

# Curso 2005/2006. Estructura de Datos y de la Información

## I. Informática, I. T. Informática de Gestión y de Sistemas

### PRÁCTICA 2

## 1. Enunciado

La compañía discográfica EDIrecords desea instalar un sistema gratuito de descargas de sus canciones vía Internet. Para ello nos encargan un programa informático que gestione automáticamente las fuentes descargables y las descargas diarias. Después de distintas reuniones con los expertos de la empresa EDIrecords, podemos resumir las reglas del sistema de descargas como sigue:

- Cada semana se actualiza el servidor de descargas con las canciones disponibles.
- Cada canción (o fuente) estará dividida en varias partes.
- Ya que las descargas son gratuitas y legales, para cada fuente se podrán realizar un número máximo de descargas al día. Efectuadas este máximo de descargas, la fuente (y por lo tanto, todas las partes) se eliminan del servidor.
- Cada usuario podrá solicitar la descarga de una o más canciones.
- Las peticiones de los usuarios serán atendidas en riguroso orden de llegada.
- Supondremos una tasa de descarga fija de 10MB/s.
- Cuando un usuario termina de descargarse una parte de una fuente, automáticamente se le sitúa en la cola de descarga de la parte siguiente, si es que ésta existe.

## 2. Implementación

Para resolver el problema será necesario implementar:

1. Una **lista ordenada** para mantener las fuentes (divididas en partes) que se pueden descargar.
2. Varias **colas de descarga**, una por cada parte de cada fuente, para los usuarios que solicitan su descarga.

La Figura 1 muestra un ejemplo de estas estructuras y sus interrelaciones.

Por lo tanto, como parte de la práctica, se deberá implementar utilizando **variables dinámicas** las siguientes estructuras:

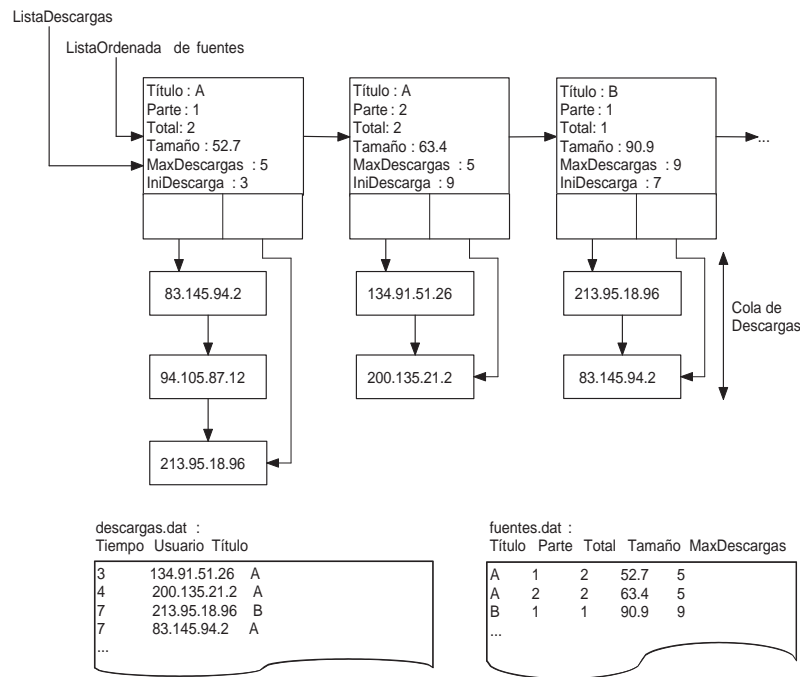


Figura 1: Lista de fuentes y colas de descarga para cada una de ellas.

