

2

TAREA-EXAMEN 3	
Título:	Desarrollo del control electrónico de una máquina registradora
Fecha:	27-03-2020
Preparado por:	Fiel Muñoz Teresa Elpidia
Evaluación:	

I. Planteamiento del proyecto

Objetivo:

Diseñar el control electrónico de una caja registradora que cumpla con los requerimientos especificados en los puntos “Requerimientos de hardware” y “Requerimientos funcionales”.

Descripción del proyecto:

El presente proyecto consiste en el diseño de una caja registradora en lenguaje ensamblador para el microcontrolador Tiva C Series TM1294NCPDT, el cual tendrá como única entrada un teclado matricial de 4x4 y de salida un display de 7 segmentos de 4 dígitos.

La caja registradora realizará diferentes funciones que serán habilitadas con ciertas teclas del teclado, tal como lo indica la siguiente tabla:

Tabla 1. Asignación de operación a teclas

Tecla	Operación
A	SUMA
B	TOTAL
C	IVA
D	DESCUENTO

Se realizará una lectura y escritura de datos constante de tal manera que todo valor ingresado se mostrará en el display de 7 segmentos, por lo que los valores estarán constantemente cambiando entre una representación decimal y una de 7 segmentos.

Lo que realicé para este proyecto fue asignar un valor directo a 7 segmentos dependiendo del número de tecla que se presionaba, de tal manera que pudiera mostrar en todo momento los valores ingresados.

Una vez que se logró una comunicación de lectura-escritura con el teclado se procedió al diseño de operaciones en la cual el primer reto fue decodificar de 7 segmentos a un valor hexadecimal para ser operado.

Muchos de los algoritmos que diseñe para el proyecto presentan una complejidad lineal por lo que hicieron del código algo sumamente grande. Después de la decodificación de 7 segmentos a hexadecimal pude operar los valores en dos registros distintos (R7 para enteros, R11 para unidades) y al acumular en estos registros los valores ingresados la parte final fue operar ambos en conjunto para poder después representarlos de nuevo en formato 7 segmentos.

Para la lectura y escritura de números se utilizaron 4 registros distintos, siendo:

- R4 el registro de decenas
- R8 el registro de unidades
- R2 el registro de décimas
- R1 el registro de centésimas

II. Requerimientos del proyecto

Para la construcción del proyecto se presentaron los siguientes requerimientos:

Requerimientos de Hardware

1. Como unidad de procesamiento de la caja registradora utilice el microcontrolador TM4C1294NCPDT.
2. El Ingreso a la máquina de los valores de costos de los productos y las operaciones correspondientes se deben realizar mediante un teclado matricial como el que se muestra en la Figura 1.



Figura 1.

3. La visualización de los valores que se ingresen y el resultado de la operación se presentarán en un “Display de 7 segmentos” de 4 dígitos como el que se muestra en la Figura 2.

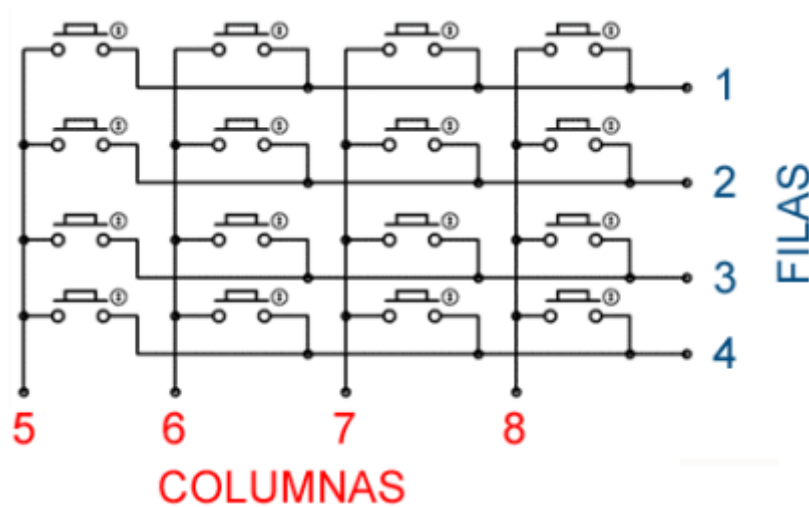


Figura 2.

4. Con un led verde se indicará que la caja registradora abre el compartimiento para guardar dinero.
5. Con un led rojo se indicará que la caja registradora cierra el compartimiento para guardar dinero.
6. Las operaciones que se deseen realizar, se controlarán con las teclas A,B,C y D del teclado matricial de acuerdo a la asignación de la Tabla 1.

III. Marco teórico (antecedentes necesarios para el diseño)

Para el diseño del proyecto se utilizaron conceptos del lenguaje ensamblador como son las subrutinas, los puertos de entrada y salida, y el uso de operaciones aritméticas y lógicas. Sin embargo, como antecedentes teóricos más que el manejo del lenguaje de programación se necesitaron ciertos temas de la materia de Diseño Digital. Los antecedentes que considero necesarios abordar son:

- Decodificación de un teclado matricial
- Decodificación de un display 7 segmentos

Decodificación de un teclado matricial

Para la decodificación del teclado matricial se debe saber que la lectura de un teclado matricial se realiza por filas y columnas, donde cada tecla presionada representa la unión de una fila con una columna.

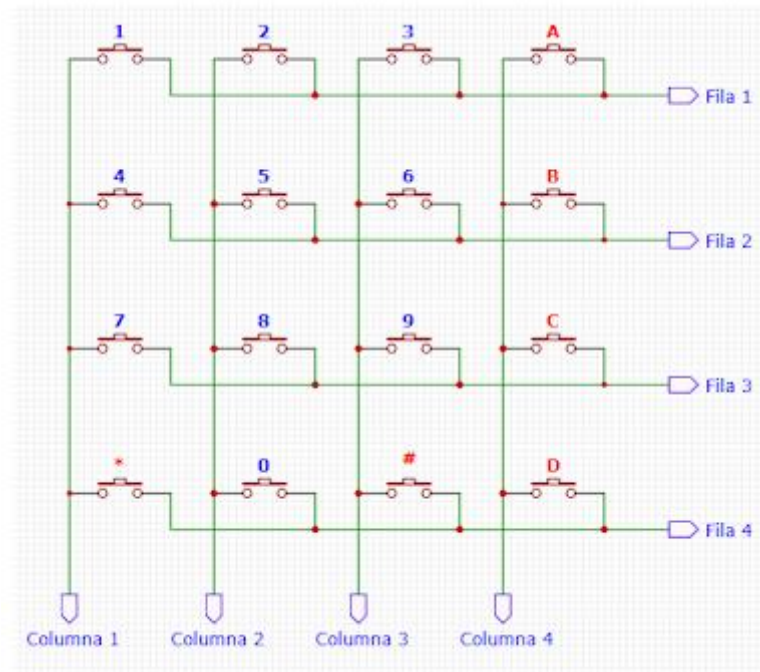


Figura 3. Estructura de un teclado matricial

Lo que se realiza es un barrido en las filas habilitando una por una para que en todo momento se estén habilitando. Dado que el ciclo de reloj es corto, se puede presionar una tecla y en el instante que la fila se habilite con el valor habrá una coincidencia fila-columna.

Siguiendo el diagrama se puede ver con esta coincidencia a que tecla corresponden los valores presionados. Una vez conseguido este valor se puede decodificar a 7 segmentos o a hexadecimal según sea el caso.

Decodificación de un display 7 segmentos

Para la decodificación de un display 7 segmentos primero se debe comprender la diferencia entre un display cátodo común y ánodo común.

El display cátodo común posee un valor 1 común para cada terminal del display por lo que se necesita mandar un cero para el bit de habilitación o aterrizar a tierra el pin común para poder encender con un 1 cada uno de los 7 segmentos del display. A diferencia del display cátodo común, el ánodo común tiene un valor 0 común para cada terminal del display, por lo que se necesita mandar un 1 lógico al bit de habilitación para mostrar un dígito o bien mandar el pin común a un 1 lógico y encender con un 0 cada uno de los 7 segmentos del display.

Ahora bien, una vez que se identifica qué tipo de display se posee, se procede a identificar que led representa cada valor, las terminales se configuran de la siguiente manera:

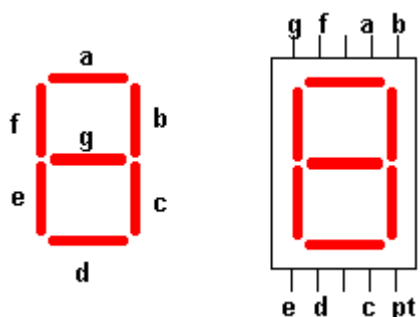


Figura 4. Terminales del display 7 segmentos

Cuando las terminales se han determinado, cada número tendrá una habilitación distinta dependiendo de qué leds deben prenderse y en qué configuración. De esta manera se determinan los dígitos para el display ánodo y cátodo común.

		Catodo Comun							
		Numero	A	B	C	D	E	F	G
Enable	0	1	1	1	1	1	1	1	0
0	1	0	1	1	0	0	0	0	0
0	2	1	1	0	1	1	0	1	1
0	3	1	1	1	1	0	0	1	1
0	4	0	1	1	0	0	1	1	1
0	5	1	0	1	1	0	1	1	1
0	6	1	0	1	1	1	1	1	1
0	7	1	1	1	0	0	0	0	0
0	8	1	1	1	1	1	1	1	1
0	9	1	1	1	1	0	1	1	1

		Anodo Comun							
		Numero	A	B	C	D	E	F	G
Enable	0	0	0	0	0	0	0	0	1
1	1	1	0	0	1	1	1	1	1
1	2	0	0	1	0	0	1	0	0
1	3	0	0	0	0	1	1	0	0
1	4	1	0	0	1	1	0	0	0
1	5	0	1	0	0	1	0	0	0
1	6	0	1	0	0	0	0	0	0
1	7	0	0	0	1	1	1	1	1
1	8	0	0	0	0	0	0	0	0
1	9	0	0	0	0	1	0	0	0



Figura 5. Tabla de números para el display 7 segmentos

IV. Diseño

El diseño realizado para el proyecto consistió principalmente en el barrido del teclado matricial y en la lectura de éste. Se realizó un barrido de las filas y para cada fila se igualó cada posibilidad del teclado (16 opciones), siendo 10 teclas utilizadas para los números, dos teclas para símbolos (decimales y caja) y 4 teclas para operaciones (Suma, Total, IVA y Descuento).

Una vez realizado este barrido cada valor numérico se asignó a cuatro registros distintos, dos para unidades y dos para decimales. Los registros ocupados para este fin fueron:

- R4 el registro de decenas
- R8 el registro de unidades

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 6 de 41	

- R2 el registro de décimas
- R1 el registro de centésimas

Los valores se registraron en cada uno de los registros en formato 7 segmentos cátodo común, y una vez que se realizó esto, se prosiguió a diseñar cada una de las operaciones.

Suma

Para la operación de la suma se realizó una decodificación de 7 segmentos a hexadecimal, para convertir en un principio cuatro registros numéricos. Una vez realizado esa conversión de R4, R8, R2 y R1 se procedió a sumar estos registros y agruparlos en dos registros:

- R7 (registro de enteros)
- R11 (registro de decimales)

Una vez realizada esta conversión se poseerán dos registros para las siguientes operaciones, se representan estos en hexadecimal para ser operables. Una vez realizado esto la cuestión fue que siempre el registro R11 no sobrepasara el valor de 99 para que al realizarse la operación total se puedan convertir enteros y decimales con el mismo algoritmo, además de que se necesita que cuando la suma de decimales sobrepase la unidad los enteros se reinicien y el registro de unidades aumente en uno.

Total

Para la operación total se realizó un proceso parecido al de la suma solo que de manera inversa. Partimos de dos registros donde se obtiene el resultado de manera hexadecimal, se realiza pues una conversión de hexadecimal a 7 segmentos con los enteros y después con los decimales.

IVA

La operación IVA realizó dos porcentajes y restas distintas, una para los enteros y otra para los decimales. Dado que la multiplicación es una propiedad distributiva podemos separar los valores decimales de enteros, sacar su porcentaje de cada uno, restarlo al original y después sumar estos valores.

Sin embargo, lo que yo realicé fue obtener el porcentaje de los enteros y ese porcentaje obtener su parte entera y restarla al valor original para los enteros. Y después el valor decimal restante acumularlo en otro registro. Posteriormente realicé lo mismo con el valor decimal y luego sume a este el valor acumulado del decimal entero para obtener un resultado. Se validó también si los decimales del resultado ya con IVA sobrepasaban la unidad para sumarlo al registro de los enteros.

