

**ST0242 – Fundamentos de programación**  
**Práctica 2: Sistema de gestión de estudiantes universitarios**

MBA, I.S. José Luis Montoya Pareja  
Departamento de Informática y Sistemas  
Universidad EAFIT  
Medellín, Colombia, Suramérica

**RESUMEN**

*En este proyecto, estarás construyendo un sistema simple de administración de clases de estudiantes usando los principios de la programación orientada a objetos, arreglos, strings y otros elementos vistos en el curso. El objetivo es crear el programa que permita administrar los cursos y los asistentes al mismo de manera sencilla.*

**PALABRAS CLAVE**

*Programación orientada a Objetos, Clases, Atributos, Métodos, Manipulación de arreglos, administración de datos, creación de menú.*

**CONTEXTO**

Se requiere implementar un programa en Java que permita crear un sistema de administración de clases de estudiantes en la universidad Unimateria. La característica principal de esta universidad es que los estudiantes solo ven una materia por semestre.

Para lograr la implementación completa, vamos a trabajar mediante el concepto de historias en el siguiente orden.

**Historia 1: Menú de la aplicación**

**Yo** como profesor, **requiero** que se cree un menú de la aplicación **para que** se pueda elegir cualquiera de las opciones del menú.

1. Semestre
2. Cursos
3. Estudiantes
4. Terminar el programa

Escriba el número de la opción deseada y presiona Enter.

*Criterios de aceptación*

1. Si el usuario está empezando el programa, solo debe aparecer el menú principal que son las cuatro primeras opciones: Semestre, Cursos, Estudiantes y Terminar el programa.
2. Si existen en disco los archivos Semestre.txt, Cursos.txt, Estudiantes.txt y Matriculados.txt debe leer los archivos y crear los objetos correspondientes con la información guardada en el archivo (ver Historia 1 criterio de aceptación 4.b).
3. Si el usuario ingresa un número de opción, presiona Enter y la opción no es ni 1, 2, 3 ni 4: debe mostrar un mensaje en pantalla diciendo *Opción seleccionada no está en la lista*.

4. Si el usuario ingresa un número de opción, presiona Enter y la opción es 1, 2, 3 ó 4:
  - a. Si la opción tiene submenú (1, 2 y 3, ver historias 2, 3 y 9), debe mostrar en pantalla el contenido del submenú y un mensaje diciendo *Escriba el número de la opción deseada y presiona Enter.*
  - b. Si la opción es la 4, el programa debe terminar.
5. Si el usuario está en un submenú y se presiona el número de la opción *Volver al menú principal*, se debe mostrar en pantalla el menú principal con las cuatro opciones.

### Historia 2: Semestre

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Semestre de la aplicación **para que** se pueda realizar las acciones para operar un semestre en la universidad.

#### *Criterios de aceptación*

1. Si se elige *Crear un semestre* debe verificar que no haya otro semestre abierto.
  - a. Si hay un semestre abierto, debe mostrar en pantalla un mensaje en una línea independiente que diga *No se puede crear un semestre nuevo cuando el actual todavía está abierto.*
  - b. Si no hay un semestre abierto, pregunta al usuario por el año y el semestre (si es el primero o el segundo), crea el semestre y lo marca como semestre activo. Debe aparecer en pantalla en una línea independiente el mensaje *Semestre creado.* Luego, debe volver al menú principal.
2. Si se elige *Cerrar un semestre* debe cerrar el semestre actual y debe aparecer en pantalla en una línea independiente el mensaje *Semestre cerrado* y se vuelve al menú principal.

### Historia 3: Agregar un curso

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Cursos – Agregar un curso de la aplicación **para que** se pueda realizar las acciones que permitan agregar un curso que se dictará durante un semestre en la universidad.

#### *Criterios de aceptación*

1. Si se elige *Agregar un curso*:
  - a. Si el semestre está cerrado, debe mostrar en pantalla en una línea independiente el mensaje *No se puede agregar cursos a un semestre cerrado.*
  - b. Si el semestre está abierto, pregunta al usuario el nombre del curso.
    - i. Si el nombre del curso se deja vacío debe aparecer un mensaje en pantalla diciendo *Ingrese un nombre para el curso.*
    - ii. Si se escribe la palabra CANCELAR en mayúscula o minúscula, se debe regresar al submenú.

