# Laboratorio de PRED – Práctica 2 (3-12-2008)

Se desea implementar una estructura de datos que guarde información bancaria. En concreto, la información contenida en esa estructura de datos nos ha de suministrar información de quien es el titular de cada cuenta de un cierto conjunto de bancos, donde cada cuenta se identifica por un string alfabético (un código asociado a cada banco) y una cadena de caracteres numéricos que identifica la cuenta en el banco y cada titular es identificado por otro string (su NIF). Es decir, la información sobre cada cuenta es guardada en objetos de la siguiente clase:

```
class Cuenta{
string banco;
string id_cuenta;
titular * L_titulares;};
```

donde **L_titulares** corresponde a la lista de titulares de la cuenta.
La estructura de datos se ha de implementar usando una clase llamada Bancario. Esto es, la práctica ha de tener una clase:

```
class Bancario
{/* implementación de la estructura*/}
```

Esta clase ha de incluir las siguientes operaciones:

```
void incluir (string banco,string id-cuenta, string NIF)
{/* instrucciones de incluir */}

void elimina_cuenta(string banco,string id_cuenta)
{/* instrucciones de venta */}

void consulta_titulares (string banco,string id_cuenta)
{/* instrucciones de consulta */}

void consulta_cuentas (string p)
{/* instrucciones de consulta */}
```

Además de la constructora que ha de crear la estructura vacía.

- Dados los datos de una cuenta y uno de sus titulares, la operación **incluir** la añade a la estructura. Si ya hubiera una cuenta con el mismo código se debe admitir la nueva inclusión. Es decir, puede haber cuentas con más de un titular. Si se repite el titular se ha de señalar el error escribiendo "Error: titular ya existente".

- La operación **elimina_cuenta**, dado un código de cuenta (banco, id_cuenta) elimina la cuenta del sistema bancario. Si la cuenta no estuviera en la estructura se ha de señalar un error escribiendo el mensaje "Error: cuenta inexistente".

- La operación **consulta_titulares**, dado un código de cuenta (banco, id_cuenta) nos escribe los titulares, un titular por línea. Si no hubiera una cuenta con ese código en la estructura se ha de señalar un error escribiendo el mensaje "Error: cuenta inexistente".

- Finalmente, la operación **`consulta_cuentas`**, dado un NIF nos escribe los códigos de las cuentas de las cuales es titular (cada código en una línea nueva y las dos partes del código separado por un espacio). Si no hubiera ningún titular con ese NIF en la estructura entonces no se escribe nada.

La estructura se ha de implementar creando, en primer lugar, una clase llamada **`titular`** donde cada objeto de esta clase incluye un campo de tipo string para el NIF y los enlaces que se crea conveniente para almacenar la lista de titulares de una cuenta. Asimismo se ha de crear una clase llamada **`Nodo`**, donde cada objeto de esa clase incluye un campo de tipo **`Cuenta`** y los campos de encadenamiento que se considere oportuno tener.
A los **`Nodo`** accedemos por medio de dos tablas de hash encadenadas. En una se accede a través de los códigos de cuenta (el banco concatenado con el id_cuenta) y en la otra a través del NIF de los titulares. En los dos casos se accede usando la misma función de hash que siempre devuelve un valor entre 0 y 22. Esta función, que no forma parte de lo que hay que entregar en la práctica, ha de llamarse h y estar declarada dentro de una clase que se tiene que llamar Fhash. En concreto, esta clase estaría declarada así:

```
class Fhash{
public:
    static int h (string c)
     {/* código de la función */}
}
```

En particular, ésto quiere decir que para calcular el valor de hash del NIF 66666666F habría que hacer una llamada a Fhash::h("66666666F").

La práctica ha de contener, como comentarios, la especificación de Bancario, de su implementación y la justificación de la corrección de la implementación. Además, las declaraciones de las clases, atributos y funciones que se realicen deben de incluir el atributo de visibilidad (public, private, etc.) que se considere adecuado. Como en el caso de la primera práctica, todas las clases han de estar incluidas en un único fichero que estará encabezado por un comentario que contenga los nombres y números de DNI del autor o los autores de la práctica (las prácticas se hacen individualmente o en grupos de dos personas). Una práctica debe entregarse en la cuenta de uno solo de los autores. Además, tanto dichas clases, como las operaciones y los campos mencionados más arriba han de tener **exactamente** los nombres que tienen en este enunciado. Sin embargo, el fichero que se entregue, que ha de llamarse "Bancario.cpp", **NO** ha de incluir la implementación de la función de hash h, **ni se ha de importar** ninguna clase no estándar (salvo la clase string, los vectores y el espacio de nombres estándar). El incumplimiento de estas especificaciones será penalizado.