Curso: **Programación orientada a objetos**

Laboratorio 1C

A usted se le contrata para hacer una aplicación para la Universidad que permita manejar la reserva de los laboratorios. Básicamente lo que se desea es que, desde la aplicación, el personal de TI pueda manejar la reserva y asignación de laboratorios al inicio y durante el transcurso del cuatrimestre.
Como todavía hay algunas reservas de que usted pueda entregar el proyecto a tiempo, el gerente de TI decidió que lo que usted deberá de entregar primero una aplicación que permita administrar el ingreso, modificación y eliminación de los laboratorios (lo que se conoce como un mantenimiento), así como el ingreso, modificación y eliminación de carreras y cursos.
Como a su jefe no le gustó mucho eso de que no tuvieran la confianza de que lo puede concluir a tiempo, le indicó que, si bien no va a usar todos los conocimientos de la orientación a objetos, deberá de modificar el prototipo para que maneje estas clases: Laboratorio (que tiene los atributos código (entero), nombre (String), capacidad (entero)), la clase Carrera (que tiene código (String), nombre (String), grado (que puede ser técnico, diplomado, Bachillerato, Licenciatura y Maestría), acreditada (bolleano)) y la clase Curso (que tiene código (String), nombre (String), créditos (entero)).
Siempre siguiendo la idea del cliente, su jefe le pidió que mantuviera la funcionalidad anterior, pero que en lugar de usar ArrayList de String, utilizara ArrayList Laboratorios, ArrayList de Cursos y ArrayList de Carreras.
Debido al éxito de su prototipo, le pidieron que añadiera la posibilidad de manejar profesores y empleados. De los empleados le pidieron manejar cédula, nombre, apellido, dirección, teléfono y puesto (todos los datos de tipo String). De los profesores se le solicitó manejar cédula, nombre, apellido, dirección, teléfono, lugar de trabajo y años de experiencia. (todos String menos los años de experiencia). Su jefe le indicó que, para manejar estos dos nuevos elementos, debe crear una clase Persona y que empleado y profesor hereden de dicha clase.
Para manejar su diseño, a usted se le pide que haga un diagrama de clases en UML que le permita representar las asociaciones entre los diferentes conceptos y hacer las modificaciones en el código que usted ha manejado hasta ahora.
En ese diseño debe poder incluir el ingreso de solicitudes de reserva por parte de los profesores. En la solicitud deberá de seleccionarse el laboratorio, el curso y el

profesor que lo imparte, y la cantidad de estudiantes y la fecha en que se ocupa el laboratorio. Cada solicitud de reserva tendrá un identificador único.

El programa debe ser hecho en capas, lo que implica que deberá de usar un Gestor para separar la capa lógica de la capa de presentación. Además, deberá trabajar usando una librería para la CL y para probar, deberá de almacenar los empleados, los profesores y las carreras en archivos de texto.

Para hacer la transición al almacenamiento de la información utilizando bases de datos, su jefe le solicitó que en la arquitectura agregara una nueva serie de clases que se conocen como MULTIS. Los multis son clases que van a tener la funcionalidad de crear las consultas a la base de datos y comunicarse con la base de datos. Usted debe crear un Multi para cada clase, y poner ahí los métodos de insertar, listar y modificar.

Para ello se le brinda un ejemplo en un archivo adjunto. Usted deberá de probar el uso de multis en el registro, listado y modificación de empleados, profesores y laboratorios solamente. El resto de las funcionalidades las puede manejar en memoria. En el menú deber incluir las siguientes opciones.

1. Registrar Empleados
2. Registrar Profesores
3. Registrar carrera. (si ya existe no debe poder registrarla)
4. Listar Carreras
5. Registrar cursos (si ya existe no debe poder registrarlo. De momento no importa la carrera a la que está asociado el curso)
6. Listar Cursos
7. Registrar Laboratorios (si ya existe no debe poder registrarlo).
8. Listar Laboratorios.
9. Eliminar Laboratorio (debe validar que exista)
10. Registrar reserva de laboratorio
11. Listar reserva de laboratorio