

## INDICE

1. INTRODUCCION	PÁG. 1
2. PSEUDOCÓDIGO	PÁGS. 2 Y 3
3. DIAGRAMAS:	
3.1.    ORDINOGRAMAS	PÁGS. 4-7
3.2.    DIAGRAMA ARBORESCENTE	PÁG. 8
3.3.    DIAGRAMA NS-CHAPIN	PÁG. 9

---

### **1. INTRODUCCION**

Se pide diseñar un programa para llevar un control horario de una empresa de 5 trabajadores.

Este programa lo he llamado "Fichador". Permitirá introducir las horas de entrada y salida de cada trabajador a lo largo de los días. Además, permitirá sacar un informe de horas totales de cualquiera de los trabajadores.

El programa está compuesto del proceso principal y tres subprocesos, que se corresponden con las opciones del menú.

En la cabecera del proceso principal se declaran las variables tipo arreglo y se inicializa todas las variables. Seguidamente se muestra el menú y se pide elegir entre las distintas opciones. No permite continuar si se elige otra opción. Se sale del programa eligiendo la opción 0.

Los tres subprocesos ejecutan las tres opciones principales del menú. Se establecen los controles oportunos para evitar datos no válidos: que el código de empleado corresponda a uno de los definidos, que las horas y los minutos sean válidos, que no se permita fichar una salida sin fichar entrada, que la hora de salida sea mayor que la de entrada.

He llevado una serie de pruebas siguiendo todos los posibles caminos del programa e introduciendo distintos tipos de datos de entrada para comprobar que las validaciones funcionan correctamente.

Se incluyen tres diagramas distintos:

- los ordinogramas de todo el programa, realizados con DIA.
- los diagramas arborescentes y de NS-Chapin del proceso principal, realizados con Ms PowerPoint.

Se adjunta código realizado en PSeInt.

## 2. PSEUDOCÓDIGO

```
// Programa Fichador para empresa de 5 empleados
Algoritmo AA1
    // Declaración de las variables principales
    Dimension nombre[5]
    Dimension hora_entrada[5]
    Dimension hora_salida[5]
    Dimension horas_totales[5]
    // Asignación de valores iniciales
    Para i<-1 Hasta 5 Hacer
        hora_entrada[i]<- 0
        hora_salida[i]<- 0
        horas_totales[i] <- 0
    FinPara
    nombre[1] <- 'Juan'
    nombre[2] <- 'Lorena'
    nombre[3] <- 'Pepe'
    nombre[4] <- 'Eva'
    nombre[5] <- 'Mari'
    salir <- Falso

    // Ejecución del programa
    Repetir
        // Mostrar menu
        Escribir ' '
        Escribir 'SISTEMA DE ALQUILER DE COCHES CON CONDUCTOR - MENÚ DE USUARIO'
        Escribir '1. Fichar entrada'
        Escribir '2. Fichar salida'
        Escribir '3. Ver horas totales de un trabajador'
        Escribir '0. Finalizar el programa'
        Escribir 'Introduzca opción 1, 2, 3 (0 para finalizar el programa):'
        Leer opcion_menu
        Segun opcion_menu Hacer
            1:
                fichar_entrada(nombre,hora_entrada)
            2:
                fichar_salida(nombre,hora_entrada,hora_salida,horas_totales)
            3:
                totales(nombre,horas_totales)
            0:
                salir <- Verdadero
        De Otro Modo:
            Escribir 'Opción no válida'
        FinSegun
    Hasta Que salir
FinAlgoritmo

// opción 1 del menu: registrar horas de entrada
SubProceso fichar_entrada (nombre,hora_entrada)
    Escribir 'Fichar entrada'
    Repetir
        Escribir 'Introduce código del empleado)'
        Leer numemple
    Hasta Que (numemple>0 Y numemple<=5)
    Repetir
        Escribir 'Introduce hora de entrada (hh:mm) de ',nombre[numemple]
```

FUNDAMENTOS DE PROGRAMACIÓN  
Actividad de aprendizaje (AA1). Práctica de las unidades 1 y 2  
JUAN DEL RIO HUERTAS

