**PSEUDOCÓDIGO**

**Algoritmo “Pizzamiga”**

**Declaración:**

* **Tipos:**

fecha: Arreglo [3] de entero

hora: Arreglo [2] de entero

minutos\_loc: Arreglo [3] de entero

solicitud: registro de

num,desc,monto\_ini,monto,min\_entr: entero

fecha\_solic: fecha

hora\_ped,hora\_entr:hora

nomb\_loc,ced\_cli\_est\_entr: cadena

fin registro

cliente: registro de

cedula,nombre,apellido,dirección: cadena

fin registro

localidad: registro de

nombre: cadena

tiempo1,tiempo2,tiempo2: entero

fin registro

* **Variables:**

solic:solicitud

solici: Archivo de solicitud

clien,lin\_cli:cliente

client: Archivo de cliente

local,lin\_local: localidad

locali: Archivo de localidad

min\_loc:minutos\_loc

hoy,fch:fecha

apert,cierre: booleano

opc:cadena

num,err:entero

salir\_apl:booleano

**función** localidad\_rep(↓nombre): booleano

**variables**

encontrado: booleano

**inicio**

abrir (locali,’localidades.dat’)

encontrado←falso

mientras (NO (FinArchivo(locali) )) hacer

leer (local,lin\_local)

si (lin\_local.nombre=nombre) entonces

encontrado←verdadero

fin si

fin mientras

cerrar (locali)

localidad\_rep←encontrado

**fin función**

**función** validar\_min\_loc: booleano

**inicio**

si (err<>0) O (num<=0) entonces

validar\_min\_loc←falso

sino

validar\_min\_loc←verdadero

fin si

**fin función**

**proceimiento** mostrar\_localidades

**inicio**

abrir (locali,’localidades.dat’)

escribir(‘Listado de localidades registradas’)

mientras (NO (FinArchivo(locali))) hacer

leer (local,lin\_local)

escribir (‘\*Nombre: ’,lin\_local.nombre)

escribir (‘ Tiempo nr. 1 (min): ’,lin\_local.tiempo1)

escribir (‘ Tiempo nr. 2 (min): ’,lin\_local.tiempo2)

escribir (‘ Tiempo nr. 3 (min): ’,lin\_local.tiempo3)

fin mientras

cerrar (locali)

**fin procedimiento**

**procedimiento** modificar\_loc

**inicio**

mostrar\_localidades

escribir ('Nombre de la localidad a modificar:')

leer (local.nombre)

mientras (localidad\_rep(local.nombre)=falso) hacer

escribir('Nombre de localidad no encontrado, ingrese nuevamente dicho

nombre')

leer (local.nombre)

fin mientras

abrir (locali,’localidades.dat’)

mientras (NO (FinArchivo(locali))) hacer

leer (locali,lin\_local)

si (lin\_local.nombre=local.nombre) entonces

escribir (‘tiempo nr. 1 (min): ’)

leer (local.tiempo1)

min\_loc[1]←local.tiempo1

escribir (‘tiempo nr. 2 (min):’)

leer (local.tiempo2)

mientras (local.tiempo1=local.tiempo2) hacer

escribir (‘Cada tiempo asignado debe ser distinto a los

demás. Ingrese nuevamente dicho valor’)

leer (local.tiempo2)

fin mientras

min\_loc[2]←local.tiempo2

si (min\_loc[1]>min\_loc[2]) entonces

num← min\_loc[2]

min\_loc[2]← min\_loc[1]

min\_loc[1]←num

local.tiempo1← min\_loc[1]

local.tiempo2← min\_loc[2]

fin si

escribir (‘tiempo nr. 3 (min):’)

leer (local.tiempo3)

mientras(local.tiempo1=local.tiempo3) O

(local.tiempo2=local.tiempo3) hacer

escribir (‘Cada tiempo asignado debe ser distinto a los

demás. Ingrese nuevamente dicho valor’)

leer (local.tiempo3)

fin mientras

min\_loc[3]←local.tiempo3

si (min\_loc[3]<min\_loc[2]) entonces

num← min\_loc[3]

min\_loc[3]← min\_loc[2]

min\_loc[2]← min\_loc[1]

min\_loc[1]←num

local.tiempo1← min\_loc[1]

local.tiempo2← min\_loc[2]

local.tiempo3← min\_loc[3]

sino si (min\_loc[3]<min\_loc[2]) entonces

num← min\_loc[3]

min\_loc[3]← min\_loc[2]

min\_loc[2]← num

local.tiempo2← min\_loc[2]

local.tiempo3← min\_loc[3]

fin si

escribir (locali,local)

fin si

fin mientras

cerrar (locali)

