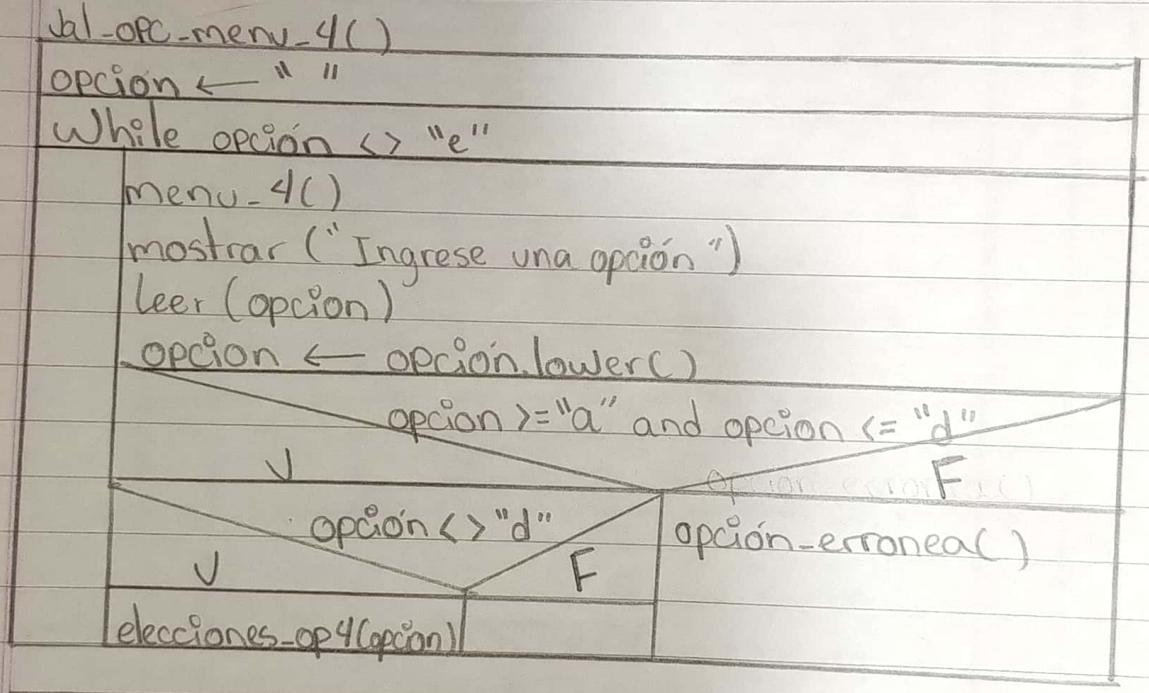
Modelado en Chapin

```
Class novedades():
    def __init__(self):
        self.codNovedad <= " "
        self.fechaDesdeNovedad <= "ØØ-ØØ-ØØØØ"
        self.fechaHastaNovedad <= "ØØ-ØØ-ØØØØ"
        self.tipoUsuario <= " "
        self.estado <= "A"
```



		Opción
"a"	"b"	"c"
Crear_nove()	modifi_nove()	eliminar_nove()

menu_4()

```
mostrar ("a) Crear novedades")
mostrar ("b) Modificar novedad")
mostrar ("c) Eliminar novedad")
mostrar ("e) Volver
```

Crear_nove()
Ask_show()
codigo_nove ← -1
While codigo_nove <> Ø
mostrar ("Ingrese el codigo de la novedad")
mostrar ("-si ingresa Ø vuelve al menú anterior")
Leer (codigo_nove)
hasta condigo_nove > -1
codigo_nove <> Ø
V
F
Pos ← buscar (codigo_nove, "code")
V
Pos == -1
F
repetidos()
texto ← ingreso_text()
desde ← get_fecha()
hasta desde >= datetime.datetime.today()
hasta ← get_fecha()
hasta hasta >= desde
tipo ← tipo_choice()
estado ← "A"
novedad ← NOVEDADES()
novedad.textoNovedad ← texto
novedad.codNovedad ← codigo_nove
novedad.fechaDesdeNovedad ← desde
novedad.fechaHastaNovedad ← hasta
novedad.tipoUsuario ← tipo
novedad.estado ← estado
pickle.dumb(novedad, archnove)
archnove.flush
ordenar_nove()

