

# EjercicioSinColisiones

## Estructuras de Datos

### Tema 3: conjuntos

1º Grado en Ingeniería de la Computación

© Profesor Dr. Carlos Grima Izquierdo ([www.carlosgrima.com](http://www.carlosgrima.com))

URJC ([www.urjc.es](http://www.urjc.es))

Programa en C++ una base de datos de estudiantes, implementada mediante una sencilla tabla hash sin colisiones. Un estudiante es simplemente su nombre, guardado en un objeto de la clase "string".

La clase "string" nos la proporciona la biblioteca estándar de C++ (busca en [www.cplusplus.com](http://www.cplusplus.com) cómo usarla), y así de este modo no tenemos que usar arrays de caracteres. Para crear un array de objetos de una determinada clase (ej: "string"), es conveniente usar "new" en vez de "malloc". El primero llamaría automáticamente al constructor de cada uno de los objetos del array, pero el segundo se limitaría únicamente a reservar memoria, sin llamar a los constructores (y por lo tanto los objetos quedarían en estado inconsistente al no haber llamado a sus constructores). Por lo tanto new hace dos trabajos a la vez: reserva memoria y llama a los constructores. Para liberar memoria reservada con "new", es necesario llamar a "delete", el cual no sólo liberará la memoria sino que también llamará automáticamente a los destructores. Más información y ejemplos de "new" y "delete" en [http://c.conclase.net/curso/?cap=013b#OP2\\_NewDelete](http://c.conclase.net/curso/?cap=013b#OP2_NewDelete)

Los nombres de los estudiantes se guardarán en la tabla hash según su número de matrícula. Así pues, la clave de cada estudiante consistirá en su número de matrícula, un número entero de tipo "int".

Nuestra clase "TablaHashDeEstudiantes" tendrá los siguientes métodos y atributos:

- Un atributo "capacidad" que contenga la capacidad de la tabla hash. Esta capacidad no cambiará en la vida del objeto.
- Un constructor en el cual se proporciona dicha capacidad.
- Uno o varios arrays (o listas contiguas) que guardarán la información de la tabla hash. Cada posición de la tabla contiene tres informaciones:
  - La clave del elemento: el número de matrícula (int). Nos permite buscar.
  - El valor de cada elemento: el nombre del estudiante (string). Es la información importante que estamos guardando.
  - El bit "ocupado" (dato de tipo bool). Nos indica si en esa posición hay algo (está ocupada) o bien hay basura (está libre).

Para guardar toda esta información, se puede crear una clase o struct llamado "Elemento" que contenga estos tres campos (y hacer un array de esos structs), o bien la tabla hash puede tener tres atributos: un vector de enteros para las claves, un vector de strings para los nombres, y un vector de datos booleanos para los bits de ocupado.

- Un método privado llamado "obtenerPosicion" que hará el hashing, es decir, transformará un número de matrícula pasado como parámetro en la posición del

vector en la cual debería estar (en caso de encontrarse) el nombre del estudiante que tiene dicho número de matrícula (dicha posición debe ser devuelta por la función). Esta función hash calculará el módulo de “capacidad”, es decir el resto de la división entera de la clave entre la capacidad de la tabla.

- Un método para comprobar si un estudiante está o no en la tabla hash. Se le pasa su número de matrícula y nos devuelve verdadero o falso. Recordemos que comprobar que un elemento está en una tabla hash quiere decir ir a la posición en donde debería estar (según la función de hashing) y mirar si la clave que hay allí es la clave que se está buscando... ya que recuerda que puede haber varias claves distintas que se meterían en la misma posición, debido a que la función hashing nos devolvería el mismo valor.
- Un método para obtener un estudiante. Se le pasa su número de matrícula y nos devuelve el nombre del estudiante. La precondition es que el estudiante esté en la tabla hash.
- Un método para meter un estudiante en la tabla hash. Se le proporciona el número de matrícula y el nombre del estudiante. La precondition es que no haya nada en la posición en donde metemos al estudiante (en posteriores ejercicios eliminaremos esta precondition).
- Una función para eliminar un estudiante. Se proporciona la clave del estudiante y la función lo elimina de la tabla. La precondition es que el estudiante exista previamente en la tabla hash. Recuerda que eliminar es simplemente poner el bit “ocupado” a falso.
- Un método para imprimir la tabla hash. Cada posición de la tabla se imprimirá en una fila. En cada fila, imprimirá: posición (empieza en 0 y acaba en capacidad-1), flag (que indica si esa posición tiene algo o no), clave (el número de matrícula) y contenido (el nombre del estudiante). Todas las posiciones de la tabla hash se imprimirán independientemente de si tienen algo o no. Ojo porque esta función viola el principio de separación entre modelo e interfaz.

Realizaremos un main con un menú que nos permita meter un estudiante, eliminarlo, comprobar si está, obtenerlo, e imprimir la tabla hash.

```

Introduce la capacidad de la tabla: 5
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 1
Posicion 0 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 1 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 2 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 3 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 4 : Flag: 0 : Clave: 3452816845 : Contenido:
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 2
Introduzca el numero de matricula del estudiante: 106
El estudiante NO esta
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 4
Introduzca el numero de matricula del estudiante: 105
Introduzca el nombre del estudiante: juan
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 4
Introduzca el numero de matricula del estudiante: 207
Introduzca el nombre del estudiante: maria
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 1
Posicion 0 : Flag: 1 : Clave: 105 : Contenido: juan
Posicion 1 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 2 : Flag: 1 : Clave: 207 : Contenido: maria
Posicion 3 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 4 : Flag: 0 : Clave: 3452816845 : Contenido:
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 3
Introduzca el numero de matricula del estudiante: 207
El estudiante se llama maria

```

```

0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 1
Posicion 0 : Flag: 1 : Clave: 105 : Contenido: juan
Posicion 1 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 2 : Flag: 1 : Clave: 207 : Contenido: maria
Posicion 3 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 4 : Flag: 0 : Clave: 3452816845 : Contenido:
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 3
Introduzca el numero de matricula del estudiante: 207
El estudiante se llama maria
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 5
Introduzca el numero de matricula del estudiante: 105
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 1
Posicion 0 : Flag: 0 : Clave: 105 : Contenido: juan
Posicion 1 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 2 : Flag: 1 : Clave: 207 : Contenido: maria
Posicion 3 : Flag: 0 : Clave: 3452816845 : Contenido:
Posicion 4 : Flag: 0 : Clave: 3452816845 : Contenido:
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 2
Introduzca el numero de matricula del estudiante: 105
El estudiante NO esta
0. Salir
1. Imprimir tabla
2. Comprobar si esta un estudiante
3. Ver un estudiante
4. Meter un estudiante
5. Eliminar un estudiante
Introduzca opcion: 0
Presione una tecla para continuar . . .

```