segundos en la mayoría de los cristales iónicos. A una temperatura baja se necesita un tiempo muy largo para llegar a la polarización de saturación ya que los dipolos están "congelados" en sus posiciones. En los cristales iónicos, a temperaturas de nitrógeno líquido, τ puede llegar a ser de semanas o meses.

Si no hay interacciones mutuas entre dipolos, el número de dipolos desorientados en la unidad de tiempo bajo calentamiento es proporcional al número de dipolos aún orientados. Esto es, la cinética de despolarización es de primer orden:

$$-\frac{dP}{dt} = \frac{P}{\tau} = P \tau_0^{-1} \exp(-E/KT) \quad (1.4)$$

si la rapidez de calentamiento es lineal:

$$\beta = \frac{dT}{dt} \quad (1.5)$$

combinando estas dos últimas ecuaciones se tiene:

$$dP = -\frac{P}{\beta \tau} dT \quad (1.6)$$

integrandola y usando (1.1) se llega a:

$$P = P_0 \exp \left\{ - \int_{T_0}^T (\beta \tau_0)^{-1} \exp [-E/KT'] dT' \right\} \quad (1.7)$$

La densidad de corriente debida a la despolarización esta dada por:

$$J(t) = - \frac{dP}{dt} \quad (1.8)$$

suponiendo una razón de calentamiento constante ($T = T_0 + \beta t$) se obtiene:

$$J(T) = -\beta \frac{dP}{dT} \quad (1.9)$$

sustituyendo (1.6) en (1.9) :

$$J(T) = -\beta \left\{ - \frac{P}{\beta \tau} \right\} = \frac{P}{\tau} \quad (1.10)$$

usando (1.7):

$$J(T) = (\tau_0 \exp[E/KT])^{-1} P_0 \exp \left[- \int_{T_0}^T \frac{1}{\beta \tau_0} \exp \left(- \frac{E}{KT'} \right) dT' \right] \quad (1.11)$$

utilizando (1.3):

$$J(T) = \frac{N \mu^2 \alpha E_p}{K T_p \tau_0} \exp \left[- \frac{E}{KT} - \int_{T_0}^T \frac{1}{\beta \tau_0} \exp \left(- \frac{E}{KT'} \right) dT' \right] \quad (1.12)$$

Recordando que $i = JA$ con A el área de la muestra:

$$i(T) = \frac{N \mu^2 \alpha E_p A}{K T_p \tau_0} \exp \left[- \frac{E}{KT} - \int_{T_0}^T (\beta \tau_0)^{-1} \exp \left(- \frac{E}{KT'} \right) dT' \right] \quad 1.13$$

Los aspectos más importantes de esta función son⁽²⁾:

- 1) La corriente como función de la Temperatura es una banda asimétrica tipo pico con un máximo (obtenido haciendo $di/dT=0$ en la ecuación 1.13) a la temperatura⁽³⁾:

$$T_M = \sqrt{\beta E \tau(T_M) / K} \quad (1.14)$$

independiente de T_p y E_p .

Despejando $\tau(T_M)$ de (1.14) y sustituyendo en (1.1):

$$\tau_0 = \frac{K T_M^2}{\beta E} \exp \left\{ - \frac{E}{K T_M} \right\} \quad (1.15)$$

2) La energía de activación puede obtenerse tomando logaritmos en la ecuación (1.13):

$$\ln i(T) = \ln \left[\frac{N \mu^2 \alpha E_p A}{K T_p \tau_0} \right] - \int_{T_0}^T \frac{1}{\beta \tau_0} \exp \left\{ - \frac{E}{K T'} \right\} dT' - E/KT \quad (1.16)$$

para T bajas (tales que $[E/KT] \gg 1$) la integral en la ecuación (1.16) se anula, por lo que:

$$\ln i(T) = \text{cte} - E/KT \quad (1.17)$$

De esta ecuación se obtiene fácilmente E y con ella τ_0 de (1.15), este es el llamado *método de subida inicial (initial rise)*, el cual se usa en el análisis de muchos procesos termoestimulados.

3) Integrando la ecuación (1.9) entre T_0 y T_f donde se cumple que $P(T_f)=0$ se tiene:

$$\int_{T_0}^{T_f} J(T') dT' = -\beta \int_{P_0}^0 \frac{dP}{dT'} dT' = \beta P_0 \quad (1.18)$$

entonces, el área delimitada por la banda resulta proporcional al momento dipolar inicial por unidad de volumen; la polarización inducida en saturación es:

$$P_0 = \frac{N \mu^2 \alpha E_p}{K T_p} = \frac{1}{\beta} \int_{T_0}^{T_f} J(T') dT' \quad (1.19)$$

donde la integral es el área bajo el pico de ITC. Esta ecuación permite la evaluación del número de dipolos contenidos en el dieléctrico haciendo ciertas suposiciones sobre el tamaño del dipolo para evaluar μ_{med} , siendo d la distancia que

separa las componentes del dipolo. Despejando de (1.19) la concentración de dipolos resulta:

$$N = \frac{K T_p}{\alpha \mu^2 E_p \beta A} \int_{T_0}^{T_f} i(T') dT' \quad (1.20)$$

4) Despejando P de la ecuación (1.10):

$$P_0 = \tau J(T_0) \quad (1.21)$$

sustituyendo en (1.18):

$$\int_T^{T_f} J(T') dT' = \beta P_0 = \beta J(T) \tau_0 \exp(E/KT) \quad (1.22)$$

tomando logaritmos en ambos miembros de la ecuación (1.22):

$$\ln \left\{ \frac{\int_T^{T_f} J(T') dT'}{J(T)} \right\} = \ln(\tau_0 \beta) + \frac{E}{KT} \quad (1.23)$$

como $i=JA$ entonces:

$$\ln(\tau_0 \beta) + \frac{E}{KT} = \ln \left\{ \int_T^{T_f} i(T') dT' \right\} - \ln i(T) \quad (1.24)$$

Con (1.24) pueden estimarse los valores de E y de τ_0 .

Como experimentalmente es difícil tener una razón de calentamiento β constante se hace el truco de integrar sobre el tiempo y no sobre la temperatura, para lo cual se integra la ecuación (1.8) entre los tiempos t y $t_f^{(5)}$:

$$\int_t^{t_f} J(t') dt' = -(P_f - P) \quad (1.25)$$

sustituyendo P de la ecuación (1.10):

$$\int_t^{tr} J(t') dt' = -(J(tr)\tau - J(t)\tau) \quad (1.26)$$

seleccionando t_r como el tiempo final del pico en el que $J(t_r)=0$ y usando (1.1) en (1.26) se encuentra:

$$\int_t^{tr} J(t') dt' = J(t)\tau = J(t)\tau_0 e^{E/KT} \quad (1.27)$$

tomando logaritmos a ambos miembros de (1.27) resulta:

$$\ln \left\{ \frac{\int_t^{tr} i(t') dt'}{i(T)} \right\} = \ln \tau_0 + \frac{E}{KT} \quad (1.28)$$

La ecuación (1.28) conduce al método llamado *método de áreas* que se utiliza en este trabajo para encontrar los parámetros en el programa ITC. A la integral $\int_t^{tr} i(t') dt'$ se le conoce como el área residual y para calcularla numericamente se utiliza el método de Simpson. Dicho método aproxima el área residual por:

$$\int_t^{tr} i(t') dt' \cong \sum_{n=t}^{NTP} \{ [i(n) + 4i(n+1) + i(n+2)] (t(n+1) - t(n)) / 3 \} \quad (1.29)$$

Si se varía el tiempo t a partir del cual se calcula el área residual (y en consecuencia T), la ecuación (1.28) establece que la relación entre el logaritmo del cociente del área residual entre la corriente y el inverso de la temperatura es una línea recta con pendiente E/K y ordenada al origen τ_0 . Usando el procedimiento de mínimos cuadrados se hace el mejor ajuste a dicha recta. Una vez hecho el ajuste se determinan los valores de E y τ_0 , que son precisamente los parámetros más importantes del pico de ITC.

II

Adquisición Y

Análisis

De

Datos.

II.1. LA ADQUISICION DE DATOS MEDIANTE LA TARJETA PCL-712

II.1.1. DESCRIPCION DE LA TARJETA DE ADQUISICION.

La tarjeta PCL-712 es una tarjeta de tipo analógico-digital I/O que tiene 5 funciones de medición y control, 16 entradas analógicas, 2 salidas analógicas, 16 entradas digitales, 16 salidas digitales y un reloj programable. La resolución para las conversiones analógico-digitales es de 12 bits y el tiempo de conversión es menor a 30 μ s. El rango de entrada es bipolar de ± 5 V, con una precisión de 0.2 %. La impedancia de entrada es mayor a 10 mega-ohms.

Como la entrada de la tarjeta es bipolar de ± 5 V y su resolución es de 12 bits, entonces en el intervalo de 10 V habrá 2^{12} (4096) valores distinguibles, es decir, la tarjeta discierne hasta voltajes de 2.4 mV (este es el valor más pequeño que puede medirse).

La mayoría de los periféricos y los adaptadores de interfase en una microcomputadora PC compatible son controlados usando los puertos digitales de entrada y salida (puertos I/O). Estos puertos se seleccionan usando el *espacio de direcciones de puertos I/O* del microprocesador 8086. Puede elegirse la dirección base de la tarjeta PCL-712 mediante una serie de interruptores que tiene a un costado, siendo la dirección escogida en fábrica la hexadecimal 220. En todos los programas se asigna esta dirección a la variable entera BASE. Es importante en Turbo-Pascal definir la variable BASE como del tipo WORD para que sea identificada como una dirección cuando se utilice como parámetro de la instrucción de direccionamiento PORTW.

La tarjeta PCL-712 tiene 12 puertos que permiten controlar las distintas funciones, estos puertos se conocen como registros y son direccionables desde la dirección base del puerto I/O seleccionada. Cada registro esta identificado por la dirección $BASE+j$ con $j=0, 1, \dots, 7, 10, 11, 13$ y 14 .

II.12. CONVERSION ANALOGICO/DIGITAL

Para realizar conversiones analógico-digitales (A/D) con la tarjeta se necesitan usar cuatro de los registros: $BASE+4$, $BASE+5$, $BASE+10$ y $BASE+11$. Cada uno de ellos es un byte de 8 bits. El resultado de la conversión A/D se encuentra en los registros $BASE+4$ y $BASE+5$; además en este último se guarda el estado de la conversión (si ya se efectuó o no). El registro $BASE+10$ permite escoger el canal donde se hará la conversión A/D. El registro $BASE+11$ controla la forma de disparo de la conversión: directa o periódica programada. La forma detallada de estos registros es la siguiente:

BASE + 10.

Los bits 0 a 3 de este registro son usados para seleccionar el canal en el que se desea se realice la conversión A/D. Los bits 4 a 7 son ignorados.

BASE + 10								
Bit	7	6	5	4	3	2	1	0
	Ignorados				Selección del canal de conversión			

BASE + 11.

Este byte controla el modo de disparo:

Bit 0 = 0 : No hay un disparo directo.

Bit 0 = 1 : Se hace un disparo directo.

Bit 1 = 0 : Impide el disparo regulado por contadores.

Bit 1 = 1 : Permite que el disparo sea regulado por los contadores.

Los bits 2 a 7 son ignorados.

BASE + 4.

Este byte, junto con los bits 0 a 3 del byte de BASE+5, forma un número de 12 bits, el cual es el resultado de la conversión A/D. El número puede ir de 0 a 4096 (2^{12}).

	BASE + 5				BASE + 4							
Bit	3	2	1	0	7	6	5	4	3	2	1	0
Número	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Formato del número resultante de la conversión

BASE + 5.

Los bits 0 a 3 son el byte alto de la conversión A/D. El bit 4 refleja el estado de la conversión:

bit 4 = 0 conversión lista

bit 4 = 1 conversión no lista.

De esta manera, si el contenido de BASE+5 es mayor a 16 (2^4) no se ha realizado la conversión A/D.

	BASE + 5							
Bit	7	6	5	4	3	2	1	0
	0	0	0	Estado de la conversión	Byte alto de la conversión			

Para determinar el voltaje leído en la tarjeta se guarda el contenido de los registros BASE+4 (que es el byte bajo de la conversión A/D) en la variable LO y BASE+5 (que es el byte alto) en la variable HI. Viendo el formato del número resultante de la conversión, se llega a:

$$m = HI * 2^8 + LO = HI * 256 + LO \quad (2.1)$$

Como la entrada de la tarjeta es bipolar, este número debe ir desde -2048 hasta 2048, el número en (2.1) va de 0 a 4096 para que pase a la escala indicada se le debe restar 2048, entonces:

$$m = HI * 256 + LO - 2048 \quad (2.2)$$

es el resultado de la conversión A/D. El intervalo de entrada va de -5 V a +5 V comprendiendo 10 V en los cuales se pueden discernir 4096 valores, para determinar el voltaje leído basta pues con multiplicar (2.2) por (10/4096) con lo que:

$$\text{Voltaje} = (HI * 256 + LO - 2048) * 10 / 4096 \quad (2.3)$$

Ejemplo: Si en el byte de BASE+5 aparece

decimal 10 = 0 0 0 0 1 0 1 0

y en el byte de BASE+4 se encuentra:

decimal 141 = 1 0 0 0 1 1 0 1

la conversión analógico digital se ha realizado y el voltaje leído en la tarjeta es 1.594±0.003 V.

II.1.2.1. Programando La Conversión A/D con Disparos Directos.

Cuando se escribe un byte con un 1 en el bit del registro de BASE+11, la conversión A/D es disparada directamente, con lo que una lectura estará lista después de que se complete la conversión A/D. Los siguientes programas, escritos en BASIC, C y

PASCAL respectivamente, toman 10 lecturas (una lectura cada segundo) de un mismo canal utilizando el reloj de la computadora e imprimen los resultados.

' PROGRAMA DE ADQUISICION DE DATOS DE ITC EN BASIC

```
10 Dim VO!(250),           'Dimensionaliza el arreglo real VO!
20 J% = &H220              'Le asigna a J% la dirección de la tarjeta
30 ChCor% = 0              'Se asigna el canal 0 como el de medición
                             de corrientes
100 Print : Print "Punto", "Corriente"
110 i=1
120 OUT J%+10, ChCor%      'Selección del canal donde se realiza
                             la conversión A/D
150 While (i<11)          'Se procede a la adquisición de datos
160   Delay (1000)        'Se cuentan con el reloj de la compu-
                             tadora 1000 ms (1 segundo)
170   OUT J%+11, 1        'Hace un disparo directo
180   GoSub 1000
190   VO!(i) = (HI*256+LO-2048)*10/4096 'Se determina el voltaje
200   Print i, VO!(i)     'Se imprime la lectura
210   Next i
220 Wend
250 END                    ' Termina el programa Adquisición

1000 ' Rutina en la que se chequea si se realizó la conversión A/D
1010 ' y se leen los bytes Hi y Lo de la misma.
1020   HI=INP(J%+5)
1030   While HI>=16        ' Si no se ha realizado la conversión
1040     HI=INP(J%+5)      ' el byte Hi sera mayor de 2^=16
1050   Wend
1060   LO=INP(J%+4)
1100 Return
```

```

/* ----- */
/* Programa de prueba del manejo de puertos de un */
/* convertidor analógico-digital en C */
/* ----- */

#include <dos.h>
#include <conio.h>
#include <stdio.h>

/* ----- */
float leepuerto(void);
/* ----- */

int base;
float v[20];
/* ----- */

void main( )
{
    unsigned char ch;
    int i;
    base = 0x220; /* Dirección de la tarjeta */
    ch = 0; /* Selección del canal 0 como el de conversión */
    outportb(base+10, ch);
    for ( i = 0; i < 11; i++ )
        ( delay( 1000 ); /* Espera 1 seg */
          v[i] = leepuerto( ); /* Hace la lectura de la conversión */
        );
    for ( i = 0; i < 16; i++ ) /* Imprime los datos adquiridos */
        printf( "i = %d, v = %f", i, v[i] );
    getch( );
} /* Termina el programa principal */

/* ----- */
float leepuerto( ) /* Se hace la lectura de la conversión */
{
    int lo, hi;
    outportb( base + 11, 1 ); /* Selección de disparo directo */
    do /* Espera a que se realice la conversión */
        hi = inportb( base + 5 );
    while( hi > 15 );
    lo = inportb( base + 4 );
    return( ( hi * 256 + lo - 2048.0 ) * 10 / 4096.0 );
}
/* ----- */

```

PROGRAM Adquisición;

(Haciendo disparos directos se adquieren datos de ITC mediante la tarjeta PCL-712 usando Turbo-Pascal.)