menu_modificar(aux)

```
while eleccion <> Ø
    mostrar ("dija la propiedad a modificar")
    mostrar ("1. fecha desde")
    mostrar ("2. fecha hasta")
    mostrar ("3. tipo usuario")
    mostrar ("Ø. terminar")
        leer (elección)
    hasta eleccion<="3" and elección >="Ø"
    V  deccion == "1"  F
        desde ← get_fecha()
        hasta desde >= (fecha ← hoy())
        aux.fechaDesdeNovedad ← desde
        V  eleccion == "2"  F
            until ← get-fecha()
            hasta until > aux.fechaDesdeNovedad
            aux.fechaHastaNovedad ← until
            V  eleccion == "3"  F
                tipo ← tipo_choice()
                aux.tipoUsuario ← tipo
```

is_int(x)

```
try:
    x ← int(x)
    return true
except:
    return false
```

Eliminar_nove( )

os.path.getsizes(archnove) — V / F

ask_show( )

code ← -1

while code != Ø

mostrar ("Ingrese el codigo de promocion a eliminar (ingresando Ø vuelve al menu)

leer(code)

while is_int(code)

leer (code)

Code != Ø — V / F

Pos ← buscar (code)

Pos != -1 — V / F

archnove.seek(pos, Ø)

aux ← NOVEDAD ( )

aux ← pickle.load(archnove)

aux.estado != "B" — V / F

mostrar ("(S) confirmar (N) cancelar")

leer (answer)

hasta answer == "S" or answer == "N"

answer != "S" — V / F

aux.estado ← "B"

archnove.seek (pos, Ø)

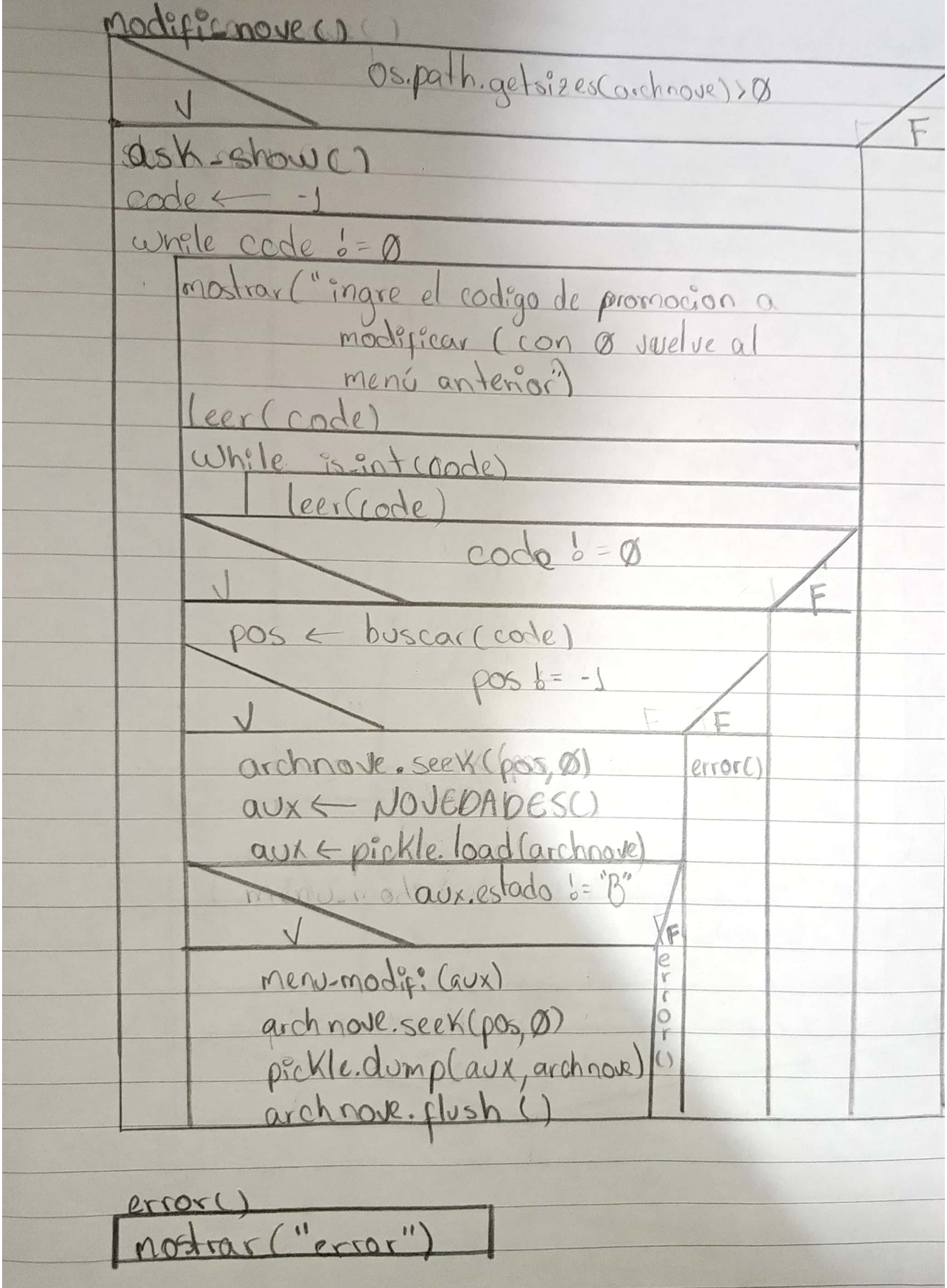
pickle.dump(aux, archnove)

