

## Fracciones Dinámicas Genéricas

El objetivo de este control es implementar una versión genérica del TAD de manejo de fracciones, de forma que pueda especificarse el tipo del numerador y del denominador. Además debe realizarse mediante una implementación dinámica (la clase plantilla tendrá atributos dinámicos, es decir, los atributos serán punteros al parámetro tipo).

### Trabajo a realizar

Debe implementarse el TAD genérico mediante una plantilla `template <class T> Racional`, que permita especificar el tipo a utilizar para representar el numerador y el denominador de la fracción. De esta forma, mediante `Racional<short>` se obtendrá una clase de fracciones que representa dichos valores utilizando **short int**, mediante `Racional<int>` una clase que los representa con **int**, y mediante `Racional<long>` una clase que los implementa con **long**.

Para realizar el control se proporciona los siguientes archivos:

- `main.cpp`. Este archivo contiene la función `main` que realiza las pruebas de la implementación del TAD. Para ello procesa un archivo que contiene, en cada línea, la directiva para realizar una determinada operación con fracciones, siguiendo los mismos convenios que en el primer control. Se incluyen, así mismo, tres nuevas directivas, `s`, `i` y `l`, que permiten fijar, respectivamente, a **short**, **int** y **long** la precisión con la que desea operarse (además de cambiar la precisión, escriben por pantalla `PRECISION_SHORT`, `PRECISION_INT` y `PRECISION_LONG` respectivamente). Se asegura que, si la precisión activa es de un determinado tipo, todos los valores numéricos que aparezcan en adelante, y hasta que se cambie la precisión, serán literales de dicho tipo.  
A continuación, se muestra un ejemplo de entrada procesable por este programa, y de la salida producida (se asume que los valores de tipo **short** ocupan 2 bytes y que la representación de dichos valores es en complemento a 1):

Entrada	Salida
s	PRECISION_SHORT
+ 32767/2 5/2	-16382/1
i	PRECISION_INT
+ 32767/2 5/2	16386/1
l	PRECISION_LONG
+ 32767/2 5/2	16386/1

Importante: No se permite realizar ningún tipo de modificación en el archivo `main.cpp` (en éste se implementa ya completamente el proceso descrito).

- `Racional.h`. El código de este archivo se corresponde con una implementación de `Racional` acorde con el primer control. Por tanto, el trabajo a realizar en este control es **modificar este archivo para obtener la implementación de la plantilla `template <class T> Racional` incorporando el uso de atributos dinámicos ( $T^*$ ). Se sugiere incorporar los cambios (genericidad, implementación dinámica) en pasos sucesivos. ¡Recordad además las implicaciones que tiene el uso de punteros!**