Uses Crt; Dos;

Var

Base : word;
ChCor, ChTemp,
Alto, Bajo, Lb, Hb : byte;
Voltaje : real;

Procedure Conversión;

(Se chequea si se realizó la conversión analógico-digital y se leen los bytes Hi (Alto) y Lo (Bajo) de la misma.)

Begin

Repeat (Hasta que el byte alto)
Alto := PortW [Base + 5] (sea menor que 16 se ha)
Until (Alto<16); (realizado la conversión A/D)
Bajo := PortW [Base + 4]; (Lectura del byte bajo)
Voltaje := (Alto*256+Bajo-2048) * 10 / 4096 (Almacenamiento de la lectura realizada)
End; (Termina conversión.)

Begin

Base := \$220; (Se asigna a Base la dirección de la tarjeta)
ChCor := 0; (El canal 0 se toma como el de medición de corrientes)
PortW [Base + 10] := ChCor; (Selección canal de adquisición)
contador := 1; (Se inicia la adquisición de datos)
While (contador<11) Do
begin
Delay (1000); (Se cuenta 1 s con el reloj de la computadora)
PortW [Base + 11] := 1; (Se hace un disparo directo)
Conversión; (Se toma 1 dato)
WriteLn (contador, ' ', 5, Voltaje); (Se imprime)
contador := contador + 1
end;
End.

II.1.2.2. Programando la conversión A/D empleando los Contadores.

Cuando se le asigna un 2 al registro en BASE+11, la conversión A/D es disparada usando el contador de intervalos de tiempo programable 8253 de Intel, el cual ocupa 4 direcciones I/O en la tarjeta PCL-712. El 8253 está organizado como 3 contadores totalmente independientes de 16 bits cada uno, tienen un intervalo de conteo desde 0 hasta 2.6 MHz, pueden programarse para hacer un conteo binario o decimal y su entrada, puerta y salida son configuradas por la selección de modo (hay 6 posibles) almacenado en el byte de control. Los contadores son regresivos, esto es, el valor puesto en el registro de conteo será disminuido hasta llegar a cero, después de lo cual reiniciarán el conteo a partir del valor inicial.

Para programar la conversión periódica A/D se selecciona el contador, el modo, la forma y tipo de conteo con el byte de control:

Byte de control : Base + 3.

Acepta información del buffer de datos y la almacena en un registro que controla el modo de operación de cada contador, la selección binaria o conteo decimal y la lectura/carga de cada registro de conteo. En este byte sólo puede escribirse, no hay disponible ninguna operación para leer su contenido. Su estructura es la siguiente:

Formato del byte de control.

Bit	7	6	5	4	3	2	1	0
	SC1	SC0	RL1	RL0	M2	M1	M0	BCD

La selección del contador se hace mediante los bits 7 y 6:

SC1=0, SC0=0 : contador 0

SC1=0, SC0=1 : contador 1

SC1=1, SC0=0 : contador 2

SC1=1, SC0=1 : ilegal

La forma de lectura o carga del contador seleccionado se hace mediante los bits 5 y 4:

RL1=0, RLO=0 : Forma 1: *Operación de carga de contadores.*

RL1=1, RLO=0 : Forma 2: *Lectura/Carga sólo el byte alto.*

RL1=0, RLO=1 : Forma 3: *Lectura/Carga sólo el byte bajo*

RL1=1, RLO=1 : Forma 4: *Lectura/Carga primero el byte bajo luego el alto, en ese orden y ambos.*

Se elige el modo de la conversión A/D con los bits 1, 2 y 3:

M2=0, M1=0, M0=0 : Modo 0 - *Interrupción del conteo final.*

M2=0, M1=0, M0=1 : Modo 1 - *Un disparo programable. Para hacer una sola conversión A/D programada*

M2=0 ó 1, M1=1, M0=0 : Modo 2 - *Generador de pulsos. Para hacer conversiones A/D periódicas programadas*

M2=0 ó 1, M1=0, M0=1 : Modo 3 - *Generador de ondas cuadradas.*

M2=1, M1=0, M0=0 : Modo 4 - *Estrobo disparado por software.*

M2=1, M1=0, M0=1 : Modo 5 - *Estrobo disparado por hardware.*

Los únicos modos que se utilizan en este trabajo son el 1 y 2.

Se selecciona el tipo de conteo mediante el bit 0:

0 : *Conteo binario de 16 bits (cuenta entre 0 y 65,535 (2^{16}))*

1 : *Conteo decimal (BCD) (se realiza entre 0 y 9999 (10^4))*

Al programar el 8253 se necesita tener presente que:

1. Para cada contador, el byte de control debe escribirse antes de que se escriba el conteo inicial.
2. El conteo inicial debe seguir el formato de conteo especificado en el byte de control (si se estableció entrada del byte menos significativo seguida por el byte más significativo deben hacerse dos escrituras en ese orden).

Seleccionado el modo 2 y la forma 4 se programa la frecuencia de las conversiones A/D. Empleando los contadores 1 y 2 en cascada puede hacerse más pequeña la frecuencia de las conversiones A/D (período más largo). Esta es la forma que se emplea en el presente caso. Después de cargarse el contador 1 empieza a contar regresivamente desde la cuenta inicial y cuando llega a cero dispara el proceso de carga del contador 2. Cuando éste llega a cero dispara la conversión A/D y también el proceso de carga del contador 1, repitiéndose el ciclo. Como los contadores 1 y 2 actúan en cascada, si se denotan por HB1 y LB1 los bytes alto y bajo del contador 1 y por HB2 y LB2 los respectivos al contador 2, el tiempo de conteo será:

$$\text{Tiempo} = (\text{HB1} \times 256 + \text{LB1}) \times (\text{HB2} \times 256 + \text{LB2}) \times 0.5 \mu\text{s}$$

Para programar la conversión A/D utilizando los contadores los pasos a seguir son:

1. Se hace un direccionamiento hacia la tarjeta PCL-712. Para ello se le asigna a la variable entera BASE la dirección de la tarjeta que corresponde al hexadecimal 220.
2. Se designa el canal donde se efectuará la conversión A/D. Esto se hace escribiendo en el registro BASE+10 el número del canal deseado.
3. Se selecciona la forma de realizar la conversión A/D, en este caso, se le indica a la tarjeta que el disparo de la conversión A/D lo harán los contadores, para lo cual se asigna un 2 al

registro BASE+11.

4. Se elige el modo 2, el contador 1 la forma 4, y el tipo de conteo binario con el byte de control, para ello se carga éste con el hexadecimal 74 que corresponde a las instrucciones:

Bit	7	6	5	4	3	2	1	0
74 =	0	1	1	1	0	1	0	0

Elige el contador 1; Lectura y carga del byte menos significativo primero y luego el más significativo; modo 2 y conteo binario.

5. Se le da al contador 1 el número de cuentas cargando primero el valor bajo LB1 y luego el alto HB1.
6. Se selecciona el modo 2, el contador 2 la forma 4 y el tipo de conteo binario con el byte de control, para ello se carga éste con el hexadecimal B4 que corresponde a las instrucciones:

Bit	7	6	5	4	3	2	1	0
B4 =	1	0	1	1	0	1	0	0

Elige el contador 2; Lectura y carga del byte menos significativo primero y luego el más significativo; modo 2 y conteo binario.

7. Se le da al contador 2 el número de cuentas cargando primero el valor bajo LB2 y luego el alto HB2.

Después de que se cargan los contadores se inicia automáticamente el conteo, la salida de los contadores cambia de un valor alto a un valor bajo. Al llegar la cuenta regresiva de pulsos del reloj de la tarjeta para el contador 1 a cero, la salida del contador 1 cambia del valor bajo en que estaba a un valor alto (da un pulso) que provoca el inicio de conteo del

contador 2. Cuando éste llega a cero, su salida cambia del valor bajo que tenía a un valor alto, disparando la conversión y hace que el contador 1 regrese automáticamente a la cuenta inicial. Este proceso de conteo se repite para hacer otra conversión A/D.

Por ejemplo, si el contador 1 se carga de manera que la cuenta inicial sea 3 y el contador 2 se carga de manera que la cuenta inicial sea 2 tendremos una conversión A/D cada $4 \times 3 = 12$ pulsos de reloj de la tarjeta, como indica el siguiente diagrama:

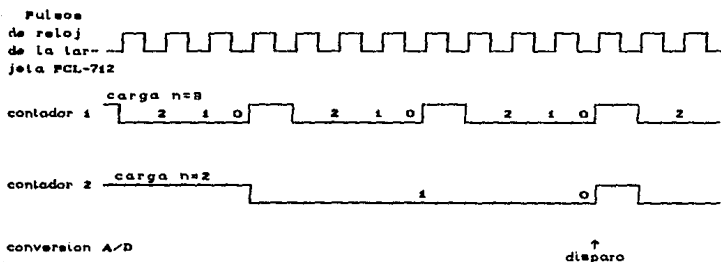


FIGURA II.1: Funcionamiento del disparo regulado (diagrama de tiempos)

A continuación se listan tres programas, uno en Basic, otro en C y el último en Pascal que toman 10 lecturas de un mismo canal utilizando el 8253 para disparar las conversiones A/D, imprimiéndolas. El tiempo entre lecturas lo elige el usuario.

' PROGRAMA DE ADQUISICION DE DATOS DE ITC EN BASIC
' UTILIZANDO LOS CONTADORES

```

10 Dim VO!(250),           ' Dimensionaliza el arreglo real VO!
20 J% = &H220              'Le asigna a J% la dirección de la tarjeta
30 ChCor% = 0             'Se asigna el canal 0 como el de medición de corrientes

40 INPUT "Intervalo de tiempo para la adquisición de datos", Tiempo
45 n% = Int (Sqrt (Tiempo/(5E-7))) ' Se establece el tiempo
50 Hb% = n% \ 256;        'de adquisición de datos
55 Lb% = n% - Hb%;

60 OUT J%+3, &H74         'Se escribe el byte de control
65 OUT J%+1, Lb%          ' Se carga el contador 1
70 OUT J%+1, Hb%

80 OUT J%+3, &HB4         'Se escribe el byte de control
85 OUT J%+2, Lb%          ' Se carga el contador 2
90 OUT J%+2, Hb%

100 OUT J%+10, ChCor%     'Selección del canal donde se realiza
                          'la conversión A/D.

110 OUT J%+11, 2          'Permite que el disparo lo hagan los contadores
140 i=1

150 While (i<11)          'Se procede a la adquisición de datos
180   GoSub 1000           'Lectura de la conversión
190   VO!(i) = (HI*256+LD-2048)*10/4096
200   Print i, VO!(i)     'Se imprime el voltaje leído
210   Next i
220 Wend

250 END                    ' Termina el programa Adquisición

1000 ' Rutina en la que se chequea si se realizó la conversión A/D
1010 ' y se leen los bytes HI y LO de la misma.
1020   HI=INP(J%+5)
1030   While HI>=16        ' Si no se ha realizado la conversión
1040     HI=INP(J%+5)      ' el byte HI sera mayor de 24=16
1050   Wend
1060   LD=INP(J%+4)
1100   Return

```

```

/* ----- */
/*      Programa de prueba del manejo de puertos de un      */
/*      convertidor analógico-digital en C                  */
/*      utilizando los contadores.                          */
/* ----- */
#include <dos.h>
#include <conio.h>
#include <stdio.h>
float leepuerto(void);
void contadores( int cont1, int cont2 );
int base;
float v[500];
/* ----- */
void main( )
{
    int i, tiempo1, tiempo2;
    base = 0x220;                               /*Dirección de la tarjeta*/
    contadores(tiempo1, tiempo2);              /*Se cargan los contadores*/
    outportb( base + 10, 0 );                  /*Selección del canal 0*/
    for (i=0; i<11; i++) v[i]=leepuerto();     /*Lectura de datos*/
    for (i=0; i<11; i++) printf("i=%d,v=%f",i,v[i]); /*Imprime datos*/
    getch( );
}                                               /*Termina el programa principal*/
/* ----- */

float leepuerto( )
{
    int lo, hi;
    do                                           /*Espera hasta que se realice la conversión*/
        hi = inportb( base + 5 );
    while( hi > 15 );
    lo = inportb( base + 4 );
    return((hi*256+lo-2048.0)/2048.0);          /*Regresa el voltaje leído*/
}

/* ----- */

void contadores( int cont1, int cont2 )
{
    char hi, lo;
    hi = cont1 >> 8; lo = cont1 & 0xff;
    outportb(base+3, 0x74);                    /*Escribe el byte de control*/
    outportb(base+1, lo); outportb(base+1, hi); /*Carga contador 1*/
    hi = cont2 >> 8; lo = cont2 & 0xff;
    outportb( base + 3, 0xB4 );                /*Escribe el byte de control*/
    outportb(base+2, lo); outportb(base+2, hi); /*Carga contador 2*/
    outportb(base+11, 2); /*Selección del disparo regulado por cont.*/
}

```

PROGRAM Adquisición;

(Se adquieren datos de ITC mediante la tarjeta PCL-712 con Pascal programando el intervalo de tiempo con los contadores)

Var

Base : word;
ChCor, ChTemp,
Alto, Bajo, Lb, Hb : byte;
Voltaje : real;

Procedure CargoContador;

(Se pide el intervalo de tiempo que se desea entre la toma de dato y dato y se cargan los contadores de la tarjeta.)

Begin

Write('Intervalo de tiempo entre la adquisición de dato y dato');

ReadLn (DelTiempo);

n := Round (Sqrt (DelTiempo/(5E-7))) ;

Hb:= n div 256;

Lb:= n - Hb;

PortW [Base + 3] := \$74; (Se escribe el byte de control)

PortW [Base + 1] := Lb; (Se carga el contador 1)

PortW [Base + 1] := Hb;

PortW [Base + 3] := \$B4; (Se escribe el byte de control)

PortW [Base + 2] := Lb; (Se carga el contador 2)

PortW [Base + 2] := Hb

End; (Termina CargoContador)

Procedure Conversión;

(Se checa si se realizó la conversión analógico-digital y se leen los bytes Hi (Alto) y Lo (Bajo) de la misma.)

Begin

Repeat

Alto := PortW [Base + 5]

Until (Alto<16);

Bajo := PortW [Base + 4];

Voltaje := (Alto*256+Bajo-2048) * 10 / 4096

End; (Termina Conversión)

Begin

Base := \$220; (Se asigna a Base la dirección de la tarjeta)

ChCor:=0; (Selecciono al canal 0 como el de medición de corrientes)

CargoContador; (Se establece el tiempo entre dato y dato)

contador:=1; (Se inicia la adquisición de datos)

PortW [Base+10]:=ChCor;

PortW [Base+11]:=2; (Se permite que los contadores de la tarjeta hagan el disparo)

While (contador<(11)) Do

begin

conversión;

WriteLn (i, Voltaje); (Se imprimen los voltajes)

end;

End.

II.2. UN PROGRAMA DE ADQUISICION Y ANALISIS : ITC.EXE

II.2.1 DESCRIPCION DEL PROGRAMA ITC.EXE

El programa ITC.EXE está escrito en Turbo-Pascal versión 5.5 y su objetivo es hacer la adquisición y ajuste de datos de ITC. El programa inicia desplegando en pantalla un menú con 11 opciones posibles. A continuación se describe cada una de ellas.