archnove.flush( )

mostrar ("no existe")

```
Show_nove()
        os.path.getsizes(archnove) > 0
        V                                   F
tam ← os.path.getsizes(arch nove)
aux ← NOVEDADES()
archnove.seek(0,0)
mostrar("Codigo de novedad      ")
mostrar("desde                  ")
mostrar("hasta                  ")
mostrar("tipo de usuario        ")
mostrar("estado                 ")
While archnove.tell() < tam
    aux ← pickle.load(archnove)
    mostrar(aux.codNovedad,"    ")
    mostrar(aux.fechaDesdeNovedad,"    ")
    mostrar(aux.fechaHastaNovedad,"    ")
    mostrar(aux.tipoUsuario,"    ")
    mostrar(aux.estado)
mostrar("A = Activo   B = dado de baja")
```

```
ask_show()
mostrar("desea ver las novedades? (S/N)")
    leer(answer)
    answer ← answer.lower()
hasta answer == "s" or answer == "n"
        answer == "s"
        V                   F
show_nove()
```



error()

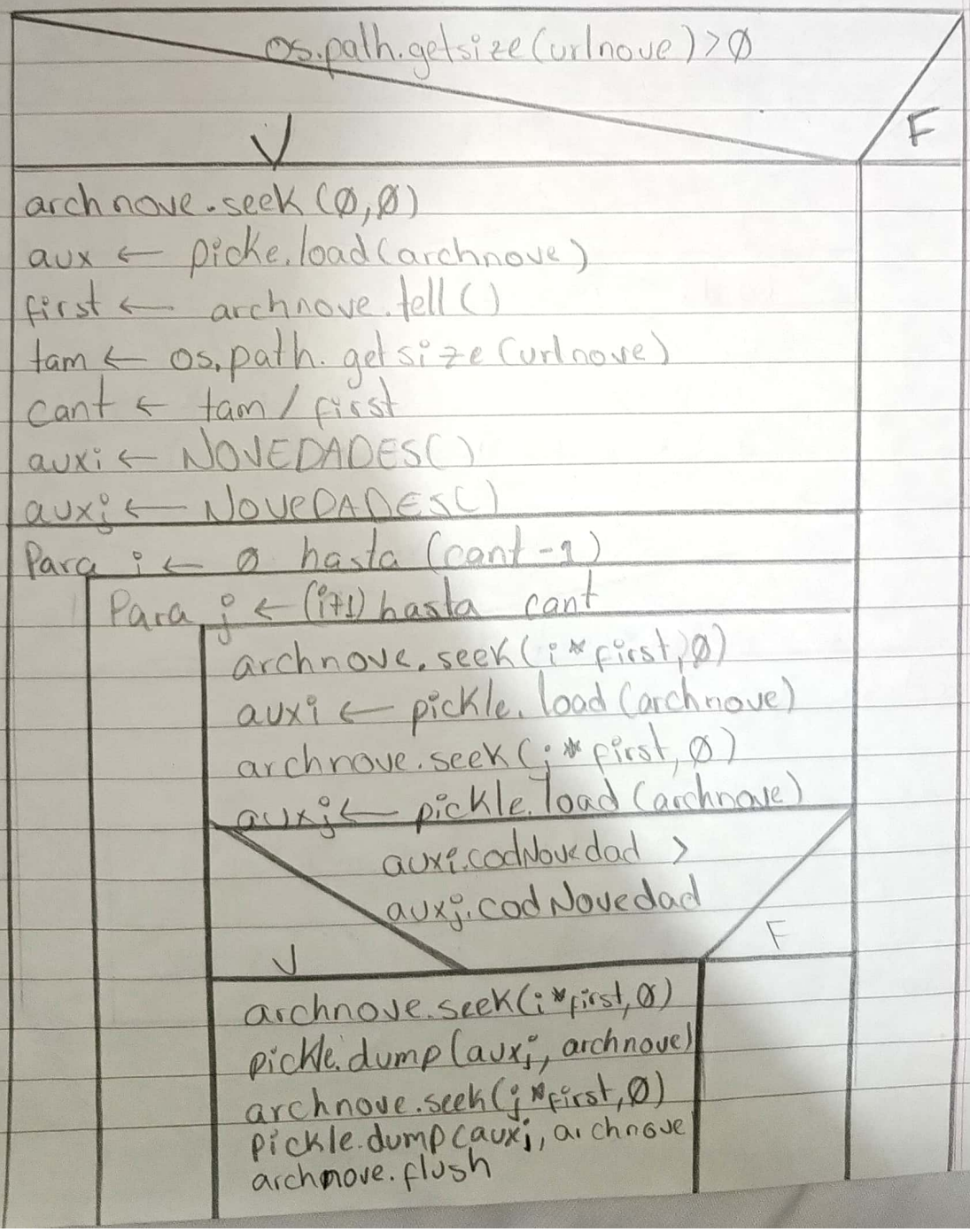
mostrar("error")

```
buscar(x)
archnove.seek(0,0)
aux ← NOVEDADES()
aux ← pickle.load(archnove)
tam ← os.path.getsize(urlnove)
first ← archnove.tell()
cantidad ← int(tam/first)
desde←0
hasta ← cantidad - 1
medio ← (desde + hasta) // 2
archnove.seek(first*medio, 0)
aux ← pickle.load(archnove)
