



# DISEÑO DE ALGORITMOS

## 3ª Relación de ejercicios (Temas 4 y 5)

1. Terminar el programa que implementa un array dinámico de estructuras en el tema 4. Añadir al programa el código necesario para ampliar el vector de personas, preguntando al usuario cuántas personas más quiere, ampliar el tamaño del vector y leer los datos de las nuevas personas.

Modificar el programa para que, además del nombre y la edad, se tenga también almacenada información de la edad de todos los hijos que tenga cada persona. Cada persona puede tener un número diferente de hijos por lo que también se almacenarán en un array dinámico de enteros. Cuando el programa pida los datos de cada persona preguntará el nombre, edad de la persona y edad de todos sus hijos.

2. Se propone simular mediante una lista enlazada un servidor de gestión de impresión de una impresora conectada en red. Para cada impresión que se vaya a realizar, habrá que crear un nodo en la lista. Para ello, queremos ofrecer un menú con las siguientes opciones:
  - a) Crear impresión: en esta opción hay que introducir el nombre del archivo que se va a imprimir, dirección IP de la máquina que imprime y nivel de prioridad (0,1 o 2). Siempre el elemento insertado se situará en última posición de su nivel, sabiendo que el nivel 2 es el más urgente y número de copias.
  - b) Crear impresión urgente: esta opción es igual a la anterior pero inserta el mensaje en la primera posición del nivel correspondiente.
  - c) Realizar impresión: esta opción elimina el primer archivo de la lista.
  - d) Listar archivos: mostrar una lista con todos los archivos, así como cuantos archivos existen de cada nivel.
  - e) Contar trabajos: nos informa cuantos trabajos en cola tiene una dirección IP.
  - f) Cambiar categoría archivo: esta opción pide el nombre de un archivo y lo cambia de nivel.
  - g) Borrar archivo; esta opción pide el nombre de un archivo y su IP y la elimina.
  - h) Incrementa/Decrementa nº copias: esta opción pide el nombre del archivo y la IP y modifica el número de copias.
  - i) Eliminar todo: esta opción elimina todos los trabajos de impresión.
3. Codificar un programa que simule el funcionamiento de una oficina. En dicha oficina hay un número de administrativos desconocido y variable que se encarga de llevar a cabo las tareas que surgen. Dichas tareas se acumulan en las mesas de cada empleado, y éstos las van realizando por orden de llegada. Para controlar la simulación, deberá implementarse un menú como el siguiente:

- |   |
|---|
| <ol style="list-style-type: none"><li>A. Contratar empleado.</li><li>B. Introducir tarea I</li><li>C. Introducir tarea II</li><li>D. Cambiar tarea.</li><li>E. Avanzar trabajo.</li><li>F. Mostrar estado.</li><li>G. Intercambio</li><li>H. Fichero</li><li>I. Salir</li></ol> <p>Introduzca opción:</p> |
|---|



## DISEÑO DE ALGORITMOS

### 3ª Relación de ejercicios (Temas 4 y 5)

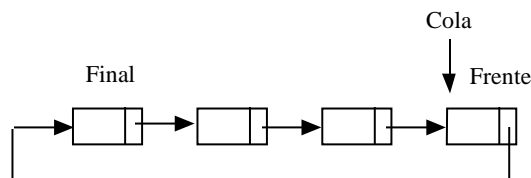
- A. Contratar empleado: al seleccionar esta opción el programa debe pedir al usuario que introduzca la información del nuevo empleado (Nombre y Edad). Tras la contratación el empleado dispondrá de una mesa en la cual se acumulará el trabajo. Al comienzo, no habrá ningún empleado contratado.
- B. Introducir tarea I: al seleccionar esta opción el programa debe pedir al usuario que introduzca la tarea, el tiempo estimado y el nombre del empleado al que se le asignará dicha tarea. Si dicho empleado no existe debe indicar un mensaje de Error: "No hay empleados".
- C. Introducir tarea II: al seleccionar esta opción el programa debe pedir al usuario que introduzca el nombre de una tarea y su duración estimada. Dicha tarea se asignará al empleado que tenga menos trabajo pendiente (en total de tiempo, no en número de tareas), mostrándose en pantalla a quien se le asignó la tarea introducida. Si no hay ningún empleado, aparecerá el mensaje Error: "No hay empleados".
- D. Cambiar tarea: Al seleccionar esta opción el programa debe pedir al usuario que introduzca el nombre del empleado al cual le queremos ayudar en alguna tarea, una vez elegida la tarea que se le va a quitar, se le asigna a otro empleado, el cual también ha sido solicitado por teclado.
- E. Avanzar trabajo: Al seleccionar esta opción el simulador hará que cada empleado realice la primera tarea que tenga pendiente, mostrando por pantalla cada empleado que tarea ha realizado.
- F. Mostrar estado: Al seleccionar esta opción el programa debe mostrar en pantalla el estado de la oficina, indicando las tareas que cada empleado tiene pendiente. En caso de que un empleado no tenga tareas pendientes, aparecerá el mensaje ocioso.
- G. Intercambiar: Al seleccionar esta opción aparecerá una lista de los empleados. Posteriormente se deben indicar dos empleados, los cuáles van a intercambiar todas sus tareas.
- H. Fichero: Esta opción crea un fichero EMPLEADO.DAT con toda la información de los empleados contratados. Para cada empleado se almacena también el número de tareas que tienen pendientes, así como su duración total.

Deberán definirse los TAD's Empleado, Tarea, ColaTareas, ListaEmpleados. Si no utilizas memoria dinámica, suponemos un máximo de 20 empleados, y en cada mesa un máximo de 10 tareas.

4. Supongamos que una Cola va a representarse mediante una estructura dinámica circular utilizando las siguientes declaraciones:

```
typedef struct nodo
{
    TipoElem valor;
    struct nodo *sig;
} tiponodo;

typedef tiponodo *TipoCola;
```



Cola apunta al dato del Frente de la misma, y Cola->sig apunta al Final.

Con esta representación implementa todas las operaciones del TAD Cola. (CrearCola, DestruirCola, ColaVacía, InsertarCola, SacarCola)

5. Implementar un menú con todas las opciones de una pila dinámica.

**Mínimo: 3 ejercicios 1, (2 o 3) y (4 o 5).**

**Debe entregarse a la tarea abierta a tal efecto en el tema 5.**