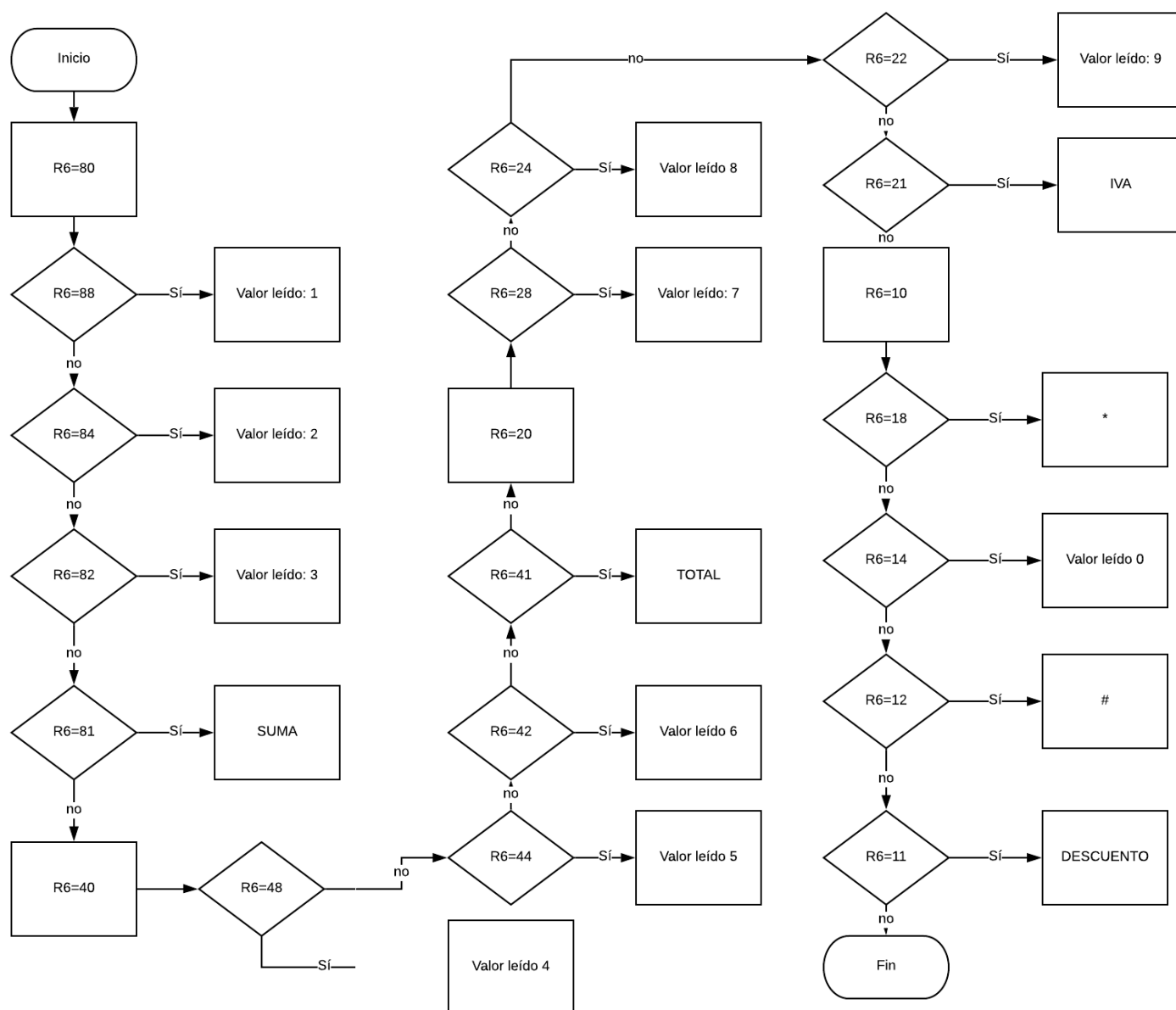
Descuento

Se utilizó el mismo algoritmo de IVA para la programación del descuento, la diferencia fue que previamente se tuvo que decodificar de 7 segmentos a hexadecimal el valor del porcentaje. Cabe resaltar que mi diseño solamente admite porcentajes de descuento enteros dado que solo decodifique los registros para estos valores, es decir, R4 y R8.

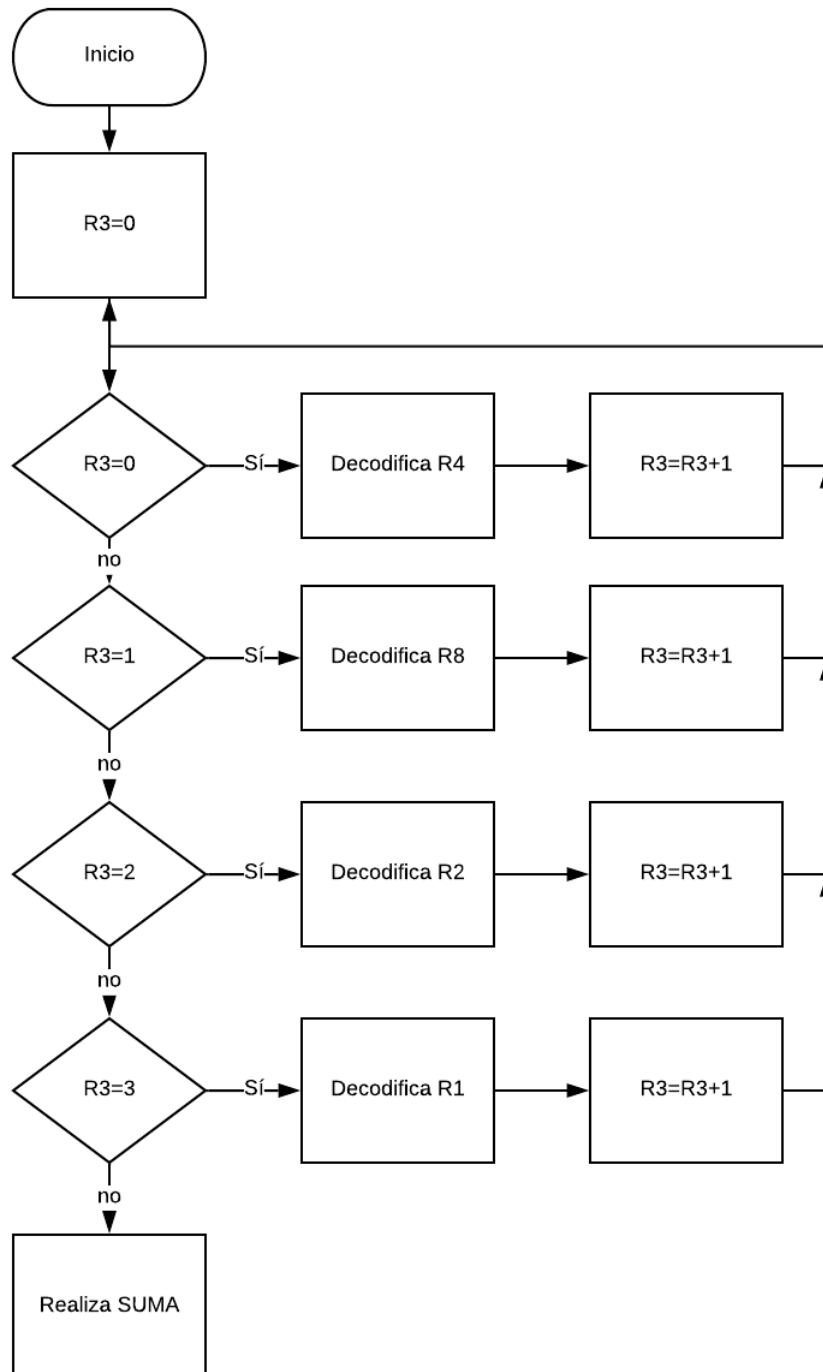
V. Diagrama de flujo y código comentado (en caso de VHDL o código de programa).

