BCD

7 Segmentos

El display de 7 segmentos es un indicador luminoso muy común en las máquinas de turnos (figura 1)



Figura 1 (máquina de turnos)

Cada dígito de este dispositivo está constituido por 8 leds conectados a 8 salidas de una computadora de decodificación

Los nombres de cada segmento se muestran en la figura 2

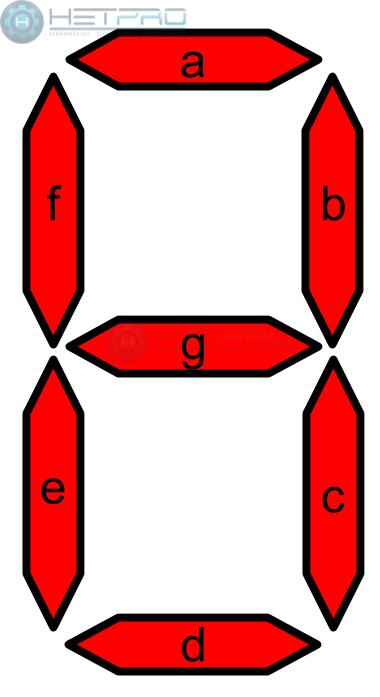
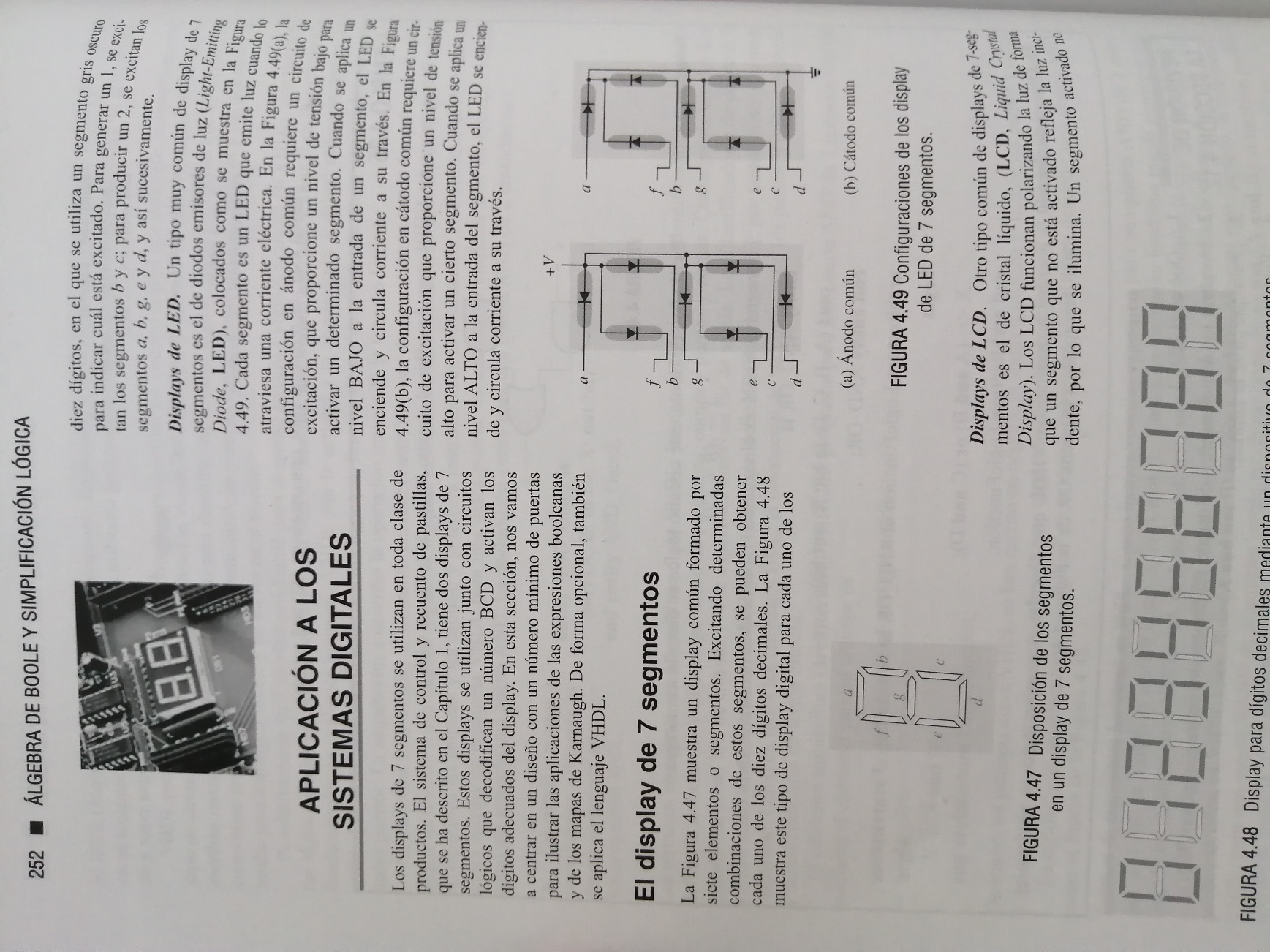
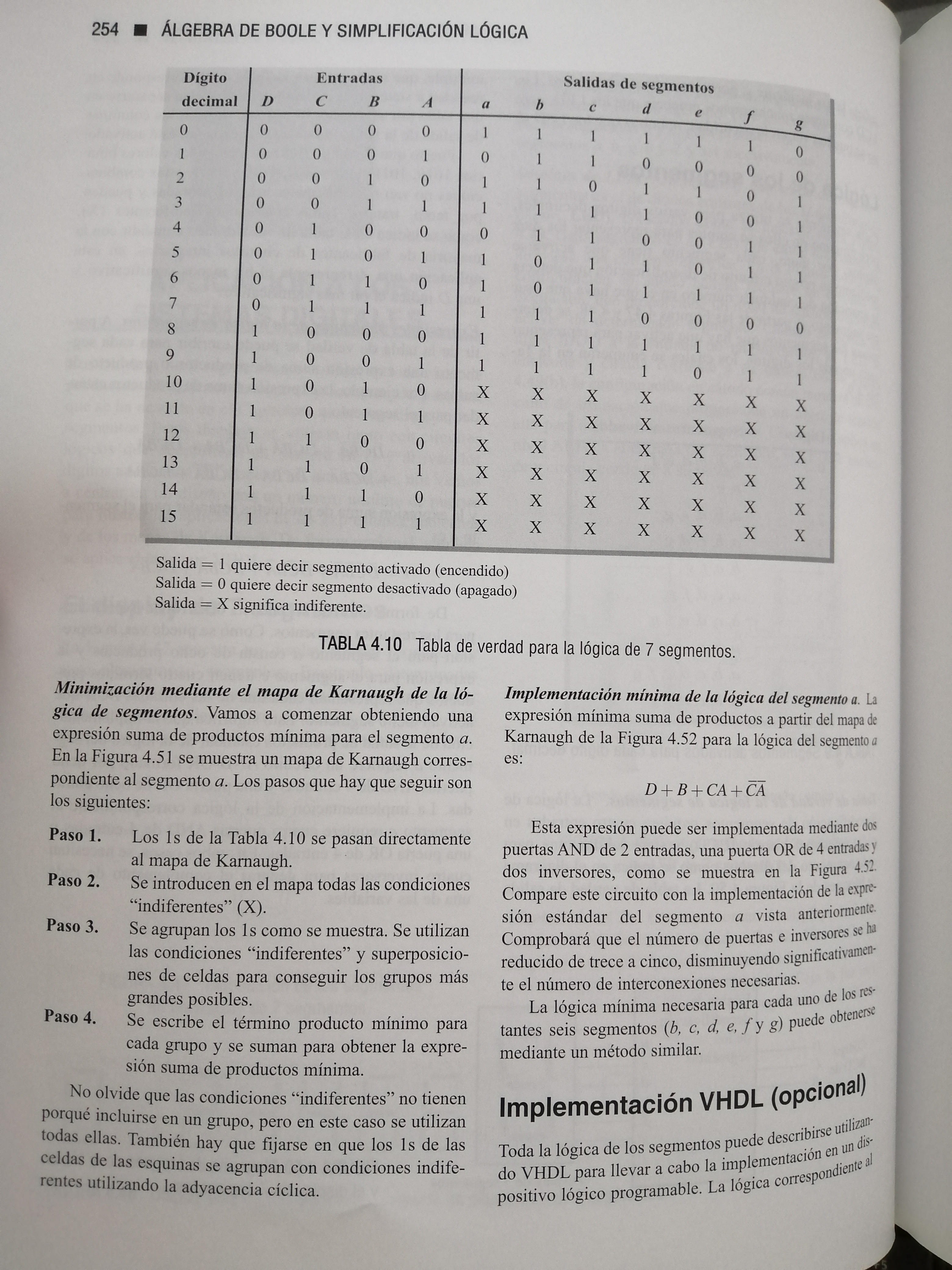
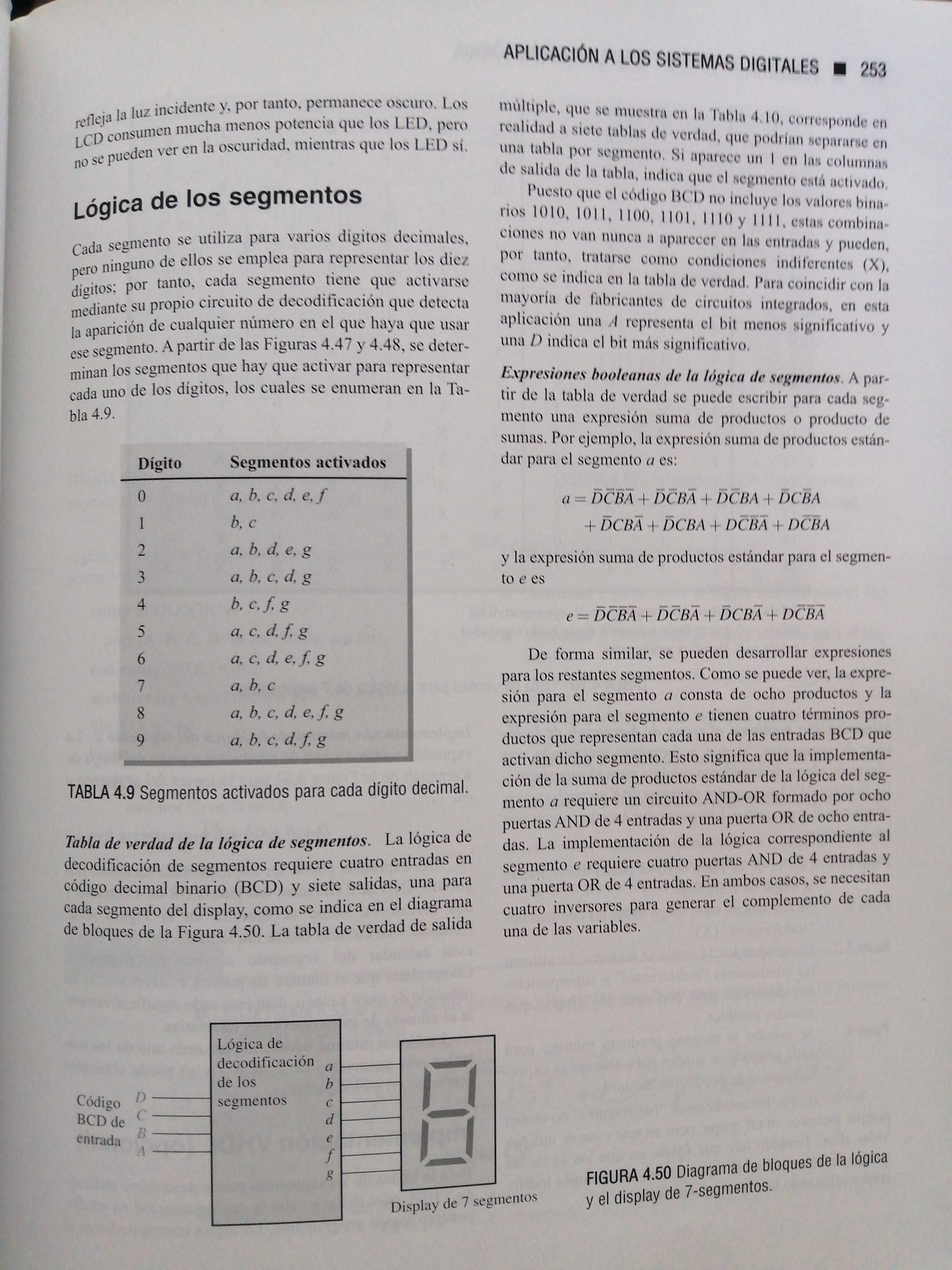