While aux.codnovedad != x and desde < hasta
        x >= aux.codnovedad
        V                    F
    hasta ← hasta-1          desde ← desde+1
    medio ← (hasta + desde) // 2
    archnove.seek(medio * first, 0)
    aux ← pickle.load(archnove)
        aux.codnovedad == x
        V                F
    Pos ← medio*first    pos ← -1
return pos
```

```
Ingreso_texto()
mostrar("Ingrese el texto de la novedad")
    leer(text)
hasta len(text)>10 an len(text)<200
return text
```

```
tipo_choice()
mostrar("Elija el tipo de usuario")
mostrar("Dueño (D) o Cliente (C)")
    Leer(tipo)
    tipo.upper()
hasta tipo=="C" or tipo=="D"
return tipo
```

ordenar_nove()



Ver-Novedades()

V — os.path.getsizes(archnove) — F

aux ← NOVEDADES,

fecha ← Hoy()

global user #en nuestro programa al inicia #sesión se guarda el usuario en una variable

archnove.seek(0,0)

tam ← os.path.getsizes(archnove)

while archnove.tell() < tam

aux ← pickle.load(archnove)

aux.tipoUsuario == user.tipoUsuario and aux.estado != "B" and aux.fechaDesdeNovedad < fecha < aux.fechaHastaNovedad

V / F

mostrar(aux.codNovedad)

mostrar("desde", aux.fechaDesdeNovedad)

mostrar("hasta", aux.fechaHastaNovedad)

mostrar("Estado", aux.estado)