---

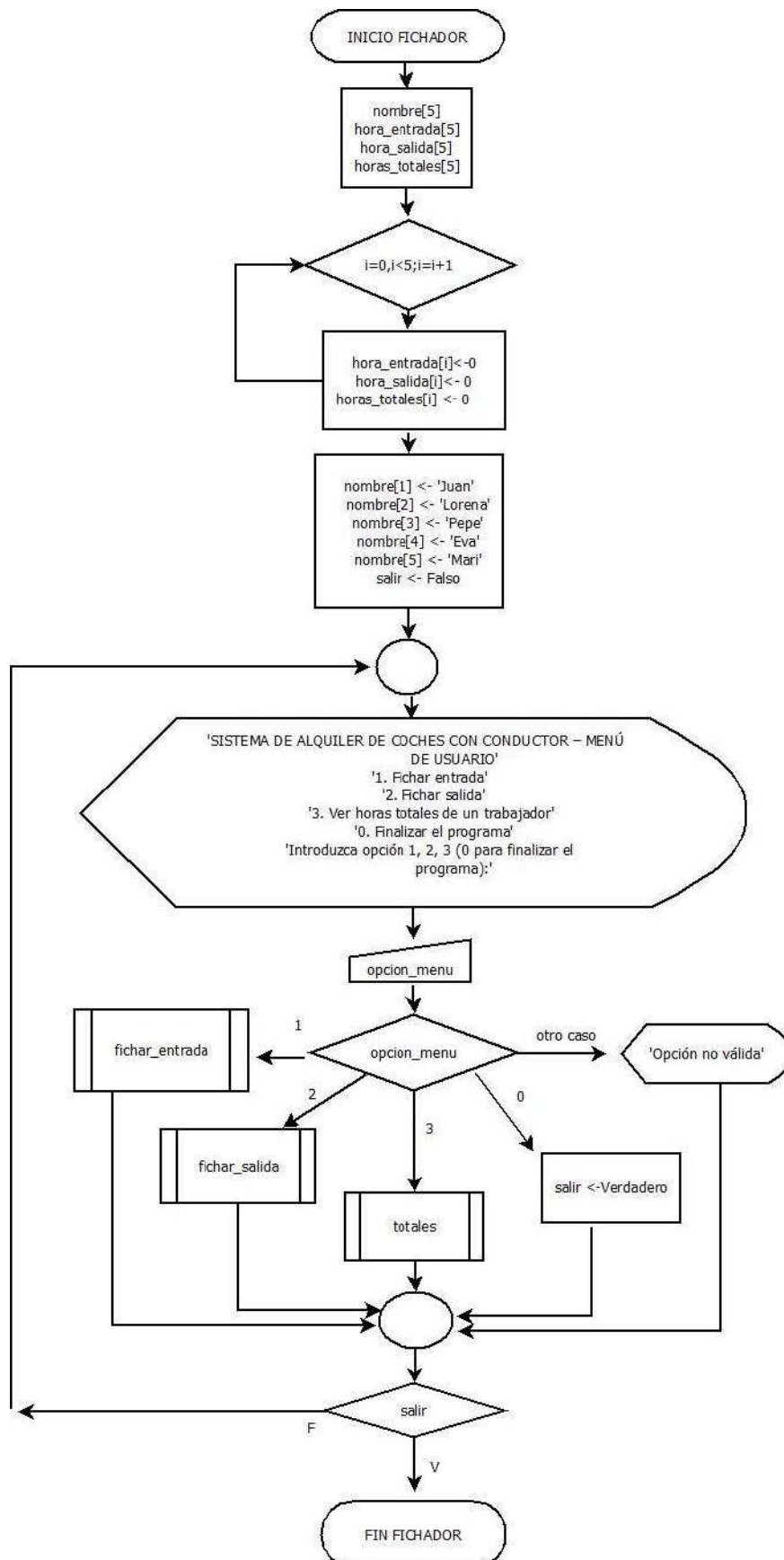
```
    Leer hh
Hasta Que (hh>=0 Y hh<24)
Repetir
    Escribir 'Introduce minutos de entrada (hh:mm) de ',nombre[numemple]
    Leer mm
Hasta Que (mm>=0 Y mm<60)
hora_entrada[numemple] <- hh+mm/60

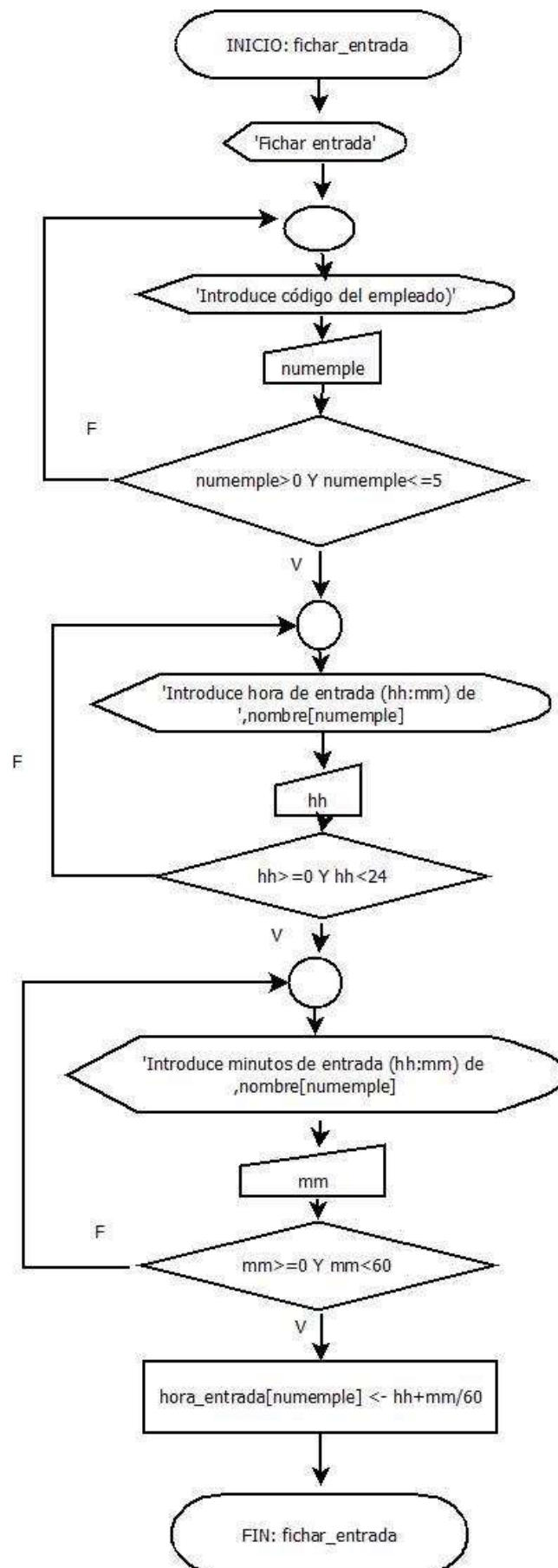
FinSubProceso

// opción 2 del menu: registrar horas de salida
SubProceso fichar_salida (nombre,hora_entrada,hora_salida,horas_totales)
    Escribir 'Fichar salida'
    Repetir
        Escribir 'Introduce código del empleado)'
        Leer numemple
Hasta Que (numemple>0 Y numemple<=5)
Si (hora_entrada[numemple]>0) Entonces // comprobar que ha fichado a la entrada
    Repetir // comprobar que no viaja al pasado