Diagramas de flujo

Lectura del teclado matricial

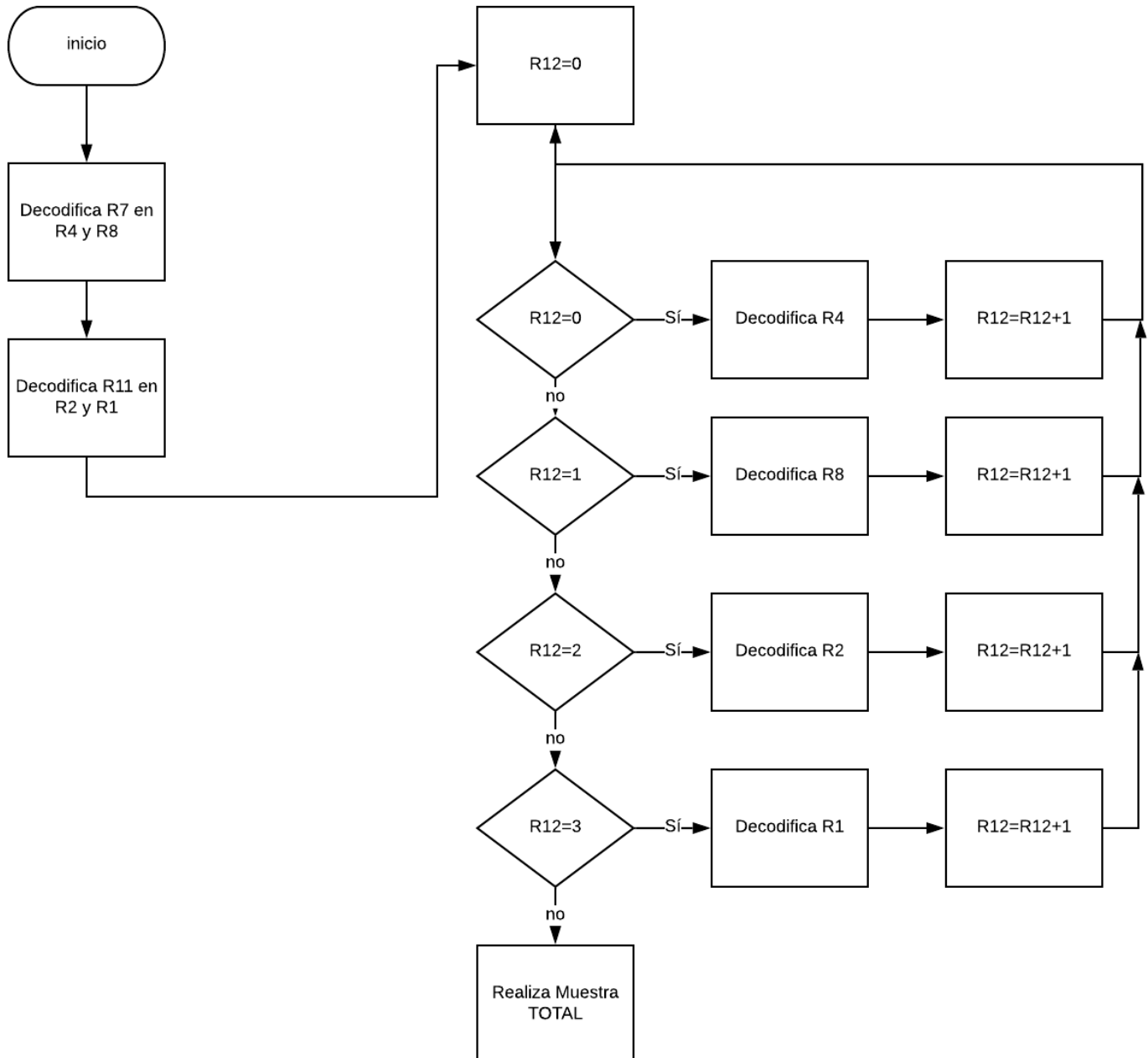


Suma



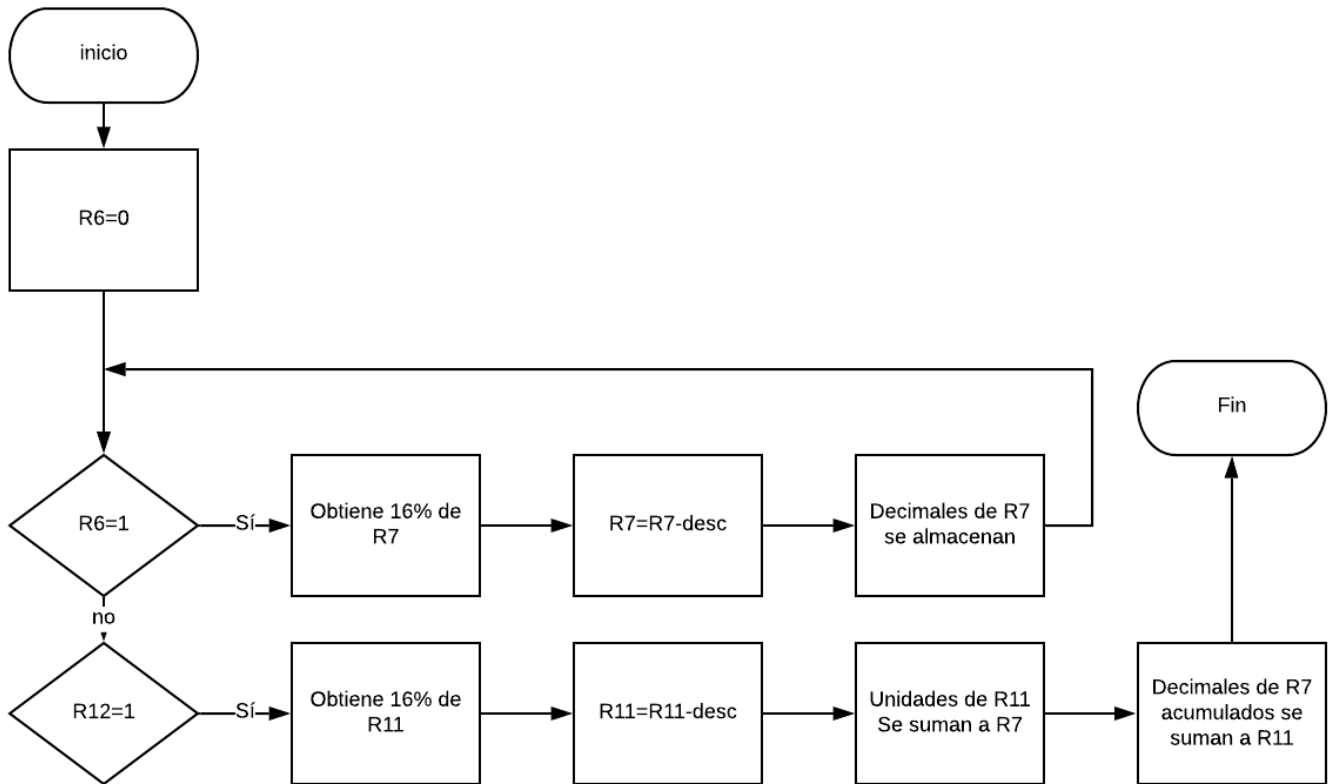


Total

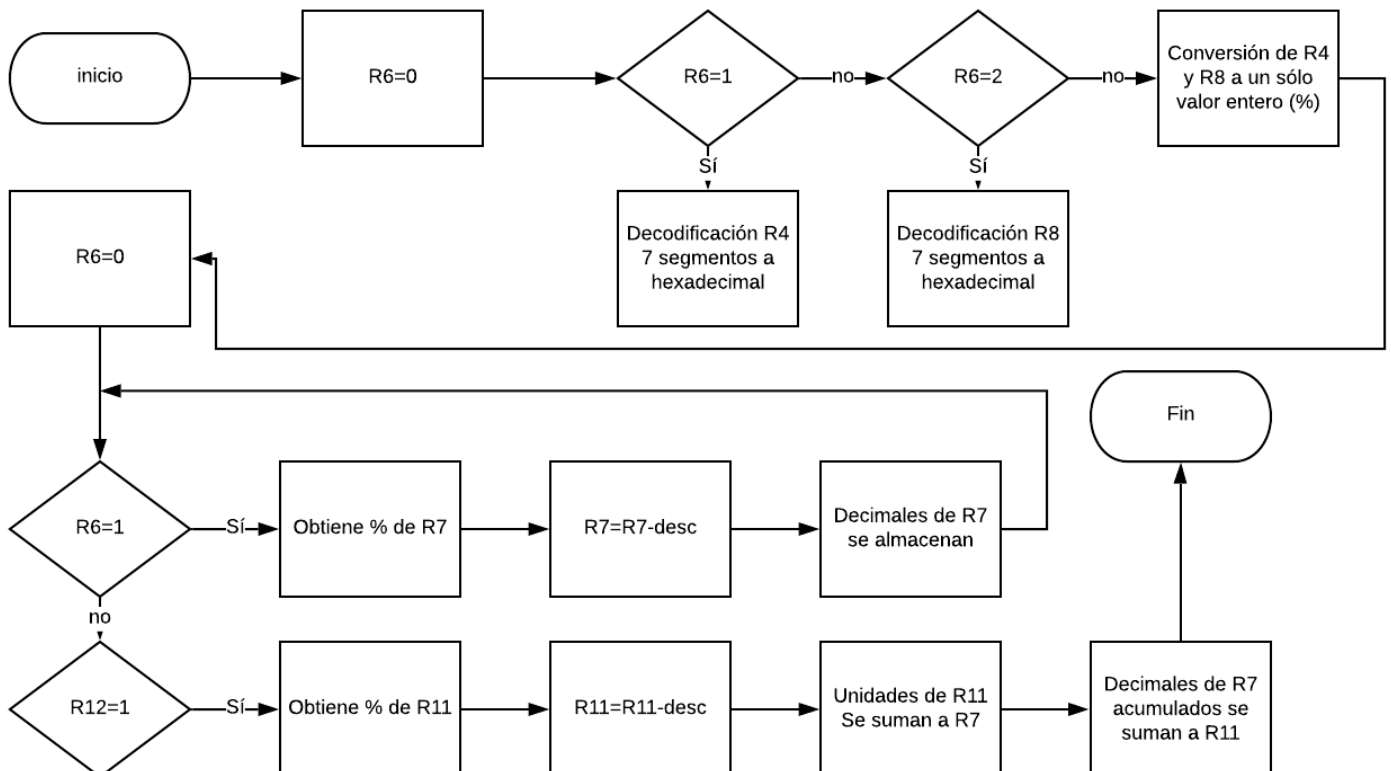






IVA



Descuento



	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 11 de 41	

Código comentado

```

;-----UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO-----
;-----FACULTAD DE INGENIERÍA-----
;-----Microprocesadores y Microcontroladores-----
;-----TAREA-EXAMEN 3 DE CAJA REGISTRADORA-----
;-----SUMA, TOTAL, IVA Y DESCUENTO-----
;----- CREADO POR: TERESA ELPIDIA FIEL MUÑOZ-----
;----- Semestre: 2020-2-----

.global main

.text                ;define codigo del programa y lo ubica en flash

SYSCTL_RCGCGPIO_R   .field 0x400FE608,32    ; REGISTRO DEL RELOJ

GPIO_PORTE_DIR_R     .field 0x4005C400,32    ; Registro de Dirección E
GPIO_PORTE_DEN_R     .field 0x4005C51C,32    ; Registro de habilitación E
GPIO_PORTE_DATA_R    .field 0x4005C03C,32    ; Registro de Datos E
GPIO_PORTE_PUR_R     .field 0x4005C510,32    ; Registro de Pull-Up E

GPIO_PORTK_DIR_R     .field 0x40061400,32    ; Registro de Dirección K
GPIO_PORTK_DEN_R     .field 0x4006151C,32    ; Registro de habilitación K
GPIO_PORTK_DATA_R    .field 0x400613FC,32    ; Registro de Datos K
GPIO_PORTK_PDR_R     .field 0x40061514,32    ; Registro de Pull-Down K

GPIO_PORTM_DIR_R     .field 0x40063400,32    ; Registro de Dirección M
GPIO_PORTM_DEN_R     .field 0x4006351C,32    ; Registro de habilitación M
GPIO_PORTM_DATA_R    .field 0x400633FC,32    ; Registro de Datos M
GPIO_PORTM_PDR_R     .field 0x40063514,32    ; Registro de Pull-Down M

GPIO_PORTH_DIR_R     .field 0x4005F400,32    ; Registro de Dirección H
GPIO_PORTH_DEN_R     .field 0x4005F51C,32    ; Registro de habilitación H
GPIO_PORTH_DATA_R    .field 0x4005F00C,32    ; Registro de Datos H
GPIO_PORTH_PDR_R     .field 0x4005F514,32    ; Registro de Pull-Down H

LEDS1 .EQU 0x02      ; Enciende D2 PUERTO N
LEDS2 .EQU 0x01      ; Enciende D1 PUERTO N
LEDS12 .EQU 0x03     ; Enciende D1 y D2 PUERTO N
LEDS4 .EQU 0x01      ; Enciende D4 PUERTO F
LEDS3 .EQU 0x10      ; Enciende D3 PUERTO F
LEDS34 .EQU 0x11     ; Enciende D3 y D4 PUERTO F
LEDS .EQU 0X00       ; RESET DE LEDS o APAGARLOS TODOS



main

;----- PROGRAMACIÓN DEL CLK -----

comienzo
    LDR R1,SYSCTL_RCGCGPIO_R    ; 1) activar el reloj de los puertos a utilizar
    LDR R0, [R1]
    ORR R0,R0,#0x0A90           ; se valida el bit 11,12,9,7 y 5 para habilitar el reloj
    STR R0, [R1]
    NOP
    NOP                        ; se da tiempo para que el reloj se habilite

;----- DIRECCIONES DE PUERTOS -----

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores		Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves	Grupo 3	Página 12 de 41	

```

LDR R3,GPIO_PORTE_DIR_R ; Determina la dirección del puerto E
LDR R5,GPIO_PORTK_DIR_R ; Determina la dirección del puerto K
LDR R7,GPIO_PORTH_DIR_R ; Determina la dirección del puerto H
LDR R9,GPIO_PORTM_DIR_R ; Determina la dirección del puerto M
MOV R8,#0xFF             ; PM0-PM7 salidas
MOV R6,#0x03             ; PH0 PH1 salidas
MOV R4,#0xF0             ; PK0-PK3 entradas y PK4-PK7 salidas
MOV R2,#0x0F             ; PE3-PE0 salidas
STR R2,[R3]
STR R4,[R5]
STR R6,[R7]
STR R8,[R9]

```

;----- DIGITAL ENABLE DE PUERTOS -----

```

LDR R3,GPIO_PORTE_DEN_R ; Habilita al puerto digital E
LDR R5,GPIO_PORTK_DEN_R ; Habilita al puerto digital K
LDR R7,GPIO_PORTH_DEN_R ; Habilita al puerto digital H
LDR R9,GPIO_PORTM_DEN_R ; Habilita al puerto digital M
MOV R8,#0xFF             ; Habilita PM0-PM7
MOV R6,#0x03             ; Habilita PH0 y PH1
MOV R4,#0xFF             ; Habilita PK0-PK7
MOV R2,#0x0F             ; Habilita PE3-PE0
STR R2,[R3]
STR R4,[R5]
STR R6,[R7]
STR R8,[R9]

```

;----- PULL DOWN DE PUERTOS K,H y M -----

```

LDR R1,GPIO_PORTK_PDR_R ; Dirección de pull-down para el puerto K
MOV R0,#0x0F             ; Habilita las resistencias de pull-down para
salidas puerto K
STR R0,[R1]
LDR R7,GPIO_PORTH_PDR_R ; Dirección de pull-down para el puerto H
MOV R6,#0x03             ; Habilita las resistencias de pull-down para
salidas puerto H
STR R6,[R7]
LDR R9,GPIO_PORTM_PDR_R ; Dirección de pull-down del puerto M
MOV R8,#0xFF             ; Habilita las resistencias de pull-up para las
salidas del puerto M
STR R8,[R9]

```

;----- PULL UP DE PUERTO E -----

```

LDR R3,GPIO_PORTE_PUR_R ; Dirección de pull-up para el puerto E
MOV R2,#0x0F             ; Habilita las resistencias de pull-Up para
salidas puerto E
STR R2,[R3]



```

;----- DATA PUERTO F,N -----

```

LDR R3,GPIO_PORTE_DATA_R ; apunta al Puerto de datos E
LDR R7,GPIO_PORTH_DATA_R ; apunta al Puerto de datos H
LDR R9,GPIO_PORTM_DATA_R ; apunta al Puerto de datos M
AND R0,#LEDS             ; Reinicia valor de R0

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves Grupo 3	Página 13 de 41	

```

AND R1,#LEDS           ; Valor de unidades decimales
AND R2,#LEDS           ; Valor de decenas decimales
AND R8,#LEDS           ; Valor de unidades enteras
AND R4,#LEDS           ; Valor de decenas enteras
AND R5,#LEDS           ; Valor de habilitación para el dígito

```

```

;----- NOMBRE DE TIENDA -----

```

```

MOV R0,#0x00
B etiquetaencendido    ; Ciclo para la duración del nombre de la tienda

```

```

etiquetaencendido

```

```

MOV R6,#0x00           ;
MOV R1,#0x30           ; Valor del display 7 segmentos para la I
MOV R2,#0x67           ; Valor del display 7 segmentos para la P
MOV R8,#0x0E           ; Valor del display 7 segmentos para la L
MOV R5,#0x4F           ; Valor del display 7 segmentos para la E
MOV R4,#0xFE           ; Valor del para el cuarto dígito del display 7

```

```

segmentos

```

```

STR R4,[R3]            ; Habilita puerto E para habilitar el cuarto dígito
STR R1,[R9]            ; Enciende la I en el cuarto dígito
STR R6,[R9]            ; Apaga la I en el cuarto dígito
MOV R4,#0xFD           ; Valor del para el tercer dígito del display 7

```

```

segmentos

```

```

STR R4,[R3]            ; Habilita puerto E para habilitar el tercer dígito
STR R2,[R9]            ; Enciende la P en el tercer dígito
STR R6,[R9]            ; Apaga la P en el tercer dígito
MOV R4,#0xFB           ; Valor del para el segundo dígito del display 7

```

```

segmentos

```

```

STR R4,[R3]            ; Habilita puerto E para habilitar el segundo dígito
STR R8,[R9]            ; Enciende la L en el segundo dígito
STR R6,[R9]            ; Enciende la L en el segundo dígito
MOV R4,#0xF7           ; Valor del para el primer dígito del display 7

```

```

segmentos

```

```

STR R4,[R3]            ; Habilita puerto E para habilitar el primer dígito
STR R5,[R9]            ; Enciende la E en el primer dígito
STR R6,[R9]            ; Apaga la E en el primer dígito
CMP R0,#0x100000        ; Compara R0 con 100,000 para el retardo del nombre de

```

```

la tienda

```

```

BNE etiquetaapagado    ; Si no son iguales salta a etiqueta apagado
AND R0,#LEDS            ; Reinicia valor de R0
AND R1,#LEDS            ; Valor de unidades decimales
AND R2,#LEDS            ; Valor de decenas decimales
AND R8,#LEDS            ; Valor de unidades enteras
AND R4,#LEDS            ; Valor de decenas enteras
AND R5,#LEDS            ; Valor de conteo para las unidades
AND R7,#LEDS            ; Valor que almacena el total de unidades en

```

```

hexadecimal

```

```

AND R11,#LEDS          ; Valor que almacena el total de decimales en

```

```

hexadecimal

```

```

AND R10,#LEDS          ; Valor de conteo para los decimales
MOV R14,#0x00          ; Valor para la caja con LED rojo y LED verde

```

```

B teclado

```

```

etiquetaapagado

```

```

ADD R0,#0x01           ; Suma al valor del ciclo 1

```

```



B etiquetaencendido    ; Regresa a etiquetaencendido para volver a mostrar el

```

```

nombre de la tienda

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves Grupo 3	Página 14 de 41	

----- LECTURA DE TECLADO -----
 ----- VALORES 1-A -----

teclado

```

      LDR R3,GPIO_PORTE_DATA_R      ; R3 apunta a la dirección de datos del puerto E,
habilita los digitos 7 segmentos
      LDR R0,GPIO_PORTK_DATA_R      ; R0 apunta a la dirección de datos del puerto K, puerto para
el teclado matricial
      LDR R9,GPIO_PORTM_DATA_R      ; R9 apunta a la dirección de datos del puerto M, habilita
los led del 7 segmentos
      MOV R6,#0x80                  ; Se realiza el barrido para la primera fila
      STR R6,[R0]                   ; Nos direccionamos a la fila cero
      LDR R6,[R0]                   ; Asignamos el valor a la dirección del puerto K
      CMP R6,#0x88                  ; Comparamos R6 con la tecla 1 presionada
      BNE etiquetaif1              ; Si no son iguales saltamos a etiquetaif1
      MOV R12,#0x30                 ; Igualamos R12 a 30 que es el equivalente a 1 en 7
segmentos
      B start                       ; Saltamos a la etiqueta start
etiquetaif1
      CMP R6,#0x84                  ; Comparamos R6 con la tecla 2 presionada
      BNE etiquetaif2              ; Si no son iguales saltamos a la etiquetaif2
      MOV R12,#0x6D                 ; Igualamos R12 a 6D que es el equivalente a 2 en 7
segmentos
      B start                       ; Saltamos a la etiqueta start
etiquetaif2
      CMP R6,#0x82                  ; Comparamos R6 con la tecla 3 presionada
      BNE etiquetaif3              ; Si no son iguales saltamos a la etiquetaif3
      MOV R12,#0x79                 ; Igualamos R12 a 79 que es el equivalente a 3 en 7
segmentos
      B start                       ; Saltamos a la etiqueta start
etiquetaif3
      CMP R6,#0x81                  ; Comparamos R6 con la tecla A presionada
      BNE etiquetaifA              ; Si no son iguales saltamos a la etiquetaifA
      MOV R3,#0x00                 ; Asignamos R3 con cero para comenzar el ciclo de
retardo de la operación SUMA
cicloA
      CMP R3,#0x90000               ; Comparamos R3 con 90,000 para dar tiempo al ingreso
de datos
      BNE ciclosumaA               ; Si no son iguales saltamos a ciclo sumaA
      MOV R3,#0x00                 ; Igualamos R3 a 0
      B decosuma                   ; Saltamos a decosuma acabado el ciclo de retardo
ciclosumaA
      ADD R3,#0x01                 ; Le suma 1 a R3
      B cicloA                     ; Regresamos al cicloA para continuar el retardo

```



----- DECODIFICACIÓN DE 7 SEGMENTOS A HEXADECIMAL PARA OPERACIÓN SUMA -----

decosuma

