Universidad Nacional
Escuela de Informática
Programa I, II ciclo 2024

II Proyecto

Academia Virtual Eureka AI School

Fecha máxima de entrega: domingo 10 de noviembre *(evite subir el proyecto a último momento,o de hacerlo queda bajo su responsabilidad cualquier eventualidad)*

Modalidad: individualmente o en parejas

Temas:

- Principios del POO
- Adecuada abstracción y encapsulamiento
- Listas enlazas en forma de clases tipo colección
- Memoria dinámica
- Relaciones
- Archivos

Descripción:
Al graduarse de Ingeniería de Sistemas usted se asociada con varios de sus compañeros y entre todos deciden abrir una academia virtual llamada **"Eureka AI School"**, cuyo objetivo es impartir cursos sobre tecnologías digitales.

La academia ofrece cuatro bloques o periodos lectivos durante el año en el que se ofrecen cursos:

- **Periodo 1:** de enero a marzo
- **Periodo 2:** de abril a junio
- **Periodo 3:** de julio a setiembre
- **Periodo 4:** de octubre a diciembre

Nota: para efectos de este proyecto, únicamente se toma en cuenta el año actual.

Una vez creadas los periodos o bloques del año, se desea poder crear y asignar cursos y grupos a dichos bloques. Recordemos que por cada curso se pueden abrir n grupos.

- De cada **curso** importa: nombre, id, horas, precio, estado *(disponible, noDisponble)* entre otros.
- De cada **grupo** importa: número de grupo, capacidad de alumnos, cantidad de alumnos, horario, entre otros.

El sistema debe permitir ingresar o registrar nuevos **estudiantes** en los grupos de los cursos ofrecidos. Dentro de la información básica requerida de cada estudiante está: nombre, id, especialidad, número telefónico, email, entre otros.

Además, se deben poder ingresar **profesores** para los cursos. Para los profesores, los datos requeridos son: nombre, id, número telefónico, email, grado académico. Tener presente que todo profesor puede estar a cargo de n grupos.

Se requiere la implementación de la clase **horario,** para la cual interesa: Hora de Inicio, Hora de Finalización, días de la semana, entre otros datos.

El sistema deberá permitir registrar: estudiantes, profesores, horarios, bloques o periodos, cursos, grupos, realizar matricula, facturas y generar informes.

Para la matrícula de estudiantes registrados en los cursos disponibles se deben solicitar los datos requeridos. Un mismo estudiante debe poder matricular n grupos *(siempre y cuando el grupo tenga cupo disponible).* Por cada periodo un estudiante solo puede matricular en un único grupo de cada curso

Al finalizar la matrícula se debe mostrar la factura por los cursos matriculados. El detalle de la factura debe incluir: curso, precio por curso, subtotal, IVA, descuento, total a pagar. En caso de aplicar descuento se deberá especificar el tipo del descuento y su monto). El sistema debe permitir realizar descuentos en los siguientes casos:

- Si un estudiante matricula dos o más cursos en un mismo periodo, tendrá un 15% de descuento en la factura.
- Si un estudiante matricula cuatro cursos o más en un año, tendrá un 25% de descuento *(adicional al anterior).*

El sistema debe permitir a los estudiantes matricular en los diferentes periodos que se ofrecen al año, crear y poblar las diferentes listas y generar los reportes o informes que se solicitan más adelante en la descripción de la interfaz.

Persistencia de datos (archivos)

- El sistema deberá permitir guardar y acceder archivos para la siguiente información:
  - Lista de cursos
  - Lista de bloques o periodos **(adecuadamente relacionada con grupos y cursos)**
  - Listas de profesores
  - Listas de estudiantes
  - Lista de grupos **(adecuadamente relacionada con el periodo, el profesor y el horario)**

Nota: No se solicita la persistencia de datos para la información de matrícula.

- Para guardar los datos de las listas en los archivos respectivos se debe ofrecer una opción en el **menú principal.**
- Para cargar y acceder la información de los archivos en las listas, no se requiere una opción del menú *(esto se realiza automáticamente en el momento que inicia el sistema).*

  Nota: En otras palabras, en todas las opciones de la Interfaz deberá reflejarse la persistencia de datos, a excepción de las opciones relacionadas con la matrícula.

