

Integrantes del equipo:
Lugo Santeliz Leonardo Daniel
Zega Juan Cruz
Goyenechea Álvaro
Lopez Frias Facundo Manuel

Programa principal

precarga()
validar_usuario (condicional)

type

COD : array[0..100] of int	✓
USER : array[0..100] of str	✓
CLAVE : array[STRING..100] of str	✓
TIPO : array[STRING..100] of str	✓
COD_LOCAL : array[ENTERO..50] of int	✓
NOMBRE_LOCAL : array[STRING..50] of str	1.1
UBICACION_LOCAL : array[STRING..50] of str	
COD_USUARIO : array[ENTERO..50] of int	
ESTADO : array[STRING..50] of str	
RUBRO : array[STRING..50] of str	

VAR

INT
LIM_USERS_RAW ← 100
LIM_USERS_COL ← 3
LIM_LOCALS_ROW ← 50
LIM_LOCALS_COL ← 5
i ← 3
cont_indu ← 0
cont_per ← 0
cont_comida ← 0
max_locales ← 0
STR
nombreLocal ← " "
rubroLocal ← " "
usr ← " "
contr_input ← " "
BOOL
condicional ← True
ARRAY
tipos_user ← ["administrador", "dueñoLocal", "cliente"]
tipo_local ← ["indumentaria", "perfumeria", "comida"]
locales ← [COD_LOCAL, NOMBRE_LOCAL, UBICACION_LOCAL, RUBRO, COD_USUARIO, ESTADO]
usuarios : array[COD, USER, CLAVE, TIPO]

validar_usuario (condicional)

while condicional = true

exh (“ingrese su nombre de usuario:”)			
leer (usr)			
usr_aprob ← busqueda(usuarios, LIM_USERS_RAW, LIM_USERS_COL,usr, 'codi')			
num_fila_usr ← busqueda(usuarios, LIM_USERS_RAW, LIM_USERS_COL,usr, 'fila')			
exh (“ingrese su contraseña”)			
contr_input ← (ocultar_pass)			
contr_aprob = busqueda(usuarios , LIM_USERS_RAW, LIM_USERS_COL, contr_input, 'condi')			
num_fila_pass = busqueda(usuarios , LIM_USERS_RAW, LIM_USERS_COL, contr_input, 'fila')			
V \usr_aprob != False and contr_aprob != False and num_fila_usr == num_fila_pass			F
exh ("ha iniciado sesion satisfactoriamente")		V \i > 1 / F	
V \if usuarios[3][num_fila] = tipos_user[0] /F		i ← i-1	exh("Máximo de intentos permitidos alcanzado")
condicional ← val_menu_admin(condicional)	usuarios[3][num_fila] == tipos_user[1] V \ / F	exh ("Tiene", i, "cantidad de intentos restantes")	condicional ← False
	condicional ← val_menu_owner (condicional)	condicional ← val_menu_client (condicional)	

ocultar()

menu_prin_client

exh ("1. Registrarme")
exh ("2. Buscar descuentos en locales")
exh ("3. Solicitar descuento")
exh ("4. Ver novedades")
exh ("0. Salir")

menu_prin_admin

exh ("1. Gestión de Locales")
exh ("2. Crear cuentas de dueños")
exh ("3. Aprobar / Denegar solicitud de descuento")
exh ("4. gestión de novedades")
exh ("5. Reporte")
exh ("0. salir ")

menu_prin_owner

exh ("1. Gestión de Descuentos")
exh ("2. Aceptar / Rechazar pedido de descuento")
exh ("3. Reporte de uso de descuentos")
exh ("0. Salir")

val_menu_client(condicional)

while condicional = true	
menu_prin_client	
exh ("Ingrese un numero:")	
if opcion != "0" and opcion != "1" and opcion != "2" and opcion != "3" and opcion != "4"	
opcion_erronea	elecciones_client

elecciones_client (opcion)

opcion					
'1'	'2'	'3'	'4'	'5'	'0'
menu_op1 (condicional)	en_construccion () return True	en_construccion () return True	en_construccion () return True	en_construccion () return True	return False exh ("Saliendo del programa")

val_menu_admin (condicional)

while condicional = True	
menu_prin_admin()	
leer(opcion)	
opcion <> '0' and opcion <> '1' and opcion <> '2' and opcion <> '3' and opcion <> '4' and opcion <> '5'	
opcion_erronea()	condicional ← elecciones_admin (opcion)