```

      CMP R3,#0x00                  ; Iniciamos el ciclo para decodificar los valores
con cero
      BEQ decosumaR4                ; Si son iguales decodificamos el valor de decenas R4
      CMP R3,#0x01                  ; Comparamos R3 con 1
      BEQ decosumaR8                ; Si son iguales decodificamos el valor de unidades R8
      CMP R3,#0x02                  ; Comparamos R3 con 2
      BEQ decosumaR2                ; Si son iguales decodificamos el valor de decimas R2
      CMP R3,#0x03                  ; Comparamos R3 con 3



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 15 de 41	

```

R1      BEQ decosumaR1          ; Si son iguales decodificamos el valor de centésimas
decosumaR1
    MOV R6,R1                ; Iguala R6 al valor R1
    B decoRN                 ; Salta para decodificar el registro R6
decosumaR4
    MOV R6,R4                ; Iguala R6 al valor R4
    B decoRN                 ; Salta para decodificar el registro R6
decosumaR8
    MOV R6,R8                ; Iguala R6 al valor R8
    B decoRN                 ; Salta para decodificar el registro R6
decosumaR2
    MOV R6,R2                ; Iguala R6 al valor R2
    B decoRN                 ; Salta para decodificar el registro R6
decoRN
    CMP R6,#0x30              ; Compara R6 con el valor 1 en 7 segmentos
    BNE conversion2          ; Si no son iguales salta a conversion2
    MOV R6,#0x01              ; Iguala R6 con 1
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion2
    CMP R6,#0x6D              ; Compara R6 con el valor 2 en 7 segmentos
    BNE conversion3          ; Si no son iguales salta a conversion3
    MOV R6,#0x02              ; Iguala R6 con 2
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion3
    CMP R6,#0x79              ; Compara R6 con el valor 3 en 7 segmentos
    BNE conversion4          ; Si no son iguales salta a conversion4
    MOV R6,#0x03              ; Iguala R6 con 3
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion4
    CMP R6,#0x33              ; Compara R6 con el valor 4 en 7 segmentos
    BNE conversion5          ; Si no son iguales salta a conversion5
    MOV R6,#0x04              ; Iguala R6 con 4
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion5
    CMP R6,#0x5B              ; Compara R6 con el valor 5 en 7 segmentos
    BNE conversion6          ; Si no son iguales salta a conversion6
    MOV R6,#0x05              ; Iguala R6 con 5
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion6
    CMP R6,#0x5F              ; Compara R6 con el valor 6 en 7 segmentos
    BNE conversion7          ; Si no son iguales salta a conversion7
    MOV R6,#0x06              ; Iguala R6 con 6
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion7
    CMP R6,#0x70              ; Compara R6 con el valor 7 en 7 segmentos
    BNE conversion8          ; Si no son iguales salta a conversion8
    MOV R6,#0x07              ; Iguala R6 con 7
    ADD R3,#0x01              ; Al contador del ciclo le suma una unidad
    B conversionRN           ; Salta a subrutina conversionRN
conversion8

```


	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 16 de 41	



```

    CMP R6,#0x7F                ; Compara R6 con el valor 8 en 7 segmentos
    BNE conversion9             ; Si no son iguales salta a conversion9
    MOV R6,#0x08                ; Iguala R6 con 8
    ADD R3,#0x01                ; Al contador del ciclo le suma una unidad
    B conversionRN              ; Salta a subrutina conversionRN
conversion9
    CMP R6,#0x7B                ; Compara R6 con el valor 8 en 7 segmentos
    BNE conversion0             ; Si no son iguales salta a conversion0
    MOV R6,#0x09                ; Iguala R6 con 9
    ADD R3,#0x01                ; Al contador del ciclo le suma una unidad
    B conversionRN              ; Salta a subrutina conversionRN
conversion0
    MOV R6,#0x00                ; Iguala R6 a 0, único valor faltante y al que se
igual si no hay registro
    ADD R3,#0x01                ; Al contador del ciclo le suma una unidad
conversionRN
    CMP R3,#0x01                ; Compara R3 con 1
    BEQ conversionR4            ; Si son iguales convertirá decenas
    CMP R3,#0x02                ; Compara R3 con 2
    BEQ conversionR8            ; Si son iguales convertirá unidades
    CMP R3,#0x03                ; Compara R3 con 3
    BEQ conversionR2            ; Si son iguales convertirá decimas
    CMP R3,#0x04                ; Compara R3 con 4
    BEQ conversionR1            ; Si son iguales convertirá centésimas
conversionR4
    MOV R4,R6                    ; Iguala el valor obtenido a las decenas
    B decosuma                  ; Salta a subrutina decosuma para convertir el
siguiente dígito
conversionR8
    MOV R8,R6                    ; Iguala el valor obtenido a las unidades
    B decosuma                  ; Salta a subrutina decosuma para convertir el
siguiente dígito
conversionR2
    MOV R2,R6                    ; Iguala el valor obtenido a las decimas
    B decosuma                  ; Salta a subrutina decosuma para convertir el
siguiente dígito
conversionR1
    MOV R1,R6                    ; Iguala el valor obtenido a las decenas
    B SUMA                      ; Salta a subrutina SUMA

;----- SUMA DE R4, R8, R2, R1 - R7 (ENTEROS) Y R11 (DECIMALES) -----

SUMA
    MOV R6,#0x0A                ; Iguala R6 con A
    MUL R4,R4,R6                ; Multiplica el valor de decenas por 10
    ADD R4,R4,R8                ; Se suman decenas con unidades y se almacenan en
R4
    ADD R7,R7,R4                ; Al valor acumulado de enteros le suma el nuevo
valor obtenido en R4
    MUL R2,R2,R6                ; Multiplica el valor de decimales por 10
    ADD R2,R2,R1                ; Al valor de decimas le suma las centésimas y se
almacenar en R2
    ADD R11,R11,R2              ; Al valor acumulado de decimales le suma el nuevo
valor obtenido en R2
    MOV R8,R11                  ; Iguala R8 al valor de decimales totales
    MOV R4,R8                    ; Iguala R4 a R8
    MOV R1,#0x64                ; Iguala R1 a 100 decimal

```


	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 17 de 41	



```

UDIV R4,R8,R1      ; Divide entre 100 los decimales para obtener
cuántos enteros hay en éste
MUL R2,R1,R4        ; Multiplica el valor de esos enteros obtenidos
por 100
SUB R11,R8,R2        ; Se resta al valor de decimales el equivalente a
los enteros de la línea anterior
ADD R7,R7,R4         ; Esos enteros que salen de los decimales se
acumulan con el valor total entero
MOV R8,#0x00         ; Se reinicia el valor R8, unidades
MOV R4,#0x00         ; Se reinicia el valor R4, decenas
MOV R2,#0x00         ; Se reinicia el valor R2, decimales
MOV R1,#0x00         ; Se reinicia el valor R1, centésimas
MOV R10,#0x00        ; Se reinicia el valor R10, acumulador de
decimales
B teclado           ;
etiquetaifA
NOP
NOP
NOP

;====VALOR CUATRO-B====

LDR R0,GPIOK_DATA_R ; El valor R0 apunta al registro de datos del puerto K
MOV R6,#0x40         ; Sensa el valor de la fila uno
STR R6,[R0]          ; Nos direccionamos a la fila uno
LDR R6,[R0]          ; Apunta el valor de la fila uno en el puerto K
CMP R6,#0x48         ; Comparamos R6 con el valor de la tecla 4
presionada
BNE etiquetaif4      ; Si no son iguales saltamos a la etiquetaif4
MOV R12,#0x33         ; Igualamos R12 al valor equivalente a 4 en 7 segmentos
B start              ; Saltamos a la etiqueta start
etiquetaif4
CMP R6,#0x44         ; Comparamos R6 con el valor de la tecla 5
presionada
BNE etiquetaif5      ; Si no son iguales saltamos a la etiquetaif5
MOV R12,#0x5B         ; Igualamos R12 al valor equivalente a 5 en 7 segmentos
B start              ; Saltamos a la etiqueta start
etiquetaif5
CMP R6,#0x42         ; Comparamos R6 con el valor de la tecla 6
presionada
BNE etiquetaif6      ; Si no son iguales saltamos a la etiquetaif6
MOV R12,#0x5F         ; Igualamos R12 al valor equivalente a 6 en 7 segmentos
B start              ; Saltamos a la etiqueta start
etiquetaif6
CMP R6,#0x41         ; Comparamos R6 con el valor de la tecla B
presionada
BNE etiquetaifB      ; Si no son iguales saltamos a la etiquetaifB
MOV R0,#0x00         ; Reiniciamos el valor de R0 con cero para
decodificar el resultado de R7 y R11
etiquetaB
CMP R0,#0x00         ; Comparamos R0 con 0
BEQ convunidades     ; Si son iguales saltamos a convunidades
CMP R0,#0x01         ; Comparamos R0 con 1
BEQ convdecimales    ; Si son iguales saltamos a convdecimales
convunidades
AND R3,R7,#0x0F      ; Realizamos una AND con 0F para guardar el valor menos
significativo de R7 en R3



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 18 de 41	

```

AND R9,R7,#0xF0      ; Realizamos una AND con F0 para guardar el valor más
significativo de R7 en R9
B asignacion          ; Saltamos a asignacion
convdecimales
AND R3,R11,#0x0F      ; Realizamos una AND con 0F para guardar el valor menos
significativo de R11 en R3
AND R9,R11,#0xF0      ; Realizamos una AND con F0 para guardar el valor más
significativo de R11 en R9
B asignacion          ; Saltamos a asignacion
asignacion
CMP R9,#0x00           ; Comparamos el valor más significativo con 0
BEQ saltoa00           ; Si son iguales saltamos a saltoa00
CMP R9,#0x10           ; Comparamos el valor más significativo con 16
BEQ saltoa16           ; Si son iguales saltamos a saltoa16
CMP R9,#0x20           ; Comparamos el valor más significativo con 32
BEQ saltoa32           ; Si son iguales saltamos a saltoa32
CMP R9,#0x30           ; Comparamos el valor más significativo con 48
BEQ saltoa48           ; Si son iguales saltamos a saltoa48
CMP R9,#0x40           ; Comparamos el valor más significativo con 64
BEQ saltoa64           ; Si son iguales saltamos a saltoa64
CMP R9,#0x50           ; Comparamos el valor más significativo con 80
BEQ saltoa80           ; Si son iguales saltamos a saltoa80
CMP R9,#0x60           ; Comparamos el valor más significativo con 96
BEQ saltoa96           ; Si son iguales saltamos a saltoa96
B error               ; Si el valor más significativo excede los 96 no
se representará, por eso error
saltoa00
CMP R0,#0x00           ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales
BEQ unidades00        ; Si es cero saltamos a unidades00 para convertir
unidades
CMP R0,#0x01           ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales
BEQ decimales00       ; Si es uno saltamos a decimales00 para convertir
decimales
unidades00
MOV R4,#0x00           ; Hacemos el valor de decenas 0
ADD R3,#0x00           ; Sumamos al valor menos significativo un 0
B decounidades        ; Saltamos a decounidades
decimales00
MOV R2,#0x00           ; Hacemos el valor de decimales 0
ADD R3,#0x00           ; Sumamos al valor menos significativo un 0
B decounidades        ; Saltamos a decounidades
saltoa16
CMP R0,#0x00           ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales
BEQ unidades16        ; Si es cero saltamos a unidades16 para convertir
unidades
CMP R0,#0x01           ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales
BEQ decimales16       ; Si es uno saltamos a decimales16 para convertir
decimales
unidades16
MOV R4,#0x01           ; Hacemos el valor de decenas 1
ADD R3,#0x06           ; Sumamos al valor menos significativo un 6
B decounidades        ; Saltamos a decounidades
decimales16



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves	Página 19 de 41	

```

MOV R2,#0x01          ; Hacemos el valor de decimales 1
ADD R3,#0x06          ; Sumamos al valor menos significativo un 6
B decounidades        ; Saltamos a decounidades
saltoa32
CMP R0,#0x00          ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales  ; Si es cero saltamos a unidades32 para convertir
BEQ unidades32
unidades
CMP R0,#0x01          ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales  ; Si es uno saltamos a decimales32 para convertir
BEQ decimales32
decimales
unidades32
MOV R4,#0x03          ; Hacemos el valor de decenas 3
ADD R3,#0x02          ; Sumamos al valor menos significativo un 2
B decounidades        ; Saltamos a decounidades
decimales32
MOV R2,#0x03          ; Hacemos el valor de decimales 3
ADD R3,#0x02          ; Sumamos al valor menos significativo un 2
B decounidades        ; Saltamos a decounidades
saltoa48
CMP R0,#0x00          ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales  ; Si es cero saltamos a unidades48 para convertir
BEQ unidades48
unidades
CMP R0,#0x01          ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales  ; Si es uno saltamos a decimales48 para convertir
BEQ decimales48
decimales
unidades48
MOV R4,#0x04          ; Hacemos el valor de decenas 4
ADD R3,#0x08          ; Sumamos al valor menos significativo un 8
B decounidades        ; Saltamos a decounidades
decimales48
MOV R2,#0x04          ; Hacemos el valor de decimales 4
ADD R3,#0x08          ; Sumamos al valor menos significativo un 2
B decounidades        ; Saltamos a decounidades
saltoa64
CMP R0,#0x00          ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales  ; Si es cero saltamos a unidades64 para convertir
BEQ unidades64
unidades
CMP R0,#0x01          ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales  ; Si es uno saltamos a decimales64 para convertir
BEQ decimales64
decimales
unidades64
MOV R4,#0x06          ; Hacemos el valor de decenas 6
ADD R3,#0x04          ; Sumamos al valor menos significativo un 4
B decounidades        ; Saltamos a decounidades
decimales64
MOV R2,#0x06          ; Hacemos el valor de decimales 6
ADD R3,#0x04          ; Sumamos al valor menos significativo un 4
B decounidades        ; Saltamos a decounidades
saltoa80
CMP R0,#0x00          ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 20 de 41	

