Universidad del Valle – Sede Tuluá **21-02-2022**
Facultad de Ingeniería
Ingeniería de Sistemas
IPOO
Doc. M.Sc. Adrian Lasso C – luis.lasso@correounivalle.edu.co

Enunciado Trabajo Final. Valor 30%

Objetivo: aplicar el Paradigma Orientado a Objetos y estructuras de datos para resolver un problema que requiere de un programa informático.

Enunciado: desarrollar una aplicación en C++ que aplique el Paradigma Orientado a Objetos y estructuras de datos primitivas tipo arrays, matrices o colecciones de datos como vector, list, stack, queue, set o cualquier otra disponible en https://www.cplusplus.com/reference/stl/ para permitir la gestión de datos de cualquier sistema de información propuesto por el equipo de trabajo.

Requerimientos básicos:

- La aplicación debe contar con al menos tres tipos de usuario. Uno debe ser tipo Administrador, que tiene acceso total a la aplicación, y puede crear otros usuarios.
- Cada usuario debe estar descrito por al menos cinco atributos (campos).
- Todos los usuarios deben loguearse para utilizar la aplicación.
- La aplicación muestra un menú con opciones dependiendo del tipo de usuario logueado.
- El sistema de información propuesto debe estar compuesto por al menos 7 entidades, que se representan por medio de clases. Por ejemplo: Cliente, Articulo, Vendedor, Proveeedor, Estudiante, Docente, Actividad, etc. A su vez pueden existir diferentes tipos de Cliente, por ejemplo: Afiliado, NoAfiliado, Cortesia, etc.
- La aplicación debe usar todo lo visto en clase: Clases, objetos, colecciones de datos para almacenar los objetos, Sobrecarga de métodos y constructores, métodos que reciban argumentos por valor y por referencia, Composición, Herencia de al menos 3 niveles, los tres tipos de modificadores de acceso (public, private y protected), Polimorfismo y métodos virtuales. El uso de estructuras de datos primitivas tipo arrays o matrices es opcional, es decir queda a potestad del equipo de trabajo.
- El proyecto se debe estructurar separando la interfaz de la implementación.
- Para cada tipo de usuario y entidad se debe declarar un atributo de tipo bool llamado activo que se asigna por defecto con el valor *true*. Este sirve para representar el estado del usuario o la entidad en el sistema de información. Por ejemplo, saber si un Cliente está activo o no.
- Se debe usar el modelo CRUD (Crear, Leer/Consultar, Actualizar y Borrar) para todos los usuarios y entidades de la aplicación. En la acción de borrar, no se borran los datos realmente, solo se cambia el valor del campo activo por *false*
- En la acción de consultar por lo menos se debe poder consultar por: número de identificación o código (según sea el caso), y un listado general.

- Demás especificados por el equipo de trabajo.
- Demás especificados por el docente después de socializada la propuesta en la semana 4 (14-03-2022).

Observaciones generales:

- La propuesta de trabajo debe ser socializada en la semana 4 (14-03-2022), por medio de un documento que describa el sistema a implantar, las funcionalidades, los tipos de usuarios y las entidades.
- Metodología de trabajo: grupos de máximo cuatro estudiantes
- Entregable: proyecto en CodeBlocks o DevC++
- Fecha y hora de entrega: 20-06-2022 a las 14:00
- Método de entrega: enlace campus virtual
- Horario y forma de sustentación: por definir