1. *Introducir datos manualmente en un archivo.*

Acepta datos de tiempo, temperatura y corriente de un proceso de ITC introducidos por el teclado y los almacena en un archivo cuyo nombre (dado por el usuario) se verifica que no exista en el disco para evitar encimar datos.

2. *Leer datos de un archivo ya existente.*

Se abre el archivo de trabajo mediante el procedimiento *Abro-Archivo*, en el que se puede ver el directorio del disco en la unidad "A:\\" con solo teclear "D". Si se proporciona un nombre de archivo no existente, se pregunta al usuario por el nombre del archivo que se quiere leer hasta que éste exista. Ya abierto el archivo se guardan en unos arreglos los datos de tiempo, corriente y temperatura. El programa cierra el archivo y dice el número total de puntos leídos. Mediante el procedimiento *TemMax*, el programa calcula el punto donde ocurre el máximo del pico de corriente.

3. Cambiar las lecturas de voltaje a temperatura.

Los datos adquiridos de ITC tienen como valores de temperatura, las lecturas de voltaje de un Termopar de Kromel-Alumel, las cuales requeriran ser transformadas a valores de temperatura en grados Kelvin. Esto se realiza mediante la comparación de cada uno de los valores de voltaje adquiridos con los voltajes de una tabla de conversión guardada en el disco duro en un archivo llamado *TermoK.Dat* y que para mayor rapidez se almacena en memoria en dos arreglos temporales al iniciar el programa.

En caso de que a una misma temperatura le correspondan varios valores de corriente, se le asigna un único valor que es el promedio de los mismos. Los datos ya convertidos se guardan en un archivo en disco, pudiendo cambiarse el nombre de éste o dejar el mismo.

4. Adquirir datos de ITC por la computadora.

Se activa la adquisición de datos con la tarjeta PCL-712. Para ello, primero se asigna a una variable de tipo *word*, la dirección de la tarjeta (que es el hexadecimal 220). Luego se cargan los contadores, permitiendo que el usuario proporcione el tiempo de adquisición entre dato y dato, el cual corresponde a:

$$\text{Tiempo} := (\text{HB1}\#256 + \text{LB1}) * (\text{HB2}\#256 + \text{LB2}) * 0.5 \mu\text{s} \quad (2.4)$$

donde HB1 y LB1 son los bytes alto y bajo respectivamente del contador 1 y HB2 y LB2 son los correspondientes para el contador 2, los cuales trabajan en cascada (ver sección II.1.2 donde se describe la tarjeta). El procedimiento *CargoContador* hace la transformación requerida para determinar los valores HB1, LB1, HB2 y LB2, y los envía a los contadores en el orden correcto. Por ejemplo, si el tiempo solicitado es

de 2 segundos el programa calcula HB1, LB1, HB2 y LB2 apropiadas para que éste sea el intervalo de tiempo entre disparo y disparo de la conversión.

El programa presenta en la pantalla del monitor los ejes de la gráfica en los que aparecerán los datos adquiridos después de desplegar la leyenda "para iniciar apriete 'ENTER' ". Al apretar ENTER el programa indica a la tarjeta que el canal donde realizará la conversión analógica-digital será el de corrientes (donde se tiene conectado el electrómetro) y que el disparo de estas conversiones lo darán los contadores. Se toman 10 datos de este canal, los cuales se van graficando en el monitor conforme se van leyendo. Después de estas 10 mediciones se toma una de temperatura, el programa le indica a la tarjeta que la conversión se hará en el canal de temperaturas (donde está conectado el termopar de Kromel-Alumel) y será disparada directamente. Se repite este procedimiento hasta que se han adquirido los datos solicitados. Cuando la adquisición ha concluido la computadora avisa mostrando un letrero con la palabra "ACABE" y emite un sonido de 220 Hertz de frecuencia durante un segundo.

Las lecturas de corriente pueden presentar cambios de escala, propios del electrómetro, que son corregidos con el procedimiento *CambioEscala*. El procedimiento *Huecos* asigna la temperatura a cada dato de corriente mediante una interpolación lineal entre los datos leídos de temperatura (uno cada diez lecturas de corriente).

Por último, se le da al usuario la opción de almacenar los datos en disco por medio del procedimiento *ArchivoDatos*. En él se verifica que el nombre que se le da al archivo no exista en el disco para evitar dañar información previa.

5. *Ver la gráfica de los puntos experimentales.*

Se traza la gráfica de los datos de corriente en función de la temperatura con la escala adecuada y ésta permanece en pantalla hasta que el usuario apriete "ENTER".

6. *Tener un listado de los datos.*

Si el usuario lo solicita, se imprime un listado de los datos en el que se incluye: el número de punto, el tiempo, la corriente y la temperatura. En caso de pedirlo por pantalla, cada dato aparecerá cada medio segundo, pudiéndose detener el listado apretando "PAUSA".

7. *Modificar la línea base.*

Se cambian los valores extremos de la línea base, modificando todos los valores de corriente con los datos extremos tecleados por el usuario, mostrándose la gráfica de los puntos modificados de acuerdo con la nueva línea base.

8. *Trabajar con una porción de los datos.*

Con esta opción se elige una región de los datos como zona de trabajo. Se permite que el usuario diga los puntos extremos que delimitan la región o bien determine esos puntos al detener líneas verticales que se mueven a través de la gráfica. La zona de trabajo elegida se muestra en pantalla entre dos rectas verticales. Si no se está de acuerdo con esos límites, se permite corregirlos. Si el usuario lo desea se almacena esta porción de los datos en un archivo.

9. *Corregir algunos puntos experimentales.*

Se permiten cambiar los valores de corriente o temperatura de los puntos experimentales que se deseen, ya sea dando el número del punto o encontrándolo con un cursor en la pantalla. Si se quiere se enlistan puntos vecinos.

10. *Hacer algún ajuste a los datos.*

Al elegir esta opción se muestra un nuevo menú. En el que se tiene la oportunidad de:

10.1 *Suponiendo una cinética de primer orden hacer un ajuste de una sola banda.*

De acuerdo con la ecuación:

$$\ln \left\{ \frac{\int_t^{t_f} i(t') dt'}{i(T)} \right\} = \ln \tau_0 + \frac{E}{K T} \quad (1.28)$$

Se hace un ajuste por mínimos cuadrados a la recta que relaciona el logaritmo del Área residual (calculada como $\int_t^{t_f} i dt$) (donde que se encuentra por el método de Simpson) contra el inverso de la temperatura, quitando un punto en cada extremo hasta que se determina la recta con la r de Pearson más cercana a uno (la mejor recta según el criterio de mínimos cuadrados). Se grafica dicha recta y los puntos experimentales. La pendiente de dicha recta por la constante de Boltzmann es la energía de activación y la exponencial de la ordenada al origen es el factor pre-exponencial, el programa calcula estos valores y los reporta.

Si el usuario proporciona la temperatura, el campo eléctrico de polarización, el Área de la muestra y la distancia dipolar, la computadora calcula el número de dipolos que tiene la muestra usando la ecuación:

$$N = \frac{K T_p}{\alpha \mu^2 E_p \beta A} \int_{T_0}^{T_f} i(T') dT' \quad (1.20)$$

Estos resultados pueden imprimirse si son solicitados por el usuario.

10.2 Restar dos picos.

A los datos con los que se está trabajando se les restan los de un archivo que se lee de disco. Se presentan en el monitor en una misma gráfica los puntos de los dos archivos y con una curva continua se muestra el resultado de la diferencia. Si se desea se pueden guardar los datos en disco. De esta forma pueden encontrarse los picos individuales de un espectro complejo obtenidos mediante borrados térmicos.

10.3 Sumar varios picos.

Se suman todos los datos de los archivos que el usuario desee, mostrándose en el monitor en una misma gráfica a todos ellos y con una línea continua el resultado de la suma. En caso de que el usuario lo solicite se almacenan los datos en disco. De esta manera puede observarse si los picos individuales han sido bien determinados, pues su superposición debe dar el espectro envolvente original.

10.4 Mostrar la gráfica de varios picos.

Se muestran en una misma gráfica los picos de diversos archivos (todos con la misma escala).

11. Trabajar con un nuevo archivo.

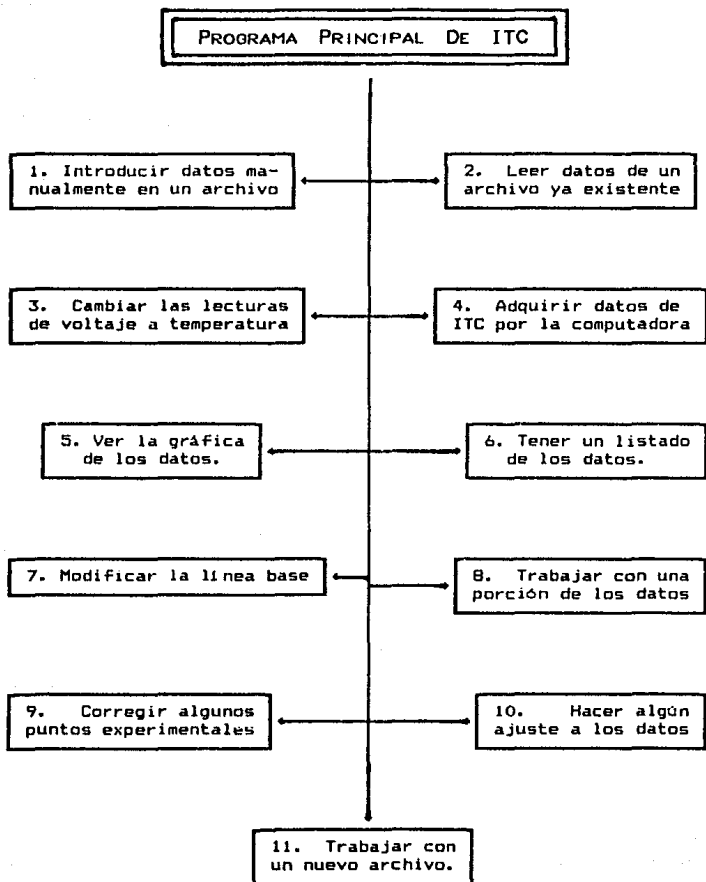
Se borra el archivo de trabajo de memoria y se leen los datos de un nuevo archivo seleccionando el procedimiento *Lectura*.

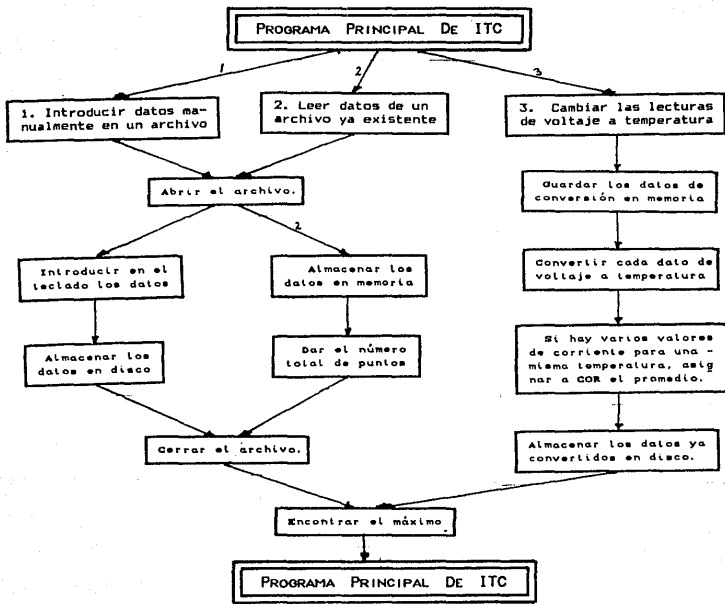
Para salir del programa, cuando se pregunta al usuario si desea volver al menú principal, basta con una "N" para regresar al sistema operativo. Desde cualquier parte del programa, exceptuando cuando se hace la adquisición de datos con la tarjeta o la lectura de un archivo de disco, se puede salir del programa apretando CONTROL C o CONTROL PAUSA.

El programa ITC.EXE ocupa 60 K de memoria y utiliza el coprocesador matemático 8087 para todas las operaciones, con lo cual se vuelve un programa muy rápido (hasta 10 veces más rápido en los procedimientos que realizan operaciones matemáticas solamente, comparado con corridas del programa en el mismo tipo de máquinas sin coprocesador). Al iniciar aparta 8 K para los llamados que hace al sistema operativo MSDOS (desplazar directorio del disco en el drive "A:\ " y verificar la existencia de archivos en él). Para agilizar conversiones carga las tablas necesarias leyendolas del disco duro y almacenandolas en arreglos que son con los que trabaja. Al principio del programa se inhibe la generación del código de chequeo de rangos, evitando demoras innecesarias cuando se trabaja con arreglos.

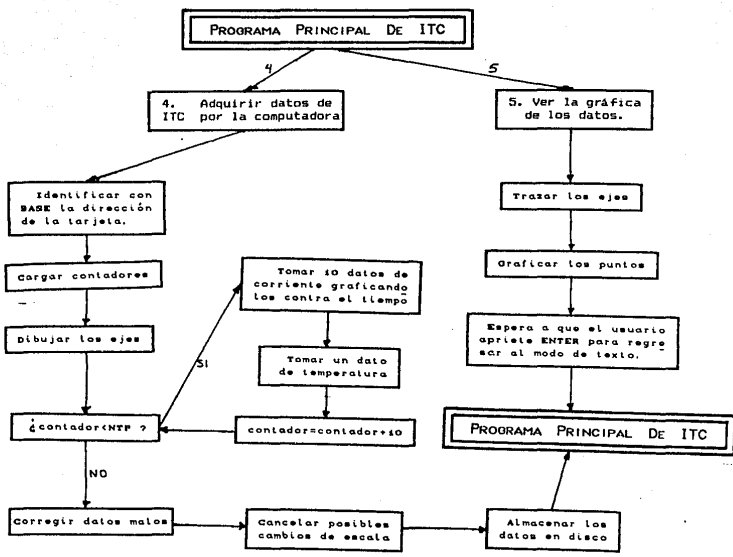
A continuación se muestra un diagrama de flujo del programa y luego se lista éste con los comentarios pertinentes en cada procedimiento en *itálicas encerrados entre llaves ()*. Se proporciona también el listado del programa fuente ITC.Pas escrito en TURBO-PASCAL para que cualquier persona que lo desee haga las modificaciones que necesite para adaptarlo a su problema específico o extienda su uso hacia el análisis de otros fenómenos termoestimulados.