**fin procedimiento**

**procedimiento agregar\_loc**

**inicio**

escribir (‘tiempo nr. 1 (min): ’)

leer (local.tiempo1)

min\_loc[1]←local.tiempo1

escribir (‘tiempo nr. 2 (min):’)

leer (local.tiempo2)

mientras (local.tiempo1=local.tiempo2) hacer

escribir (‘Cada tiempo asignado debe ser distinto a los demás.

Ingrese nuevamente dicho valor’)

leer (local.tiempo2)

fin mientras

min\_loc[2]←local.tiempo2

si (min\_loc[1]>min\_loc[2]) entonces

num← min\_loc[2]

min\_loc[2]← min\_loc[1]

min\_loc[1]←num

local.tiempo1← min\_loc[1]

local.tiempo2← min\_loc[2]

fin si

escribir (‘tiempo nr. 3 (min):’)

leer (local.tiempo3)

mientras(local.tiempo1=local.tiempo3) O (local.tiempo2=local.tiempo3) hacer

escribir (‘Cada tiempo asignado debe ser distinto a los demás.

Ingrese nuevamente dicho valor’)

leer (local.tiempo3)

fin mientras

min\_loc[3]←local.tiempo3

si (min\_loc[3]<min\_loc[2]) entonces

num← min\_loc[3]

min\_loc[3]← min\_loc[2]

min\_loc[2]← min\_loc[1]

min\_loc[1]←num

local.tiempo1← min\_loc[1]

local.tiempo2← min\_loc[2]

local.tiempo3← min\_loc[3]

sino si (min\_loc[3]<min\_loc[2]) entonces

num← min\_loc[3]

min\_loc[3]← min\_loc[2]

min\_loc[2]← num

local.tiempo2← min\_loc[2]

local.tiempo3← min\_loc[3]

fin si

abrir (locali,’localidades.dat’)

escribir (locali,local)

cerrar (locali)

**fin procedimiento**

**función** num\_reg\_ped: entero

**variables**

cont:entero

**inicio**

abrir (solici,’pedidos.dat’)

cont←tamaño(solici)

cerrar(solici)

num\_reg\_ped←cont

**fin función**

**procedimiento** ingresar\_fecha(↓↑hoy:fecha)

**inicio**

escribir(‘Ingrese el año’)

leer (hoy[3])

escribir (‘Ingrese el mes (1 al 12)’)

leer (hoy[2])

mientras (hoy[2]<1) O (hoy[2]>12) hacer

escribir (‘Valor fuera de rango. Ingrese nuevamente el mes’)

leer (hoy[2])

fin mientras

escribir (‘Ingrese el dia’)

leer (hoy[1])

si ((hoy[2]==1) O (hoy[2]==3) O (hoy[2]==5) O (hoy[2]==7) O (hoy[2]==8) O

(hoy[2]==10) O (hoy[2]==12) ) entonces

mientras (hoy[1]<1) O (hoy[1]>31) hacer

escribir (‘El dia debe ser numerico y que oscile entre 1 y 31. Ingrese

nuevamente el dia:’)

leer (hoy[1])

sino si ( (hoy[2]==4) O (hoy[2]==6) O (hoy[2]==9) O (hoy[2]==11) ) entonces

mientras (hoy[1]<1) O (hoy[1]>30) hacer

escribir (‘El dia debe ser numerico y que oscile entre 1 y 30.

Ingrese nuevamente el dia:’)

leer (hoy[1])

fin mientras

sino si ( (hoy[2]==2) O (hoy[2] **mod** 4 <>0) ) entonces

mientras (hoy[1]<1) O (hoy[1]>28) hacer

escribir (‘El dia debe ser numerico y que oscile entre 1 y 28.