3.1



elecciones_admin (opcion)

opcion					
'1'	'2'	'3'	'4'	'5'	'0'
menu_op1 (condicional)	en_construccion () return True	en_construccion () return True	val_opc_menu_4 (condicional) return True	en_construccion () return True	exh ("saliendo del programa") return False

val_menu_owner (condicional)

while condicional = true	
menu_prin_owner	
exh ("Ingrese un numero:")	
if (opcion != "0" and opcion != "1" and opcion != "2" and opcion != "3")	
opcion_erronea()	elecciones_owner (opcion)

elecciones_owner (opcion)

opcion			
"1"	"2"	"3"	"0"
menu_gestion (condicional) return True	en_contruccion() return True	en_contruccion() return True	exh("saliendo del programa") return False

gestion_descuentos ()

exh ("a) Crear descuento para mi local")
exh ("b) Modificar descuento de mi local")
exh ("c) Eliminar descuento de mi local")
exh("d) Volver")

menu_gestion (condicional)

while condicional = true	
gestion_descuentos()	
exh ("escriba una opcion: ")	
opcion != "a" and opcion != "b" and opcion != "c" and opcion != "d"	
opcion_erronea()	elecciones_desc(opcion)

elecciones_desc(opcion)

opcion			
"a"	"b"	"c"	"d"
en_construccion() return True	en_construccion() return True	en_construccion() return True	return False

menu_1 ()

exh("a) Crear locales")
exh("b) Modificar local")
exh("c) Eliminar local")
exh("d) Mapa de locales")
exh("e) Volver")



menu_op1 (condicional)

while condicional = true	
menu_1()	
exh ("escriba una opcion: ")	
leer (opcion)	
V	F
opcion != "a" and opcion != "b" and opcion != "c" and opcion != "d"	
opcion_erronea()	condicional ← elecciones_op1 (opcion)

elecciones_op1 (opción)

opcion				
"a"	"b"	"c"	"d"	"e"
crear_locales() return True	modificar() return True	borrar() return True	mapa_locales() return True	return False



crear_locales()		
global max_locales		
nombreLocal ← ""		
pregunta ← "¿desea ver los locales?"		
aux ← ask_continue(pregunta)		
V/	aux = 's'	/F
ver_locales()		
while nombreLocal <> "**"		
column ← 1		
nombreLocal ← ingreso_nombre()		
V/ nombreLocal <> "**" and max_locales != 50 /F		
fila ← max_locales + 1		
column ← column+1		
ingreso_ubi(column, fila)		
column ← column+1		
ingreso_rubro(column, fila)		
column ← column+1		
ingreso_codigos(column, fila)		
column ← column+1		
asignar_codigo_active(fila, column)		
max_locales ← max_locales + 1		
ordenar()		
V/	max_locales == 50	/F
exh("Máxima cantidad alcanzada")		
contador_rubro()		
return		

ingreso_nombre ():			
cont ← 0			
nombreLocal ← “ ”			
while cont != 1 and nombreLocal != “*”			
exh ("---- ingresando un ' * ' se termina el ingreso de locales ----")			
exh ("Ingrese el nombre del local)			
leer (nombreLocal)			
V \		len(nombreLocal) > 2:	/F
special ← char_allow (nombreLocal)		V \	nombreLocal != “*” /F
V \	if special != True /F	exh ("---- mínimo de caracteres permitidos: 3 ----")	
aprob ← busqueda (locales, LIM_LOCALS_ROW, LIM_LOCALS_COL, nombreLocal, 'condi')		exh ("Carácter no permitido\n")	
fila ← busqueda(locales, LIM_LOCALS_ROW, LIM_LOCALS_COL, nombreLocal, 'fila')			
V \	aprob != True /F		
locales[1][fila] ← nombreLocal	repetido (“el nombre”)		
cont ← 1			

