

Programación 4 - 2023

Laboratorio 4 :: Implementación

Consideraciones generales:

- La entrega podrá realizarse hasta el **lunes 19 de junio de 2023 a las 15 hrs.**
- El código fuente y el archivo Makefile [1] deberán ser entregados mediante el EVA del curso [2] dentro de un archivo con nombre `<número de grupo>_lab4.zip` (o `tar.gz`).
- Más allá de que se sugiere el uso de un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para contar con un ambiente de desarrollo adecuado, el archivo Makefile entregado debe ser independiente de cualquier IDE, permitiendo la compilación aislada de la solución.
- El código deberá poder ejecutarse sin errores de compilación en las máquinas Linux de la Facultad de Ingeniería.
- Las entregas que no cumplan estos requerimientos no serán consideradas. El hecho de no realizar una entrega implica la insuficiencia del laboratorio completo.
- Es posible utilizar la herramienta `valgrind`¹ para verificar el uso correcto de memoria.

Como última etapa del desarrollo de la prueba de concepto, se necesita implementar el prototipo funcional del sistema utilizando el lenguaje C++ [3], en un entorno Linux [4,5].

Se pide:

1. Implementar todos los casos de uso del Laboratorio 2 y del Laboratorio 3. Notar que esto incluye la implementación de casos de uso para los cuales no se pidió realizar el diseño en el Laboratorio 3.
2. Implementar un método `main` que incluya un menú sencillo e interactivo con el usuario que permita ejecutar los distintos casos de uso.
3. Incluir, dentro del menú, una opción que permita inicializar/cargar un conjunto predefinido de datos de prueba. Esto permitirá realizar pruebas partiendo de un estado válido del sistema. Más adelante se proporcionará el conjunto de datos que deberá ser cargado y que será requerido al momento de realizar la evaluación de este laboratorio.
4. Realizar un archivo Makefile que permita compilar y ejecutar el código. La ejecución del laboratorio, que se empleará para la evaluación de la última entrega, se realizará con todo el grupo presente y se ejecutará en una máquina Linux de la Facultad de Ingeniería.

Notas:

- Se puede utilizar el tipo `std::string` [6] para implementar los atributos de tipo string, así como estructuras de datos de la biblioteca STL [7], tales como vector, set, map, etc.
- Se debe solucionar problemas de dependencias circulares entre las clases, por ejemplo, con relaciones bidireccionales. Para ello es necesario utilizar declaraciones en avanzada (forward declarations) [8].

¹ <https://valgrind.org/docs/manual/quick-start.html>

Referencias

- [1] Programación 4. Instructivo de Compilación
URL: <https://eva.fing.edu.uy/course/view.php?id=413§ion=3>
- [2] EVA Programación 4. URL: <https://eva.fing.edu.uy/course/view.php?id=413>
- [3] C++. URL: <https://www.cplusplus.com/>
- [4] Unidad de Recursos Informáticos, Configuraciones.
URL: <https://www.fing.edu.uy/sysadmin/configuraciones>
- [5] Unidad de Recursos Informáticos, Gestión Estudiantil.
URL: <https://www.fing.edu.uy/sysadmin/gestion-estudiantil>
- [6] Tipo `std::string`
URL: https://en.cppreference.com/w/cpp/string/basic_string
- [7] C++ Standard Template Library (STL)
URL: <https://cplusplus.com/reference/stl/>
- [8] Programación 4. Referencias Circulares y Namespaces
URL: <https://eva.fing.edu.uy/course/view.php?id=413§ion=3>