Figura 2 (nombres de cada segmento)

El siguiente segmento del libro (Floyd, 2009) muestra lo que es el display de 7 segmentos desde el punto de vista lógico. Lea detenidamente la información





Se le solicita desarrolle un script para Python (Raspberry) el cual reciba un número insertado bit por bit (que serían las 4 entradas), lo decodifique y diga que segmentos debe encender y despliegue el número en pantalla

Por ejemplo, si el usuario inserta un 9, 1001, deben encender los segmentos que se muestran para este número en la figura 3

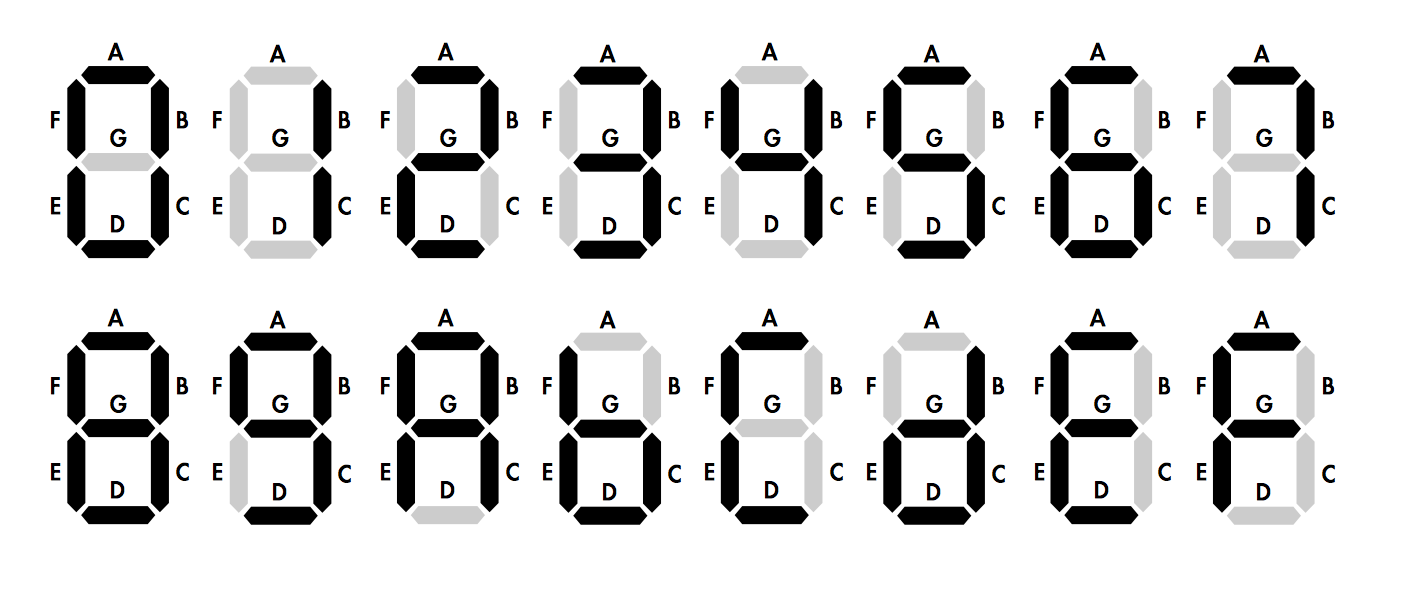


Figura 3 (segmentos encendidos para cada número)

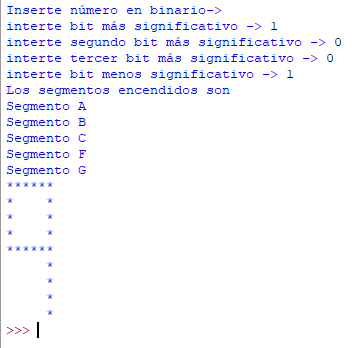


Figura 4 (Salida del script)

Este script debe ser lo más corto posible, es decir, cada una de las salidas, debe ser modelada con un mapa de Karnaugh y las condiciones lógicas lo más simplificadas que sea posible

El documento debe contener lo siguiente:

* Los mapas de karnaugh para cada una de las salidas (es decir para cada segmento)
* El circuito para cada segmento en caso de que se requiera hacer el diseño digital para alguna plataforma (PLC, Compuertas, circuitos lógicos, etc)
* El script y una imagen de cada uno de los casos

El script también debe ser enviado aparte

El proyecto se desarrolla por equipo y cuenta con 40 minutos para realizarlo

def ui():

decimal = 0

posicion = 1

potencia = 3

booleanos = []

while True:

boolean = True

if potencia < 0:

break

inputt = int(input((str(posicion) + "° bit mas significativo: ")))