ingreso_ubi(column, fila)		
cont ← 0		
while cont != 1		
exh("Ingrese la ubicacion: ")		
leer (ubicacionLocal)		
V \		if len(ubicacionLocal) > 3 /F
special ← char_allow		exh ("---- mínimo de caracteres permitidos: 4 ---- ")
V \	if special != True: /F	
locales[column][fila] ← ubicacionLocal	exh ("Carácter no permitido")	
cont ← 1		

ingreso_rubro(column, fila)

aux ← False		
while aux <> True		
exh("Escoja un rubro: indumentaria, perfumería o comida")		
leer(rubroLocal)		
aux ← busqueda_uni(tipo_local, 3, rubroLocal, aux)		
V/	aux <> False	/F
locales[column][fila] ← rubroLocal		exh("Rubro invalido")

contador_rubro()

contador_max_min ← [cont_indu, cont_per, cont_comida]	
para i ← 1 to 3	
	contador_max_min[i] ← contador(i)
contador_max_min, tipos ← max_min_arrays (contador_max_min, tipo_local)	
para i← 1 to 3	
	exh ("Cantidad de locales de",tipos[i],":", contador_max_min[i])
	exh ("-----")


contador (type)

fila ← 1														
cont ← 0														
while fila <= max_locales														
<table><tr><td>V</td><td colspan="2">locales[3][fila] == tipo_local[type] and locales[5][fila] == "A"</td><td>F</td></tr><tr><td colspan="2">cont ← cont + 1</td><td colspan="2"></td></tr><tr><td colspan="2">fila ← fila + 1</td><td colspan="2"></td></tr></table>			V	locales[3][fila] == tipo_local[type] and locales[5][fila] == "A"		F	cont ← cont + 1				fila ← fila + 1			
V	locales[3][fila] == tipo_local[type] and locales[5][fila] == "A"		F											
cont ← cont + 1														
fila ← fila + 1														
return cont														

max_min_arrays(dato, tipo)

for j ← 1 a 3		
for i ← 1 a 2		
V/	dato[i] < dato[i+1]	/F
aux ← dato[i]		
dato[i] ← dato[i+1]		
dato[i+1] ← aux		
aux ← tipo[i]		
tipo[i] ← tipo[i+1]		
tipo[i+1] ← aux		
return dato, tipo		

ingreso_codigos(column, fila)

cont ← 0 while cont != 1		
exh("Ingresa el codigo")		
leer(cod_owner)		
aprob ← verif_num(cod_owner)		
V/ aprob!= False /F		
cod_owner ← int(cod_owner)		exh("Codigo invalido!!!")
aux ← busqueda (usuarios, LIM_USERS_RAW, LIM_USERS_COL, cod_owner, 'condicional')		
x ← busqueda(usuarios, LIM_USERS_RAW, LIM_USERS_COL, cod_owner, 'fila')		
V/ aux = True and usuarios[3][x] = tipos_users[1] /F		
locales[column][fila] ← cod_owner	exh("Codigo invalido!!!")	
cont ← cont + 1		

asignar_codigo_active(fila, column)

locales[0][fila] ← fila	
locales[column][fila] ← "A"	

modificar()


V\		max_locales > 0		/F	
modificar ← ""				exh ("---- No se ha ingresado ningun local al sistema ----")	
pregunta ← "desea ver los locales?"					
aux ← ask_continue(pregunta)					
V\		aux == 's'		/F	
ver_locales()					
cont ← 0					
while modificar != "*" and cont != 1					
exh ("---- ingrese '*' para salir ----\n")					
exh ("ingrese el codigo del local a modificar: ")					
leer (modificar)					
V\		modificar != '*'		/F	
aprob ← verif_num(modificar)					
V\		aprob != False		/F	
modificar ← int(modificar)			exh("codigo invalido!!!\n")		
condi ← busqueda(locales, LIM_LOCALS_ROW, 0, modificar, 'condi')					
fila ← busqueda(locales, max_locales, 0, modificar, 'fila')					
V\		locales[5][fila] != "A"		/F	
pregunta = "----El local que ingreso se encuentra dado de baja (B) ----\n\nDesea activarlo?"					
ask = ask_continue(pregunta)					
V\		ask != "n"		/F	
locales[6][fila] = "A"					
V\		condi != False and locales[5][fila] != "B"		/F	
choice_modifi(fila,modificar)		V\		condi != True	
				/F	
ordenar()		exh("codigo invalido!!!\n")	exh("--- Para modificar un local debe estar Activo (A) ---\n")		
cont ← 1					
exh("---- Local Modificado ----\n")					