- **Unit cola.pas:** implementación del TAD Cola mediante punteros. Considerar los tipos siguientes:
  - **tInfoCola** → tipo de los datos almacenados en los nodos de las colas de descargas. Está compuesto por:
    - *tIdUsuario* o identificador de la persona que solicita la descarga (string[15])
  - **tPosCola** → tipo de los punteros a los nodos de la cola de descarga
  - **tNodoCola** → tipo de los nodos de la cola de descarga
  - **tCola** → tipo de la cola de descarga

En esta unit se implementarán las cinco operaciones permitidas en un TAD Cola:

- **ColaVacía (tCola)** → **tCola**: Crea una Cola vacía.
- **EsColaVacía (tCola)** → **Boolean**: Determina si la Cola está vacía.
- **InsertarEnCola (tCola, tInfoCola)** → **tCola**: Inserta un elemento en la Cola. *Precondición:* Se supone memoria suficiente.
- **Frente(tCola)** → **tInfoCola, boolean**: Devuelve el contenido del elemento más antiguo de la Cola y un valor verdadero/falso indicando si ha podido extraer dicho elemento.
- **EliminarDeCola(tCola)** → **tCola, boolean**: Elimina el elemento que está en el frente de la cola y un valor verdadero/falso indicando si ha podido eliminar dicho elemento.

Además de las indicadas, una *precondición* común para las cuatro últimas operaciones es que la cola debe estar inicializada.

- **Unit listaOrdenada.pas:** implementación del TAD `ListaOrdenada` mediante punteros. Además, hace uso de la unit de colas para mantener las descargas pendientes para cada fuente. Considerar los tipos siguientes:
  - **tInfoLista:** tipo de los datos almacenados en los nodos de la lista de fuentes. Está compuesto por:
    - *tTitulo*: identificador de la fuente (`string[30]`)
    - *tParte*: número de parte de la fuente (entero)
    - *tTotal*: número de partes en que está dividida la fuente (entero)
    - *tTamano*: tamaño en MB que ocupa esta parte de la fuente (entero)
    - *tMaxDescargas*: número máximo de descargas que se pueden realizar sobre ese fuente (entero)
    - *tIniDescarga*: tiempo en que se inició la descarga actual (entero). Su valor es 0 si ningún usuario se encuentra descargando la fuente.
    - *tCola*: tipo del campo que permite acceder a la cola de descargas pendientes para una parte de una fuente (definido en la `unit cola.pas`).
  - **tPosLista:** tipo de los punteros a los nodos de la lista.
  - **tNodoLista:** tipo de los nodos de la lista.
  - **tListaOrdenada:** tipo de la lista de fuentes descargables.

En esta unit se implementarán las siguientes operaciones:

- `listaVacía (tListaOrdenada) → tListaOrdenada`: Crea una lista vacía
- `esListaVacía (tListaOrdenada) → boolean`: Determina si la lista está vacía
- `primero (tListaOrdenada) → tPosLista`: Devuelve la posición del primer nodo de la lista
- `siguiente (tListaOrdenada, tPos) → tPosLista`: Devuelve la posición del siguiente nodo en la lista, o nulo si tPos es el final de la lista.
- `ultimo (tListaOrdenada) → tPosLista`: Devuelve la posición del ultimo nodo de la lista
- `insertarOrdenado (tListaOrdenada, tInfoLista) → tListaOrdenada`: Inserta ordenadamente en la lista un elemento con contenido tInfoLista. El orden se hará por el campo tTitulo y, para elementos con el mismo título, se hará por tParte. *Precondiciones*: no puede existir en la lista un elemento con el mismo tTitulo y tParte. Se supone memoria suficiente para realizar la inserción.
- `borrar (tListaOrdenada, tPosLista) → tListaOrdenada`: Borra de la lista el nodo que está en la posición indicada.
- `obtenerDato (tListaOrdenada, tPosLista) → tInfoLista`: Devuelve la información del elemento situado en la posición indicada de la lista.
- `buscarDato (tListaOrdenada, tTitulo, tParte) → tPosLista`: Dado un título y una parte de la fuente devuelve la posición que ocupa en la lista de fuentes dicho elemento. La posición que devuelve será *nulo* si el elemento no existe.