```

    BEQ unidades80          ; Si es cero saltamos a unidades80 para convertir
unidades
    CMP R0,#0x01            ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales
    BEQ decimales80        ; Si es uno saltamos a decimales80 para convertir
decimales
unidades80
    MOV R4,#0x08            ; Hacemos el valor de decenas 8
    ADD R3,#0x00            ; Sumamos al valor menos significativo un 0
    B decounidades         ; Saltamos a decounidades
decimales80
    MOV R2,#0x08            ; Hacemos el valor de decimales 8
    ADD R3,#0x00            ; Sumamos al valor menos significativo un 0
    B decounidades         ; Saltamos a decounidades
saltoa96
    CMP R0,#0x00            ; Comparamos R0 con 0 para saber si convertimos
unidades o decimales
    BEQ unidades96         ; Si es cero saltamos a unidades96 para convertir
unidades
    CMP R0,#0x01            ; Comparamos R0 con 1 para saber si convertimos
unidades o decimales
    BEQ decimales96        ; Si es uno saltamos a decimales96 para convertir
decimales
unidades96
    MOV R4,#0x09            ; Hacemos el valor de decenas 9
    ADD R3,#0x06            ; Sumamos al valor menos significativo un 6
    CMP R3,#0x07            ; Comparamos el valor obtenido con un 7
    BEQ decounidades       ; Si son iguales la cifra no supera los 99 por lo que
saltamos a decounidades
    CMP R3,#0x08            ; Comparamos el valor obtenido con un 8
    BEQ decounidades       ; Si son iguales la cifra no supera los 99 por lo que
saltamos a decounidades
    CMP R3,#0x09            ; Comparamos el valor obtenido con un 9
    BEQ decounidades       ; Si son iguales la cifra no supera los 99 por lo que
saltamos a decounidades
    B error                 ; Si el valor es mayor a 99 no se muestra por lo
que saldrá un error
decimales96
    MOV R2,#0x09            ; Hacemos el valor de decimales 9
    ADD R3,#0x06            ; Sumamos al valor menos significativo un 6
    CMP R3,#0x07            ; Comparamos el valor obtenido con un 7
    BEQ decounidades       ; Si son iguales la cifra no supera los 99 por lo que
saltamos a decounidades
    CMP R3,#0x08            ; Comparamos el valor obtenido con un 8
    BEQ decounidades       ; Si son iguales la cifra no supera los 99 por lo que
saltamos a decounidades
    CMP R3,#0x09            ; Comparamos el valor obtenido con un 9
    BEQ decounidades       ; Si son iguales la cifra no supera los 99 por lo que
saltamos a decounidades
    B error
decounidades
    CMP R3,#0x0A            ; Comparas el nuevo valor de unidades con A
    BEQ saltoa10            ; Si son iguales salta a saltoa10
    CMP R3,#0x0B            ; Comparas el nuevo valor de unidades con B
    BEQ saltoa11            ; Si son iguales salta a saltoa11
    CMP R3,#0x0C            ; Comparas el nuevo valor de unidades con C
    BEQ saltoa12            ; Si son iguales salta a saltoa12



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 21 de 41	

```

CMP R3,#0x0D      ; Comparas el nuevo valor de unidades con D
BEQ saltoa13      ; Si son iguales salta a saltoa13
CMP R3,#0x0E      ; Comparas el nuevo valor de unidades con E
BEQ saltoa14      ; Si son iguales salta a saltoa14
CMP R3,#0x0F      ; Comparas el nuevo valor de unidades con F
BEQ saltoa15      ; Si son iguales salta a saltoa15
CMP R3,#0x10      ; Comparas el nuevo valor de unidades con 10
BEQ saltoa16d     ; Si son iguales salta a saltoa16d
CMP R3,#0x11      ; Comparas el nuevo valor de unidades con 11
BEQ saltoa17      ; Si son iguales salta a saltoa17
CMP R3,#0x12      ; Comparas el nuevo valor de unidades con 12
BEQ saltoa18      ; Si son iguales salta a saltoa18
CMP R3,#0x13      ; Comparas el nuevo valor de unidades con 13
BEQ saltoa19      ; Si son iguales salta a saltoa19
CMP R3,#0x14      ; Comparas el nuevo valor de unidades con 14
BEQ saltoa20      ; Si son iguales salta a saltoa20
CMP R3,#0x15      ; Comparas el nuevo valor de unidades con 15
BEQ saltoa21      ; Si son iguales salta a saltoa21
CMP R3,#0x16      ; Comparas el nuevo valor de unidades con 16
BEQ saltoa22      ; Si son iguales salta a saltoa22
CMP R3,#0x17      ; Comparas el nuevo valor de unidades con 17
BEQ saltoa23      ; Si son iguales salta a saltoa23
B decodecimas     ; Saltamos a decodecimas
saltoa10
CMP R0,#0x00      ; Comparamos R0 con 0
BEQ unidades10    ; Si son iguales convierte unidades saltando a
unidades10
CMP R0,#0x01      ; Comparamos R0 con 1
BEQ ddecimales10  ; Si son iguales convierte decimales saltando a
ddecimales10
unidades10
ADD R4,#0x01      ; Se le agrega un 1 a las decenas
MOV R3,#0x00      ; Se iguala el valor menos significativo a 0
B decodecimas     ; Saltamos a decodecimas
ddecimales10
ADD R2,#0x01      ; Se le agrega un 1 a las decimas
MOV R3,#0x00      ; Se iguala el valor menos significativo a 0
B decodecimas     ; Saltamos a decodecimas
saltoa11
CMP R0,#0x00      ; Comparamos R0 con 0
BEQ unidades11    ; Si son iguales convierte unidades saltando a
unidades11
CMP R0,#0x01      ; Comparamos R0 con 1
BEQ ddecimales11  ; Si son iguales convierte decimales saltando a
ddecimales11
unidades11
ADD R4,#0x01      ; Se le agrega un 1 a las decenas
MOV R3,#0x01      ; Se iguala el valor menos significativo a 1
B decodecimas     ; Saltamos a decodecimas
ddecimales11
ADD R2,#0x01      ; Se le agrega un 1 a las decimas
MOV R3,#0x01      ; Se iguala el valor menos significativo a 1
B decodecimas     ; Saltamos a decodecimas
saltoa12
CMP R0,#0x00      ; Comparamos R0 con 0
BEQ unidades12    ; Si son iguales convierte unidades saltando a
unidades12



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 22 de 41	

```

    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales12           ; Si son iguales convierte decimales saltando a
ddecimales12
dunidades12
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x02                ; Se iguala el valor menos significativo a 2
    B decodecimas              ; Saltamos a decodecimas
ddecimales12
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x02                ; Se iguala el valor menos significativo a 2
    B decodecimas              ; Saltamos a decodecimas
saltoa13
    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ dunidades13           ; Si son iguales convierte unidades saltando a
dunidades13
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales13         ; Si son iguales convierte decimales saltando a
ddecimales13
dunidades13
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x03                ; Se iguala el valor menos significativo a 3
    B decodecimas              ; Saltamos a decodecimas
ddecimales13
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x03                ; Se iguala el valor menos significativo a 3
    B decodecimas              ; Saltamos a decodecimas
saltoa14
    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ dunidades14           ; Si son iguales convierte unidades saltando a
dunidades14
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales14         ; Si son iguales convierte decimales saltando a
ddecimales14
dunidades14
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x04                ; Se iguala el valor menos significativo a 4
    B decodecimas              ; Saltamos a decodecimas
ddecimales14
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x04                ; Se iguala el valor menos significativo a 4
    B decodecimas              ; Saltamos a decodecimas
saltoa15
    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ dunidades15           ; Si son iguales convierte unidades saltando a
dunidades15
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales15         ; Si son iguales convierte decimales saltando a
ddecimales15
dunidades15
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x05                ; Se iguala el valor menos significativo a 5
    B decodecimas              ; Saltamos a decodecimas
ddecimales15
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x05                ; Se iguala el valor menos significativo a 5
    B decodecimas              ; Saltamos a decodecimas
saltoa16d



```


	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 23 de 41	

```

    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ unidades16              ; Si son iguales convierte unidades saltando a
unidades16
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales16           ; Si son iguales convierte decimales saltando a
ddecimales16
unidades16
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x06                ; Se iguala el valor menos significativo a 6
    B decodecimas              ; Saltamos a decodecimas
ddecimales16
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x06                ; Se iguala el valor menos significativo a 6
    B decodecimas              ; Saltamos a decodecimas
saltoa17
    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ unidades17             ; Si son iguales convierte unidades saltando a
unidades17
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales17          ; Si son iguales convierte decimales saltando a
ddecimales17
unidades17
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x07                ; Se iguala el valor menos significativo a 7
    B decodecimas              ; Saltamos a decodecimas
ddecimales17
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x07                ; Se iguala el valor menos significativo a 6
    B decodecimas              ; Saltamos a decodecimas
saltoa18
    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ unidades18             ; Si son iguales convierte unidades saltando a
unidades18
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales18          ; Si son iguales convierte decimales saltando a
ddecimales18
unidades18
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x08                ; Se iguala el valor menos significativo a 8
    B decodecimas              ; Saltamos a decodecimas
ddecimales18
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas
    MOV R3,#0x08                ; Se iguala el valor menos significativo a 8
    B decodecimas              ; Saltamos a decodecimas
saltoa19
    CMP R0,#0x00                ; Comparamos R0 con 0
    BEQ unidades19             ; Si son iguales convierte unidades saltando a
unidades19
    CMP R0,#0x01                ; Comparamos R0 con 1
    BEQ ddecimales19          ; Si son iguales convierte decimales saltando a
ddecimales19
unidades19
    ADD R4,#0x01                ; Se le agrega un 1 a las decenas
    MOV R3,#0x09                ; Se iguala el valor menos significativo a 9
    B decodecimas              ; Saltamos a decodecimas
ddecimales19
    ADD R2,#0x01                ; Se le agrega un 1 a las decimas



```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 24 de 41	

```

MOV R3,#0x09           ; Se iguala el valor menos significativo a 9
B decodecimas          ; Saltamos a decodecimas
saltoa20
CMP R0,#0x00           ; Comparamos R0 con 0
BEQ unidades20         ; Si son iguales convierte unidades saltando a
unidades20
CMP R0,#0x01           ; Comparamos R0 con 1
BEQ ddecimales20       ; Si son iguales convierte decimales saltando a
ddecimales20
unidades20
ADD R4,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x00           ; Se iguala el valor menos significativo a 0
B decodecimas          ; Saltamos a decodecimas
ddecimales20
ADD R2,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x00           ; Se iguala el valor menos significativo a 0
B decodecimas          ; Saltamos a decodecimas
saltoa21
CMP R0,#0x00           ; Comparamos R0 con 0
BEQ unidades21         ; Si son iguales convierte unidades saltando a
unidades21
CMP R0,#0x01           ; Comparamos R0 con 1
BEQ ddecimales21       ; Si son iguales convierte decimales saltando a
ddecimales21
unidades21
ADD R4,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x01           ; Se iguala el valor menos significativo a 1
B decodecimas          ; Saltamos a decodecimas
ddecimales21
ADD R2,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x01           ; Se iguala el valor menos significativo a 1
B decodecimas          ; Saltamos a decodecimas
saltoa22
CMP R0,#0x00           ; Comparamos R0 con 0
BEQ unidades22         ; Si son iguales convierte unidades saltando a
unidades22
CMP R0,#0x01           ; Comparamos R0 con 1
BEQ ddecimales22       ; Si son iguales convierte decimales saltando a
ddecimales22
unidades22
ADD R4,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x02           ; Se iguala el valor menos significativo a 2
B decodecimas          ; Saltamos a decodecimas
ddecimales22
ADD R2,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x02           ; Se iguala el valor menos significativo a 3
B decodecimas          ; Saltamos a decodecimas
saltoa23
CMP R0,#0x00           ; Comparamos R0 con 0
BEQ unidades23         ; Si son iguales convierte unidades saltando a
unidades23
CMP R0,#0x01           ; Comparamos R0 con 1
BEQ ddecimales23       ; Si son iguales convierte decimales saltando a
ddecimales23
unidades23
ADD R4,#0x02           ; Se le agrega un 2 a las decenas
MOV R3,#0x03           ; Se iguala el valor menos significativo a 3

```


	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 25 de 41	



```

    B decodecimas                ; Saltamos a decodecimas
ddecimales23
    ADD R2,#0x02                ; Se le agrega un 2 a las decimas
    MOV R3,#0x03                ; Se iguala el valor menos significativo a 3
    B decodecimas                ; Saltamos a decodecimas
decodecimas
    ADD R0,#0x01                ; Al valor de R0 le sumamos 1
    CMP R0,#0x01                ; Comparamos con 1 para saber si sumamos unidades o
decimales
    BEQ regresou                ; Si comparamos con unidades saltamos a regresou
    MOV R1,R3                    ; Si comparamos con decimales igualamos
centésimas a R3
    B total                      ; Saltamos a total
regresou
    MOV R8,R3                    ; Si comparamos con unidades igualamos unidaes a
R3
    B etiquetaB                  ; Repetimos todo el ciclo de decodificación para
los decimales

;----- ERROR -----

error
    LDR R3,GPIO_PORTE_DATA_R    ; R3 apunta al registro de datos E
    LDR R9,GPIO_PORTM_DATA_R    ; R9 apunta al registro de datos M
    MOV R0,#0x00                ; R0 lo hacemos 0 para comenzar el ciclo de error
cicloerror
    CMP R0,#0x100000             ; Comparamos R0 con 100,000 para 100,000 iteraciones
    BNE cicloerrorsun           ; Si no son iguales saltamos a cicloerrorsun
    B teclado                    ; Si son iguales volvemos a teclado
cicloerrorsun
    MOV R6,#0xF7                 ; Asigna a R6 el valor del primer dígito del
display 7 segmentos
    STR R6,[R3]                  ; Habilita el primer digito del display 7
segmentos
    MOV R4,#0x5B                 ; Asigna R4 a la letra S
    STR R4,[R9]                  ; Enciende la letra S en el primer digito
    MOV R6,#0x00                 ; Asigna R6 con 0 para apagar el primer dígito
    STR R6,[R9]                  ; Apaga el primer dígito
    MOV R6,#0xFB                 ; Asigna a R6 el valor del segundo dígito del
display 7 segmentos
    STR R6,[R3]                  ; Habilita el segundo digito del display 7
segmentos
    MOV R8,#0x7E                 ; Asigna R4 a la letra O
    STR R8,[R9]                  ; Enciende la letra O en el segundo digito
    MOV R6,#0x00                 ; Asigna R6 con 0 para apagar el segundo dígito
    STR R6,[R9]                  ; Apaga el segundo dígito
    MOV R6,#0xFD                 ; Asigna a R6 el valor del tercer dígito del
display 7 segmentos
    STR R6,[R3]                  ; Habilita el tercer digito del display 7
segmentos
    MOV R2,#0x77                 ; Asigna R4 a la letra R
    STR R2,[R9]                  ; Enciende la letra R en el tercer digito
    MOV R6,#0x00                 ; Asigna R6 con 0 para apagar el tercer dígito
    STR R6,[R9]                  ; Apaga el tercer dígito
    MOV R6,#0xFE                 ; Asigna a R6 el valor del cuarto dígito del
display 7 segmentos