if inputt == 0:

boolean = False

booleanos.append(boolean)

decimal = decimal + (inputt \* (2\*\*potencia))

potencia = potencia - 1

posicion = posicion + 1

return decimal, booleanos

print ("---Inserte número en binario :V ---")

decimal, booleanos = ui()

print ("El numero decimal que corresponde es:", decimal)

#print(booleanos)

s = 0

a = booleanos[3]

b = booleanos[2]

c = booleanos[1]

d = booleanos[0]

# strHor = Astr = Gstr = Dstr

# strDoble = (Fstr and Bstr) or (Estr and Cstr)

# strIzq = Fstr or Estr

# strDer = Bstr or Cstr

strHor = "\*\*\*\*\*\*"

strDoble = '''\* \*

\* \*

\* \*

\* \*'''

strIzq = '''\*

\*

\*

\*'''

strDer = ''' \*

\*

\*

\*'''

#A = D + B + A'C' + AC

if d or b or (not a and not c) or (a and c):

print(strHor)

#F = D + A'B' + A'C + B'C

Fbool = d or (not a and not b) or (not a and c) or (not b and c)

#B = C' + A'B' + AB

Bbool = (not c) or (not a and not b) or (a and b)

if Fbool and Bbool:

print(strDoble)

else:

if Fbool:

print(strIzq)

else:

print(strDer)

#G = D + A'C + BC' + B'C

if (d) or (not a and c) or (b and not c) or (not b and c):

print(strHor)

#E = A'B + A'C'

Ebool = (not a and b) or (not a and not c)

#C = C + A + B'

Cbool = (c) or (a) or (not b)

if Ebool and Cbool:

print(strDoble)

else:

if Ebool:

print(strIzq)

else:

print(strDer)

#D = D + A'B + A'C' + BC' + AB'C

if d or (not a and b) or (not a and not c) or (b and not c) or (a and not b and c):

print(strHor)

print("---------------")

print("Author: Jorge, Grecia, Mike, Ruben y Victor")