Ingrese nuevamente el dia:’)

leer (hoy[1])

fin mientras

sino

mientras (hoy[1]<1) O (hoy[1]>29) hacer

escribir (‘El dia debe ser numerico y que oscile entre 1 y 29.

Ingrese nuevamente el dia:’)

leer (hoy[1])

fin mientras

fin si

**fin procedimiento**

**procedimiento** ingresar\_hora(↓↑hra:hora)

**inicio**

escribir(‘Ingrese la hora (Formato Militar de 0 a 23)’)

leer (hra[1])

mientras (hra[1]<0) O (hra[1]>23) hacer

escribir (‘Valor fuera de rango. Ingrese nuevamente la hora’)

leer (hra[1])

fin mientras

escribir(‘Ingrese los minutos (De 0 a 59)’)

leer (hra[2])

mientras (hra[2]<0) O (hra[2]>59) hacer

escribir (‘Valor fuera de rango. Ingrese nuevamente los minutos’)

leer (hra[2])

fin mientras

**fin procedimiento**

**procedimiento** registrar\_pedido

**variables**

encontrado: booleano

**inicio**

escribir(‘Cédula del cliente: ’)

leer (solic.ced\_cli)

abrir (client,’clientes.dat’)

encontrado←falso

mientras (NO (FinArchivo(client))) hacer

leer (client,lin\_cli)

si (lin\_cli.cedula=solic.ced\_cli) entonces

encontrado←verdadero

escribir (‘Nombre: ’,lin\_cli.nombre)

escribir (‘Apellido: ’,lin\_cli.apellido)

escribir (‘Dirección: ’,lin\_cli.direccion)

fin si

mientras

cerrar (client)

si (encontrado=falso) entonces

clien.cedula←solic.ced\_cli

escribir(‘Nombre del cliente: ‘)

leer (clien.nombre)

escribir(‘Apellido del cliente: ‘)

leer (clien.apellido)

escribir(‘Dirección del cliente: ‘)

leer (clien.direccion)

abrir (client,’clientes.dat’)

escribir (client,clien)

cerrar (client)

fin si

escribir (‘Localidad en donde será entregado el pedido’)

leer (solic.nomb\_loc)

mientras (localidad\_rep(solic.nomb\_loc)=falso) hacer

escribir (‘Localidad no encontrada. Ingrese nuevamente el nombre de la

localidad’)

leer (solic.nomb\_loc)

fin mientras

solic.num←num\_reg\_ped+1

solic.fecha\_solic[1]←hoy[1]

solic.fecha\_solic[2]←hoy[2]

solic.fecha\_solic[3]←hoy[3]

escribir (‘Hora de la solicitud: ’)

ingresar\_hora (solic.hora\_ped)

solic.desc←0

solic.min\_entr←0

escribir (‘Monto del pedido: ’)

leer (solic.monto\_ini)

solic.monto←0

escribir (‘Su solicitud fue guardada de manera satisfactoria’)

escribir (‘Datos del pedido)

escribir (‘Número del pedido: ‘,solic.num)

escribir (‘Localidad: ‘,solic.nomb\_loc)

escribir (‘Cédula del cliente: ‘,solic.ced\_cli)

abrir (client,’clientes.dat’)

mientras (NO (FinArchivo(client))) hacer

leer (client,lin\_cli)

si (lin\_cli.cedula=solic.ced\_cli) entonces

escribir (‘Nombre del cliente: ’,lin\_cli.nombre)

escribir (‘Apellido del cliente: ’,lin\_cli.apellido)

escribir (‘Dirección del cliente: ’,lin\_cli.direccion)

fin si

mientras

cerrar (client)

escribir (‘Fecha de solicitud: ‘,solic.fecha\_solic[1],’/’,solic.fecha\_solic[2],’/’

,solic.fecha\_solic[3],)

escribir (‘Hora de solicitud: ‘,solic.hora\_ped[1],’hr:’,solic.hora\_ped[2],’min’)

escribir (‘Monto del pedido (Bs.): ’,solic.monto\_ini)

abrir (solici,’pedidos.dat’)

escribir (solici,solic)

cerrar (solici)

**fin procedimiento**