condicional ← True	
while condicional = True	
exh("Elija Atributo a Modificar del local ", modificar)	
exh("1) nombre")	
exh("2) ubicacion")	
exh("3) rubro")	
exh("4) codigo dueño")	
exh("0) salir")	
leer(opcion)	
V/	opcion < '0' and opcion > "4" /F
opcion_erronea	condicional ← elecciones_modifi(opcion, fila, modificar)
return	

	opcion				
	1	2	3	4	0
modifi_name(1,filas)		ingreso_ubi(2,filas)	ingreso_rubro(3,filas)	ingreso_codigos(4,filas)	return False
return True		return True	return True	return True	

= 1		
ingrese el nombre del local")		
nombreLocal)		
len(nombreLocal) > 2		
al ← char_allow(nombreLocal)		exh("---- minimo de caracteres permitidos: 3 ----")
special != True		/F
o←busqueda(locales, LIM_LOCALS_ROW, LIM_LOCALS_COL, nombreLocal, 'condi')		exh("Caracter no permitido")
aprob != True		/F
les[column][fila] ← nombreLocal	repetido("El nombre")	
t ← cont + 1		

borrar()

borrar ← " "					
V/ /F max_locales > 0					
pregunta ← 'desea ver los locales?'				exh ("---- No se ha ingresado ningún local al sistema ----")	
aux ← ask_continue(pregunta)					
V/ /F aux = 's' /F					
ver_locales()					
cont ← 0					
while borrar != "*" and cont != 1					
exh ("---- ingrese '*' para salir ----\n")					
exh ("ingrese el código del local a borrar: ")					
leer (borrar)					
V\ /F borrar <> '*' /F					
aprob ← verif_num(borrar)					
V\ /F aprob != False /F					
borrar ← int(borrar)				exh ("codigo invalido!!!")	
condi ← busqueda(locales, LIM_LOCALS_ROW, 0, borrar, 'condi')					
fila ← busqueda(locales, max_locales, 0, borrar, 'fila') 					
V\ / F condi != False and locales[5][fila] != "B"					
locales[5][fila] ← "B"		V/ /F locales[5][fila] = "B" /F			
cont ← 1		exh("---- el local " + borrar + "ya se encuentra dado de baja ----")		exh("codigo invalido!!!")	
exh ("---- Local borrado ----")					

mapa_locales ():

```

all_locals ← 50
filas_map ← 5
cont ← 1
while cont <= all_locals
    column = 1
    exh ("++-++-++-++-++")
    while column <= filas_map
        if locales[0][cont] < 10
            exh ("|0"+locales[0][cont]", end←" ")
            column←column+ 1
        else
            exh ("|" + locales[0][cont], end←" ")
            column←column+ 1
        cont ← cont + 1
    exh ("|")
    exh ("++-++-++-++-++")
    cont ← cont + 1
exh ("---- presione cualquier tecla para salir ----")

```

ask_continue(pregunta)

aux2 ← True		
while aux2 = True		
exh ({pregunta} (s/n)		
aux1 ← aux1.lower()		
V \	if aux1 = "s" or aux1 = "n"	/ F
aux2 ← False	exh ("letra incorrecta") return aux1	

menu_4

exh("a) Crear novedades")
exh("b) Modificar novedad")
exh("c) Eliminar novedad")
exh("d) Ver reporte de novedad")
exh("e) Volver")

val_opc_menu_4(condicional)

while condicional != false		
menu_4 ()		
exh ("seleccione una opción")		
leer(opcion)		
opcion != 'a' and opcion != 'b' opcion != 'c' opcion != 'd' and opcion!= 'e'		
V \		/F
opcion_erronea()	condicional ← elecciones_op4(opcion)	

elecciones_op4 (opcion)

opcion				
a	b	c	d	e
en_construccion ()	en_construccion ()	en_construccion ()	en_construccion()	return False
return True	return True	return True	return True	

opcion_erronea ()

