

# Tarea 5

## Implementación de Estructuras de Datos

### Curso 2019

#### Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Materiales</b>	<b>2</b>
<b>3. ¿Qué se pide?</b>	<b>2</b>
3.1. Verificación de la implementación . . . . .	3
<b>4. Descripción de los módulos y algunas funciones</b>	<b>3</b>
4.1. Módulos nuevos . . . . .	3
4.1.1. Módulo <i>hash</i> . . . . .	3
4.1.2. Módulo <i>heap</i> . . . . .	3
<b>5. Entrega</b>	<b>3</b>
5.1. Plazos de entrega . . . . .	4
5.2. Identificación de los archivos de las entregas . . . . .	4

## 1. Introducción

En esta tarea se continúa la implementación de **Estructuras de Datos**, cumpliendo requerimientos de tiempo de ejecución. Mantenemos los módulos de las tareas anteriores. Se agregan los módulos `hash` y `heap`.

No se puede usar la representación de un tipo fuera de su propio módulo. Por ejemplo, en `uso_tads.cpp` no se puede usar `rep_cadena` sino las operaciones declaradas en `cadena.h`.

El correcto uso de la memoria y el cumplimiento de las restricciones de tiempo será parte de la evaluación.

El resto del presente documento se organiza de la siguiente forma. En la Sección 2 se presenta una descripción de los materiales disponibles para realizar la presente tarea, y en la Sección 3 se detalla el trabajo a realizar. Luego, en la Sección 4 se describen los nuevos módulos. Por último, la Sección 5 describe el formato y mecanismo de entrega, así como los plazos para realizar la misma.

## 2. Materiales

Los materiales para realizar esta tarea se encuentran en el archivo *MaterialesTarea5.tar.gz* que se obtiene en la carpeta *Materiales* de la sección *Laboratorio* del sitio EVA del curso <sup>1</sup>.

A continuación se detallan los archivos que se entregan. La estructura de directorios es igual a la de la tarea anterior.

- **info.h**: módulo de definición del tipo `info_t`
- **cadena.h**: módulo de definición de `cadena`.
- **binario.h**: módulo de definición de árbol binario de búsqueda
- **uso\_cadena.h**: módulo de definición de funciones sobre cadenas
- **pila.h**: módulo de definición de pilas de elementos de tipo `int`
- **avl.h**: módulo de definición de árboles AVL de elementos de tipo `info_t`
- **cola\_avls.h**: módulo de definición de colas de árboles AVL
- **iterador.h**: módulo de definición de iteradores sobre elementos de tipo `info_t`
- **conjunto.h**: módulo de definición de conjuntos de elementos de tipo `info_t`
- **hash.h**: módulo de definición de hash de asociaciones de enteros con textos.
- **heap.h**: módulo de definición de heaps de elementos de tipo `info_t`
- Casos de prueba en el directorio **test**
- **Makefile**: archivo para usar con el comando `make`, que provee las reglas `principal`, `testing`, `clean` y entrega como en las tareas anteriores.
- **principal.cpp**: módulo principal.

**Ninguno de estos archivos deben ser modificados.**

## 3. ¿Qué se pide?

Para cada archivo de encabezamiento `.h`, descrito en la Sección 2, se debe implementar un archivo `.cpp` que implemente **todas** las definiciones de tipo y funciones declaradas en el archivo de encabezamiento correspondiente. Los archivos `.cpp` serán los entregables y deben quedar en el directorio **cpp**.

En algunas operaciones se piden requerimientos de tiempo de ejecución. Ese tiempo es el del peor caso, a menos que se especifique otra cosa de manera explícita.

<sup>1</sup><https://eva.fing.edu.uy/course/view.php?id=132&section=6>

### 3.1. Verificación de la implementación

Para testear la implementación debe generar el ejecutable y cumplir con todas las pruebas utilizando los archivos **.in** y **.out**. Tenga en cuenta que el archivo **Makefile** proveerá las reglas **principal**, **testing** y **clean** como en las tareas anteriores.

## 4. Descripción de los módulos y algunas funciones

### 4.1. Módulos nuevos

#### 4.1.1. Módulo *hash*

En este módulo se implementa un hash de asociaciones de enteros con textos. La cantidad de asociaciones que se pueden mantener en cada momento está acotada por un parámetro pasado al crear la tabla.

#### 4.1.2. Módulo *heap*

En este módulo se implementa un heap de elementos de tipo `info_t`. La cantidad de elementos que puede contener el heap está acotada por un parámetro pasado al crearlo. El orden de cada elemento está determinada por su dato numerico. Ese valor puede ser modificado luego mediante la operacion `reducir`. No puede haber dos elementos con el mismo dato numérico. Los datos numéricos deben pertenecer a un rango determinado al crear el heap.

## 5. Entrega

Se mantienen las consideraciones reglamentarias y de procedimiento de las tareas anteriores.

En particular, la tarea otorga dos punto a los grupos que la aprueben en la primera instancia de evaluación y uno o cero a los que apruebn en la segunda instancia. La adjudicación de los puntos de las tareas de laboratorio queda condicionada a la respuesta correcta de una pregunta sobre el laboratorio en el segundo parcial. La adjudicacion se aplica de manera individual a cada estudiante del grupo.

La tare es **eliminatória**.

Se debe entregar el siguiente archivo, que contiene los módulos implementados *binario.cpp*, *cadena.cpp*, *avl.cpp*, *binario.cpp*, *cadena.cpp*, *cola\_avls.cpp*, *conjunto.cpp*, *info.cpp*, *iterador.cpp*, *hash.cpp* *heap.cpp* *pila.cpp* *uso\_cadena.cpp*:

#### ■ Entrega5.tar.gz

Este archivo se obtiene al ejecutar la regla entrega del archivo *Makefile*:

```
$ make entrega
tar zcvf Entrega5.tar.gz -C src info.cpp cadena.cpp uso_cadena.cpp binario.cpp pila.cpp avl.cpp cola
info.cpp
cadena.cpp
uso_cadena.cpp
binario.cpp
pila.cpp
avl.cpp
cola_avls.cpp
iterador.cpp
conjunto.cpp
heap.cpp
hash.cpp
```

Con esto se empaquetan los módulos implementados y se los comprime.

**Nota:** En la estructura del archivo de entrega los módulos implementados deben quedar en la raíz, NO en el directorio `src` (y por ese motivo se usa la opción `-C src` en el comando `tar`).

**NO SE PUEDEN ENTREGAR MÓDULOS ADICIONALES A LOS SOLICITADOS**

### **5.1. Plazos de entrega**

El plazo para la entrega es el **jueves 20 de junio a las 21 horas**.

**NO SE ACEPTARÁN ENTREGAS DE TRABAJOS FUERA DE FECHA Y HORA. LA NO ENTREGA O LA ENTREGA FUERA DE LOS PLAZOS INDICADOS IMPLICA LA PÉRDIDA DEL CURSO.**

### **5.2. Identificación de los archivos de las entregas**

Cada uno de los archivos a entrega debe contener, en la primera línea del archivo, un comentario con el número de cédula de el o los estudiantes, **sin el guión y sin dígito de verificación**. Ejemplo:

`/* 1234567 - 2345678 */`