Análisis del programa:

El programa permite la entrada de datos de estudiantes y sus respectivas calificaciones. Además, calcula el promedio de calificaciones de cada estudiante y genera un listado con los estudiantes que obtienen las mejores notas.

Sugerencia:

En la implementación del menú principal, se recomienda utilizar una estructura switch en lugar de if-else para optimizar el rendimiento y la legibilidad del código.

Se ha incorporado un bucle while para gestionar el cierre del programa adecuadamente, garantizando que el programa siga ejecutándose hasta que el usuario seleccione la opción de salida correspondiente

Los ingresos de estudiantes se optimizan.

En lugar de registrar a los estudiantes de forma individual, una mejor práctica sería solicitar al usuario la cantidad total de estudiantes que desea registrar. Se puede implementar esto mediante un bucle for, que permite iterar un número determinado de veces en función de la cantidad ingresada.

Al comienzo del proceso, se solicitará al usuario el número de estudiantes a registrar antes del proceso. Posteriormente, el programa llevará a cabo la ejecución del bucle, solicitando los datos correspondientes a cada estudiante en cada iteración. En caso de que el usuario ingrese un número superior o superior a 2, el sistema procederá a solicitar la información de todos los estudiantes de manera secuencial, optimizando la entrada de datos y mejorando la experiencia del usuario.

***¿Qué datos deben ser accesibles en todo el programa?***

Por ejemplo, en el código las listas de estudiantes y calificaciones deben ser accesibles en todo el código por qué es lo que almacena la información que se consulta y que también modifica en varios métodos, a comparación fueran locales tendrían gran diferencia y seria un poco mas complicado ya que cada método tendría que manejar su propia lista y no estarían compartiendo datos.

***¿Qué datos solo son necesarios dentro de una función específica?***

Las variables como nombre y calificaciones en ingreoAlumno() o suma en PromedioCalificaciones(), solo se necesitan mientras el método se ejecuta , y no es necesario almacenar las afuera, porque no afectan el resto del programa una vez que el método termina

***Modularización***

***¿Qué datos necesitan ser compartidos entre múltiples funciones?***

El uso de Mostrarcalificación porque si el usuario ingresa la calificación de ese mismo ingreso se puede sacar el promedio así se omite crear otra variable.

***¿Qué datos solo son relevantes dentro de una función específica?***

El nombre y la calificación dentro de la función aggestu() son datos que se piden al usuario para agregar a las listas de estudiantes y calificaciones. Estos valores se usan solo dentro de esa función para ingresar un nuevo estudiante en el sistema. Una vez que el nombre y la calificación se guardan en las listas globales, ya no se necesitan más dentro de la función. Es decir, solo tienen sentido durante ese proceso de agregar al estudiante, pero fuera de esa función no tienen relevancia.

***Preguntas Guía***

***¿Qué ventajas tiene dividir el código en funciones?***

* Facilidad en detectar errores en partes específicas del código
* Mejor optimización
* Ayuda a llevar el código de una manera más lógica y que sea más fácil de comprender
* Mejor entendimiento en el análisis

***¿Por qué es importante limitar el uso de variables globales?***

* Puede provocar problemas a la hora de intentar entender el programa
* Será más difícil detectar los errores
* Facilidad para agregar o modificar el código

***¿Cómo se puede mejorar la legibilidad del código?***

* Definiendo las variables de una manera correcta
* Al asignar nombre a las variables intentar abreviar los nombres

A la hora de asignar nombres a las variables o funciones que sean nombres que estén relacionados