exh(" Por favor seleccione una de las opciones correctas")
--

en_construccion ()

exh('En construcción')

repetido(x)

exh (x, " que intenta ingresar ya se encuentra guardado ")
--



14.2

```

return False

```

```
return
```

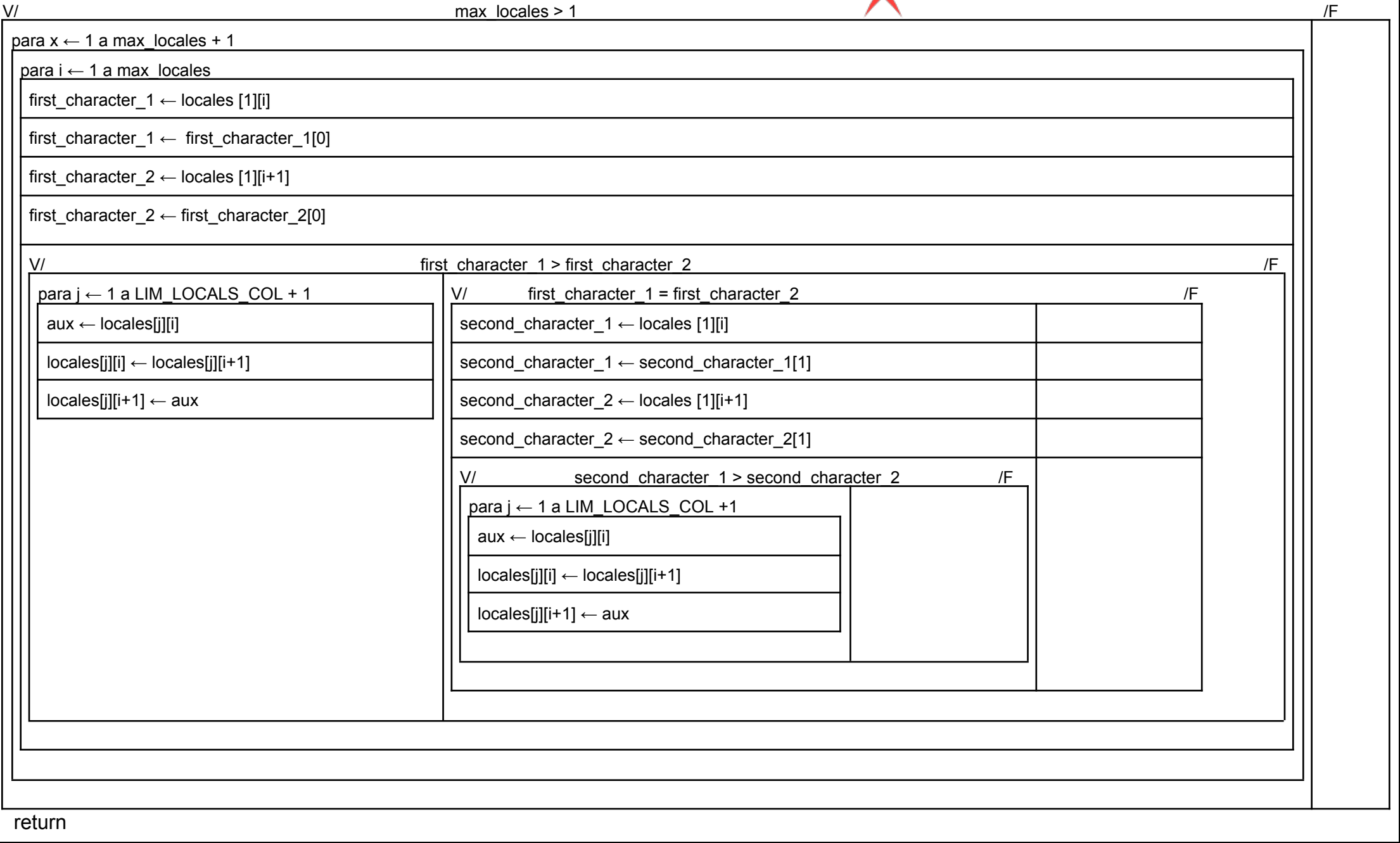
```
return aux
```

```
exh ("\\n\\n\\nA = Local activo   B = Local dado de baja")
```

