

Tema 1 – Proyecto final

Sistema para Control Escolar

Descripción

Este proyecto se centra en el manejo de información de estudiantes, asignaturas y profesores utilizando estructuras de datos estáticas como arreglos unidimensionales, bidimensionales y multidimensionales. El programa permitirá al usuario realizar las siguientes operaciones:

1. **Agregar Estudiantes:** El usuario puede ingresar los datos de los estudiantes, que incluyen el nombre, la edad, la nota promedio y el número de cursos aprobados. Los datos de los estudiantes se almacenarán en un arreglo unidimensional.
2. **Agregar Asignaturas:** El usuario puede registrar las asignaturas disponibles, incluyendo el nombre de la asignatura y el nombre del profesor que la imparte. Los datos de las asignaturas se almacenarán en un arreglo unidimensional.
3. **Agregar Profesores:** El usuario puede registrar los profesores que imparten las asignaturas. Los datos de los profesores incluyen el nombre y la especialidad. Los datos de los profesores se almacenarán en un arreglo unidimensional.
4. **Asignar Asignaturas a Estudiantes:** El usuario podrá asignar asignaturas a estudiantes, indicando qué estudiantes están inscritos en cada asignatura. Esto se realizará utilizando una estructura de datos bidimensional o multidimensional que relaciona estudiantes y asignaturas.
5. **Mostrar Lista de Estudiantes:** El programa mostrará una lista de todos los estudiantes ingresados con sus respectivos datos.
6. **Mostrar Lista de Asignaturas:** Se mostrará una lista de todas las asignaturas disponibles junto con el nombre de los profesores que las imparten.
7. **Buscar Estudiante por Nombre:** El usuario puede buscar un estudiante por su nombre y ver su información detallada, incluyendo las asignaturas en las que está inscrito.
8. **Calcular Estadísticas:** El programa calculará estadísticas sobre la población de estudiantes, como el promedio de edad, la nota promedio y el número total de estudiantes.
9. **Ordenar Estudiantes:** El usuario puede ordenar la lista de estudiantes por nombre, edad, nota promedio o número de cursos aprobados utilizando arreglos bidimensionales.
10. **Filtrar Estudiantes:** El usuario puede filtrar estudiantes según criterios específicos, como estudiantes mayores de una cierta edad o con una nota promedio superior a cierto valor.

Consideraciones Adicionales

- Utiliza clases y objetos para representar a estudiantes, asignaturas y profesores.
- Implementa métodos para realizar las operaciones mencionadas, como agregar estudiantes, asignaturas y profesores, asignar asignaturas a estudiantes, calcular estadísticas y más.
- Proporciona una interfaz de usuario (consola) para que el usuario pueda interactuar con el programa.

Requisitos de funcionalidad (Ampliación descriptiva de las operaciones a realizar)

1. Registro de Estudiantes:
 - a. El sistema debe permitir a los usuarios agregar estudiantes con la siguiente información: nombre, edad, nota promedio y número de cursos aprobados.
 - b. Los estudiantes registrados deben almacenarse en una lista o arreglo unidimensional.
2. Registro de Asignaturas:
 - a. El sistema debe permitir a los usuarios registrar asignaturas con la siguiente información: nombre de la asignatura y nombre del profesor que la imparte.
 - b. Las asignaturas registradas deben almacenarse en una lista o arreglo unidimensional.
3. Registro de Profesores:
 - a. El sistema debe permitir a los usuarios registrar profesores con la siguiente información: nombre y especialidad.
 - b. Los profesores registrados deben almacenarse en una lista o arreglo unidimensional.
4. Asignación de Asignaturas a Profesores:
 - a. El sistema debe permitir a los usuarios asignar asignaturas a profesores, indicando qué profesor imparte cada asignatura.
 - b. Debe ser posible asignar múltiples asignaturas a un mismo profesor y viceversa.
 - c. Esta información de asignación debe almacenarse en una estructura de datos adecuada, como un arreglo bidimensional o una relación entre profesores y asignaturas.
5. Visualización de Listas:
 - a. El sistema debe permitir a los usuarios ver listas de estudiantes, asignaturas y profesores, junto con sus respectivos detalles.
 - b. Las listas deben mostrarse de manera legible y organizada en la interfaz de usuario.
6. Búsqueda de Estudiantes:
 - a. Los usuarios deben poder buscar estudiantes por su nombre y ver su información detallada, incluyendo las asignaturas en las que están inscritos.
7. Cálculo de Estadísticas:
 - a. El sistema debe calcular estadísticas sobre la población de estudiantes, como el promedio de edad, la nota promedio y el número total de estudiantes registrados.
8. Ordenación de Listas:
 - a. Los usuarios deben poder ordenar las listas de estudiantes, asignaturas y profesores por nombre, edad, nota promedio, número de cursos aprobados o especialidad, según corresponda.
9. Filtrado de Estudiantes:
 - a. Los usuarios deben poder filtrar estudiantes según criterios específicos, como estudiantes mayores de una cierta edad o con una nota promedio superior a cierto valor.
10. Interfaz de Usuario (UI):