- `actualizaDato(tListaOrdenada, tPosLista, tInfoLista)`  
 → `tListaOrdenada`: Actualiza la información guardada en el elemento que está en una posición de la lista con la nueva información introducida por parámetro.

Una *precondición* común a todas las funciones que reciben un parámetro de tipo `tPos` es que éste se corresponde con una posición válida en la lista.

- `Unit ListaDescargas.pas`: Utiliza las units anteriores para implementar un TAD Descargas, que mantiene la información sobre las fuentes disponibles y las descargas solicitadas y realiza, por tanto, las operaciones necesarias para su gestión. Esta unit incluirá el nuevo tipo siguiente:

`tListaDescargas` → tipo de la estructura de descargas

Implementar las siguientes operaciones, sabiendo que la precondición común para todas ellas, salvo `ListaDescargasVacía`, es que la estructura debe estar previamente inicializada:

- `ListaDescargasVacía` → `tListaDescargas`: Crea una lista de descargas vacía.
- `EsListaFuentesVacía (tListaDescargas)` → `boolean`: Determina si no hay fuentes descargables (y, por lo tanto, tampoco usuarios descargando).
- `HayDescargas (tListaDescargas)` → `boolean`: Determina si en la lista de descargas hay algún usuario descargando o esperando.
- `ImprimirListaDescargas (tListaDescargas)` →  $\emptyset$ : Imprime por pantalla el listado de fuentes disponibles, indicando para cada una de ellas si algún usuario se la está descargando y, si es así, el identificador de ese usuario (el del frente de la cola de descargas correspondiente). El formato será el siguiente:

```
Título1 (Parte x de z)
    Descargando <Usuario>
Título1 (Parte y de z)
    No tiene descargas pendientes
Título2...
```

- `InsertarFuente(tListaDescargas, tTitulo, tParte, tTotal, tTamano, tMaxDescargas)` → `tListaDescargas`: Inserta en la lista de descargas una nueva fuente para ser descargada. *PreCD*: Se supone supone memoria suficiente para realizar la operación. *Poscondiciones*: la cola de descargas correspondiente queda inicializada a vacía y el campo `tIniDescarga` a 0.
- `EliminarFuente(tListaDescargas, tTitulo, tParte)` → `tListaDescargas`: Elimina de la lista de descargas una parte de una fuente para impedir que sea descargada. Automáticamente se eliminan también los usuarios que estuviesen en la cola de descargas correspondiente. Se imprimirá un mensaje del tipo:  
 Eliminada fuente...TITULO(PARTE/TOTAL)  
 Descargas pendientes: <Usuario1><Usuario2>...  
 o bien,

Eliminada fuente...TITULO(PARTE/TOTAL)  
No tiene descargas pendientes.

- SolicitarDescarga (tListaDescargas, tTitulo, tIdUsuario, tIniDescarga) → tListaDescargas: Inserta un nuevo usuario tIdUsuario en la cola de descargas para la fuente tTitulo indicada. Si la fuente está dividida en varias partes, el usuario se sitúa en la cola de descargas de la primera de ellas. Además, si la descarga comienza inmediatamente, se almacena también el tiempo en que se inicia la descarga y se imprime el mensaje

Iniciando descarga...TITULO (PARTE/TOTAL) por USUARIO en TIEMPO

En caso contrario, se imprimirá un mensaje del tipo:

Intentando descarga...TITULO (PARTE/TOTAL) por USUARIO

Si no existe la fuente solicitada se imprime el mensaje:

Error en descarga TÍTULO(PARTE/TOTAL) por USUARIO.

*Precondición:* Se supone supone memoria suficiente para realizar la operación.