Valide que el sistema impida el ingreso datos repetidos.

Se debe diseñar, mostrar y manejar un menú de opciones adecuado, el cual permita ingresar y administrar adecuadamente la información requerida y manejar las transacciones ofrecidas por el sistema.

Implemente una interfaz agradable e intuitiva para el usuario. No olvide limpiar la pantalla cada que vez que se requiera. Utilice una interacción limpia y clara.

Detalle de Interfaz y demás funcionalidades

Menú Principal

1- Submenú Administración
2- Submenú Matrícula
3- Submenú Búsquedas e Informes
4- Guardar los Datos en Archivos
5- Salir

Ingrese la opción: _

El menú principal permitirá acceder a los diferentes submenús del programa.

Submenú Administración General

(1) Ingresar Profesor
(2) Ingresar Estudiante
(3) Ingresar Bloque o Periodo
(4) Ingresar Curso
(5) Ingresar Grupo
(6) Asignar Profesor a Grupo
(0) Regresar al Menú Principal

Ingrese la opción: _

(1) Permite ingresar un profesor con todos sus atributos
(2) Permite ingresar un estudiante con todos sus atributos
(3) Permite crear un bloque o periodo lectivo.
(4) Permite ingresar un curso.
(5) Permite crear grupos de cursos ya registrados. Cada grupo debe ser asociado un horario y a un periodo lectivo. Se pueden registrar varios grupos para un mismo curso, ejecutando repetidamente esta transacción.
(6) Permite asociar un profesor a un grupo, ambos ya registrados previamente en el sistema.

Submenú Matrícula

(1) Matricular Estudiante
(2) Desmatricular Estudiante
(0) Regresar al menú Principal

Ingrese la opción: _

(1) Se podrá matricular estudiantes existentes, en periodos, cursos y grupos previamente registrados en el sistema. No se pueden matricular estudiantes en grupo llenos. No se puede matricular un mismo estudiante en mas de un grupo de mismo curso. El estudiante podrá matricularse en n grupos siempre y cuando sean de cursos diferentes.

Al finalizar la matrícula de un estudiante se debe generar y mostrar la **factura** por los cursos matriculados, con el detalle descrito anteriormente. Se debe mostrar la factura debe ser detallada.

(2) La opcion deberá permitir buscar un estudiante y mostrar la lista de grupos matriculados en el periodo especifico por dicho estudiante, para así poder seleccionar de cuales grupos se desea desmatricular.

Si un estudiante se des-matricula de un curso, la disponibilidad de cupo en el grupo correspondiente se incrementa en 1.

Búsquedas e Informes

(1) Informe Profesores Registrados
(2) Informe Estudiantes Registrados
(3) Informe Cursos Matriculados por un Estudiante
(4) Informe Profesor Especifico
(5) Informe Periodos Habilitados para el Año
(6) Informe Grupo Especifico

(0) Regresar al Menú Principal

Ingrese la opción: _

(1) Se muestra el nombre y ID de cada profesor

(2) Se muestra el nombre y ID de cada estudiante.

3) Se busca un estudiante por Id y se muestra la información de todos los cursos matriculados: código del curso, nombre del curso, numero de grupo matriculado y horario.

(4) Se selecciona un profesor especifico y se muestran todos los datos registrados del profesor, incluyendo **los cursos/grupos que imparte en cada periodo del año.**

(5) Se debe mostrar el nombre de todos los periodos lectivos habilitados para el año *(Periodo I, Periodo II, Periodo III, Periodo IV),* incluyendo el mes de inicio y finalización y **la lista de todos los cursos y grupos disponibles en cada periodo.**

(6) Se selecciona un curso registrado y luego uno de los grupos del curso. Se debe mostrar la información del grupo seleccionado: nombre del curso, numero, capacidad o cupo del grupo, horario, profesor a cargo y en especial la **lista de estudiantes**

Tabla de Evaluación

Para la evaluación se espera el cumplimiento de todo los lineamientos y pautas detalladas anteriormente en este documento, así como el desarrollo un programa eficiente el cual haga uso adecuado del POO.