II.2.2. DIAGRAMA DE FLUJO

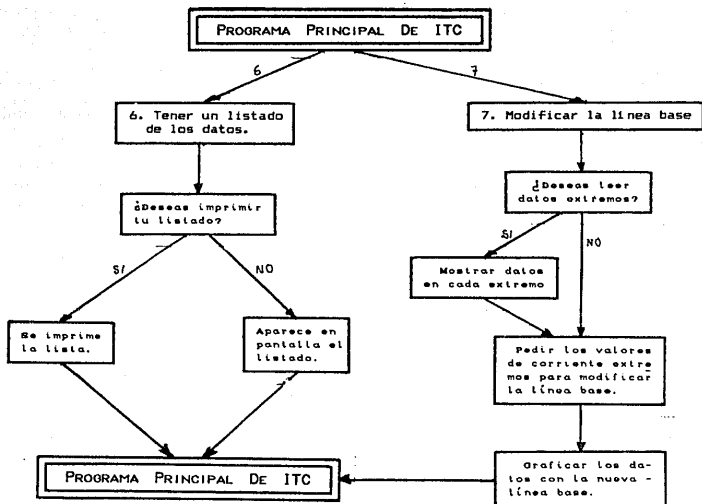




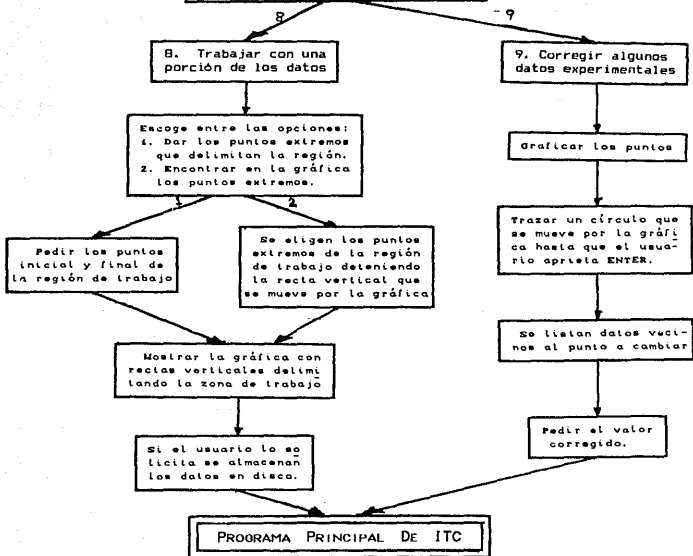
11 - 24



5.2 - 11



PROGRAMA PRINCIPAL DE ITC



PROGRAMA PRINCIPAL DE ITC

10. Hacer algún ajuste a los datos

1. Suponiendo una cinética de primer orden hacer el ajuste de una sola banda.

inicial=1, final=0, $r = -1$, $r_0 = 0$

¿Es $r_1 < r$?

SI

$r = r_1$
inicial=inicial+1
final=final+1

Para n=inicial...NTP-final:
abscisa(n) = 1/KT(n)
ordenada(n) = $\ln \frac{f(t(n))}{L(n)}$

Hacer un ajuste de mínimos cuadrados a la recta que relaciona abscisas y ordenadas. Determinar r.

NO
Graficar la mejor recta según el criterio de mínimos cuadrados.

Reportar r, la energía de activación y el factor pre-exponencial

Si el usuario proporciona T_p , E_p , A y d se calcula el número de dipolos.

2. Restar dos picos

Leer el archivo a restar

Graficar los datos del primer archivo

Graficar el archivo a restar

Calcular la diferencia de los valores de corriente de los dos archivos para cada valor de Temperatura

Graficar la resta

Si el usuario lo solicita se almacenan los datos de la diferencia en disco.

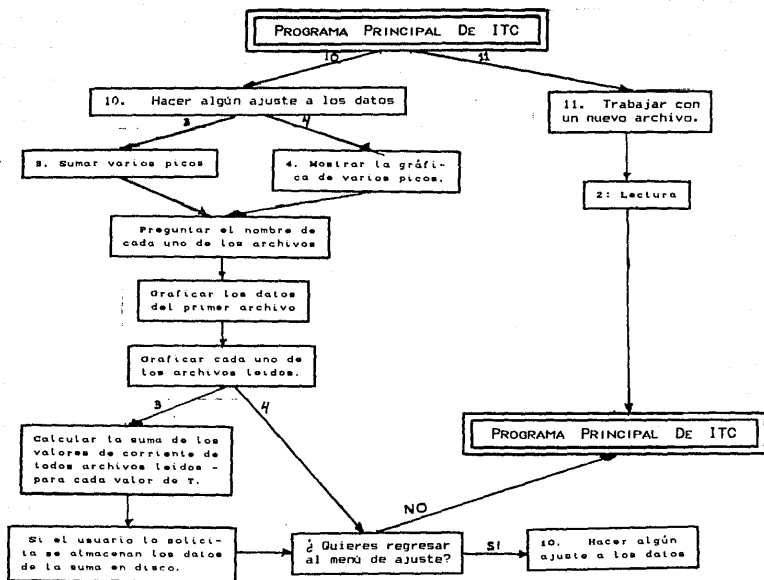
¿Quieres regresar al menú de ajuste?

SI

10. Hacer algún ajuste a los datos

NO

PROGRAMA PRINCIPAL DE ITC



II.2.3. LISTADO DEL PROGRAMA

PROGRAM ITC:

(Este programa hace el ajuste y adquisición de los datos experimentales de un proceso de despolarización térmicamente estimulada (ITC) dando la energía de activación EA y el factor pre-exponencial TAO de los dipolos contenidos en el cristal.)

(%N+) (Usa el coprocesador 8087.)
(%R-) (Inhibe la generación del código de chequeo de rangos)
(%M 8190, 0, 65520) (Deja memoria disponible para llamados al MsDos.)

USES

Crt, Dos, Graph3, Printer;

CONST

KB = 8.617E-05; (Constante de Boltzmann)
tam = 400; (Tamaño máximo del archivo)
fondo = 1; (Color del fondo)
letra = 14; (Color de las letras)
ColGraf = 8; (Color fondo de gráficas)
ColPunto = 1; (Color de los puntos)
ColRecta = 2; (Color de rectas de ajuste)
TiempoPausa = 1000; (Tiempo del Delay (ms))

TYPE

```
arreglo = array [1..Tam] of real;
Datos = record
    ti, I, T : real
end;
```

VAR

```
Data : File of Datos;
DatRec : Datos;
NombreArchivo : string[12];
respuesta : char;
Existe : boolean;
contador, i, n, NTP : integer;
elegi : byte;
RCal, Tvar, tv, Iv, Vv,
EA, TAO, DC,
RCalMax, Tm, Im,
KTE, EKT, Sigma, Nint,
Xi, Yi, Xii, Yii, Ti, Tf,
DelTiempo, AreaTotal : real;
Tiempo, Cor, Temp, Beta,
abscisa, ordenada, Z : arreglo;
```

PROCEDURE CeroArrays;

```
( Se ponen ceros en todos los arreglos a utilizar. )
Begin ( Inicia CeroArrays. )
    For contador:=1 to tam Do
        begin
            Tiempo[contador]:=0;
            Cor [contador] := 0;
            Temp[contador] := 0;
            Beta[contador] := 0
        end
    End; ( Termina CeroArrays. )
```

PROCEDURE Texto;

(Se regresa al modo de texto.)

Begin

TextMode (LastMode);
TextBackground (fondo);
TextColor (letra)

End;

PROCEDURE TemMax;

(Se determina la temperatura del máximo del pico de ITC (T_m)
y la tasa de calentamiento en su entorno ($RCalMax$).)

Var

máximo : integer;

Procedure Betas;

(Se calculan las tasas de calentamiento en cada punto y se
almacenan en el arreglo beta.)

Begin

for i := 1 to NTP Do

begin

DelTiempo := Tiempo[i+1] - Tiempo[i];

if i < 3

Then Beta[i] := (Temp[i+1]-Temp[i]) / DelTiempo;

if (i>=3) and (i<(NTP-4))

Then Beta[i] := (Temp[i+4]-Temp[i-2]) / (6*DelTiempo)

Else Beta[i] := Beta[i-1]

end

End;

```

Function CoMax : real;
  ( Se encuentra el valor de corriente máxima. )
Begin
  Im := 0;
  repeat
    if (Im <= Cor[contador])
      Then begin
        Im := Cor[contador];
        máximo := contador
      end;
    contador := contador + 1
  until (Im > Cor[contador]) or (contador = NTP);
  CoMax := Cor[máximo]
End;

Begin
  ( Inicia TemMax )
  Betas; (Se calculan las tasas de calentamiento)
  contador := 1; (Se encuentra la corriente máxima)
  Im := CoMax;
  Tm := Temp[maximo]; (Se determina la temperatura del máximo)
  RCalMax := Beta[maximo]; (y tasa de calentamiento en su entorno)
  WriteLn; WriteLn;
  WriteLn (' :5, 'La temperatura del máximo es: ', Tm:3:1, ' K');
  WriteLn;
  Write ('y la tasa de calentamiento en su entorno es: ');
  WriteLn (RCalMax:10, ' K/s');
  Delay (TiempoPausa*2)
End; ( Termina TemMax. )

```

PROCEDURE AbroArchivo (Llama: char);

(Se abre el archivo de trabajo y se inicializan los arreglos con 0)

```

Begin
  ( Inicia AbroArchivo )
  WriteLn;
  WriteLn ('Con "D" aparecera el directorio del disco en "A:"');

```

Repeat

```
WriteLn; (Se lee el nombre del archivo)
Case Llama of
  'R' : Write ('Al archivo ', NombreArchivo, ' le vas a
           restar : ');
  'S' : Write ('Archivo ', n:3, ' a sumar : ');
  'M' : Write ('Archivo ', n:3, ' : ');
  else Write ('Dame el nombre de tu archivo: ')
end;
```

ReadLn (NombreArchivo);

WriteLn;

If (NombreArchivo='D') or (NombreArchivo='d')

```
Then begin ( En caso de que el usuario )
  ( lo solicite se le da el directorio del disco en "A:")
  SwapVectors;
  Exec (GetEnv('COMSPEC'), '/C Dir A:');
  SwapVectors;
  Existe := False
end
```

Else begin (Se asigna el archivo.)

NombreArchivo := 'A:' + NombreArchivo;

Assign (Data, NombreArchivo); (Se abre el archivo)

(\$I-) (Se checa si existe el archivo)

Reset (Data); (\$I+)

Existe := (IOresult = 0);

WriteLn;

If (Llama = 'A') and Existe Then

begin

WriteLn;

WriteLn('Bajo ese nombre hay un archivo ya existente.');

WriteLn('Dquieres escribir tus datos sobre ese archivo?');

ReadLn (respuesta);

If (UpCase(respuesta)='S') Then Existe:=True

end

Else If Not Existe Then

WriteLn('No existe ningún archivo llamado ', NombreArchivo)

end

Until Existe;

If (Llama='L')(Se inicializan los arreglos con ceros)

Then CeroArrays

End;

(Termina AbroArchivo.)

PROCEDURE ArchivoDatos;

```
( Se guardan en un archivo de disco los datos )

Begin ( Inicia ArchivoDatos )
  AbroArchivo('A'); ( Se abre el archivo )
  for contador:=1 to NTP do
    begin ( Se guardan los datos )
      with DatRec Do
        begin
          I := Cor[contador];
          T := Temp[contador];
          ti:= Tiempo[contador]
        end;
      Seek (Data, contador-1);
      Write (Data, DatRec)
    end;
  Close (Data) ( Se cierra el archivo )
End; ( Termina ArchivoDatos )
```

PROCEDURE Ejes;

```
( Se trazan los ejes X y Y. )

Begin ( Inicia Ejes )
  GraphColorMode; ( Se llama al modo gráfico )
  GraphBackground (ColGraf);
  Palette (0);
  Draw (15, 15, 15, 175, ColRecta); ( Se dibujan los ejes. )
  Draw (15, 175, 275, 175, ColRecta);
  Draw (275, 173, 275, 177, ColRecta)
End; ( Termina Ejes )
```

PROCEDURE Mensajes;

(Se le agregan a los ejes letreros y escalas.)

```
Begin ( Inicia Mensajes )
  GotoXY (1, 6); Write ('I');
  GotoXY (1, 7); Write ('n');
  GotoXY (1, 8); Write ('t');
  GotoXY (1, 9); Write ('e');
  GotoXY (1, 10); Write ('n');
  GotoXY (1, 11); Write ('s');
  GotoXY (1, 12); Write ('i');
  GotoXY (1, 13); Write ('d');
  GotoXY (1, 14); Write ('a');
  GotoXY (1, 15); Write ('d');
  i := (40-length('Temperatura (K)')) div 2;
  GotoXY (1, 25); Write ('Temperatura (K)');
  GotoXY (1, 23); Write (0);
  GotoXY (1, 2); Write (Im18);
  GotoXY (2, 24); Write (Temp[1]:5:1);
  GotoXY (33, 24); Write (Temp[NTP]:5:1)
End; ( Termina Mensajes )
```

PROCEDURE GrafPuntos;

(Se trazan los puntos con la escala adecuada.)

```
Begin ( Inicia GrafPuntos. )
  T1 := Temp[1];
  Tf := Temp[NTP] - T1;
  For contador := 1 to NTP do
    begin
      Xi := (Temp[contador] - T1) * 250 / Tf + 20;
      Yi := 170 - (Cor[contador] * 150 / Im);
      if ((contador mod 10) = 0)
        then Draw(Round(Xi), 173, Round(Xi), 177, ColRecta);
      Plot (Round(Xi), Round(Yi), ColPunto)
    end
End; ( Termina GrafPuntos. )
```

PROCEDURE GrafArchivo (parámetro : char);

{ Grafica los datos de un archivo. }

Begin { Inicia GrafArchivo. }

contador := 1;

repeat

Seek (Data, contador-1);

Read (Data, DatRec);

Xi := (DatRec.T - T1) * 250 / Tf + 20;

Yi := 170 - (DatRec.I * 150 / Im);

Plot (Round(Xi), Round(Yi), (Colpunto+1));

If (Temp[contador] = DatRec.T) Then

Case parámetro of

'R' : Cor[contador] := Cor[contador] - DatRec.I;

'S' : Cor[contador] := Cor[contador] + DatRec.I

end;

contador := contador + 1

until (DatRec.ti=0) and (DatRec.I=0) and (DatRec.T=0);

End; { Termina GrafArchivo. }

PROCEDURE GrafOperación;

{ Se grafica la suma o resta de datos de archivos realizada. }

Begin { Inicia GrafOperación }

contador := 1;

repeat

Xi := ((Temp[contador] - T1) * 250 / Tf) + 20;

Xii:= ((Temp[contador+1]-T1) * 250 / Tf) + 20;

Yi := 170 - (Cor[contador] * 150 / Im);

Yii:= 170 - (Cor[contador+1]*150 / Im);

Draw (Round(Xi), Round(Yi), Round(Xii), Round(Yii), ColRecta);

contador := contador + 1

until (contador=NTP);

ReadLn; { Se espera a que el usuario apriete "ENTER"}

End; { Termina GrafOperación }

PROCEDURE Delimito (j : byte);

(Se marca en la gráfica uno de los extremos de la región de trabajo)

```
Begin                                     ( Inicia Delimito.           )
Ejes;                                     ( Se dibujan los ejes.       )
Mensajes;
GrafPuntos;                               ( Se dibujan los puntos.     )
Palette (j+1);
contador := elegi;
if (j=1)                                  ( Si ya se eligió un extremo )
  then begin                               ( se dibuja una vertical en él )
    Xi := (Tvar - T1) * 250 / Tf + 20;
    Draw (Round(Xi), 15, Round(Xi), 175, (ColRecta+1))
  end;
repeat                                     (Se traza una vertical que se mueve por la pantalla)
  Xi := (Temp[contador] - T1) * 250 / Tf + 20;
  Draw (Round(Xi), 15, Round(Xi), 170, (ColRecta+1));
  Delay (TiempoPausa);
  Draw (Round(Xi), 15, Round(Xi), 170, 0);
  Yi := 170 - (Cor[contador] * 150 / Im);
  Plot (Round(Xi), Round(Yi), ColPunto);
  contador := contador + 1
until KeyPressed;
ReadLn
End;                                       ( Termina Delimito       )
```

FUNCTION AreaRes (n:integer) : real;

(Usando el método de Simpson se calcula el área bajo el pico de ITC desde el punto n hasta el dato final.)

```
Var
  h, suma : real;
Begin
  Suma := 0;
  repeat
    h := (Tiempo[n+1] - Tiempo[n]) / 3;
    suma := suma + h * (Cor[n] + 4*Cor[n+1] + Cor[n+2]);
    n := n + 2
  until (n=(NTP-final)) or (n=(NTP-final-1));
  AreaRes := suma
End;
```

PROCEDURE IntroDatosMano;

