

# Tarea 1

## Modularización

### Curso 2022

## 1. Introducción y objetivos

Esta tarea tiene los siguientes objetivos específicos:

- Familiarizarse con el ambiente de trabajo del laboratorio.
- Realizar un primer acercamiento al lenguaje C.
- Comenzar a familiarizarse con la división del código de programas compuestos por módulos distribuidos en múltiples archivos.

Como cada tarea, es individual y eliminatoria.

## 2. Materiales

Los archivos para la tarea se extraen de *MaterialesTarea1.tar.gz*. Para conocer la estructura de los directorios ver la Sección **Materiales** de [Estructura y funcionamiento del Laboratorio](#).

En esta tarea los archivos en el directorio `include` son **utils.h**, **info.h**, **cadena.h** y **colCadenas.h**. Estos archivos no se pueden modificar.

En el directorio `src` se incluyen ya implementados **utils.cpp** e **info.cpp**. Además, se incluyen los archivos **plantilla\_cadena.cpp** y **plantilla\_colCadenas.cpp**. Estos dos archivos contienen las funciones que se deben implementar pero con lo mínimo para que pueda compilar. Se pueden usar para lo cual se les debe cambiar el nombre (esto es remover `plantilla`), poner la cédula y completar el cuerpo de las funciones y el `struct`.

## 3. Desarrollo

Ver la Sección **Desarrollo** de [Estructura y funcionamiento del Laboratorio](#).

En esta tarea se deben implementar los archivos **cadena.cpp** y **colCadenas.cpp**.

La generación del ejecutable se hace compilando los archivos implementados **cadena.cpp**, **colCadenas.cpp**, **info.cpp**, **utils.cpp** y **principal.cpp** y enlazando los archivos objeto obtenidos:

El archivo **Makefile** (ver el material disponible sobre [Makefile](#)) provee para la compilación la regla **principal**, que es la predeterminada. Por lo tanto el ejecutable se obtiene mediante

```
$ make
```

Se sugiere probar cada uno de los test provistos y al final confirmar mediante la regla **testing**:

```
$ make testing
```

## 4. Entrega

Ver la Sección **Entregas** de [Reglamento del Laboratorio](#).

#### 4.1. Plazos de entrega

El plazo para la entrega es el **viernes 18 de marzo a las 13:00**.

#### 4.2. Archivo a entregar y procedimiento

Se debe entregar el archivo **Entrega1.tar.gz**, que contiene los módulos a implementar **cadena.cpp** y **colCadenas.cpp**.

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

```
$ make entrega
```

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

El archivo a entregar **DEBE** ser generado mediante este procedimiento. Si se lo genera mediante alguna otra herramienta (por ejemplo, usando un entorno gráfico) **la tarea no será corregida**, independientemente de la calidad del contenido. Tampoco será corregida si el nombre del archivo se modifica en el proceso de entrega. Si alguna de estas dos irregularidades ocurre en la entrega la calificación será *Insuficiente* lo que obligará a hacer la reentrega. Si ocurre en la reentrega la calificación será *No Aprobado* lo que implicará la pérdida de la tarea, y por lo tanto del curso.

#### 4.3. Identificación de los archivos de las entregas

Cada uno de los archivos a entregar debe contener, en la primera línea del archivo, un comentario con el número de cédula del estudiante, sin el guión y sin dígito de verificación.

Ejemplo:

```
/* 1234567 */
```

#### 4.4. Individualidad

Ver la Sección **Individualidad** de [Reglamento del Laboratorio](#).

### 5. Descripción de los módulos

En esta sección se describen los módulos que componen la tarea. Estos son *utils*, *info*, *cadena*, *colCadenas* y *principal*.

#### 5.1. Módulo *utils*

En este módulo se declara el tipo de los enteros no negativos, arreglos de tipos simples, y operaciones de lectura desde la entrada estándar.

#### 5.2. Módulo *info*

En este módulo se declara el tipo `TInfo`. Este tipo representa información compuesta por un natural (tipo `nat`) y un punto flotante (tipo `double`).

Se debe notar que para acceder a los componentes de un elemento de tipo `TInfo` se deben usar las operaciones declaradas en `info.h`.

#### 5.3. Módulo *cadena*

En este módulo se declara el tipo `TCadena`. Este tipo representa secuencias acotadas de elementos de tipo `TInfo`. O sea, una cadena puede contener desde 0 hasta `L` elementos, donde `L` es una constante definida en el mismo módulo. Hay operaciones para insertar, remover y hacer búsquedas.

#### 5.4. Módulo *colCadenas*

En este módulo se declara el tipo `TColCadenas`. Este tipo representa una colección de `CANT_COLS` `TCcadenas`, donde `CANT_COLS` es una constante definida en el mismo módulo. Cada una de las `TCadenas` corresponde a una posición en el rango  $0 \dots \text{CANT\_COLS} - 1$ . Hay operaciones para insertar, remover y hacer búsquedas, que incluyen la posición que identifica la cadena.

#### 5.5. Módulo *principal*

El módulo *principal* consiste en un único archivo, `principal.cpp`. Es un intérprete de comandos utilizado para probar las funciones implementadas en los otros módulos. Se crea una instancia de `TCadena` y una de `TColCadena` y luego se ejecuta un ciclo en el que se leen comandos que se ejecutan. Está definido un comando para cada operación de `TCadena` y de `TColCadenas` (excepto para `CrearCadena` y `CrearColCadenas`).