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 26 de 41	



```

STR R6,[R3] ; Habilita el cuarto dígito del display 7
segmentos
MOV R1,#0x30 ; Asigna R4 a la letra I
STR R1,[R9] ; Enciende la letra I en el cuarto dígito
MOV R6,#0x00 ; Asigna R6 con 0 para apagar el cuarto dígito
STR R6,[R9] ; Apaga el cuarto dígito
ADD R0,#0x01 ; A R0 le suma 1 para reducir las iteraciones
MOV R7,#0x00 ; Reinicia el valor de unidades
MOV R11,#0x00 ; Reinicia el valor de decimales
MOV R1,#0x00 ; Reinicia el registro de decenas
MOV R2,#0x00 ; Reinicia el registro de unidades
MOV R8,#0x00 ; Reinicia el registro de decimas
MOV R4,#0x00 ; Reinicia el registro de centésimas
B cicloerror ; Salta al ciclo de error para cumplir las
iteraciones



;----- DECODIFICACIÓN HEXADECIMAL-7 SEGMENTOS -----

total
MOV R12,#0x00 ; Iguala R12 a 0 para la decodificación de
hexadecimal a 7 segmentos
decoHa7S
CMP R12,#0x00 ; Compara R12 con 0 para decodificar R4
BEQ decoHa7SR4 ; Si son iguales salta a decoHa7SR4
CMP R12,#0x01 ; Compara R12 con 1 para decodificar R8
BEQ decoHa7SR8 ; Si son iguales salta a decoHa7SR8
CMP R12,#0x02 ; Compara R12 con 2 para decodificar R2
BEQ decoHa7SR2 ; Si son iguales salta a decoHa7SR2
CMP R12,#0x03 ; Compara R12 con 3 para decodificar R1
BEQ decoHa7SR1 ; Si son iguales salta a decoHa7SR1
decoHa7SR4
MOV R6,R4 ; Iguala R6 a R4
B decoHa7 ; Salta a decoHa7
decoHa7SR8
MOV R6,R8 ; Iguala R6 a R8
B decoHa7 ; Salta a decoHa7
decoHa7SR2
MOV R6,R2 ; Iguala R6 a R2
B decoHa7 ; Salta a decoHa7
decoHa7SR1
MOV R6,R1 ; Iguala R6 a R1
B decoHa7 ; Salta a decoHa7
decoHa7
CMP R6,#0x01 ; Compara R6 con 1
BEQ Ha71 ; Salta a Ha71
CMP R6,#0x02 ; Compara R6 con 2
BEQ Ha72 ; Salta a Ha72
CMP R6,#0x03 ; Compara R6 con 3
BEQ Ha73 ; Salta a Ha73
CMP R6,#0x04 ; Compara R6 con 4
BEQ Ha74 ; Salta a Ha74
CMP R6,#0x05 ; Compara R6 con 5
BEQ Ha75 ; Salta a Ha75
CMP R6,#0x06 ; Compara R6 con 6
BEQ Ha76 ; Salta a Ha76
CMP R6,#0x07 ; Compara R6 con 7
BEQ Ha77 ; Salta a Ha77

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 27 de 41	

	CMP R6,#0x08	; Compara R6 con 8
	BEQ Ha78	; Salta a Ha78
	CMP R6,#0x09	; Compara R6 con 9
	BEQ Ha79	; Salta a Ha79
	CMP R6,#0x00	; Compara R6 con 0
	BEQ Ha70	; Salta a Ha70
Ha71	MOV R6,#0x30	; Iguala R6 al equivalente a 1 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha72	MOV R6,#0x6D	; Iguala R6 al equivalente a 2 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha73	MOV R6,#0x79	; Iguala R6 al equivalente a 3 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha74	MOV R6,#0x33	; Iguala R6 al equivalente a 4 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha75	MOV R6,#0x5B	; Iguala R6 al equivalente a 5 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha76	MOV R6,#0x5F	; Iguala R6 al equivalente a 6 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha77	MOV R6,#0x70	; Iguala R6 al equivalente a 7 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha78	MOV R6,#0x7F	; Iguala R6 al equivalente a 8 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha79	MOV R6,#0x7B	; Iguala R6 al equivalente a 9 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
Ha70	MOV R6,#0x7E	; Iguala R6 al equivalente a 0 en 7 segmentos
	ADD R12,#0x01	; A R12 le suma 1 para las iteraciones
	B decoHa7SF	; Salta a decoHa7SF
decoHa7SF	CMP R12,#0x01	; Compara R12 con 1 para igualar resultado a R4
	BEQ decoHa7SMR4	; Si son iguales salta a decoHa7SMR4 para decenas
	CMP R12,#0x02	; Compara R12 con 1 para igualar resultado a R8
	BEQ decoHa7SMR8	; Si son iguales salta a decoHa7SMR8 para unidades
	CMP R12,#0x03	; Compara R12 con 1 para igualar resultado a R2
	BEQ decoHa7SMR2	; Si son iguales salta a decoHa7SMR2 para decimas
	CMP R12,#0x04	; Compara R12 con 1 para igualar resultado a R1
	BEQ decoHa7SMR1	; Si son iguales salta a decoHa7SMR1 para centésimas
decoHa7SMR4		

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 28 de 41	



```

MOV R4,R6 ; Iguala R4 a R6
B decoHa7S ; Salta a decoHa7S
decoHa7SMR8
MOV R8,R6 ; Iguala R8 a R6
B decoHa7S ; Salta a decoHa7S
decoHa7SMR2
MOV R2,R6 ; Iguala R2 a R6
B decoHa7S ; Salta a decoHa7S
decoHa7SMR1
MOV R1,R6 ; Iguala R1 a R6
B mostrartotal ; Salta a mostrartotal

;----- MUESTRA DE R4,R8,R2,R1 (LOS VALORES DECODIFICADOS EN 7 SEGMENTOS) --

mostrartotal
MOV R12,#0x00 ; Iguala R12 a 0 para el ciclo de muestra del resultado
comptot
CMP R12,#0x100000 ; Compara R12 con 100,000 las iteraciones donde se
mostrará el total
BNE sumatot ; Si no son iguales salta a sumatot para mostrar
el resultado
MOV R1,#0x00 ; Reinicia el valor de centésimas
MOV R2,#0x00 ; Reinicia el valor de decimas
MOV R4,#0x00 ; Reinicia el valor de unidades
MOV R8,#0x00 ; Reinicia el valor de decenas
MOV R14,#0x01 ; Iguala R14 a 1 para que ya pueda abrirse la
caja
B teclado ; Salta a teclado
sumatot
LDR R3,GPIO_PORTE_DATA_R ; R3 apunta al registro de datos del puerto E
LDR R0,GPIO_PORTK_DATA_R ; R0 apunta al registro de datos del puerto K
LDR R9,GPIO_PORTM_DATA_R ; R9 apunta al registro de datos del puerto M
MOV R6,#0xF7 ; R6 se iguala a la habilitación del primer
dígito del display
STR R6,[R3] ; Se habilita el primer dígito del display 7
segmentos
STR R4,[R9] ; Se muestra el valor de decenas en el primer
dígito
MOV R6,#0x00 ; Se iguala R6 a 0 para apagar los leds
STR R6,[R9] ; Apaga los LEDS
MOV R6,#0xFB ; R6 se iguala a la habilitación del segundo
dígito del display
STR R6,[R3] ; Se habilita el segundo dígito del display 7
segmentos
STR R8,[R9] ; Se muestra el valor de unidades en el segundo
dígito
MOV R6,#0x00 ; Se iguala R6 a 0 para apagar los leds
STR R6,[R9] ; Apaga los LEDS
MOV R6,#0x80 ; Se muestra unicamente el punto del segundo
dígito
STR R6,[R9] ; Se muestra el punto en el segundo dígito
MOV R6,#0x00 ; Se iguala R6 a 0 para apagar los leds
STR R6,[R9] ; Apaga los LEDS
MOV R6,#0xFD ; R6 se iguala a la habilitación del tercer
dígito del display
STR R6,[R3] ; Se habilita el tercer dígito del display 7
segmentos

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 29 de 41	

```



    STR R2,[R9]                ; Se muestra el valor de decimas en el tercer
dígito
    MOV R6,#0x00                ; Se iguala R6 a 0 para apagar los leds
    STR R6,[R9]                ; Apaga los LEDS
    MOV R6,#0xFE                ; R6 se iguala a la habilitación del cuarto
dígito del display
    STR R6,[R3]                ; Se habilita el cuarto dígito del display 7
segmentos
    STR R1,[R9]                ; Se muestra el valor de centésimas en el cuarto
dígito
    MOV R6,#0x00                ; Se iguala R6 a 0 para apagar los leds
    STR R6,[R9]                ; Apaga los LEDS
    ADD R12,#0x01               ; Suma R12 con 1 para que se continuen las
iteraciones
    B comptot                   ; Salta a la subrutina del ciclo
etiquetaifB
    NOP                         ; No operacion
    NOP                         ; No operacion
    NOP                         ; No operacion

;----- VALOR SIETE-C -----
--

    LDR R0,GPIOPORTK_DATA_R    ; R0 apunta al registro de datos del puerto K
    MOV R6,#0x20                ; Sensamos la fila 3 del teclado
    STR R6,[R0]                ; Nos direccionamos a la fila dos
    LDR R6,[R0]                ; R6 habilita la fila 3 del teclado
    CMP R6,#0x28                ; Compara R6 con el valor 7 del teclado matricial
    BNE etiquetaif7            ; Si no son iguales salta a la etiquetaif7
    MOV R12,#0x70               ; Iguala R12 al valor equivalente a 7 en el display 7
segmentos
    B start                     ; Salta a start
etiquetaif7
    CMP R6,#0x24                ; Compara R6 con el valor 8 del teclado matricial
    BNE etiquetaif8            ; Si no son iguales salta a la etiquetaif8
    MOV R12,#0x7F               ; Iguala R12 al valor equivalente a 8 en el display 7
segmentos
    B start                     ; Salta a start
etiquetaif8
    CMP R6,#0x22                ; Compara R6 con el valor 9 del teclado matricial
    BNE etiquetaif9            ; Si no son iguales salta a la etiquetaif9
    MOV R12,#0x7B               ; Iguala R12 al valor equivalente a 9 en el display 7
segmentos
    B start                     ; Salta a start
etiquetaif9
    CMP R6,#0x21                ; Compara R6 con el valor C del teclado matricial
    BNE etiquetaifC            ; Si no son iguales salta a la etiquetaifC
    MOV R1,#0x00                ; Iguala R1 a 0 para iniciar el ciclo de retardo
del IVA
cicloIVA
    CMP R1,#0x90000              ; Compara R1 con 90,000 para 90,000 iteraciones
    BEQ IVA                    ; Si son iguales salta a IVA
    ADD R1,#0x01                ; Si no son iguales suma 1 a R1 para la siguiente
iteración
    B cicloIVA                  ; Salta al ciclo

;----- OBTENCIÓN DE IVA -----

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves Grupo 3	Página 30 de 41	



```

IVA
MOV R1,R7                ; R1=R7 conversion enteros
MOV R4,#0x10              ; R4=16
MUL R8,R4,R1              ; R8=R4*R1
MOV R3,#0x64              ; R3=100
MOV R2,R8                  ; R2=R8
UDIV R2,R2,R3              ; R2=R2/100
SUB R7,R7,R2              ; R7=R7-R2 Obtenemos valor entero
MUL R2,R2,R3              ; R2=R2*100
SUB R2,R8,R2              ; R2=R8-R2
MOV R1,R11                ; R1=R11 conversión decimales
MOV R3,#0x10              ; R3=16
MUL R1,R1,R3              ; R1=R1*R3
MOV R3,#0x64              ; R3=100
UDIV R1,R1,R3              ; R1=R1/100
ADD R1,R1,R2              ; R1=R1+R2
CMP R1,R11                ; Compara si el valor decimal original con el que se
restará superará la unidad ; Si es superior se restará 1 al entero y a 100 se le
BHS resta1                ; Si es inferior solo se resta el valor decimal
restará el decimal obtenido
BLO resta2
original con el obtenido descuento
resta1
SUB R2,R1,R11              ; R2=R1-R11
MOV R11,#0x01              ; R11=1
SUB R7,R7,R11              ; R7=R7-1 realiza la resta de unidad
MOV R11,#0x64              ; R11=100
SUB R11,R11,R2             ; R11=R11-R2 se obtienen los nuevos decimales
MOV R14,#0x01              ; R14=1
MOV R8,#0x00              ; Reiniciamos valor de unidades
MOV R4,#0x00              ; Reiniciamos valor de decenas
MOV R2,#0x00              ; Reiniciamos valor de decimas
MOV R1,#0x00              ; Reiniciamos valor de centésimas
B teclado                  ; Salto a teclado
resta2
SUB R11,R11,R1             ; R11=R11-R1 valor de decimales
MOV R8,#0x00              ; Reiniciamos el valor de unidades
MOV R4,#0x00              ; Reiniciamos valor de decenas
MOV R2,#0x00              ; Reiniciamos valor de decimas
MOV R1,#0x00              ; Reiniciamos valor de centésimas
B teclado                  ; Salto a teclado
etiquetaifC
NOP                        ; No operación
NOP                        ; No operación
NOP                        ; No operación

;----- VALOR *-D -----

LDR R0,GPIO_PORTK_DATA_R  ; R0 apunta al registro de datos del puerto K
MOV R6,#0x10              ; Igualamos R6 a la habilitación de la fila tres
STR R6,[R0]               ; Nos direccionamos a la fila tres
LDR R6,[R0]               ; Habilitamos la fila tres
CMP R6,#0x18              ; Comparamos R6 con el valor * presionado
BNE etiquetaifCAJA        ; Si no es igual saltamos a etiquetaifCAJA
LDR R3,GPIO_PORTH_DATA_R  ; R3 apunta al registro de datos H

```




	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 31 de 41	

;----- LEDS CAJA REGISTRADORA -----;

