

Oliver Graf
17190
Mayra Silva
17276

Análisis Previo Laboratorio 5. Temario A

Contexto

El propósito de este ejercicio es crear un programa que sea de utilidad para el Ministerio de Educación en el cual se busca automatizar el control de aspirantes a Peritos en computación, para ello se elaborará un escalafón. Un escalafón es un listado de cada aspirante con su respectivo promedio.

Sección A. Requisitos funcionales

1. El usuario puede ingresar cualquier tipo de estudiante en el mismo programa, los cuales incluyen DPI y nombre.
2. Si el usuario desea ingresar un estudiante recién graduado de secundaria, debe poder ingresar el promedio de los tres años de estudio (primero a tercero básico). Como estos estudiantes realizan exámenes de ingreso de matemática, historia y español, se pueden ingresar las notas correspondientes.
3. Si el usuario desea ingresar un estudiante desvinculado de la secundaria, se puede ingresar el promedio de sus tres últimos años de estudio (primero a tercero básico), las notas de los exámenes de ingreso, los cuales son matemática historia y español. Aparte, realizan un examen de aptitud así que su nota debe de ser ingresada también.
4. Si el usuario desea ingresar un estudiante recién graduado de bachillerato, el cual sólo realiza examen de historia; debe de colocar el promedio de cuarto y quinto bachillerato y la nota obtenida en el examen de historia.
5. Si el usuario desea ingresar un estudiante desvinculado del bachillerato, el cual solo realiza examen de ingreso de historia, se ingresa el promedio obtenido en cuarto y quinto bachillerato y la nota obtenida en el examen de ingreso correspondiente.
6. El programa hace todos los cálculos, como calcular el promedio dependiendo del tipo de estudiante.
7. Ordena los estudiantes aspirantes de mayor a menor, en cuanto a su puntuación.
8. Muestra si el promedio de notas de los aspirantes desvinculados graduados de secundaria es mayor a un valor ingresado.
9. Muestra si el 50% de los aspirantes desvinculados graduados de bachillerato tienen un promedio de notas mayor a 80 puntos.

Sección B. Posible solución basada en las estructuras a diseñar

**Se encuentran en orden, como en la lista de requisitos funcionales.

1. Se utilizará la clase scanner para poder recibir los datos que el usuario ingrese. Por otro lado, se utilizará herencia de tal forma que todos los estudiantes tengan los mismos atributos de nombre y DPI. Además se creará un arraylist de tipo estudiante el cual almacenará a todos los estudiantes sin importar el tipo que sea.
Se implementará polimorfismo, de tal forma que todos los estudiantes tengan el método para calcular el promedio de los años de estudio, y el método para calcular la nota para el escalafón.
2. Se implementará una clase, que por medio de herencia y la palabra reservada *super*, contenga los atributos de nombre y dpi. Además en esta clase, se añadirán los atributos correspondientes para las notas, todos con sus respectivos sets y gets, toString y el constructor. Esta clase, por medio de polimorfismo, calculará la nota para colocar en el escalafón.
3. Se implementará una clase que por medio de herencia, tenga los atributos de nombre y DPI, esta clase extiende a la clase descrita en el inciso 2. Esto es almacenado en el arreglo de estudiantes. Además tiene un atributo adicional que hace referencia a la nota del examen de aptitud.
4. Se implementará una clase que por medio de herencia, tenga los atributos de nombre y DPI Esta clase, contiene atributos adicionales que hacen referencia a la nota obtenida en los últimos dos años de estudio. Es un objeto que también es agregado al arreglo de estudiantes.
5. Se implementará una clase, que por herencia tenga los atributos de nombre y DPI, esta clase también extiende a la clase descrita en el inciso 4. Se agrega al arreglo de estudiantes.
6. La interfaz se encargará de crear "el contrato" para que todas las clases tengan el método tanto para calcular el promedio de los años anteriores como para calcular la nota para el escalafón.
7. Por medio de un método que se llama CompareTo, se implementará una clase que contenga el arreglo de estudiantes y en ella se colocará este método de tal forma que los estudiantes se puedan ordenar de mayor a menor sin importar el tipo de estudiante que sea.
8. En la clase descrita en el inciso 7, se agregará un método que permita verificar el promedio de las notas de los estudiantes desvinculados de secundaria es mayor a un valor dado. Es por ello, que este método si tiene un parámetro ingresado por el usuario.
9. En la misma clase descrita en el inciso 7, se agregará otro método que verifica si un 50% de los estudiantes que son graduados de bachillerato tienen un promedio superior a 80 puntos.