(Se introducen manualmente datos adquiridos de algún proceso termalmente estimulado en un archivo.)

```

Begin                                     ( Inicia IntroDatosMano. )
  AbraArchivo('A');                       ( Se abre el archivo. )
  For i := 1 To 3                          ( Se introducen en el teclado )
    Do WriteLn;                            ( el número de datos. )
  WriteLn ('Dame la cantidad de datos que vas a introducir. ');
  Write (' ': 15);                          ReadLn (NTP);
  WriteLn;                                  (Se introducen en el teclado los datos)
  WriteLn ('Dame tus valores de : ');
  WriteLn;
  Write ('TIEMPO');                         GotoXY(25, WhereY);
  Write ('CORRIENTE');                      GotoXY(50, WhereY);
  WriteLn ('TEMPERATURA'); WriteLn;
  for contador := 1 to NTP Do
    begin
      Read (tv);                            GotoXY(25, WhereY-1);
      Read (Iv);                            GotoXY(50, WhereY-1);
      ReadLn (Vv);
      Tiempo[contador]:= tv;
      Cor [contador] := Iv;
      Temp[contador] := Vv;
      with DatRec do                        ( Se almacenan los datos en )
        begin                               ( el archivo. )
          ti := tiempo[contador];
          I := Cor[contador];
          T := Temp[contador]
        end;
      Seek (Data, contador-1);
      Write (Data, DatRec)
    end;
  with DatRec do
    begin
      ti := 0;                               I := 0;                               T := 0
    end;
  Write (Data, DatRec);
  Close (Data);                             ( Se cierra el archivo. )
  TemMax                                     ( Se encuentran los máximos. )
End;                                         ( Termina IntroDatosMano. )

```

PROCEDURE Lectura;

{ Se hace la lectura de los datos del archivo que se analizará. }

Begin

```
ClrScr;                GotoXY (5, 6);
If (NombreArchivo=' ') or (Not Existe)
  Then AbroArchivo ('L');      {Se abre el archivo de Trabajo}
contador:=1;                { Se guardan los datos de }
Repeat                       {tiempo, corriente y tempera-}
  Seek (Data, contador-1);    {tura en los arreglos Tiempo.}
  Read (Data, DatRec);       {Cor y Temp. }
  tiempo[contador]:= DatRec.ti;
  Cor [contador] := DatRec.I;
  Temp[contador] := DatRec.T;
  contador := contador + 1
until (DatRec.ti=0) and (DatRec.I=0) and (DatRec.T=0);
Close (Data);                { Se cierra el archivo. }
NTP := contador - 2;         {Se da el número total de puntos}
WriteLn;                     WriteLn;
WriteLn (' El número total de puntos es: ', NTP);
TemMax                        {Se encuentra el punto máximo}
End;                           { Termina Lectura. }
```

PROCEDURE Adquisición;

{Se toman datos de ITC mediante la tarjeta de adquisición PCL-712}

Var

```
Base : word;
ChCor, ChTemp,
Alto, Bajo, Lb, Hb : byte;
Voltaje : real;
```

Procedure CargoContador;

(Se pide el intervalo de tiempo que se desea entre la toma de dato y dato y se cargan los contadores de la tarjeta.)

Begin

Write ('Intervalo de tiempo entre la adquisición de dato y dato');
ReadLn (DelTiempo);

n := Round (Sqrt (DelTiempo/(SE-7))));(Se encuentran los)
Hb:= n div 256;(valores del byte Hí y Lo del conteo)
Lb:= n - Hb;

PortW [Base + 3] := \$74; (Se carga el contador 1.)
PortW [Base + 1] := Lb; (dando la instrucción \$74 que)
PortW [Base + 1] := Hb; (indica que se daran primero
el byte Lo y luego el Hí al contador 1.)

PortW [Base + 3] := \$B4; (Se carga el contador 2.)
PortW [Base + 2] := Lb;
PortW [Base + 2] := Hb

End;

Procedure Conversión;

(Se checa si se realizo la conversión analógico-digital y se leen los bytes Hí (Alto) y Lo (Bajo) de la misma.)

Begin

Repeat

Alto := PortW [Base + 5]

Until (Alto < 16);

Bajo := PortW [Base + 4];

Voltaje := (Alto*256+Bajo-2048) * 10 / 4096

End;

Procedure CambioEscala;

(Se verifican todos los datos de corriente para determinar si hubo algún cambio de escala y se cancela este efecto.)

Begin

For i := 1 To (NTP-1) do

If ((Cor[i+1]*10 - Cor[i]) < 0.5)

Then Cor[i+1] := Cor[i+1] * 10

End;

Procedure Huecos;

(Se interpolan los datos de temperatura no tomados suponiendo que en ese intervalo la tasa de calentamiento fue constante)

Var

m : real;

Begin

contador := 10;

Repeat

m := (Temp[contador+10] - Temp[contador]) / 10;

For i := (contador+1) To (contador+9) Do

Temp[i] := Temp[contador] + m*(i-contador);

contador := contador + 10

Until (contador>=NTP);

End;

Procedure MalData;

(Se corrigen los datos cuyo valor este muy por encima o muy por debajo del valor promedio entre datos vecinos.)

Var

bueno : real;

Begin

i := 2;

repeat

bueno := (Cor[i-1] + Cor[i+1]) / 2;

Iv := ABS(bueno - Cor[i]) - ABS(Cor[i+1] - Cor[i-1])/2;

if (Iv>0) Then Cor[i]:=bueno;

i := i + 1

until (i=(NTP-1))

End;

Begin

(Inicia Adquisición.)

ClrScr;

WriteLn;

For contador:=1 to tam do Temp[contador]:=0;

Base := \$220; (Se asigna a Base la dirección de la tarjeta)

ChCor := 0; (Canal de medición de corrientes.)

ChTemp:= 2; (Canal de medición de temperaturas.)

Write ('Número de puntos a adquirir ');

ReadLn (NTP);

CargoContador; (Se establece el tiempo entre dato y dato.)


```

WriteLn;
WriteLn ('Cuando veas la grafica aprieta "ENTER" para empezar a
         tomar datos');
Delay (2*TiempoPausa);

For contador:=1 to (NTP+1) Do
  Tiempo[contador] := contador * DelTiempo;
Im := 2;           Tf:= Tiempo[NTP+1];
Temp[NTP]:=Tiempo[NTP+1];
Ejes;              ( Se trazan los ejes.           )
Mensajes;

ReadLn;            (Se inicia la adquisicion de datos)
contador:=1;
While (contador<(NTP+1)) Do
  begin
    PortW [ Base + 10 ] := ChCor; ( Se prepara la tarjeta para
que el canal donde se realice la conversión analógico-digital sea ChCor)
    PortW [ Base + 11 ] := 2;    ( Se permite que los contadores
de la tarjeta hagan el disparo)

    For i:=contador To (contador+9) Do
      begin
        Conversión;              (Se toman 10 datos de corriente)
        Cor[i] := voltaje;
        Xi := Tiempo[i] * 250 / Tf + 20; (Se gráfica cada dato)
        Yi := 170 - (Cor[i] * 150 / Im);
        Plot (Round(Xi), Round(Yi), ColPunto)
      end;

    PortW [ Base + 10 ] := ChTemp; (Se cambia al canal ChTemp)
    PortW [ Base + 11 ] := 1;    ( Se hace un disparo directo )

    Conversión;                  (Se toma 1 dato de temperatura)
    Temp [ contador+9 ] := voltaje;
    contador := contador + 10
  end;

CambioEscala; (Se detecta si se realizo algún cambio de escala)
Huecos;       ( Se completan los datos de Temp. )
MalDatos;     ( Se corrigen los datos "malos" )

(Avisá que término de tomar datos con un Beep durante un segundo)
WriteLn ('ACABE');
Sound(220);   Delay(TiempoPausa);           NoSound;

ReadLn;      ( Se continúa hasta que el usuario apriete "ENTER")
Texto;      ( Se regresa al modo de texto )

Write ('Quieres almacenar los datos en disco? ');
ReadLn (respuesta);
If (UpperCase(respuesta)='S') (Se almacenan los datos en disco)
  Then ArchivoDatos;

TemMax      ( Se calculan los máximos. )

```

End;

(Termina Adquisición.)

PROCEDURE CambioVaT;

(Se transforman las lecturas de voltaje del termopar type K de "chromel-alumel" con una unión de referencia a 0 Celsius a valores de temperatura en Kelvin.)

Const

tam1 = 280;

Type

valores = record
 Vc, Tc : real
end;

Var

Termopar : file of valores;
ValRec : valores;
DatoV, DatoT : array [1..tam1] of real;
j, repetidos : integer;
Dif, Res : real;

Begin

(Inicia CambioVaT.)

contador := 1; (Se guardan los datos del)
Assign (Termopar, 'C:TermoK.Dta'); (archivo de conversión en los)
Reset (Termopar); (arreglos DatoV y DatoT.)

repeat

 Seek (Termopar, contador-1);
 Read (Termopar, ValRec);
 DatoV[contador] := ValRec.Vc;
 DatoT[contador] := ValRec.Tc;
 contador := contador + 1

until (contador=tam1);

Close (Termopar);

If (NombreArchivo=' ') (Si no se tiene ningún archi-
 Then Lectura; vo en memoria se hace lectura)

contador := 1; j := 5; (Se hace la conversión de)
repeat (voltajes a temperaturas.)

 j := j - 5;

 repeat

 j := j + 1;

```

    res:= Temp[contador] - DatoV[j];
    dif:= (DatoV[j+1]-DatoV[j]) / 2
until (ABS(res)<dif) Or (j=tam1);
Temp[contador] := DatoV[j];
contador := contador + 1
Until (contador=NTP);

contador := 1;                                     ( En caso de que para una mis)
                                                    ( ma temperatura haya varios )
repeat                                             ( valores de corriente, se le )
  j := contador;                                  ( asigna a Cor el promedio. )
  while (Temp[contador]=Temp[j+1])
    Do j := j + 1;
  repetidos := j - contador;
  if (repetidos>0) Then
    begin
      Iv := 0;          Tv := 0;
      for j := 1 to repetidos Do
        begin
          tv := tv + Tiempo[contador+j];
          Iv := Iv + Cor[contador+j]
        end;
      Cor[contador] := Iv / repetidos;
      Tiempo[contador] := tv / repetidos;
      for j := (contador+repetidos+1) To (NTP+repetidos) Do
        begin
          Tiempo[j-repetidos]:=Tiempo[j];
          Cor [j-repetidos] := Cor[j];
          Temp[j-repetidos] := Temp[j]
        end;
      NTP := NTP - repetidos
    end;
  contador := contador + 1
until (Tiempo[contador]=0) and (Temp[contador]=0)
and (Cor[contador]=0);

for j:=NTP to tam do
  begin
    Tiempo[j]:=0;          Cor[j] := 0;          Temp[j] := 0
  end;

ArchivoDatos;                                     ( Se guardan en el archivo )
                                                    ( los datos ya convertidos. )

NTP := NTP - 1;

TemMax                                             ( Se calcula la Temperatura )
                                                    ( del máximo y su beta. )

End;                                               ( Termina CambioVaT. )

```

PROCEDURE Gráfica;

(*Se grafican los puntos experimentales.*)

```
Begin ( Inicia Gráfica. )
  If (NombreArchivo=' ') ( Si no se tiene ningún archi-
    Then Lectura; vo en memoria se hace lectura)
  For i:=1 to 3 Do WriteLn;
  WriteLn ('Tu gráfica permanecerá en pantalla hasta que aprietes
    "ENTER"');
  Ejes; ( Se dibujan los ejes. )
  Mensajes;
  GrafPuntos; ( Se grafican los puntos. )
  ReadLn;
  Texto ( Se regresa al modo de texto)
End; ( Termina Gráfica. )
```

PROCEDURE Listado;

(*Se da un listado de los datos experimentales.*)

```
Begin ( Inicia Listado. )
  If (NombreArchivo=' ') ( Si no se tiene ningún archi-
    Then Lectura; vo en memoria se hace lectura.)
  WriteLn;
  WriteLn ('Deseas que tu listado aparezca por la impresora? ');
  Write(' ':15); ReadLn (respuesta);
  WriteLn;
  if (UpCase(respuesta)='S')
  Then begin ( Se imprime la lista. )
    WriteLn (' ': 10, ' I M P R I M I E N D O . . . ');
    Write (LST, 'Punto', ' ':5, 'TIEMPO', ' ':5, 'CORRIENTE');
    WriteLn (LST, ' ':8, 'TEMPERATURA');
    WriteLn (LST);
  end
```

```

for contador := 1 to NTP do
  begin
    Write (LST, contador:4, Tiempo[contador]:10:2, ' ':7);
    WriteLn(LST, Cor[contador]:14, ' ':4, Temp[contador]:5:3)
  end
end
Else begin
  (Se saca la lista por pantalla)
  Write ('Punto', ' ':5, 'TIEMPO', ' ':5, 'CORRIENTE');
  WriteLn (' ':8, 'TEMPERATURA');
  WriteLn;
  for contador := 1 to NTP do
    begin
      Delay (TiempoPausa/2);
      Write (contador:4, Tiempo[contador]:10:2, ' ':7);
      WriteLn (Cor[contador]:14, ' ':4, Temp[contador]:5:3)
    end
  end
End;
( Termina Listado. )

```

PROCEDURE Porción;

(Se elige una región de datos para trabajar con ella.)

Var

```

Tempfin : real;
inicio : byte;

```

Procedure Región;

Begin

```

elegi := 1;
Delimito(0);
elegi := contador + 1;
inicio := contador - 1;
TVar := Temp[inicio];
Delimito(1);
TempFin := Temp[contador-1];
elegi := contador - 1

```

End;

```

Begin                                     ( Inicia Porción. )
  If (NombreArchivo=' ')                 ( Si no se tiene ningún archi-
    Then Lectura;                          vo en memoria se hace lectura. )

  ClrScr;                                GotoXY (5, 3);
  WriteLn (' Que prefieres?');
  WriteLn;
  WriteLn ('1. Darme los puntos extremos de la región con la que
    trabajarás');
  WriteLn;
  WriteLn ('2. Delimitar la región con líneas verticales en la
    gráfica.');
```

```

  WriteLn;                                Write (' ':15);
  ReadLn (elegi);
  If (elegi=1)
    Then begin
      repeat
        WriteLn;                            Write (' Punto inicial : ');
        ReadLn (Inicio);
        Write (' Punto final : ');
        ReadLn (elegi);
        If (elegi < inicio)
          Then WriteLn ('No seas tramposo ', elegi, ' es menor
            que ', inicio);
          TempFin := Temp[elegi]
        until (elegi >= inicio)
      end
    Else región;

  Ejes;                                    ( Se trazan rectas vertica-)
  Mensajes;                                ( Les delimitando la región )
  GrafPuntos;                              ( de trabajo. )
  Xi := (Temp[inicio] - T1) * 250 / Tf + 20;
  Draw (Round(Xi), 15, Round(Xi), 175, (ColRecta+1));
  Xi := (TempFin - T1) * 250 / Tf + 20;
  Draw (Round(Xi), 15, Round(Xi), 175, (ColRecta+1));
  ReadLn;

  Texto;                                    ( Se regresa al modo de texto)
  WriteLn('Estas de acuerdo con estos limites o quieres corregirlos?')
  ReadLn (respuesta);
  If (UpCase(respuesta)='S')
    Then begin
      Write ('Cuál deseas cambiar el izquierdo (0). el derecho');
      WriteLn (' (1) o ambos (2)?');
      Write(' ':15); ReadLn (elegi);
      If (elegi=2) Then Región
        Else Delimito(elegi);
      If (elegi=0) Then inicio := contador - 1
        Else TempFin:= Temp[contador-1];

      Texto
    end;
end;
```

```

i := 1;           ( Se guardan los datos en los arreglos. )
For contador := inicio To elegi Do
  begin
    Tiempo[i]:= Tiempo[contador];
    Cor[i]   := Cor[contador];
    Temp[i]  := Temp[contador];
    i := i + 1
  end;
NTP := i - 1;
For contador := NTP To tam Do
  begin
    Tiempo[i]:= 0;      Cor[i] := 0;      Temp[i] := 0
  end;
WriteLn ('Deseas guardar esta porción de datos en un archivo?');
Write(' ':10);          ReadLn (respuesta);
If (UpCase(respuesta)='S')
  Then ArchivoDatos;

TemMax           ( Se determina el máximo. )
End;             ( Termina Porción. )

```

PROCEDURE LineaBase;

```

( Se modifica la línea base. )

Var
  LBO, LBN : real;

Begin           ( Inicia LíneaBase. )
  If (NombreArchivo=' ') ( Si no se tiene ningún archi-
    Then Lectura;        vo en memoria se hace lectura)

  ClrScr;      GotoXY (5, 3);
  WriteLn ('Deseas leer datos extremos?');          ( Se permiten leer )
  Write (' ':15);                                     ( datos en cada extremo. )
  ReadLn (respuesta);
  WriteLn;
  if (UpCase(respuesta) = 'S')
    Then begin
      WriteLn ('Numero de puntos en cada extremo. ');
      Write (' ':5);   ReadLn (elegi);
      WriteLn ('Temperatura', ' ':6, 'Corriente');
    end;

```

```

    for contador := 1 to elegi Do
      WriteLn (Temp[contador]:9:1, ' ':5, Cor[contador]:14);
    WriteLn;
    for contador := (NTP-elegi+1) to NTP do
      WriteLn (Temp[contador]:9:1, ' ':5, Cor[contador]:14);
    WriteLn
  end;
WriteLn;
Write ('Dame los valores de corriente extremos para modificar');
WriteLn (' la línea base. ');
ReadLn (LBO);          ReadLn (LBN);
for contador := 1 to NTP Do
  Cor[contador] := Cor[contador]-LBO+contador*(LBO-LBN)/(NTP-1);
TemMax;                (Se encuentra el punto máximo)
Grafica                ( Se muestra la gráfica de los
                        puntos con la nueva línea base)
End;                   ( Termina LíneaBase. )

```

PROCEDURE Corrección;

(Se permiten cambiar los valores de ciertos puntos experimentales introduciéndolos desde el teclado.)