```

MOV R6,#0x00          ; Iguala R6 a 0 para el ciclo de retardo de la caja
registradora
cicloLED
    CMP R6,#0x90000    ; Compara R6 con 90,000 para 90,000 iteraciones
    BNE sumaLED        ; Si no son iguales salta a sumaLED
    B encendido        ; Salta a encendido
sumaLED
    ADD R6,#0x01        ; Suma 1 a R6 para la siguiente iteración
    B cicloLED        ; Regresa a cicloLED
encendido
    CMP R14,#0x00      ; Si R14 es 0 no se ha presionado ni IVA ni total por
lo que no pasa nada
    BEQ cajacerrada    ; Salta a cajacerrada
    CMP R14,#0x01
    BNE etiquetaroja
    MOV R6,#0x00
    STR R6,[R3]        ; Escribe en el registro de datos del Puerto H
    MOV R6,#LEDS1      ; R6 se iguala al valor de encendido del led verde
    STR R6,[R3]        ; Enciende LED verde
    MOV R14,#0x02      ; R14 se iguala a dos para que a la siguiente vez que
se presione se prenda ledrojo
    B teclado          ; Salto a teclado
cajacerrada
    B teclado
etiquetaroja
    MOV R4,#0x00        ; R4 se iguala a cero para el retardo del led
rojo
    MOV R6,#0x00        ; R6 se iguala a cero para apagar el led verde
    STR R6,[R3]        ; Escribe en el registro de datos del Puerto H
cicloLEDrojo
    CMP R4,#0x900000    ; Compara R4 con 90,000 para el ciclo de retardo del
reinicio del programa
    BEQ apagamosled    ; Si son iguales procedemos al reinicio
    MOV R6,#LEDS4      ; Iguala R6 al valor de habilitación del led rojo
    STR R6,[R3]        ; Encendemos led rojo
    ADD R4,#0x01        ; Sumamos 1 a R4 para la siguiente iteración
    B cicloLEDrojo     ; Volvemos a cicloLEDrojo
apagamosled
    MOV R6,#0x00        ; R6 se reinicia con 0
    STR R6,[R3]        ; Apagamos el led rojo
    MOV R14,#0x00      ; R14 se reinicia
    B comienzo        ; Reiniciamos programa
etiquetaifCAJA
    CMP R6,#0x14        ; Comparamos R6 con el valor correspondiente a 0
del teclado
    BNE etiquetaif0    ; Si no son iguales saltamos a etiquetaif0
    MOV R12,#0x7E      ; Igualamos R12 al valor correspondiente a 0 en 7
segmentos
    B start            ; Saltamos a start
etiquetaif0
    CMP R6,#0x12        ; Comparamos R6 con el valor correspondiente a #
del teclado
    BNE etiquetaifGATO ; Si no son iguales saltamos a etiquetaifGATO
    MOV R10,#0x01      ; R10 lo igualamos a 1 para que ahora los valores
ingresados sean como decimales

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves Grupo 3	Página 32 de 41	

```



    B start                                ; Saltamos a start
etiquetaifGATO
    CMP R6,#0x11                          ; Comparamos R6 con el valor correspondiente a la
D del teclado
    BNE etiquetaifD                       ; Si no son iguales saltamos a etiquetaifD

;----- DESCUENTO -----

    MOV R6,#0x00                          ; Igualamos R6 a cero para iniciar el retardo del
descuento
cicloDESCUENTO
    CMP R6,#0x90000                       ; Comparamos R6 con el valor 90,000 para 90,000
iteraciones
    BEQ DESCUENTO                         ; Si son iguales saltamos a DESCUENTO
    ADD R6,#0x01                          ; Si no son iguales sumamos 1 a R6 para la
siguiente iteración
    B cicloDESCUENTO                     ; Saltamos a cicloDESCUENTO
DESCUENTO
    MOV R6,#0x00                          ; Iniciamos la operación descuento con R6=0
descuentazo
    CMP R6,#0x00                          ; Se compara R6 con 0 para saber si se
descontaran decenas
    BEQ descuentodec                     ; Saltamos a descuentodec para decenas
    CMP R6,#0x01                          ; Se compara R6 con 0 para saber si se
descontaran unidades
    BEQ descuentoun                      ; Si son iguales salta a descuento dec para unidades
    B descuentohexa                      ; Se salta a descuentohexa ya convertidas decenas y
unidades
descuentodec
    MOV R3,R4                            ; Se iguala R3 con decenas
    B convdescuento                       ; Salto a conv descuento
descuentoun
    MOV R3,R8                            ; Se iguala R3 con unidades
    B convdescuento                       ; Salto a conv descuento
convdescuento
    CMP R3,#0x30                          ; Comparamos R3 con 1 en 7 segmentos
    BEQ descuento1                        ; Si son iguales saltamos a descuento1
    CMP R3,#0x6D                          ; Comparamos R3 con 2 en 7 segmentos
    BEQ descuento2                        ; Si son iguales saltamos a descuento2
    CMP R3,#0x79                          ; Comparamos R3 con 3 en 7 segmentos
    BEQ descuento3                        ; Si son iguales saltamos a descuento3
    CMP R3,#0x33                          ; Comparamos R3 con 4 en 7 segmentos
    BEQ descuento4                        ; Si son iguales saltamos a descuento4
    CMP R3,#0x5B                          ; Comparamos R3 con 5 en 7 segmentos
    BEQ descuento5                        ; Si son iguales saltamos a descuento5
    CMP R3,#0x5F                          ; Comparamos R3 con 6 en 7 segmentos
    BEQ descuento6                        ; Si son iguales saltamos a descuento6
    CMP R3,#0x70                          ; Comparamos R3 con 7 en 7 segmentos
    BEQ descuento7                        ; Si son iguales saltamos a descuento7
    CMP R3,#0x7F                          ; Comparamos R3 con 8 en 7 segmentos
    BEQ descuento8                        ; Si son iguales saltamos a descuento8
    CMP R3,#0x7B                          ; Comparamos R3 con 9 en 7 segmentos
    BEQ descuento9                        ; Si son iguales saltamos a descuento9
    CMP R3,#0x7E                          ; Comparamos R3 con 0 en 7 segmentos
    BEQ descuento0                        ; Si son iguales saltamos a descuento0

descuento1



```


	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 33 de 41	

```

MOV R3,#0x01                ; Igualamos R3 a 1
B descuentounidades         ; Saltamos a descuentounidades
descuento2
MOV R3,#0x02                ; Igualamos R3 a 2
B descuentounidades         ; Saltamos a descuentounidades
descuento3
MOV R3,#0x03                ; Igualamos R3 a 3
B descuentounidades         ; Saltamos a descuentounidades
descuento4
MOV R3,#0x04                ; Igualamos R3 a 4
B descuentounidades         ; Saltamos a descuentounidades
descuento5
MOV R3,#0x05                ; Igualamos R3 a 5
B descuentounidades         ; Saltamos a descuentounidades
descuento6
MOV R3,#0x06                ; Igualamos R3 a 6
B descuentounidades         ; Saltamos a descuentounidades
descuento7
MOV R3,#0x07                ; Igualamos R3 a 7
B descuentounidades         ; Saltamos a descuentounidades
descuento8
MOV R3,#0x08                ; Igualamos R3 a 8
B descuentounidades         ; Saltamos a descuentounidades
descuento9
MOV R3,#0x09                ; Igualamos R3 a 9
B descuentounidades         ; Saltamos a descuentounidades
descuento0
MOV R3,#0x00                ; Igualamos R3 a 0
B descuentounidades         ; Saltamos a descuentounidades
descuentounidades
ADD R6,#0x01                ; Sumamos a R6 1
CMP R6,#0x01                ; Comparamos R6 con 1 para igualar decenas
BEQ igualardecenas         ; Saltamos a igualardecenas
CMP R6,#0x02                ; Comparamos R6 con 1 para igualar unidades
BEQ igualarunidades        ; Saltamos a igualarunidades
igualardecenas
MOV R4,R3                   ; Igualamos decenas a R3
B descuentazo              ; Saltamos a descuentazo
igualarunidades
MOV R8,R3                   ; Igualamos unidades a R3
B descuentazo              ; Saltamos a descuentazo
descuento0hexa
MOV R3,#0x0A                ; Igualamos R3 con 10
MUL R1,R4,R3                ; R1=R4*R3
ADD R1,R1,R8                ; R1=R1+R8
MOV R3,#0x10                ; R3=16
MOV R4,#0x00                ; Reiniciamos valor de decenas
MOV R8,#0x00                ; Reiniciamos valor de unidades
ciclorestad
CMP R1,R3                   ; Comparamos R1 con R3
BHS sumadecena              ; Si es mayor o igual va a sumadecena
BLO sumaunidad             ; Si es menor va a sumaunidad
sumadecena
SUB R1,R1,R3                ; R1=R1-R3
ADD R4,#0x01                ; R4=R4+1
B ciclorestad              ; Salto a ciclorestad
sumaunidad

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves Grupo 3	Página 34 de 41	

```



      ADD R8,R1,R8                ; R8=R1+R8
      B aplicaciones              ; Salto a aplicaciones
aplicaciones
      MOV R6,#0x10                ; R6=16
      MUL R4,R4,R6                ; R4=R4*R6
      ADD R4,R4,R8                ; R4=R4+R8
      MOV R1,R7                  ; R1=R7
      MUL R8,R4,R1                ; R8=R4*R1
      MOV R3,#0x64                ; R3=100
      MOV R2,R8                  ; R2=R8
      UDIV R2,R2,R3               ; R2=R2/R3
      SUB R7,R7,R2                ; R7=R7-R2
      MUL R2,R2,R3                ; R2=R2*R3
      SUB R2,R8,R2                ; R2=R8-R2
      MOV R1,R11                  ; R1=R11
      MOV R3,R4                  ; R3=R4
      MUL R1,R1,R3                ; R1=R1*R3
      MOV R3,#0x64                ; R3=100
      UDIV R1,R1,R3               ; R1=R1/R3
      ADD R1,R1,R2                ; R1=R1+R2
      CMP R1,R11                  ; Comparamos el valor decimal actual con el de
descuento para ver si sobrepasa 1
      BEQ resta2d                 ; Si son iguales salta a resta2d
      BHS resta1d                 ; Si es mayor o igual salta a resta1d
      BLO resta2d                 ; Si es menor salta a resta2d
resta1d
      SUB R2,R1,R11                ; R2=R1-R11
      MOV R11,#0x01                ; R11=1
      SUB R7,R7,R11                ; R7=R7-R11
      MOV R11,#0x64                ; R11=100
      SUB R11,R11,R2               ; R11=R11-R2
      MOV R8,#0x00                ; Reinicia el valor de unidades
      MOV R4,#0x00                ; Reinicia el valor de decenas
      MOV R2,#0x00                ; Reinicia el valor de decimas
      MOV R1,#0x00                ; Reinicia el valor de centésimas
      B teclado                   ; Salto a teclado
resta2d
      SUB R11,R11,R1               ; R11=R11-R1
      MOV R8,#0x00                ; Reinicia el valor de unidades
      MOV R4,#0x00                ; Reinicia el valor de decenas
      MOV R2,#0x00                ; Reinicia el valor de decimas
      MOV R1,#0x00                ; Reinicia el valor de centésimas
      B teclado                   ; Salto a teclado
etiquetaifD

;----- ESCRITURA DE UNIDADES -----

      CMP R5,#0x00                ; Identifica que dígito se va a recibir
      BNE unidades                ; Si no es igual a cero salta a unidades
      CMP R10,#0x05               ; Comparamos el valor de decimales con 5 para
muestra de 4 dígitos simultáneos
      BEQ muestratodo              ; Si son iguales salta a muestratodo
      B unidades                  ; Sino salta a unidades

unidades
      CMP R5,#0x01                ; Compara R5 con un dígito
      BEQ habilitaciondecenas      ; Si son iguales habilita solo decenas

```

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores		Semestre: 2020-2	 INGENIERÍA ELÉCTRICA ELECTRÓNICA
	Prof. Dr. Saúl De La Rosa Nieves	Grupo 3	Página 35 de 41	

```

MOV R6,#0xF7          ; R6 es el valor del primer dígito
STR R6,[R3]           ; Habilita primer dígito
STR R4,[R9]           ; Muestra decenas
MOV R6,#0x00          ; R6 es cero para apagar leds
STR R6,[R9]           ; Apaga primer dígito
MOV R6,#0xFB          ; R6 es valor del segundo dígito
STR R6,[R3]           ; Habilita segundo dígito
STR R8,[R9]           ; Muestra unidades
MOV R6,#0x00          ; R6 es cero para apagar leds
STR R6,[R9]           ; Apaga segundo dígito
MOV R5,#0x00          ; Iguala a R5 con 0
CMP R10,#0x04         ; Compara para mostrar decimales
BEQ reinicioR10       ; Si son iguales se reinician decimales
B decimales           ; Salta a decimales

```

habilitaciondecenas

```

MOV R6,#0xF7          ; Iguala R6 a primer dígito
STR R6,[R3]           ; Habilita primer dígito
STR R4,[R9]           ; Muestra decenas
MOV R6,#0x00          ; Iguala a 0 para apagar leds
STR R6,[R9]           ; Apaga leds
CMP R10,#0x04         ; Compara decimales
BEQ reinicioR10       ; Si son iguales los reinicia
B decimales           ; Salto a decimales

```

reinicioR10

```

MOV R10,#0x00         ; Ciclo de reinicio para muestra de decimales

```

comp

```

CMP R10,#0x90000      ; Comparamos para 90,000 iteraciones
BNE sumaR10           ; Si no son iguales salta a sumaR10
MOV R10,#0x05         ; Iguala R10 a 5 para reiniciar decimales
B decimales           ; Salta a decimales



```

sumaR10

