

## Trabajo práctico 3: Implementación en C++ “Wolfe”

### Normativa

**Límite de entrega:** Desde el 30 de Junio de 2014 a las 13:30 hs. (TM) o las 17:00 hs. (TT) hs.

**Normas de entrega:** Ver “Información sobre la cursada” en el sitio Web de la materia.

(<http://www.dc.uba.ar/materias/aed2/2014/1c/informacion>)

### 1. Normas de entrega adicionales

- **La entrega es presencial;** si alguien no puede asistir, con justificativo debe comunicarse con los JTPs.
- El día de la entrega deberán disponer de una copia del código que pueda ejecutarse en los laboratorios.
- Para aprobar, además del correcto funcionamiento del TP deben estar presentes todos los integrantes y responder las preguntas del corrector.

### 2. Enunciado

Tomando como base el diseño realizado en el TP2 (con el enunciado del TP2, sin incluir las modificaciones del RTP2) y utilizando las correcciones realizadas por los docentes, se pide:

1. Implementar en C++ todo lo diseñado en el TP2, prestando atención a lo siguiente, justificando en cada caso que corresponda.
  - Respetar las interfaces de cada módulo, las funciones de la interfaz deben ser **public**.
  - Respetar estructura de representación elegida en el TP2. La misma debe ser **private**. Pueden adaptar la representación a la implementación de C++ por cuestiones de memoria y no por cuestiones de performance.
  - Respetar los algoritmos diseñados y sus complejidades. Pueden modificarlos solo ante problemas encontrados por uds. y no detectados en la corrección por el docente.
  - Siempre que se pueda, corresponder los parámetros in/out de los algoritmos de diseño con parámetros por referencia y referencia constante en la implementación de C++.
2. Implementar las funciones de test que crean necesarias para verificar la correctitud de la implementación (con respecto a la especificación) de cada módulo. El correcto funcionamiento de los test **no es garantía** de aprobación. Se sugiere realizar al menos un test por función y por módulo de diseño. Para facilitar el testing y la evaluación se provee un *driver* de test en C++ en la página web de la materia: <http://www.dc.uba.ar/materias/aed2/2014/1c/descargas/tps/tp3>
3. La implementación dada no debe perder memoria en ningún caso. Al momento de la corrección se hará el chequeo pertinente usando *valgrind*.
4. Se sugiere chequear las precondiciones con **assert** para facilitar la depuración de errores.
5. Para los módulos que utilicen STRING, considerar los 256 posibles caracteres del código ascii, tipo **char** en C++.