- a. El sistema debe tener una interfaz de usuario (consola u otra interfaz gráfica) que permita a los usuarios interactuar con las funciones mencionadas de manera intuitiva.
- 11. Comentarios y Documentación:
 - a. El código del proyecto debe estar adecuadamente comentado y documentado para facilitar su comprensión y mantenimiento.
- 12. Asignación de Asignaturas a Estudiantes:
 - a. El sistema debe permitir a los usuarios asignar asignaturas a estudiantes, indicando qué estudiantes están inscritos en cada asignatura.
 - b. Debe ser posible asignar múltiples estudiantes a una misma asignatura y viceversa.
 - c. Esta información de asignación debe almacenarse en una estructura de datos adecuada, como un arreglo bidimensional o una relación entre estudiantes y asignaturas.

Fecha de entrega

Jueves, 21 de septiembre de 2023. Horario de clase.

Rubrica para evaluación del proyecto

Categoría de Evaluación	Insuficiente (0-69)	Suficiente (70-74)	Bueno (75-84)	Notable (85-94)	Excelente (95-100)
Funcionalidad General	El programa no se ejecuta correctamente y no realiza las operaciones básicas.	El programa se ejecuta pero tiene errores significativos que afectan su funcionalidad.	El programa se ejecuta y realiza las operaciones básicas, pero con algunos errores menores.	El programa se ejecuta sin errores y realiza todas las operaciones especificadas correctamente.	El programa se ejecuta sin errores y supera las expectativas en funcionalidad y usabilidad.
Interfaz de Usuario	La interfaz de usuario es confusa, poco intuitiva o inexistente.	La interfaz de usuario es funcional pero no es muy intuitiva ni amigable.	La interfaz de usuario es clara y permite al usuario interactuar de manera efectiva con el programa.	La interfaz de usuario es altamente intuitiva, bien diseñada y facilita la interacción del usuario con el programa.	La interfaz de usuario es excepcionalmente intuitiva y estéticamente agradable.
Uso de Estructuras de Datos	No se utilizan estructuras de datos de manera adecuada o no se utilizan en absoluto.	Se utilizan algunas estructuras de datos, pero su implementación es deficiente o inadecuada.	Se utilizan estructuras de datos de manera adecuada y eficiente en la mayoría de las operaciones.	Se utilizan estructuras de datos de manera altamente eficiente y se eligen las estructuras más apropiadas para cada caso.	La elección y el uso de estructuras de datos son excepcionales y contribuyen significativamente a la eficiencia del programa.
Representación de Clases y Objetos	No se utilizan clases y objetos para representar estudiantes, asignaturas y profesores.	Se utilizan clases y objetos de manera básica, pero la encapsulación y la estructura pueden mejorarse.	Se utilizan clases y objetos para representar adecuadamente a estudiantes, asignaturas y profesores, con una buena encapsulación.	Se utilizan clases y objetos de manera excepcional, con una excelente encapsulación y diseño orientado a objetos.	La implementación de clases y objetos es ejemplar y demuestra un alto nivel de dominio de la programación orientada a objetos.

Funcionalidad Adicional	No se implementan características adicionales o las implementadas son no funcionales.	Se implementan algunas características adicionales, pero su funcionalidad es limitada o con errores.	Se implementan varias características adicionales con funcionalidad aceptable.	Se implementan múltiples características adicionales con funcionalidad destacada y sin errores.	Las características adicionales implementadas son excepcionales y añaden un valor significativo al proyecto.
Comentarios y Documentación	El código carece de comentarios y documentación.	El código incluye algunos comentarios y documentación, pero es insuficiente o incoherente.	El código incluye comentarios y documentación adecuados para entender su funcionamiento.	El código incluye comentarios detallados y documentación exhaustiva para facilitar la comprensión y el mantenimiento.	La documentación es excepcionalmente completa y profesional, y facilita enormemente la comprensión y el mantenimiento del proyecto.
Cumplimiento de Requisitos	El proyecto no cumple con la mayoría de los requisitos especificados.	El proyecto cumple con algunos requisitos, pero no con todos.	El proyecto cumple con la mayoría de los requisitos, pero pueden faltar detalles.	El proyecto cumple con todos los requisitos especificados de manera completa y precisa.	El proyecto no solo cumple con todos los requisitos, sino que también supera las expectativas y agrega características valiosas.