- iii. Si el nombre del curso tiene algún nombre, se debe validar que no existe otro curso con el mismo nombre.
  - 1. Si hay un curso con el mismo nombre, debe aparecer en pantalla un mensaje diciendo *El curso con ese nombre ya fue creado* y debe volver a preguntar el nombre del curso.
- iv. Si el nombre del curso no existe, se solicita por pantalla el número de estudiantes del curso.
  - 1. Si el número de estudiantes es un número menor que cero debe aparecer un mensaje en pantalla diciendo *Debe ingresar un número mayor o igual a cero*. Y debe volver a pedir el número de estudiantes del curso.
  - 2. Si el número es mayor o igual a cero, se crea el curso y debe aparecer en pantalla en una línea independiente el mensaje *Curso creado* y se vuelve al menú principal.

#### **Historia 4:** Modificar un curso

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Cursos – Modificar un curso de la aplicación **para que** se pueda realizar las acciones para modificar la información de un curso que se dictará durante un semestre en la universidad.

#### *Criterios de aceptación*

- 1. Si se elige *Modificar un curso* debe aparecer una lista de los cursos que tiene el semestre y se debe poder elegir de la lista el curso que se quiere modificar. Cada curso debe aparecer con un número que lo identifique para seleccionarlo y una opción final de *Volver al menú anterior*.
  - a. Si el número de curso no aparece en la lista, debe aparecer un mensaje en pantalla diciendo *El curso no existe* y debe mostrar la lista de cursos de nuevo.
  - b. Si el número del curso existe, debe aparecer en pantalla una opción que pregunta al usuario si quiere modificar el nombre del curso o la cantidad de estudiantes.
  - c. Si desea modificar el nombre del curso debe solicitar el nombre nuevo y realizar la misma validación del menú agregar curso punto iv y solicitar de nuevo el nombre del curso.
    - i. Si el usuario escribe cancelar en mayúscula o minúscula debe volver al submenú.
    - ii. Si el nombre nuevo no existe, lo modifica y aparece en pantalla un mensaje diciendo *Nombre del curso modificado correctamente*.
  - d. Si desea modificar la cantidad de estudiantes que puede tener el curso, se valida cuantos estudiantes tiene ya matriculados el curso.
    - i. Si ya tiene estudiantes matriculados (por ejemplo, 3), la cantidad de estudiantes a modificar no puede ser menor a

3. Si esta condición no se cumple, se muestra en pantalla un mensaje diciendo *El número de estudiantes nuevo no puede ser menor al número de estudiantes matriculados*. Se debe volver a pedir la información del número de estudiantes.
- ii. Si el número es mayor o igual al número de estudiantes matriculados o si el número nuevo es mayor o igual a cero, se modifica el número de estudiantes y debe aparecer en pantalla en una línea independiente el mensaje *Número de estudiantes del curso modificado correctamente* y se vuelve al menú principal.

#### **Historia 5:** Matricular estudiantes a un curso

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Cursos – Matricular estudiantes a un curso de la aplicación **para que** se pueda matricular estudiantes a un curso que se dictará durante un semestre en la universidad.

##### *Criterios de aceptación*

1. Si se elige *Matricular estudiantes a un curso* debe aparecer una lista de los cursos que tiene el semestre y se debe poder elegir de la lista el curso al que se quiere agregar estudiantes. Cada curso debe aparecer con un número que lo identifique para seleccionarlo y una opción final de *Volver al menú anterior*.
  - a. Si el número de curso no aparece en la lista, debe aparecer un mensaje en pantalla diciendo *El curso no existe* y debe mostrar la lista de cursos de nuevo.
  - b. Si el número del curso existe, debe aparecer en pantalla la lista de estudiantes que no están matriculados en una materia, cada uno marcado con un número que lo identifique para seleccionarlo y una opción al final de *Volver al menú anterior*.
  - c. Si se elige la opción de *Volver al menú anterior*, regresa al menú anterior.
  - d. Si desea matricular a un estudiante, lo selecciona de la lista con el número y presiona Enter.
    - i. Si el número no está en la lista, se muestra en pantalla un mensaje diciendo *El número seleccionado no existe*. Se debe pedir nuevamente el número a elegir.
    - ii. Si el número está en la lista, se asocia el estudiante al curso y debe aparecer en pantalla en una línea independiente el mensaje *Estudiante asignado correctamente al curso* y se vuelve al menú principal.