```

MOV R6,#0xF7          ; R6 dato que habilita dígito 1
STR R6,[R3]           ; Habilita dígito 1
STR R4,[R9]           ; Muestra decenas
MOV R6,#0x00          ; Iguala R6 a cero para apagar leds
STR R6,[R9]           ; Apaga leds
MOV R6,#0xFB          ; R6 dato que habilita dígito 2
STR R6,[R3]           ; Habilita dígito 2
STR R8,[R9]           ; Muestra unidades
MOV R6,#0x00          ; Iguala R6 a cero para apagar leds
STR R6,[R9]           ; Apaga leds
MOV R6,#0x80          ; Iguala R6 al valor de un punto
STR R6,[R9]           ; Enciende punto de dígito 2
MOV R6,#0x00          ; Iguala R6 a cero para apagar leds
STR R6,[R9]           ; Apaga leds
MOV R6,#0xFD          ; R6 dato que habilita dígito 3
STR R6,[R3]           ; Habilita dígito 3
STR R2,[R9]           ; Muestra decimas
MOV R6,#0x00          ; Iguala R6 a cero para apagar leds
STR R6,[R9]           ; Apaga leds
MOV R6,#0xFE          ; R6 dato que habilita dígito 4
STR R6,[R3]           ; Habilita dígito 4
STR R1,[R9]           ; Muestra centésimas
MOV R6,#0x00          ; Iguala R6 a cero para apagar leds
STR R6,[R9]           ; Apaga leds
ADD R10,#0x01         ; Suma a cuenta de decimales 1

```

	<p align="center">UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores</p>	<p>Semestre: 2020-2</p>	
	<p>Prof. Dr. Saúl De La Rosa Nieves Grupo 3</p>	<p>Página 36 de 41</p>	

B comp ; Salta a comp

;----- ESCRITURA DE DECIMALES -----;

decimales

CMP R10,#0x00 ; Compara R10 con cero para no habilitación de

decimales

BNE habilitaciondec ; Si no son iguales si los muestra

B teclado ; Si si regresa a teclado

habilitaciondec

CMP R10,#0x03 ; Compara R10 con 3

BEQ habilitaciondecenasd ; Si son iguales salta a habilitaciondecenasd

MOV R6,#0xFB ; R6 se iguala a habilitación de dígito 2

STR R6,[R3] ; Habilita dígito 2

MOV R6,#0x80 ; Iguala a punto de dígito 2

STR R6,[R9] ; Enciende punto de dígito 2

MOV R6,#0x00 ; Se iguala a cero para apagar los leds

STR R6,[R9] ; Apaga dígito 2

MOV R6,#0xFD ; R6 se iguala a habilitación de dígito 3

STR R6,[R3] ; Habilita dígito 3

STR R2,[R9] ; Muestra dígito 3

MOV R6,#0x00 ; Se iguala a cero para apagar los leds

STR R6,[R9] ; Apaga dígito 3

MOV R6,#0xFE ; R6 se iguala a habilitación de dígito 4

STR R6,[R3] ; Habilita dígito 4

STR R1,[R9] ; Muestra dígito 4

MOV R6,#0x00 ; Se iguala a cero para apagar los leds

STR R6,[R9] ; Apaga dígito 3

B teclado ; Salta a teclado

habilitaciondecenasd

MOV R6,#0xFB ; Iguala R6 a valor de habilitación dígito 2

STR R6,[R3] ; Habilita dígito 2

MOV R6,#0x80 ; Iguala al punto de dígito 2

STR R6,[R9] ; Enciende punto dígito 2

MOV R6,#0x00 ; Iguala R6 a 0 para apagar leds

STR R6,[R9] ; Apaga dígito 2

MOV R6,#0xFD ; Iguala R6 a valor de habilitación dígito 3

STR R6,[R3] ; Habilita dígito 3

STR R2,[R9] ; Enciende dígito 3

MOV R6,#0x00 ; Iguala R6 a 0 para apagar leds

STR R6,[R9] ; Apaga dígito 3

B teclado ; Salta a teclado

muestratodo

MOV R6,#0xF7 ; Iguala R6 a dígito 1

STR R6,[R3] ; Habilita dígito 1

STR R4,[R9] ; Muestra dígito 1

MOV R6,#0x00 ; Iguala R6 para apagar leds

STR R6,[R9] ; Apaga dígito 1

MOV R6,#0xFB ; Iguala R6 a dígito 2

STR R6,[R3] ; Habilita dígito 2

STR R8,[R9] ; Muestra dígito 2



MOV R6,#0x00 ; Iguala R6 para apagar leds

STR R6,[R9] ; Apaga dígito 2

MOV R6,#0x80 ; Iguala R6 punto

STR R6,[R9] ; Enciende punto dígito 2

MOV R6,#0x00 ; Iguala R6 para apagar leds

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA Microprocesadores y Microcontroladores	Semestre: 2020-2	
	Prof. Dr. Saúl De La Rosa Nieves	Página 37 de 41	

```

STR R6,[R9]                ; Apaga dígito 3
MOV R6,#0xFD               ; Iguala R6 a dígito 3
STR R6,[R3]                ; Habilita dígito 3
STR R2,[R9]                ; Muestra dígito 3
MOV R6,#0x00               ; Iguala R6 para apagar leds
STR R6,[R9]                ; Apaga dígito 1
MOV R6,#0xFE               ; Iguala R6 a dígito 4
STR R6,[R3]                ; Habilita dígito 4
STR R1,[R9]                ; Muestra dígito 4
MOV R6,#0x00               ; Iguala R6 para apagar leds
STR R6,[R9]                ; Apaga dígito 4
B teclado

start
MOV R6,#0x00               ; Iguala R6 a 0 para saber que dígito se está
ingresando
B start1                   ; Salta a ciclo start1
start1
CMP R6,#0x9000             ; Compara con 90,000 iteraciones
BNE start2                 ; Sino salta a start2
B start3                   ; Salta a start3
start2
ADD R6,#0x01               ; Suma a R6 1 para siguiente iteración
NOP                        ; No operación
NOP                        ; No operación
NOP                        ; No operación
B start1                   ; Salta a start1
start3
CMP R10,#0x05              ; Compara R10 con 5 para reinicio de decimales
BEQ saltoR5                ; Si si salta a saltoR5
CMP R10,#0x00              ; Compara R10 con 0
BNE start4                 ; Si no son iguales salta a start4
saltoR5
ADD R5,#0x01               ; Suma 1 a R5
CMP R5,#0x01               ; Compara R1 con 1
BEQ decena                 ; Si son iguales salta a decena
CMP R5,#0x02               ; Si no son iguales compara R5 con 2
BEQ unidad                 ; Si son iguales salta a unidad
decena
MOV R10,#0x00              ; Iguala R10 a 0
MOV R4,R12                 ; Iguala R4 a R2
B teclado                  ; Salto a teclado
unidad
MOV R8,R12                 ; Iguala R8 a R12
B teclado                  ; Salta a teclado
start4
ADD R10,#0x01              ; A R10 le suma 1
CMP R10,#0x02              ; Compara R10 con 2
BEQ inicio                 ; Si son iguales salta a inicio
CMP R10,#0x03              ; Compara R10 con 3
BEQ decenad                ; Si son iguales salta a decenad
CMP R10,#0x04              ; Compara R10 con #0x01
BEQ unidadd                ; Si son iguales salta a unidadd
inicio
B teclado                  ; Salta a teclado
decenad
MOV R2,R12                 ; Iguala R2 con R12
B teclado                  ; Salta a teclado

```

unidadd

```

MOV R1,R12           ; Iguala R1 con R12
B teclado            ; Salta a a teclado
; -----
.end                  ; FIN

```

VI. Construcción

La construcción de este proyecto requirió de distintos materiales para su implementación. Los materiales utilizados fueron:

- Tiva C TMC1294NCPDT
- Cable de datos del microcontrolador (USB)
- Display de 4 dígitos a 7 segmentos
- Cable jumper Hembra-Macho
- Alambre calibre 22
- Resistencias de 330 Ohms
- Led rojo
- Led verde
- Cautín
- Pasta para soldar
- Soldadura

El primer paso para la implementación fue soldar las terminales del teclado matricial que conseguí ya que éste no tenía las terminales con alambre.



Figura. Teclado matricial utilizado

Una vez soldado el teclado matricial se conecto a una protoboard utilizando las filas como bits más significativos y las columnas como bits menos significativos, de tal forma que el barrido de los valores del teclado se realizan respecto a las 4 filas del teclado.

Para la lectura del teclado se utilizó el puerto K, de tal forma que las 8 terminales del teclado se utilizaron para los puertos PK0-PK7 de esta manera:

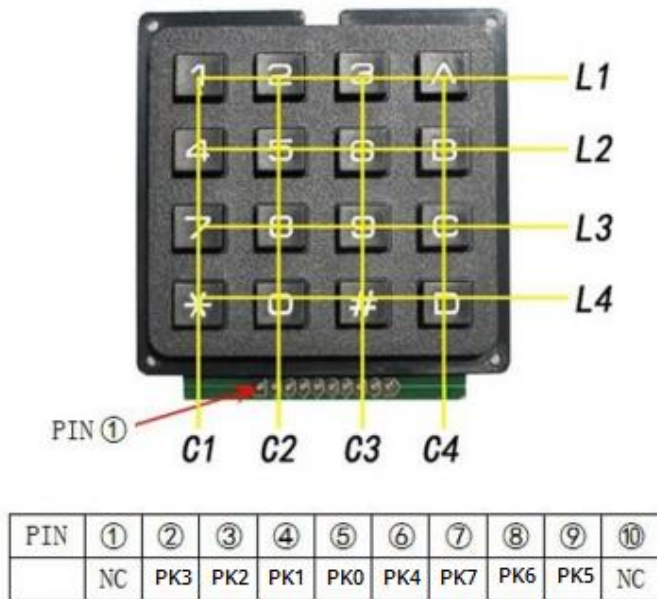


Figura. Asignación del puerto K al teclado matricial

Posterior a la asignación del teclado matricial se realizó la implementación de los leds rojo y verde para abrir y cerrar la caja. Se colocó en cada LED una resistencia de 330 ohms aunque también se habilitó una resistencia de pull down para ambas salidas.

El puerto que utilizaron los leds fue el puerto H, utilizando PH0 y PH1. Siendo el led rojo PH0 y el led verde PH1, la conexión se realizó de la siguiente manera:



Figura . Implementación del puerto H

Una vez alambrados los dos leds al puerto H se prosiguió a configurar el puerto para los 7 segmentos del display de 4 dígitos y el puerto para habilitación de cada uno de los dígitos. Para los 7 segmentos se configuró el puerto M y para la habilitación de los dígitos se habilitó el puerto E. Se configuraron todas las terminales del puerto M como salidas y los primeros 4 bits del puerto E como salidas.

Las conexiones se realizaron de la siguiente manera:

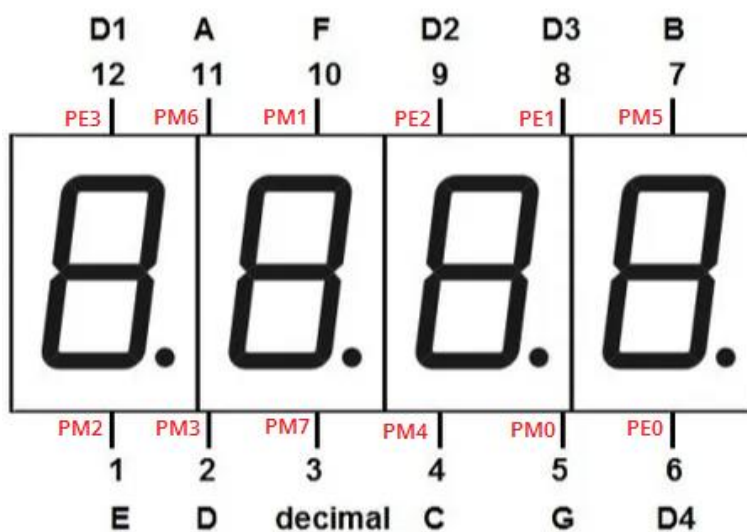
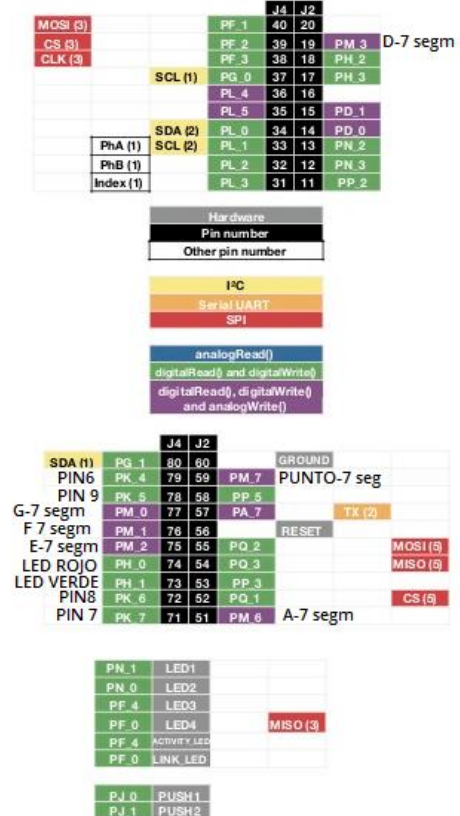
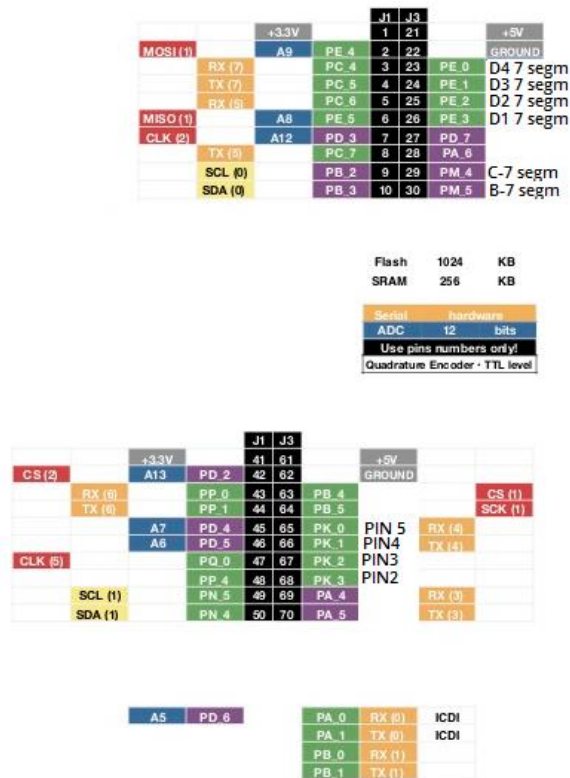


Figura. Conexión del display 7 segmentos

Por lo que la conexión final al microcontrolador fue la siguiente:



VII. Resultados y – Conclusiones

En conclusión, pude verificar que este proyecto desarrolló mucho mi capacidad de utilizar el microcontrolador como una herramienta para varias utilidades. Se utilizaron los temas que se vieron en la clase de manera que se reforzó bastante el conocimiento para el lenguaje ensamblador. Al inicio del proyecto y la programación del diseño pude notar que utilicé algoritmos bastante largos y repetitivos dado que en ese momento fue lo que mejor se me ocurrió para desarrollar las operaciones.

Sin embargo, conforme fui avanzando en la programación me di cuenta que muchas cosas pudieron quizás simplificarse y así utilizar menos líneas de código para lograr el mismo cometido. Aun así, considero que la implementación desde cero de este proyecto desarrolló mis capacidades de análisis para la programación de futuros proyectos y a su vez solidificó mis bases de diseño digital y los conocimientos de microprocesadores y microcontroladores.