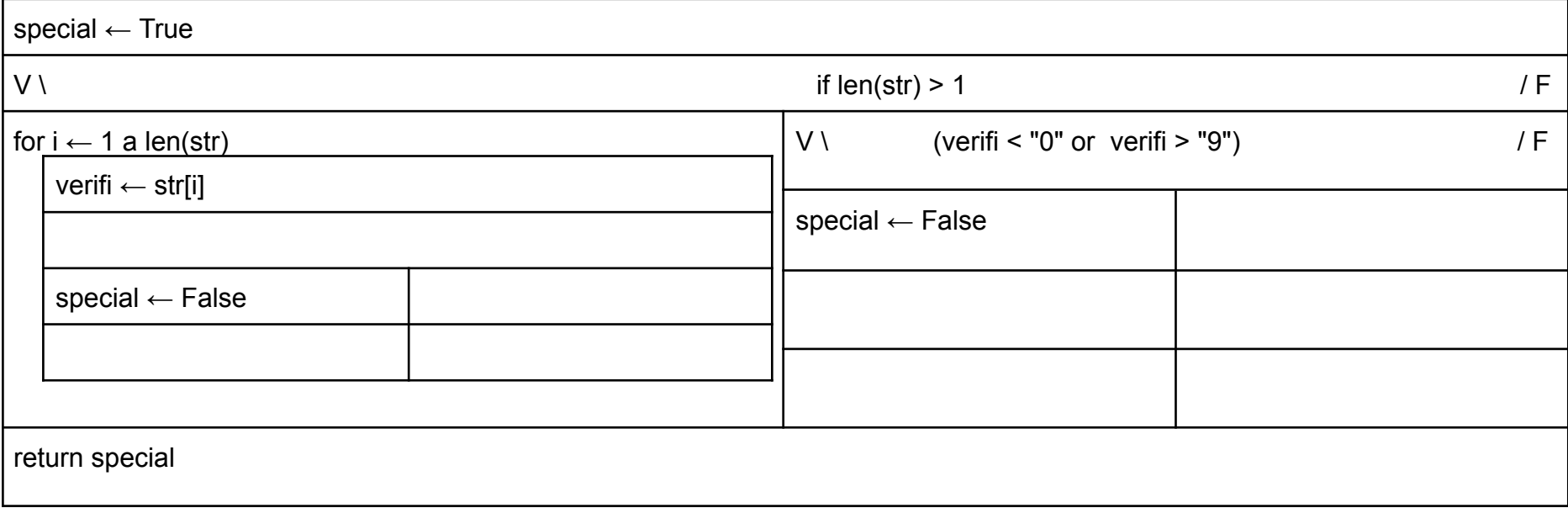
exh('no se ha ingresado ningún local al sistema)

ordenar()

15.1



verif_num(str)



char_allow (word)

special ← False		
for i ← 1 a len(word)		
verifi ← word[i]		
verifi ← verifi.upper()		
V \ i == 0 / F		
V \ if verifi = " " or verifi = " " or verifi = "" or verifi = "&" / F		
special = True		
V \ if ((verifi < "A" and verifi > "Z") or (verifi < "0" and verifi > "9")) and not(verifi != " " or verifi != "" or verifi != "&" or verifi != "."): special = True / F		
special = True		
return special		

precargar() #precarga de todos los valores (la variable locales llena de 0 todos los slots de locales [1][1])



Índice de comentarios

- 1.1 son todos del mismo tipo, se debería utiizar un arreglo bidimensional
- 1.2 uso de listas, tipo de dato no utilizado en la catedra.
Contradice la teoria que se aplica a os arreglos
- 2.1 si se define como arreglo unidimensional no podría usarse con 2 indices
- 3.1 pensar en dar la salida por un valor de la variable opcion, cuando lo elige el usuario, innecesario usar variables booleanas
- 6.1 condiciones para el while
- 6.2 faltaria mostrar las cantidades por rubro
- 14.1 No se apica busqueda dicotomica
- 14.2 la funcion debe devolver una salida por return, no usarlos rompiendo la estructura iterativa, sino al salir de ella.
- 15.1 el método de falso burbuja, simplifica este proceso, reveer.