- ActualizarDescargas (tListaDescargas, tiempoActual, tasaDescarga) → tListaDescargas: Recorre la lista de descargas para actualizar el estado del sistema. Para cada descarga finalizada (se detectan de acuerdo a la tasaDescarga, el tiempoActual y el tiempo en que se inició la descarga):
  - Se imprime el mensaje:

Descarga finalizada...TITULO (PARTE/TOTAL) por USUARIO
  - Se elimina la petición del usuario si ha descargado todas las partes de la fuente, o
  - Se introduce la petición del usuario en la cola de descargas de la siguiente parte de la fuente.
  - Si para alguna fuente se ha alcanzado el número máximo de descargas, eliminar esta fuente del sistema.

- **Programa principal.pas.** El programa principal simulará el funcionamiento del proceso de descargas. Para ello se proporcionan dos ficheros de texto.

El fichero “fuentes.dat” contiene listado de las fuentes que se podrán descargar. Cada línea de este fichero se corresponde a una fuente y tiene la estructura mostrada en la Figura 1.

El fichero “descargas.dat” contiene las descargas que se irían solicitando a lo largo del tiempo. Cada línea de este fichero contiene el identificador de un usuario, el título de la fuente que solicita y el tiempo en que hace la solicitud. Dentro del fichero, las líneas se encuentran ordenadas por este último campo (ver Figura 1).

La secuencia de ejecución del programa será la siguiente:

1. Leer el fichero “fuentes.dat” utilizando su contenido para insertarlo en la *lista de fuentes*.

A partir de este momento, el programa comenzará la simulación de descargas a lo largo del tiempo.

2. Inicializar a 0 un contador temporal que simulará el avance del tiempo.
3. Mostramos la lista de fuentes disponibles y sus descargas pendientes.
4. Incrementar contador de tiempo y mostrar: `Tiempo actual = < Tiempo >`
5. Leer todas las líneas del fichero “descargas.dat” que tengan un tiempo de solicitud igual al tiempo actual e insertarlas en las colas de descarga correspondiente.
6. Actualiza el sistema de descargas eliminando usuarios que han finalizado descargas completas, moviéndolos hacia la descarga de la siguiente parte de una fuente o eliminando fuentes para las que se ha alcanzado el máximo número de descargas.
7. Repetir desde el paso 4 hasta que no haya solicitudes nuevas ni descargas pendientes.

## NOTAS IMPORTANTES

- A través de la Facultad Virtual se suministra:
  - Los ficheros “fuentes.dat” y “descargas.dat” para probar la práctica. Inicialmente, tendréis disponibles unos ficheros más sencillos. Una vez avanzada la práctica se dejarán unos ficheros más completos.
  - El código que indica cómo realizar la lectura de los ficheros “fuentes.dat” y “descargas.dat” (`lectura_ficheros.pas`).
  - También habrá un fichero “FAQ.txt” donde se irá comentando cualquier aclaración/corrección a la práctica.
- Se recuerda que para acceder a la información contenida en cada una de las colas de descarga se hará a través de las operaciones definidas en la *unit* correspondientes que figuran en su sección de *Interfaz*. Lo mismo ocurre para acceder a la parte de la estructura donde se almacenan las fuentes, que se manejará con las funciones del TAD `ListaOrdenada`. Finalmente, el programa principal accederá a la estructura completa de fuentes y descargas a través del TAD `ListaDescargas`.
- El desarrollo de una buena interfaz, aunque importante, **NO ES EL OBJETIVO** de esta asignatura. Por lo tanto, no se permitirá el uso de la UNIT CRT en la implementación de la práctica, ni variar los formatos de los mensajes especificados.

## 3. Normas de realización y entrega

- Serán las mismas que para la primera práctica.
- **Fecha límite de entrega: 9 de Junio de 2006.**
- La forma de entrega será la misma que para la primera práctica, siendo ahora los directorios de entrega:
  - /PRACTICAS/ETIX/EDI/P2/login (Alumnos de I.T.I. de Gestión)
  - /PRACTICAS/ETIS/EDI/P2/login (Alumnos de I.T.I. de Sistemas)
  - /PRACTICAS/EI/EDI/P2/login (Alumnos de I.I.)
- Todos los componentes del grupo deberán depositar la práctica en el directorio correspondiente a su login.