        Repetir // validación de la hora
            Escribir 'Introduce hora de salida (hh:mm) de ',nombre[numemple]
            Leer hh
Hasta Que (hh>=0 Y hh<24)
        Repetir // validación de los minutos
            Escribir 'Introduce minutos de salida (hh:mm) de ',nombre[numemple]
            Leer mm
Hasta Que (mm>=0 Y mm<60)
        Si (hh+mm/60)>=hora_entrada[numemple] Entonces
            hora_salida[numemple] <- hh+mm/60
        SiNo
            Escribir 'Error: hora de salida es menor a la hora de entrada.
Inténtalo de nuevo.'
        FinSi
Hasta Que (hora_salida[numemple]>=hora_entrada[numemple])
    Escribir nombre[numemple], ' ha trabajado hoy ',hora_salida[numemple]-
hora_entrada[numemple], ' horas.'
    horas_totales[numemple] <- horas_totales[numemple]+hora_salida[numemple]-
hora_entrada[numemple]
    // reseteo valores
    hora_salida[numemple] <- 0
    hora_entrada[numemple] <- 0
SiNo
    Escribir 'Introduce primero la hora de entrada'
FinSi
FinSubProceso

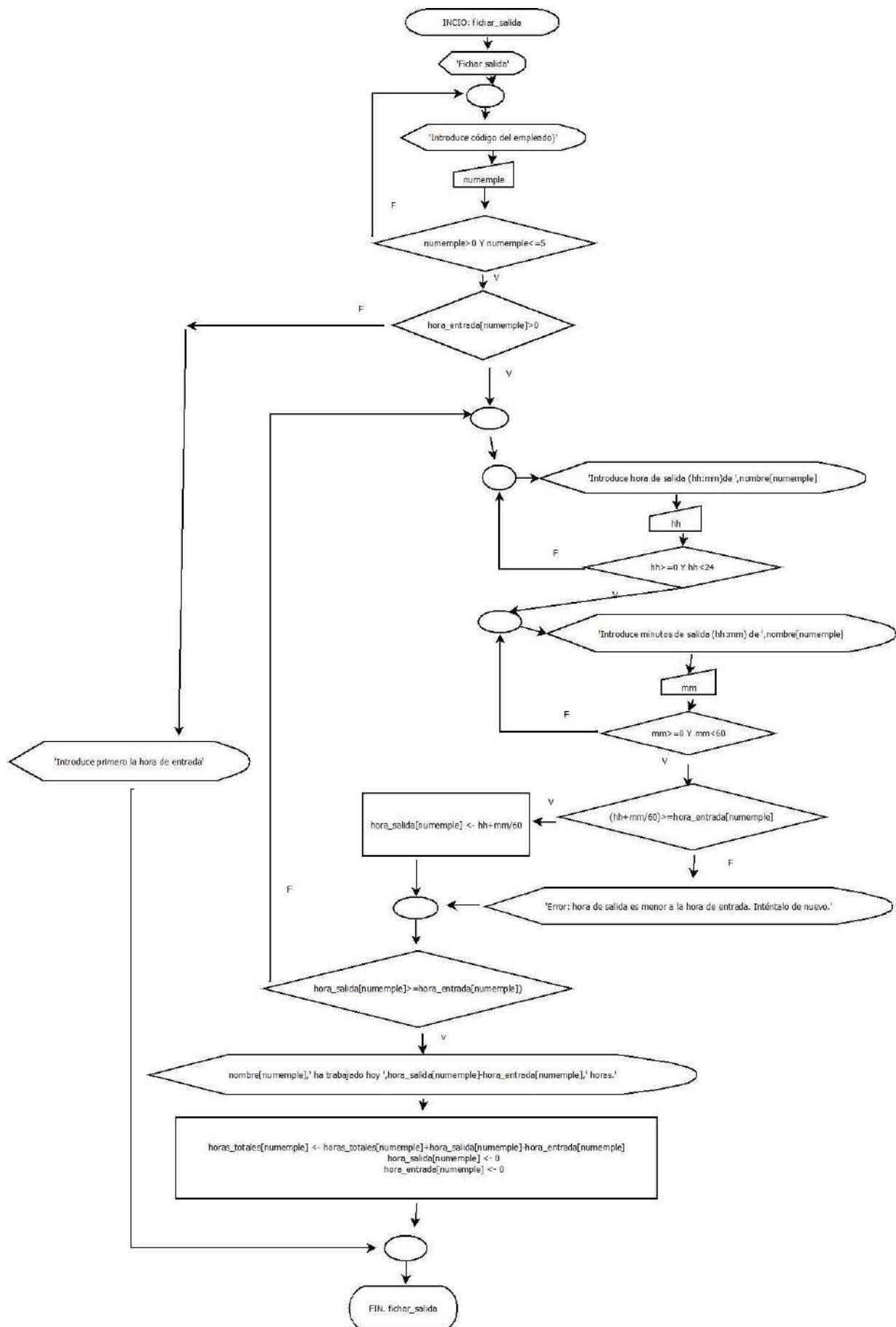
// opción 3 del menu: mostrar total de horas trabajadas
SubProceso totales(nombre,horas_totales)
    Escribir 'Ver horas totales de un trabajador'
    Repetir
        Escribir 'Introduce código del empleado)'
        Leer numemple
Hasta Que (numemple>0 Y numemple<=5)
    Escribir nombre[numemple], ' ha trabajado un total de ',horas_totales[numemple], '
horas'
FinSubProceso
```