Identificación de las clases

1. Main: esta clase solamente interactúa con el usuario y recibe información.
2. Estudiante: es la "clase madre" la cual tiene como propósito heredar los atributos de nombre y DPI.

3. ArrEstudiante: tiene como propósito crear un arrayList de objetos tipo estudiante, en esta clase se realiza la verificación y procesamiento de la información como se describe en los incisos 7-9, de las soluciones propuestas.
4. Secundaria: tiene como propósito crear un estudiante recién egresado de secundaria, como se describe en el inciso 2 de la sección B.
5. SecundariaDesvinculado: tiene como propósito extender de la clase secundaria para crear un estudiante desvinculado de secundaria.
6. Bachillerato: su propósito es crear un estudiante recién egresado de bachillerato, como se describe en el inciso 4 de la sección A.
7. BachilleratoDesvinculado: tiene como propósito extender a la clase Bachillerato, para crear un estudiante desvinculado de bachillerato.
8. NotasEscalafon: esta es una interfaz que por medio del polimorfismo, permite implementar como "contrato" los métodos para el cálculo de notas y promedios correspondientes.
9. Erro: esta clase tiene las excepciones, donde se hará la verificación de datos que ingresa el usuario.
10. DbConnection: esta clase hace la conexión con mongoDB

Sección C. Atributos de las clases

**Se encuentran enlistados de tal forma que el inciso 1 de esta sección corresponde a los atributos de la clase descrita en el inciso 1 de la sección B.

1. No tiene atributos.
2. La clase Estudiante tiene dos atributos:
 - ✓ Nombre: Es de tipo String, hace referencia al nombre del estudiante.
 - ✓ DPI: es de tipo String, hace referencia al número de DPI del estudiante.
 - ✓ notaFinal: es de tipo int, hace referencia a la nota del escalafón.
3. La clase ArrEstudiante tiene los siguientes atributos:
 - ✓ Temporal: es un objeto tipo estudiante y es el arraylist de estudiantes que permite agregar el estudiante ingresado.
 - ✓ valorComp: es de tipo float y representa el valor con el cual el usuario desea comparar el rendimiento en cuanto a las notas de los estudiantes desvinculados de secundaria.
4. La clase Secundaria utiliza los atributos previamente declarados en la clase estudiante y además implementa los siguientes atributos:
 - ✓ promSeptimo: es de tipo float y hace referencia al promedio obtenido en primero básico.
 - ✓ promOctavo: es de tipo float y se refiere al promedio obtenido en segundo básico.
 - ✓ promNoveno: es de tipo float y se refiere al promedio obtenido en tercero básico.
 - ✓ notaMate: es de tipo float y hace referencia a la nota obtenida en el examen de ingreso de matemática.

- ✓ notaHistoria: es de tipo float y se refiere a la nota obtenida en el examen de ingreso de historia.
- ✓ notaEspanol: es de tipo float y hace referencia a la nota obtenida en el examen de ingreso de español.
- 5. La clase SecundariaDesvinculado extiende los tributos enlistados en el inciso 3. Además posee un atributo adicional:
 - ✓ notaAptitud: es de tipo float y hace referencia a la nota obtenida en el examen de ingreso de aptitud.
- 6. La clase Bachillerato extiende los atributos de la clase Estudiante, además posee atributos adicionales:
 - ✓ promDecimo: es de tipo float y representa el promedio obtenido en cuarto bachillerato.
 - ✓ promOnceavo: es de tipo float y representa el promedio obtenido en quinto bachillerato.
 - ✓ notaHistoria: es de tipo float y hace referencia a la nota obtenida en el examen de ingreso de historia.
- 7. La clase BachilleratoDesvinculado extiende los atributos de la clase Bachillerato.
- 8. La clase notaEscalafon no tiene atributos.