Var

```

  punto, num1, num2 : integer;
  opción : byte;

```

Begin (Inicia Corrección.)

```

  If (NombreArchivo=' ') Then lectura;
  ClrScr;          GotoXY (3, 3);
  WriteLn ('Que quieres hacer : ');
  WriteLn;
  WriteLn (' 1. Introducir el número de punto por el teclado. ');
  WriteLn;
  WriteLn (' 2. Encontrar en la gráfica el punto a corregir. ');
  WriteLn;          Write (' ':10);
  ReadLn (Opción);

```



```

Repeat
  IF (Opción = 1)
    Then begin
      WriteLn;
      WriteLn('Dame el número de punto que deseas corregir. ');
      Write ( ' ':10);
      ReadLn (punto);
      WriteLn
    end
  Else begin
    Ejes;           ( Se dibujan los ejes. )
    Mensajes;
    GrafPuntos;    ( Se dibujan los puntos. )
    contador := elegi;
    repeat         ( Se traza un círculo que
                  se mueve por la pantalla )
      Xi := (Temp[contador] - T1) * 250 / Tf + 20;
      Yi := 170 - (Cor[contador] * 150 / Im);
      Circle (Round(Xi), Round(Yi), 2, (ColRecta+1));
      Delay (TiempoPausa);
      Circle (Round(Xi), Round(Yi), 2, 0);
      Plot (Round(Xi), Round(Yi), ColPunto);
      contador := contador + 1
    until KeyPressed;
    ReadLn;
    Delay (TiempoPausa*4);
    Punto := contador - 1;
    Texto           ( Se regresa al modo de texto)
  end;

GotoXY (3, 3);
WriteLn ('Deseas que te liste puntos vecinos? ');
ReadLn (respuesta);
IF (UpCase(respuesta)='S')
  Then begin
    WriteLn ( ' ':5, 'Temperatura', ' ':6, 'Corriente');
    WriteLn;
    if (punto>=4) Then num1:=punto-4 Else num1:=1;
    if ((punto+4)>=NTP) Then num2:=NTP Else num2:=punto+4;
    for contador:= num1 to num2 do
      begin
        If (contador = punto ) Then WriteLn;
        Write (contador, ' ':4, Temp[contador]:9:1, ' ':3);
        WriteLn (Cor[contador]:15);
        If (contador = punto ) Then WriteLn
      end
    end;

Assign (Data, NombreArchivo);
Reset (Data);
Seek (Data, punto-1);
Read (Data, DatRec);

Write ('Qué deseas cambiar la temperatura (T), la corriente (C)');
WriteLn (' o ambas (A)?');
ReadLn (respuesta);

```

```

if (UpCase(respuesta)='T')
  Then begin
    WriteLn ('Nuevo valor de temperatura ');
    ReadLn (Temp[elegi]);
  end
Else if (UpCase(respuesta)='C')
  Then begin
    WriteLn ('Nuevo valor de corriente ');
    ReadLn (Cor[elegi]);
  end
  Else begin
    WriteLn ('Nuevo valor de temperatura ');
    ReadLn (Temp[elegi]);
    WriteLn ('Nuevo valor de corriente ');
    ReadLn (Cor[elegi]);
  end;
DatRec.I := Cor[elegi];
DatRec.T := Temp[elegi];
Seek (Data, elegi-1);
Write (Data, DatRec);
WriteLn ('Deseas modificar otro punto? ');
ReadLn (respuesta)
Until (UpCase(respuesta)<>'S');
Close (Data)
End;                                     ( Termina Corrección. )

```

PROCEDURE PrimerOrden;

(Se hace un ajuste de un solo pico utilizando el método de áreas reducido suponiendo una cinética de primer orden.)

Const

min = 0.00001;
 Constante = 1.613494E+31;

Var

M, b, r,
 m1, b1, r1, OrMax : real;
 TotalPuntos, i : integer;
 Tp, Ep, A, d, Ndip : real;

Procedure Recta;

(*Se hace un ajuste por mínimos cuadrados a la recta que relaciona el logaritmo del área residual contra 1/T.*)

Var

SumX, SumY, SumXY,
SumX2, SumY2,
numerador, denom : real;

Begin

SumX := 0; SumY := 0; SumXY := 0;
SumX2 := 0; SumY2 := 0;

for contador:=1 to TotalPuntos do

begin

SumX := SumX + abscisa[contador];
SumY := SumY + ordenada[contador];
SumXY := SumXY + abscisa[contador]*ordenada[contador];
SumX2 := SumX2 + Sqr(abscisa[contador]);
SumY2 := SumY2 + Sqr(ordenada[contador])
end;

numerador := (TotalPuntos)*SumXY - SumX*SumY;

denom := (TotalPuntos)*SumX2 - Sqr(SumX);

m := numerador / denom;

b := (SumX2 * SumY - SumXY * SumX) / denom;

denom := Sqrt(denom * ((TotalPuntos)*SumY2 - Sqr(SumY)));

r := numerador / denom

End;

Procedure Puntos;

(*Se calculan las abscisas y ordenadas.*)

Begin

contador := elegi;
TotalPuntos := NTP - contador - final - 2;

i := 0;

repeat

i := i + 1;

contador := contador + 1;

abscisa[i] := 1 / Temp[contador];

ordenada[i] := Ln (AreaRes(contador)/Cor[contador])

until (i=TotalPuntos)

End;

Procedure GráficaRecta;

(*Se traza la mejor recta de acuerdo al criterio de mínimos cuadrados que ajusta a los datos experimentales.*)

Var

Xf, Yf : real;

```

Begin
  Ejes;
  T1 := abscisa[1];
  Tf := abscisa[TotalPuntos] - T1;
  OrMax:=Ordenada[1];
  contador:=2;
  repeat
    if (OrMax<=Ordenada[contador])
      Then OrMax := Ordenada[contador];
    contador := contador + 1
  until (OrMax>Ordenada[contador]);
  For contador := TotalPuntos Downto 1 Do
  begin
    Xi := abscisa[contador];
    Yi := Ordenada[contador];
    Xi := (Xi-T1)*250/Tf + 20;
    Yi := Yi * 150 / OrMax;
    Plot (Round(Xi), Round(Yi), ColPunto)
  end;

  Xi := abscisa[1];
  Yi := m * Xi + b;
  Xi := (Xi-T1)*250/Tf + 20;
  Yi := Yi * 150 / OrMax;
  Yf := m * abscisa[TotalPuntos] + b;
  Xf := (abscisa[TotalPuntos]-T1)*250/Tf + 20;
  Yf := Yf * 150 / OrMax;
  Draw (Round(Xi), Round(Yi), Round(Xf), Round(Yf), ColRecta);
  ReadLn;
  Texto ( Se regresa al modo de texto )
End;

Begin ( Inicia PrimerOrden. )
  elegi := 1;          final := 0;
  r := 0;              m := 0;          b := 0;
  repeat
    r1 := r;          m1 := m;          b1 := b;
    elegi:=elegi*2;   final:=final+1;
    Puntos;          Recta
  until (Abs(r-r1)<min) or (elegi>Round(NTP/4));
  if (elegi>Round(NTP/4)) Then
  begin
    repeat
      r1 := r;          m1 := m;          b1 := b;
      elegi := elegi + 1;
      Puntos;          Recta
    until (r1>r) or (Abs(r-r1)<min);
    repeat
      r1 := r;          m1 := m;          b1 := b;
      final := final - 1;
      Puntos;          Recta
    until (r1>r) or (Abs(r-r1)<min)
  end;
end;

```


PROCEDURE SumaVariosPicos;

(Se suma la cantidad de picos que se deseen.)

Var

nombres : array [1..tam] of string[15];

Begin (Inicia SumaVariosPicos.)

for n:=1 to 3 do WriteLn;
WriteLn (' Dame la cantidad de picos que vas a sumar.');

ReadLn (elegi);

for n:=2 to elegi do (Se lee el nombre de todos)
begin (los archivos a sumar.)
AbroArchivo('S');
nombres[n] := NombreArchivo;
Close(Data)
end;

Ejes; (Se trazan los ejes)

Mensajes;

GrafPuntos; (Se grafican los puntos de nombre[i])

n:=2; (Se grafican cada uno de)
repeat (los picos a sumar.)

Assign (Data, Nombres[n]);
Reset (Data);
GrafArchivo ('S');
Close (Data);
n := n + 1

until (n>elegi);

GrafOperacion; (Se gráfica la suma.)

Texto; (Se regresa al modo de texto)

GotoXY (5, 5);
WriteLn ('Quieres guardar los datos de la suma?');
ReadLn (respuesta);

if (UpCase(respuesta)='S') (Se guardan en disco la suma)
Then ArchivoDatos

End; (Termina SumaVariosPicos.)

PROCEDURE MuestraPicos;

(Se muestran en una misma gráfica picos de diversos archivos.)

Var

nombres : array [1..tam] of string[15];

Begin (Inicia MuestraPicos.)

for n:=1 to 3 do WriteLn;
WriteLn (' Dame la cantidad de picos que quieres ver. ');
ReadLn (elegi);

for n:=2 to elegi do (Se lee el nombre de todos)
begin (los archivos a ver.)
AbroArchivo('M');
nombres[n] := NombreArchivo;
Close(Data)
end;

Ejes; (Se trazan los ejes.)
Mensajes;

GrafPuntos; (Se grafican los puntos de nombre[i])

n:=2; (Se grafican cada uno de los picos)

repeat
Assign (Data, Nombres[n]);
Reset (Data);
GrafArchivo('M');
Close (Data);
n := n + 1

until n>elegi;
ReadLn; (Se muestra la gráfica hasta "ENTER")

Texto (Se regresa al modo de texto)

End; (Termina MuestraPicos.)

PROCEDURE MenúAjuste;

{ Se ofrece un menú con diversas opciones para ajustar los datos }

Begin *{ Inicia MenúAjuste. }*

If (NombreArchivo=' ') *{Si no se tiene ningún archivo en memoria se hace lectura}*
 Then Lectura;

Repeat

```

ClrScr;                GotoXY (20, 6);
WriteLn ('QUE DESEAS HACER ?');
WriteLn;

Write (' :5, '1. Suponiendo cinética de primer orden hacer');
WriteLn ('un ajuste de una sola banda.');
```

WriteLn;

```

WriteLn (' :5, '2. Restar dos picos.');
```

WriteLn;

```

WriteLn (' :5, '3. Sumar varios picos.');
```

WriteLn;

```

WriteLn (' :5, '4. Mostrar la gráfica de varios picos.');
```

WriteLn;

```

WriteLn (' :10, 'Escoge 1, 2, 3, o 4 ');
WriteLn;                Write (' :15);
ReadLn (elegi);
```

Case elegi of

```

  1 : PrimerOrden;
  2 : RestaDosPicos;
  3 : SumaVariosPicos;
  4 : MuestraPicos
  else WriteLn (Elegi, 'Es un valor fuera del rango [1..4]')
  end;
```

```

for contador := 1 to 3 Do WriteLn;
WriteLn (' Quieres regresar al menú de ajuste ? ');
Write (' :15);    ReadLn (respuesta)
```

Until (UpCase(respuesta)<>'S')

End; *{ Termina MenúAjuste. }*

PROCEDURE MenúPrincipal;

```
Begin                                     ( Inicia el Menú Principal. )

  ClrScr;                                Window (8, 2, 80, 24);
  WriteLn ('1. Introducir datos manualmente en un archivo. ');
  WriteLn;
  WriteLn ('2. Leer datos de un archivo ya existente. ');
  WriteLn;
  WriteLn ('3. Cambiar las lecturas de voltaje a temperatura. ');
  WriteLn;
  WriteLn ('4. Adquirir datos de ITC por la computadora. ');
  WriteLn;
  WriteLn ('5. Ver la gráfica de los puntos experimentales. ');
  WriteLn;
  WriteLn ('6. Tener un listado de los datos. ');
  WriteLn;
  WriteLn ('7. Modificar la línea base. ');
  WriteLn;
  WriteLn ('8. Trabajar con una porción de los datos. ');
  WriteLn;
  WriteLn ('9. Corregir algunos puntos experimentales. ');
  WriteLn;
  WriteLn ('10. Hacer algún ajuste a los datos. ');
  WriteLn;
  WriteLn ('11. Trabajar con un nuevo archivo. ');
  WriteLn;                                Write(' ':15);

  ReadLn (elegi);
  Window (1, 1, 80, 25);
  ClrScr;                                WriteLn;

  Case elegi of
    1 : IntroDatosMano;
    2 : Lectura;
    3 : CambioVaT;
    4 : Adquisición;
    5 : Gráfica;
    6 : Listado;
    7 : LíneaBase;
    8 : Porción;
    9 : Corrección;
    10 : MenúAjuste;
    11 : begin
          NombreArchivo := ' ';
          Lectura
        end;
```

```
else      writeln (elegi, ' es un valor diferente a las 11
              opciones posibles');
end
End;      { Termina el Menu Principal. }
```

{ EMPIEZA EL PROGRAMA PRINCIPAL }

BEGIN

```
  Texto;
  NombreArchivo:= ' ';
  CeroArrays;
  REPEAT
    MenuPrincipal;
    For i := 1 to 4 Do WriteLn;
    WriteLn ('Quieres regresar al menú principal ? (S, N) ');
    Write (' ':15);
    ReadLn (respuesta);
  UNTIL (Ucase(respuesta)<>'S');
```

END.

III

Resultados

Las mediciones de ITC se realizaron en un criostato evacuado a una presión de 10^{-5} Torr en el rango de temperaturas de 120 a 300 K. El cristal se polarizaba bajo un voltaje de 1200 V durante 5 minutos a una temperatura de 273 K. Para medir la corriente de ITC, la muestra se enfrió hasta 80 K bajo el campo eléctrico externo y entonces después de quitar el campo, se calentó por medio de un caudín con sus electrodos corto-circuitados a través del electrómetro. El electrómetro empleado fue un Keithley digital modelo 616 con una sensibilidad máxima de 10^{-16} Amperes. Un termopar de Kromel-Alumel que tiene un corrector de punta fría marca Omega se utilizó para medir temperaturas. Tanto el electrómetro como el termopar están conectados a dos canales de adquisición de una tarjeta PCL-712 que permite adquirir datos con una computadora PC compatible marca ACER.