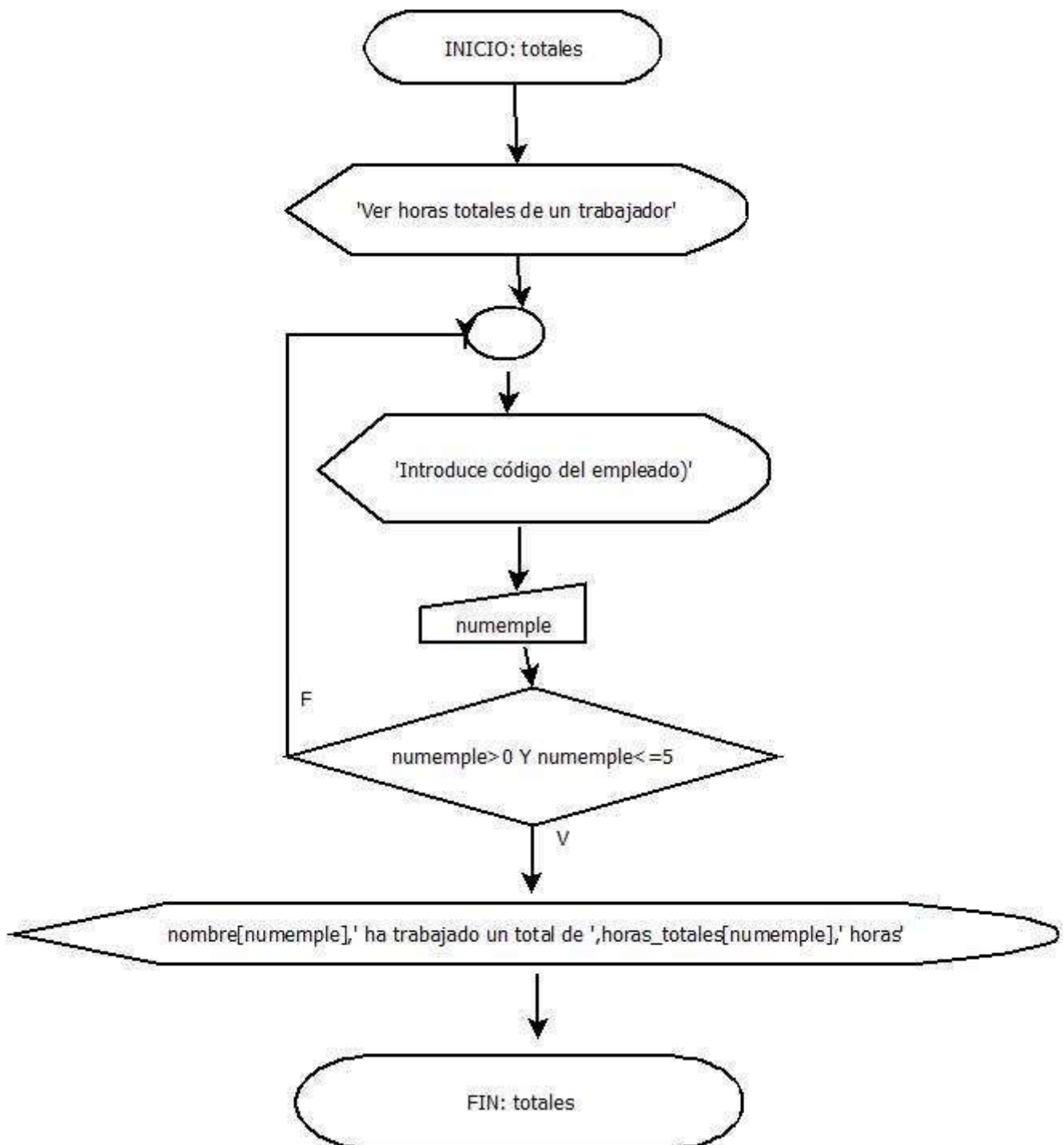
### 3.1. ORDINOGRAMAS



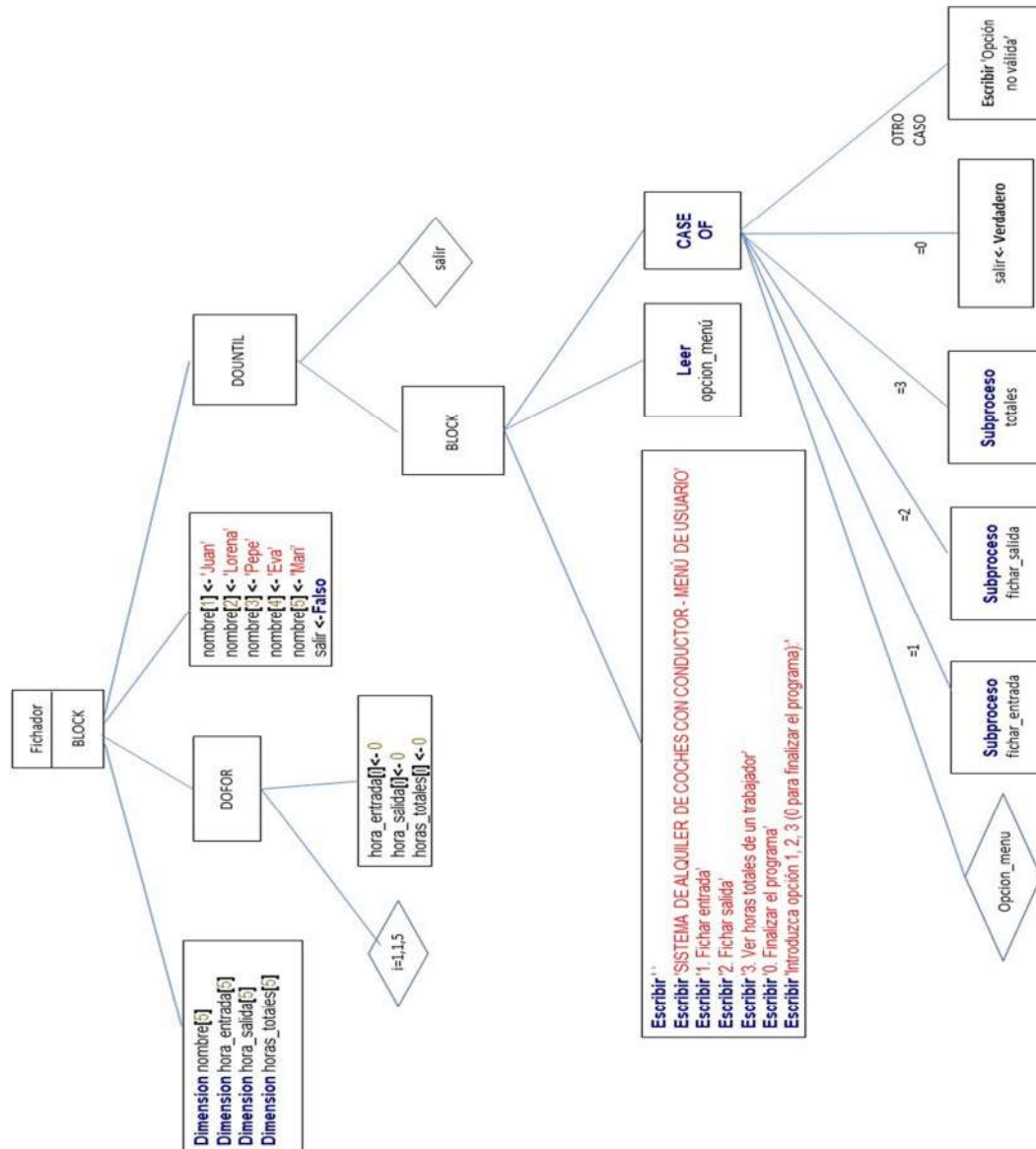


FUNDAMENTOS DE PROGRAMACIÓN  
Actividad de aprendizaje (AA1). Práctica de las unidades 1 y 2  
JUAN DEL RIO HUERTAS





### 3.2. DIAGRAMA ARBORESCENTE DEL PROCESO PRINCIPAL





3.3. DIAGRAMA NS-CHAPIN DEL PROCESO PRINCIPAL

