**Pseudocódigos**

***Ejercicio 01***

***Ejercicio 02***

***Ejercicio 03***

Registro dato

Real valor

Real peso

Fin \_ Registro

Algoritmo \_ Promedio \_ Ponderado

Dato v [100]

Entero n

Numdato (n)

Ingreso (v, n)

Reportedato (v. n)

Escribir “El promedio ponderado es: “, promedioponderado (v, n)

Fin \_ Algoritmo

Numdato (entero n (R))

Hacer

Escribir “Numero de datos: “

Leer n

Mientras (n<=0)

Fin \_ numdato

Ingreso (dato v [], entero n)

Entero i

Para (i🡨0 hasta n-1 inc 1) hacer

Escribir “Dato”, i+1, “:”

Escribir “valor [“, i+1, “]: “

Leer v[i].valor

Escribir “pes [“, i+1, “]: “

Leer v[i].peso

Fin \_ Para

Fin \_ ingreso

Reportedato (d ato v [], entero n)

Entero i

Escribir “datos ingresados”

Para (i🡨0 hasta n-1 inc 1) hacer

Escribir v[i].valor, “, “, v [i].peso

Fin \_ Para

Fin \_ reportedato

Promedioponderado (dato v [], entero n)

Entero i

Escribir “datos ingresados”

Para (i🡨0 hasta n-1 inc 1) hacer

S1🡨 s1 + v [i].valor \* v[i].peso

S2🡨s2+v[i].peso

Fin \_ Para

Fin \_ promedioponderado

***Ejercicio 04***

Algoritmo \_ Menú \_Empleados

Entero op, n

Empleado v [100]

numEmpleados (n)

ingresoEmpleados (v, n)

Hacer

Escribir “Menu”

Escribir “[1] Consultar empleado”

Escribir “[2] Eliminar empleado”

Escribir “[3] Ordenar por salario descendentemente”

Escribir “[4] Mostrar empleados”

Escribir “[5] Mostrar empleados”

Escribir “[6] Salir”

Escribir “Ingrese opción (1-6): “

Leer op

Según \_ Sea

Caso 1: consultarEmpleado (v, n)

salir

Caso 2: eliminarEmpleado (v, n)

salir

Caso 3: ordenarPorSalario (v, n)

Salir

Escribir “Datos ordenador por salario”

reporteEmpleados (v, n)

salir

case 4:

reporteEmpleados (v, n)

salir

caso 5:

Escribir “el promedio de sueldos es: “, promedioSalarios (v, n)

Fin \_ Según \_sea

Mientras (op<>6)

Fin \_ Algoritmo

numEmpleados (entero n (R))

hacer

escribir “Numero de empleados : “

leer n

Mientras (n<=0)

Fin \_ numEmpleados

ingresoEmpleados (Empleado v[], entero n)

entero i

para (i🡨0 hasta n-1 inc 1) hacer

Escribir “Datos del empleado”, i+1

Escribir “código: “

Leer v[i].c odigo

Escribir “Apellidos: “

cin.getline (v[i].apellidos, 40)

Escribir “Nombres: “

cin.getline (v[i].Nombres, 40)

Escribir “telefono: “

cin.getline (v[i].telefono, 40)

Escribir “Sueldo: “

cin.getline (v[i].salario, 40)

Fin \_ Para

Fin \_ ingresoEmpleados

reporteEmpleados (Empleado v[], entero n)

entero i

escribir “Lista de empleados”

para (i🡨0 hasta n-1 inc 1) hacer

escribir v[i].codigo, “,”, v[i].apellidos, “,”, v[i].nombres, “,”, v[i].telefono, “,”, v[i].salario

Fin \_ para

Fin \_ reporteEmpleados

buscarCodigo (empleado v[], entero n, carácter código [])

entero i

para (i🡨o hasta n-1 inc 1) hacer

si (v[i].código = código) entonces

retornar i

Fin \_ Si

Fin \_ Para

Fin \_ buscarCodigo

ConsultarEmpleado (Empleado v [], entero n)

Entero p

Carácter codigo [10]

Escribir “Ingrese codigo: “

Leer codigo

P = buscarCodigo (v, n, codigo)

Si (p<>1) entonces

Escribir “Datos del empleado : “

Escribir “Apellidos: “, v[p].apellidos

Escribir “Nombres: “, v[p].nombres

Escribir “Telefono: “, v[p].telefono

Escribir “Sueldo: “, v[p].salario

Sino

Escribir “Codigo no se encuentra”

Fin \_ Si

Fin \_ Para

eliminarEMpleado (Empleado v[], entero n (R))

Entero p

Carácter codigo [10]

Escribir “Ingrese codigo: “

Leer codigo

P = buscarCodigo (v, n, codigo)

Si (p<>1) entonces

Para (entero i🡨0 hasta n-1 inc 1) hacer

V[i]🡨 v[i+1]

Fin \_ Para

N🡨n-1

Escribir “Dato eliminado”

Sino

Escribir “El código no se encuentra”

Fin \_ Si

Fin \_ eliminarEmpleado

ordenarPorSalario (Empleado v[], entero n)

entero i, j

Empleado temp

Para (i🡨0 hasta n-1 inc 1) hacer

Para (j🡨i+1 hasta n-1 inc j) hacer

Si ((v[i].salario<v[i].salario)) entonces

Temp<- v[i]

V[i]🡨v[j]

V[j]🡨temp

Fin \_ Si

Fin \_ Para

Fin \_ Para

Fin \_ ordenarPorSalario (Empleado v[], entero n)

Entero i

Real s🡨0

Para (i🡨0 hasta n-1 inc 1) hacer

S🡨 s+ v[i].salario

Fin \_ Para

Retornar as/n

Fin \_ ordenarPorSalario