Profesores Registrados *(Se muestra la lista de todos los profesores ingresados, no pueden existir profesores matriculados)*	4
Estudiantes Registrados *(Se muestra la lista de todos los estudiantes ingresados, , no pueden existir estudiantes matriculados)*	4
Informe Cursos Matriculados por un Estudiante *Aquí se evalúa el proceso de matrícula, considerando estudiantes, periodos, cursos y grupos previamente existentes y bien relacionados*	20
Generación de factura completa	7
Detalle de profesor específico, especificando entre otros los cursos/grupos que imparte en cada periodo del año	15
Periodos Habilitados para el Año *Se debe mostrar los datos de todos los periodos lectivos habilitados para el año, incluyendo la lista de todos los cursos y grupos disponibles en cada periodo.*	15
Detalle de un Grupo *Se debe mostrar la información de grupos seleccionado, incluyendo: nombre del curso, numero, capacidad o cupo del grupo, horario, profesor a cargo y en especial lista de estudiantes matriculados, etc*	15
Recuperación de archivos y visualización de la información en el sistema	20
Total	**100**

Consideraciones Generales:

- ✓ Se evalúa el diseño, la funcionalidad y la eficiencia.
- ✓ Adecuada definición e implementación de relaciones entre las clases.
- ✓ La principal estructura de datos a utilizar debe ser **Listas Enlazadas.**
- ✓ El programa no debe tener errores de compilación. En caso de tener errores de compilación o no iniciar la ejecución, el proyecto se califica con cero.
- ✓ Cuando se requieran conjuntos de datos, se deberán implementar clases tipo colección creadas por el programador. No se podrán utilizar las colecciones provistas por la STL (Standard Template Library) o cualquier por otra biblioteca de C++. Si no se cumple esta directriz el proyecto se calificará con cero.
- ✓ En proyectos programación, el main() deberá quedar con la menor cantidad de código posible: básicamente sin funcionalidad. Funcionará únicamente como punto de inicio del programa.
- ✓ En la definición de las clases se deben separar correctamente su declaración y su implementación, utilizando archivos de cabecera (archivos .h) para la declaración y archivos de código fuente (archivos .cpp) para la implementación. Los archivos de cabecera deberán siempre usar guardas (#ifndef, #pragma once).
- ✓ Las clases tipo entidad no deberán contener código de entrada/salida (como, por ejemplo, salida a consola usando **cout**, o lectura por medio de **cin**). Sin embargo, se pueden mostrar mensajes de comprobación al efectuar las pruebas

del programa, para corroborar que se están ejecutando las funciones de manera correcta.

- ✓ Para ser evaluado, el programa debe tener implementadas de manera funcional, por lo menos, el 20% de las **funcionalidades** descritas en este enunciado.
- ✓ En caso de detectarse plagio en alguno de los métodos o en cualquier parte del programa, se asignará cero a la calificación del proyecto.
- ✓ Queda a criterio de cada profesor solicitar la defensa oral del proyecto, dentro o fuera del horario de clase del curso.
- ✓ El proyecto se realizará individualmente o en grupos de dos estudiantes máximo y se entregará por medio del aula virtual, en la actividad correspondiente. No se recibirán documentos ni materiales por otro medio, salvo indicación manifiesta del profesor.
- ✓ No se admitirá la entrega tardía del proyecto.
- ✓ Verifique que el archivo entregado sea el correcto pues no se aceptaran archivos entregados posteriormente.
- ✓ El medio oficial de entrega será por medio del aula virtual.
- ✓ Recuerde incluir en los documentos de entrega un archivo bloc de notas llamado equipo de trabajo, que incluya el nombre completo de cada integrante del equipo.
- ✓ Solo uno de los integrantes debe entregar el proyecto.
- ✓ Elabore y entregue el diseño UML de clases en formato pdf. Se debe incluir un diagrama de clases con los atributos y métodos de cada clase y las relaciones entre éstas.
- ✓ Entregue el proyecto completo en el IDE formalmente solicitado por cada profesor.