Sección D. Métodos de las clases

****Se encuentran enlistados de tal forma que el inciso 1 de esta sección corresponde a los atributos de la clase descrita en el inciso 1 de la sección A. ****

1. El Main no contiene métodos.
2. Contiene solamente el constructor.
3. Contiene 4 métodos:
 - ✓ AddEstudiante el cual es de tipo void y permite añadir a los estudiantes al arreglo y recibe como parámetro un estudiante.
 - ✓ escalafonOrdenar() es de tipo void y permite ordenar los estudiantes en un arreglo, Aquí se creará una lista, donde se concatenará el nombre y la nota final, y se irán almacenando en la lista.
 - ✓ imprimirLista() es de tipo String y devolverá el nombre de los estudiantes y su respectiva nota ya en orden, haciendo uso del método ordenarEst().
 - ✓ VerificarRendimientoSecundaria() permite comparar si el promedio de notas de los aspirantes desvinculados graduados de secundaria es superior a un valor dado, donde recibe como parámetro el valor. Retorna un string en el cual se informa si es o no válida la suposición.
 - ✓ verificarRendimientoBach() permite determinar si el 50% de los aspirante desvinculados graduados de bachillerato tienen un promedio de notas superior a 80 puntos.
4. Contiene dos métodos, los cuales son:

- ✓ calcularPromedio(), es de tipo void y retorna un valor de tipo float. Se realiza de la siguiente forma:

$$\frac{promSeptimo + promOctavo + promNoveno}{3} = promedio$$

- ✓ CalcularNota(): se realizará nuevamente un promedio de las notas de los exámenes de admisión, realizando una sumatoria de las notas y dividiéndolas dentro de 3. Siguiendo la siguiente ecuación:

$$(promedio \times 0.6) + \left(\left(\frac{notaMate + notaHistoria + notaEspanol}{3} \right) \times 0.4 \right) = nota$$

Seguido de esto, se sumará el resultado obtenido en el método anterior al resultado obtenido y de esta forma se obtendrá la nota para el estudiante egresado de secundaria.

- Esta clase, en vista de que extiende a secundaria utilizará la misma metodología de los métodos a excepción de que se añade la nota obtenida en el examen de aptitud, quedando de la siguiente forma:

$$\frac{(nota\ del\ promedio \times 0.40) + (promedio\ de\ los\ exámenes\ de\ ingreso \times 0.6) + notaAptitud}{2}$$

- Para la clase Bachillerato:

- ✓ calcularPromedio(): donde se retorna un valor de tipo float.

$$\frac{promDecimo + promOnceavo}{2} = promedio$$

- ✓ calcularNota(): se retorna un valor de tipo float.

$$(promedio \times 0.6) + (notaHistoria \times 0.4) = nota$$

- Para la clase Bachillerato Desvinculado, se utiliza el mismo método para calcular promedio, el método que cambia es el siguiente:

- ✓ calcularNota() donde se retorna un valor de tipo float.

$$(promedio \times 0.4) + (notaHistoria \times 0.6) = nota$$

- Contiene los métodos:

- ✓ calcularPromedio() al cual se le dará la estructura en cada una de las clases enlistadas anteriormente.
- ✓ calcularNota() al cual se le dará la estructura en cada una de las clases enlistadas anteriormente.

- Contiene los métodos:

- ✓ verificarString() hace el uso de excepciones para ver si el dato ingresado es un string o un número, retornará un valor booleano. Es para que el usuario solo ingrese un string.
- ✓ verificarInt() hace el uso de excepciones y valores booleanos, donde se verifica que el usuario en efecto ingrese un dato numérico.

- DBconnection, tiene los siguientes métodos:

- ✓ addEstudiante() ingresa un estudiante, uno a uno a la base de datos.

- Link de githbu: <https://github.com/silvamayra/Laboratorio5>

