



Universidad
de Huelva

ALGORITMOS Y ESTRUCTURAS DE DATOS II

Ingeniería Técnica en Informática de Gestión

Ingeniería Técnica en Informática de Sistemas

CURSO 2002/03

Práctica 2

Diseño y construcción de un TAD basado en una
lista dinámica

Objetivos

- Afianzar los conocimientos adquiridos sobre listas dinámicas.

Enunciado

Una empresa multinacional necesita realizar un proceso de reestructuración y limpieza del fichero que mantiene la información de sus empleados. Dicho fichero fue creado, a lo largo de la vida de la empresa, sin aplicar ninguna estrategia, es decir, simplemente se añadían nuevas fichas de empleado cada vez que era necesario. Por tanto, siempre que un empleado cambiaba de departamento, ubicación, ciudad, categoría, etc., se añadía al fichero una nueva ficha con la información correspondiente (DNI, Código Empleado, Teléfono). Debido a la política de movilidad de la empresa, cada empleado tiene muchas fichas distintas en el fichero de datos.

Cada ficha almacena el D.N.I., un código y un teléfono departamental. El código representa toda la información relativa al puesto: categoría, departamento, ciudad, etc. La principal característica de este código es que, siempre que un empleado cambia de ubicación o puesto, se le asigna un código de valor superior al anterior. Por tanto, la empresa necesita compactar la información de la base de datos, para quedarse sólo con las dos fichas más recientes de cada empleado, que serán las que tengan los códigos de mayor valor.

Se pide

1. Especificar e implementar un TAD *TFicha* y un TAD *TListaF* en módulos (unidades) independientes. Los ficheros se guardarán con los nombres **tficha.pas** y **tlistaf.pas** respectivamente.
2. Un pequeño programa usuario, llamado **gestionF.pas**, que lea el fichero **datos.dat** y cree dos ficheros (**FP.dat** y **FS.dat**) a partir de las listas que contienen los datos más recientes de cada empleado, las cuales se construyen mediante la operación de *inserción* del TAD *TListaF*. Debido a que puede haber errores en los datos del fichero, cada ficha leída deberá ser validada y creada mediante la operación *crearFicha* del TAD *TFicha*.

Descripción de los TAD

TAD *TFicha*

El TAD *TFicha* está formado por el tipo **Ficha**, implementado mediante un registro de 3 campos de tipo entero largo, entero y entero largo, con las operaciones siguientes:

función *crearFicha*(dni: entero largo; codigo: entero; tlf: entero largo; **var** F: Ficha): booleano;
crea una Ficha F con la información suministrada en los parámetros, y devuelve verdad si la ficha se ha creado correctamente. En caso contrario devuelve falso y no crea la ficha. Una ficha se considera correcta si cumple los siguientes requisitos:

DNI: debe empezar por 29, 41 o 37 y tener 8 dígitos (Ej.: 29345654 o 41876323)

Código: Número entero de 4 dígitos menor que 5000

Teléfono: debe empezar por 25, 28 o 23 y tener 6 dígitos (Ej.: 254321 o 234543)

función *dni*(F: Ficha): entero largo;

devuelve el DNI de la ficha F

función *codigo*(F: Ficha): entero;

devuelve el código de una ficha F

función *telefono*(F: Ficha): entero largo;

devuelve el teléfono de una ficha F

TAD TlistaF:

El TAD *TlistaF* está formado por el tipo **listaF**, basado en una lista enlazada mediante punteros. Las características y operaciones del TAD son las siguientes:

- El tipo **listaF** se implementará mediante un registro formado por 2 campos: un puntero apuntando al principio de la lista y un natural que indica el número de elementos que tiene la lista.

procedimiento crearListaF(**var** L: listaF);

crea una listaF vacía

función esVacia(L: listaF): booleano;

devuelve verdad si la listaF está vacía y falso en caso contrario

función longitud(L: listaF): natural;

devuelve el número de fichas que hay en la lista L

función esta(dni: entero largo, L: listaF): booleano;

devuelve verdad si existe un ficha con un determinado dni en la lista L y falso en caso contrario

función inserta(**var** LP,LS: listaF; F: Ficha): natural;

inserta la ficha F en la lista LP o en la lista LS según el siguiente criterio:

(a) si el DNI de F no se encuentra en LP, se inserta F en LP

(b) si el DNI de F se encuentra en LP y su código es mayor que el encontrado en LP, se sustituye la ficha de la lista LP por la nueva ficha F. La ficha sustituida de LP se intenta insertar en la lista LS, siguiendo este mismo criterio.

(c) si el DNI de F se encuentra en LP pero su código es menor que el encontrado en LP, entonces se intenta insertar F en LS. Si también se encuentra en LS y el código sigue siendo menor que el encontrado en LS, la ficha no se inserta en ninguna lista.

las inserciones se realizarán de forma ordenada en orden creciente, por el campo D.N.I.

la función devuelve "1" si la ficha se ha insertado en LP, "2" si se inserta en LS, y "0" si no se inserta en ninguna lista

función borrarDNI(dni: entero largo; **var** L: listaF): booleano;

elimina de la lista L una ficha con un determinado dni. Devuelve verdad si se ha eliminado y falso en caso contrario

función borrarTlf(tlf: entero largo; **var** L: listaF): natural;

elimina de la lista L todas las fichas con un determinado teléfono. Devuelve el número de fichas que han sido eliminadas

Además, habrá que implementar las operaciones: igualdad, asignación y liberación

- **función** iguales(L1,L2: listaF): booleano;
- **procedimiento** asigna(**var** L1:listaF; L2: listaF);
- **procedimiento** libera(**var** L: listaF);

Tiempo estimado de realización

3 clases (se entregará en la semana del 2 al 6 de diciembre en la clase de prácticas)