III.1 UNA PRUEBA DEL PROGRAMA :

EL SISTEMA KCl:Eu

Para probar el programa ITC.EXE se adquirieron y analizaron los datos de un sistema muy estudiado cuyos parámetros ya son conocidos⁽⁷⁾: KCl:Eu²⁺. Se encontró que:

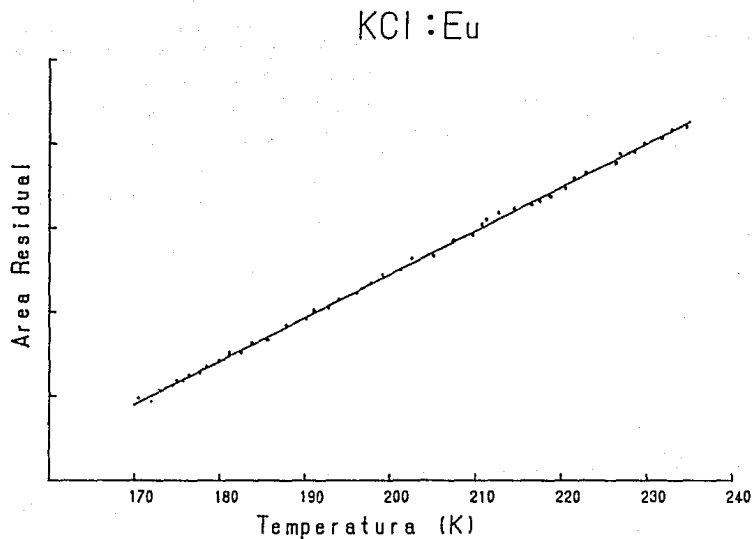
TABLA 3,1.

Analisis del pico de ITC del KCl:Eu²⁺

T _M (K)	E (eV)	τ_0 (s)	Ref.
219 ± 1	0.66 ± 0.02	$6.5 \cdot 10^{-14 \pm 0.5}$	7
224	0.68 ± 0.01	$(3.7 \pm 1) \cdot 10^{-14}$	11
221	0.66 ± 0.03	$4.95 \cdot 10^{-14}$	Este trabajo

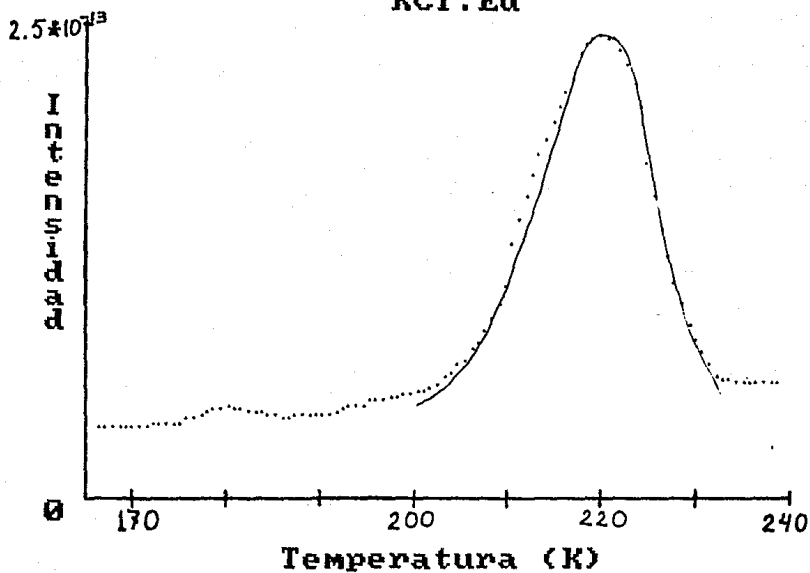
En la gráfica 3.1 se muestra el análisis del pico por el método de Áreas y en la gráfica 3.2 los datos experimentales del

pico de ITC de $KCl:Eu$ adquirido por la computadora y con una línea continua el ajuste con los parámetros reportados.



GRAFICA 1

KCl : Eu



GRAFICA 2

III.2. UNA APLICACION :

EL SISTEMA RbCaF₃:Eu:Mn

Los cristales del tipo de las fluoroperovskitas pueden describirse por la fórmula general ABF₃ donde A es Li, Na, K, etc. y B es Mg, Ba, Zn, etc. Cada ión A⁺ está rodeado por doce iones F⁻ y cada ión B²⁺ por seis iones F⁻. El cristal bajo estudio RbCaF₃:Eu:Mn tiene una estructura cúbica cuyo parámetro de red^(a₀) vale a=4.452 Å.

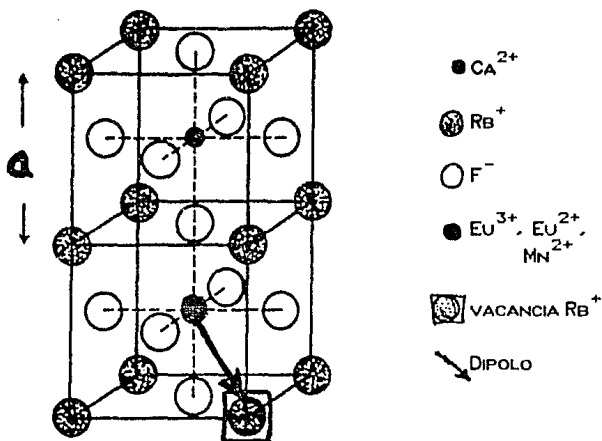


Figura 3.1
Estructura del RbCaF₃:Eu:Mn

Para la muestra de RbCaF₃:Eu:Mn estudios ópticos previos⁽⁶⁾ revelaron la presencia de europio trivalente, europio divalente y manganeso divalente con las concentraciones reportadas en la tabla 3.2. De estos resultados se concluyó que las impurezas entraron sustituyendo al ión Ca²⁺.

TABLA 3.2.
Concentración de impurezas en RbCaF₃:Eu:Mn

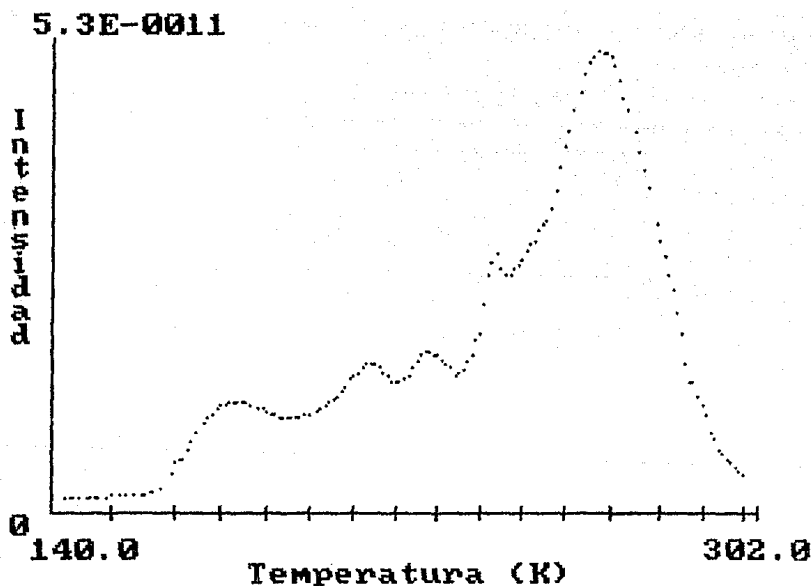
	N ($\times 10^{18}$ cm ⁻³)
Eu ³⁺	24.7
Eu ²⁺	32.0
Mn ²⁺	13.6

Al sustituir el Ca²⁺ por Eu³⁺ se requiere de un compensador de carga para mantener la neutralidad eléctrica de la red cristalina, una vacancia de ión rubidio sería suficiente para ello. Si se supone que éste es el mecanismo de compensación de carga, habrá una interacción electrostática atractiva entre el ión Eu³⁺ y la vacancia de Rb⁺ que puede hacer que la vacancia y el Eu³⁺ se ubiquen cerca uno del otro formandose un *dipolo eléctrico* I-V como se ilustra en la figura 3.1. Por saltos del ión Rb⁺ hacia la vacancia el dipolo podría girar.

Otro posible mecanismo de compensación de carga sería una vacancia de ión Ca²⁺ por cada dos iones de Eu³⁺. En este caso no habría dipolos eléctricos sino defectos cargados más complejos que serían más difíciles de orientar con un campo eléctrico.

Para averiguar si el primer mecanismo de compensación estaba presente en los cristales de RbCaF₃:Eu:Mn debía hacerse el experimento de ITC. Empleando una muestra en forma de disco de 6.4 mm de radio y 1.6 mm de espesor, usando un voltaje de polarización de 1200 V a una temperatura de 273 K durante 5 minutos, se obtuvo la gráfica 3. Como se observa, la curva de

ITC del $\text{RbCaF}_3:\text{Eu:Mn}$ consta al menos de cinco picos cuyos máximos se reportan en la tabla 3.3.



GRAFICA 3

TABLA 3.3.

Maximos de la curva de ITC del RbCaF₃:Eu:Mn

pico	T (K)	i ($\times 10^{-11}$ A)
1	180	1.140
2	213	1.609
3	227	1.737
4	244	2.435
5	267	5.347

Para aislar el pico de más baja temperatura se realizó el borrado térmico siguiente: se calentó la muestra hasta 204 K, se enfrió luego hasta 80 K y se registró la corriente durante un nuevo calentamiento hasta 302 K. El resultado aparece en la gráfica 4. Restando punto a punto ésta de la gráfica 3 mediante la opción 10.2 del programa ITC.EXE se obtuvo el pico aislado mostrado en la gráfica 5. Aplicando la opción 10.1 del programa a los datos de este pico se encontró:

TABLA 3.4.

Análisis del pico de mas baja temperatura del RbCaF₃:Eu:Mn

T _m (K)	E (eV)	τ_0 (s)
173 \pm 1	0.26 \pm 0.01	4.5 $\times 10^{-7}$

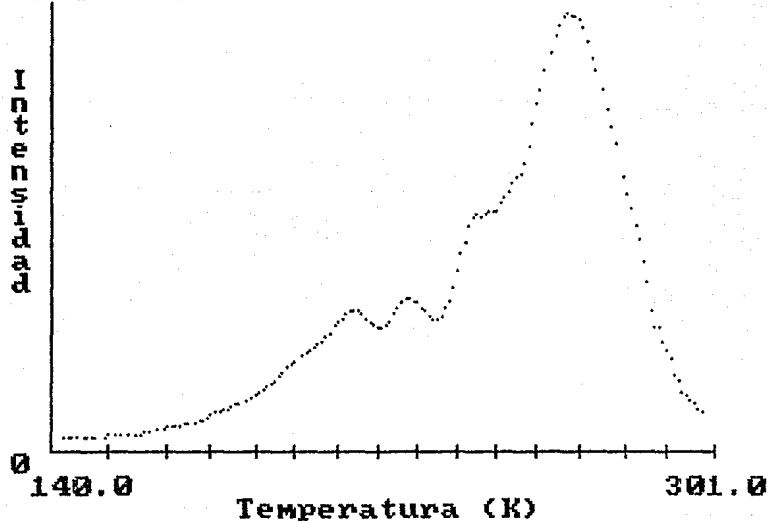
Estos valores se encuentran en el intervalo en que usualmente se detectan picos de ITC debidos a dipolos I-V en cristales iónicos⁽⁷⁾. Por ello consideramos que este primer pico es debido a los dipolos I-V.

Si se supone que la vacancia de catión se ubica a primeros vecinos del europio, esto es, que la distancia dipolar es:

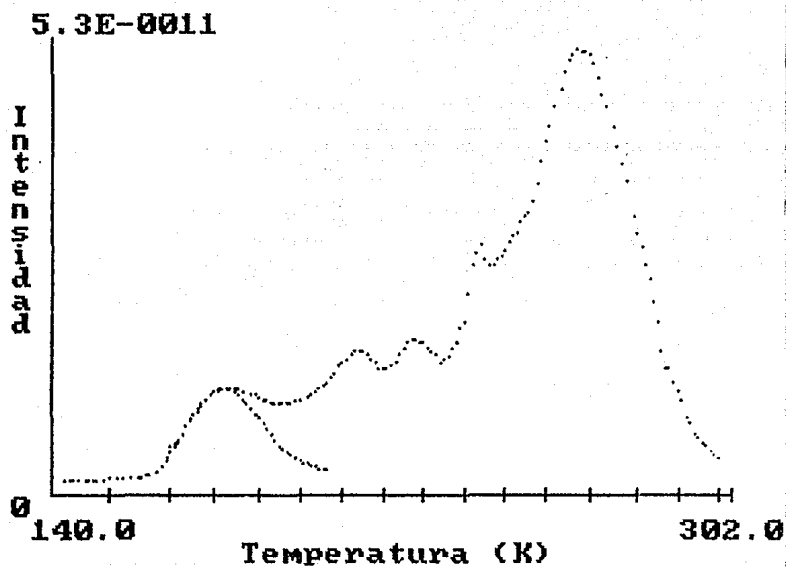
RbCaF :Eu:Mn

3

5.3E-0011



GRAFICA 4



$RbCaF_3:Eu:Mn$

GRAFICA 5

$$d = (\sqrt{3/2}) (4.45 \text{ \AA}) = 3.856 \text{ \AA}$$

Con este valor, usando los parámetros de la tabla 3.4 y la fórmula (1.20), se obtuvo una concentración de dipolos de:

$$N = (4.85 \pm 0.15) \times 10^{-17} \text{ cm}^{-3}$$

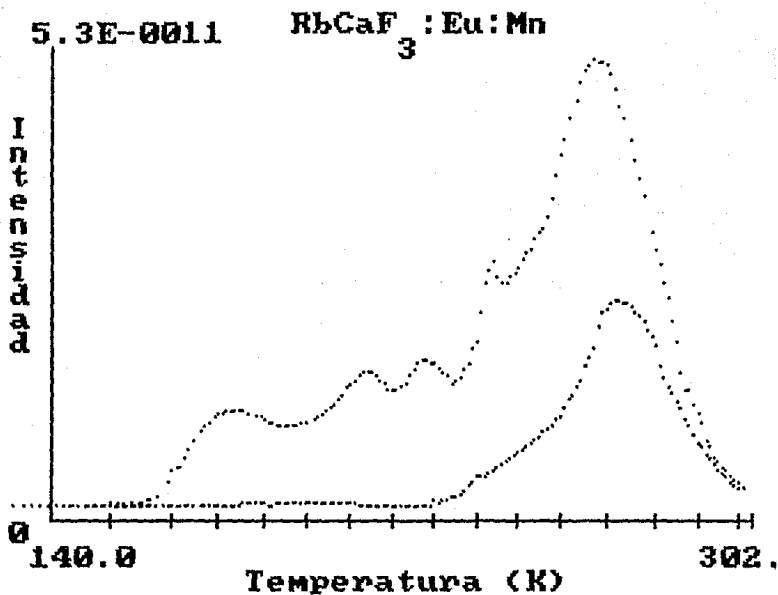
lo cual corresponde a 1/50 del total de Eu^{3+} presente en la muestra (tabla 3.2).

De los estudios ópticos previos en el $\text{RbCaF}_3:\text{Eu:Mn}$ ⁽⁶⁾ se observó también que hay una gran transferencia de energía entre el Eu^{3+} y el Mn^{2+} , así como entre el Eu^{3+} y el Eu^{2+} , también entre el Eu^{2+} y el Mn^{2+} . Esta eficiente transferencia de energía llevó a la conclusión de que los iones respectivos se encuentran ubicados muy próximos unos de otros dentro de la red.

Por lo tanto, podemos suponer que los demás picos de ITC de la gráfica 3 son debidos a dipolos Eu^{3+} -vacancia de Rb^+ perturbados por la presencia de Eu^{2+} , Mn^{2+} o agregados de impurezas vecinas al dipolo. Esto es aceptable pues se ha encontrado que la presencia de agregados cerca de los dipolos I-V provoca picos de ITC a mayor temperatura, por ejemplo para PB^{2+} en NaCl ⁽⁴⁾.

El pico de más alta temperatura apareció idéntico en diversas repeticiones del experimento, independientemente del valor del campo eléctrico de polarización, lo que permite afirmar que no es debido a carga espacial. Por esta razón consideramos que esta relacionado con agregados complejos vecinos a los dipolos Eu^{3+} -vacancia, dándole una gran estabilidad a estos dipolos pues se requiere de mucha energía para desorientarlos.

Para aislar el pico de más alta temperatura se realizó el borrado siguiente: se calentó la muestra hasta 200 K, se enfrió ésta hasta 80 K y se registró la corriente al calentar nuevamente hasta 302 K obteniéndose la curva de ITC mostrada en la gráfica 6. Aplicando la opción 10.1 del programa ITC.EXE se encontraron los parámetros reportados en la tabla 3.5.



GRAFICA 6

TABLA 3.5.

Análisis del pico de mas alta temperatura del RbCaF₂:Eu:Mn

T _m (K)	E (eV)	τ ₀ (s)
273±1	0.80±0.05	9.4*10 ⁻¹⁵

En el presente momento continuamos haciendo borrados térmicos para obtener los parámetros de los otros picos, más difíciles de aislar por el amplio traslape entre ellos. Pensamos realizar experimentos de disolución térmica de los posibles agregados, así como irradiaciones con rayos X para tratar de identificar el origen particular de los picos restantes, como parte de un proyecto de investigación de más larga duración.

IV

Conclusiones

Como conclusiones del presente trabajo tenemos las siguientes:

1. Se aprendió a usar la tarjeta PCL-712 para la adquisición programada de datos de un experimento.
2. Se puso en operación un sistema automático de adquisición de datos de ITC consistente en la tarjeta PCL-712 instalada en una computadora PC compatible marca ACER de 712 K de memoria, disco duro de 20 Mb, un controlador de disco de 360 Kb de 5¹/₄", coprocesador matemático 8087 y monitor de color con tarjeta CGA.
3. Se desarrolló el programa ITC.EXE escrito en Turbo-Pascal para la adquisición y análisis de datos de corrientes termoestimuladas (ITC). El programa tiene once opciones en el menú principal permitiendo entre otras cosas adquirir los datos, guardarlos en un archivo, graficarlos y analizarlos para calcular la energía de activación E, el factor pre-exponencial τ_0 y el número de dipolos N.
4. El programa ITC.EXE se probó adquiriendo datos de una muestra de $KCl:Eu^{2+}$, los resultados obtenidos se compararon con los reportados en la literatura, pudiéndose afirmar que dentro del error experimental coinciden con los obtenidos en este trabajo.
5. Una vez probado el programa se utilizó para el estudio del cristal $RbCaF_2:Eu:Mn$ cuyos resultados más importantes son los siguientes:
 - 5a. La curva de ITC consta de 5 picos con máximos en 173 K, 217 K, 229 K, 245 K y 270 K.
 - 5b. Mediante borrado térmico se aisló el pico de más baja temperatura. El análisis de los datos dio:

$$E = 0.26 \text{ eV}$$

$$\tau_0 = 4.5 * 10^{-7}$$

Estos valores, así como la posición del pico nos llevan a concluir que se debe a dipolos Eu^{3+} -vacancia de Rb^+ (dipolos I-V) situados a primeros vecinos aislados. Hay $1/50$ de la concentración total de Eu^{3+} que se encuentran formando este tipo de dipolos.

5c. Los demás picos son debidos a dipolos I-V perturbados por la presencia de Eu^{2+} , Mn^{2+} y agregados de ellas.

5d. Por medio de borrado térmico se logró aislar el pico de más alta temperatura. El análisis de los datos dió:

$$E = 0.80 \text{ eV}$$

$$\tau_0 = 9.4 * 10^{-15}$$

Este pico apareció independientemente del valor del campo aplicado, por lo que se concluye que no es debido a carga espacial.

Se continúan realizando experimentos para obtener los parámetros de los otros picos.

Como continuación de esta tesis es factible la extensión del programa ITC.EXE para adquirir y analizar los datos de otras técnicas como son la termoluminiscencia y la luminiscencia. Esto permitirá en breve la automatización total del Laboratorio de Propiedades Ópticas y Eléctricas de Sólidos del IFUNAM.

Apéndices

APENDICE I

INCERTIDUMBRES ASOCIADAS A E Y τ_0

El cálculo de las incertidumbres asociadas a la energía de activación y el factor pre-exponencial se hizo de dos formas:

1. Se encontró mediante la desviación cuadrática media a la recta de mínimos cuadrados ajustada la varianza de la pendiente (E/K) y de la ordenada al origen ($\ln \tau_0$) usando⁽¹⁰⁾:

$$\sigma^2 = \frac{1}{n-2} \sum_{i=1}^n \left[Y_i - \left(\frac{E}{K T_i} + \ln \tau_0 \right) \right]^2 \quad (\text{A.1})$$

$$\delta (E/K) = \frac{\sigma^2}{\sum (T_i^{-1} - \bar{T}^{-1})^2} \quad (\text{A.2})$$

$$\delta (\ln \tau_0) = \left[\frac{1}{n} + \frac{\bar{T}^{-2}}{\sum (T_i^{-1} - \bar{T}^{-1})^2} \right] \sigma^2 \quad (\text{a.3})$$

Para los datos de KCl:Eu reportados en la tabla 3.1 estas incertidumbres fueron del 5%.

2. Se observó la reproducibilidad de los datos tomando varias curvas de ITC de una misma muestra. Los resultados obtenidos son los que se reportan como incertidumbres asociadas a E y τ_0 en la tablas 3.1, 3.4 y 3.5.

APENDICE II

PROCESOS TERMOESTIMULADOS

A.II.1. ¿QUÉ SON LOS PROCESOS TERMOESTIMULADOS?

En un *proceso termicamente estimulado (TSP)* la muestra es calentada de una manera controlada y una propiedad física es monitoreada continuamente. A veces el efecto de interés aparece solo cuando la muestra ha sido excitada o tratada de una cierta manera antes de su calentamiento (por ejemplo en ITC, polarizandola). El registro del parámetro variante como una función del tiempo o de la temperatura de la muestra dará lugar a una curva que proveerá de información concerniente a los procesos que tienen lugar en la muestra durante el calentamiento.

Algunos de los procesos estimulados termicamente son: termoluminiscencia, conductividad termoestimulada, corrientes de despolarización termicamente estimuladas (ITC), emisión de electrones, termogravimetría y análisis térmico diferencial. En cada uno de estos procesos, una cantidad física relacionada con la muestra es registrada mientras la temperatura de la muestra se eleva de una forma predeterminada, desde una cierta temperatura baja T_0 . El resultado de la medición es una curva que describe el parámetro medido como una función del tiempo o de la temperatura. Esta curva se conoce comunmente como *curva de calentamiento*, *curva de brillo* o *termograma*.

A.II.2. ECUACIONES GENERALES DE LOS PROCESOS TERMOESTIMULADOS

Los diferentes procesos termicamente estimulados son debidos a distintos procesos microscópicos, no obstante, no solo los termogramas sino también las ecuaciones que describen las variables como una función de la temperatura son sorprendentemente similares. Esta similitud matemática permite el uso de los métodos desarrollados para un TSP específico (en este trabajo, por ejemplo, para ITC) para el análisis de otros. Se mostrará esta semejanza desarrollando en esta sección las ecuaciones de *Termoluminiscencia*.

La termoluminiscencia aparece cuando una muestra sólida (usualmente un aislante o un semiconductor) es calentado después de haber sido radiado a una temperatura baja, T_0 , por alguna clase de radiación ionizante (rayos X, rayos γ , luz ultravioleta, etc.). Parte de la energía absorbida por la muestra durante la radiación puede liberarse en el calentamiento, en forma de luz.

Un modelo que explica la termoluminiscencia establece que la irradiación excita electrones desde la banda de valencia a la banda de conducción; la mayoría de los electrones excitados regresan a la banda de valencia después de un tiempo muy corto (del orden de 10^{-8} segundos) produciendo una luminiscencia que puede detectarse durante la irradiación. Sin embargo, algunos electrones son capturados en niveles de trampas en la banda prohibida (estas trampas están asociadas con defectos de la red tales como vacancias, iones intersticiales o impurezas); cada electron atrapado deja un hoyo en la banda de valencia. Un electron atrapado (u hoyo) puede elevarse termicamente a la banda de conducción (valencia), moverse en el cristal, y finalmente recombinarse con un portador de carga atrapado del signo opuesto. El posible sitio de recombinación se conoce como un centro de recombinación. Si estas recombinaciones son radiativas y ocurren durante el calentamiento de la muestra (el cual se realiza después de que ha cesado la irradiación), se observa un pico termoluminiscente.

Si la intensidad de la señal de termoluminiscencia es proporcional a la razón con la cual los portadores de carga atrapados son liberados, y esta razón es proporcional en cualquier momento a la población de portadores atrapados (es decir, la recombinación es el proceso dominante), entonces se tiene un fenómeno con una cinética de primer orden:

$$I = - (dn/dt) = sn \exp(-E/KT) \quad (A2.1)$$

donde I es la intensidad de la señal de termoluminiscencia, n la concentración de electrones atrapados, E es la energía de activación (o profundidad de la trampa) y s el factor pre-exponencial) el cual se considera independiente de T .

La solución de la ecuación (A2.1) en el caso de una rapidez de calentamiento β constante es:

$$I(T) = n_0 s \exp\left[-\frac{E}{KT} - \frac{s}{\beta} \int_{T_0}^T \exp(-E/KT') dT'\right] \quad (A2.2)$$

La ecuación (A2.2) de termoluminiscencia con una cinética de primer orden es análoga a la ecuación (1.13) que describe el pico de ITC, y se aplica a una gran variedad de procesos termoestimulados.

Suponiendo una fuerte probabilidad de reatrapamiento, se tiene la ecuación para una cinética de segundo orden; esto es, la intensidad de la señal es proporcional en cualquier momento al cuadrado de la población de portadores atrapados:

$$I = - (dn/dt) = s'n^2 \exp(-E/KT) \quad (A2.3)$$

La solución de la ecuación (A2.3) para una β constante es:

$$I(T) = n_0^2 s' \exp\left[-\frac{E}{KT}\right] \left\{ 1 + \frac{n_0 s'}{\beta} \int_{T_0}^T \exp(-E/KT') dT' \right\}^{-2} \quad (A2.4)$$

Tanto en el caso de primer orden como en el de segundo, I como función de T es una curva en forma de pico cuya temperatura

en el punto de máxima intensidad (T_m) depende principalmente de E (a mayor E , mayor T_m) y en menor medida de s y de β (y también de n_0 en el caso de cinética de segundo orden).

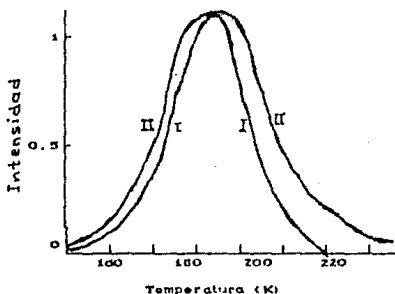


Figura A2.1

Picos de termoluminiscencia para cinéticas de primer (I) y segundo orden (II)⁽⁸⁾.

Muchos picos de termoluminiscencia no cumplen ni la ecuación (A2.2) ni la (A2.4), sino la ecuación empírica:

$$I = - (dn/dt) = s'n^b \exp(-E/KT) \quad (A2.5)$$

donde b (el orden de la cinética) es un parámetro determinado empíricamente, el cual puede tomar cualquier valor entre 1 y 2. La solución de la ecuación (A2.5) para β constante y $b \neq 1$ es:

$$I = n_0 s \exp\left[-\frac{E}{KT}\right] \left\{ 1 + \frac{(b-1)s}{\beta} \int_{T_0}^T \exp(-E/KT') dT' \right\}^{-b/(b-1)} \quad (A2.6)$$

donde $s = s'n_0^{b-1}$.

La ecuación (A2.6) incluye el caso de cinética de segundo orden ($b=2$) y, aunque no es válida para el caso $b=1$ tiende a la ecuación (A2.2) cuando $b \rightarrow 1$.

En todas las expresiones de TSP aparece una integral de la forma⁽⁹⁾:

$$\int_{T_0}^T \exp(-E/KT') dT' \quad (A2.7)$$

La evaluación de esta integral es necesaria para calcular la curva de calentamiento cuando los parámetros son dados. (A2.7) puede reescribirse como:

$$\int_{T_0}^T \exp(-E/KT') dT' = \int_0^T \exp(-E/KT') dT' - \int_0^{T_0} \exp(-E/KT') dT' \quad (A2.8)$$

Podemos definir una función $F(T, E)$ como:

$$F(T, E) = \int_0^T \exp(-E/KT') dT' \quad (A2.9)$$

con lo cual:

$$\int_{T_0}^T \exp(-E/KT') dT' = F(T, E) - F(T_0, E) \quad (A2.10)$$

La función $F(T, E)$ es una función creciente de T lo cual permite despreciar $F(T_0, E)$ para T no muy cercana a T_0 y escribir:

$$\int_{T_0}^T \exp(-E/KT') dT' \approx F(T, E) \quad (A2.11)$$

La integración por partes repetida de (A2.11) conduce a:

$$F(T, E) = T \exp\left[-\frac{E}{KT}\right] \sum_{n=1}^{\infty} (-1)^{n-1} n! \left[\frac{KT}{E}\right]^n \quad (A2.12)$$

No obstante que la expansión asintótica anterior es una serie divergente como tiene signo alternante, la desviación de la función deseada cuando se trunca (A2.12) en un cierto término n no excede el término $n+1$. Los valores absolutos de los términos en la ecuación (A2.12) decrecen con n desde $n=1$ hasta cierta $n=N$ para la cual los términos empiezan a incrementarse en una forma no acotada. Para encontrar dicha N se toma la razón entre dos términos consecutivos:

$$\left| \frac{a_{n+1}}{a_n} \right| = \frac{(KT/E)^{n+1} (n+1)!}{(KT/E)^n n!} = \left[\frac{KT}{E} \right] (n+1) \quad (A2.13)$$

Los valores absolutos de los términos, entonces, decrecen para $n \geq 1$ hasta el término para el cual $(KT/E)n \cong 1$, o sea, $N \cong E/KT$. Entonces, la integral (A2.7) puede aproximarse por:

$$\int_{T_0}^T \exp \left[-\frac{E}{KT} \right] dT' \cong T \exp \left[-\frac{E}{KT} \right] \sum_{n=1}^{E/KT} (-1)^{n-1} n! \left(\frac{KT}{E} \right)^n \quad (A2.14)$$

Una extensión que se está intentando de esta tesis es el análisis computarizado de picos termoluminiscentes. Para ello, con los datos se hace un cálculo inicial del factor preexponencial s y la energía de activación E suponiendo que el orden de la cinética es $b=1$ (utilizándose el ajuste para picos de ITC). Mediante las ecuaciones (A2.6) y (A2.14) se calcula $I(T)$ comparándose con cada uno de los datos. Si el ajuste no es bueno (el criterio se basa en las desviaciones cuadráticas medias) entonces se cambia el orden de la cinética tomando $b=b+0.1$, se verifica la calidad del ajuste (i) tomando la E y s anteriores. Se cambia E y se compara este ajuste (i+1) con el anterior. El orden de la cinética no se cambia hasta que el ajuste (i) sea mejor que el ajuste (i+1). Se repite el proceso hasta encontrar el mejor ajuste determinando b , s y E .