#### **Historia 6:** Borrar un curso

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Cursos – Borrar un curso de la aplicación **para que** se pueda retirar un curso de la lista de cursos disponibles.

##### *Criterios de aceptación*

1. Si se elige *Borrar un curso*, se muestra la lista de cursos disponibles cada uno con un número para identificarlo. Al final de la lista, debe aparecer la opción *Volver al menú anterior*.
  - a. Si se elige la opción de *Volver al menú anterior*, regresa al menú anterior.
  - b. Si el curso elegido para borrar tiene estudiantes matriculados, debe aparecer en pantalla en una línea independiente el mensaje *No se puede borrar un curso con estudiantes matriculados* y vuelve al menú de cursos.
  - c. Si el curso no tiene estudiantes matriculados, se retira el curso de la lista de cursos disponibles y se vuelve al menú principal.

#### **Historia 7:** Mostrar los cursos existentes

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Cursos – Mostrar los cursos existentes de la aplicación **para que** se pueda consultar y mirar en pantalla la lista de cursos existentes.

##### *Criterios de aceptación*

1. Si se elige *Mostrar los cursos existentes* y no existe ningún curso aún creado durante el semestre, se muestra en pantalla un mensaje diciendo que *El semestre no tiene cursos creados aún*. Se debe volver al menú principal.
2. Si se elige *Mostrar los cursos existentes* y tiene cursos disponibles, se muestra la lista de cursos disponibles y al final de la lista, debe aparecer el mensaje *Para volver al menú anterior presione A*.
  - a. Si se elige una opción diferente a la letra A, debe mostrar en pantalla un mensaje diciendo *Opción inválida* y debe mostrar nuevamente el mensaje con la opción válida.
  - b. Si se elige la opción de A de *Volver al menú anterior*, regresa al menú anterior.

#### **Historia 8:** Mostrar los cursos existentes y sus estudiantes

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Cursos – Mostrar los cursos existentes y sus estudiantes de la aplicación **para que** se pueda consultar y mirar en pantalla la lista de cursos existentes y los estudiantes que tiene cada curso.

##### *Criterios de aceptación*

1. Si se elige *Mostrar los cursos existentes y sus estudiantes* y no existe ningún curso aún creado durante el semestre, se muestra en pantalla un mensaje diciendo que *El semestre no tiene cursos creados aún*. Se debe volver al menú principal.
3. Si se elige *Mostrar los cursos existentes y sus estudiantes* y tiene cursos disponibles, se muestra la lista de cursos disponibles y para cada curso, los estudiantes que tiene y al final de la lista, debe aparecer el mensaje *Para volver al menú anterior presione A*.

- a. Si se elige una opción diferente a la letra A, debe mostrar en pantalla un mensaje diciendo *Opción inválida* y debe mostrar nuevamente el mensaje con la opción válida.
- b. Si se elige la opción de A de *Volver al menú anterior*, regresa al menú anterior.

2.

#### **Historia 9:** Agregar un estudiante

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Estudiante – Agregar un estudiante de la aplicación **para que** se pueda realizar las acciones para agregar un estudiante en la universidad.

#### *Criterios de aceptación*

1. Si se elige *Agregar un estudiante*:
  - c. Si el semestre está cerrado, debe mostrar en pantalla en una línea independiente el mensaje *No se puede agregar estudiantes a un semestre cerrado*. Debe volver al menú principal.
  - d. Si el semestre está abierto, pregunta al usuario el nombre del estudiante.
    - i. Si el nombre del curso se deja vacío debe aparecer un mensaje en pantalla diciendo *Ingrese un nombre para el estudiante*. Se debe pedir nuevamente el nombre del curso
    - ii. Si se escribe la palabra CANCELAR en mayúscula o minúscula, se debe regresar al menú anterior.
    - iii. Si el nombre del estudiante tiene algún nombre, se debe validar que no existe otro estudiante con el mismo nombre.
      1. Si hay un estudiante con el mismo nombre, debe aparecer en pantalla un mensaje diciendo *El estudiante ya está creado* y debe volver al menú principal.
      2. Si el nombre del estudiante no existe, se crea el estudiante y debe aparecer en pantalla en una línea independiente el mensaje *Estudiante creado* y se vuelve al menú principal.

#### **Historia 10:** Modificar un estudiante

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Estudiante – Modificar un estudiante de la aplicación **para que** se pueda realizar las acciones para modificar el nombre de un estudiante.

#### *Criterios de aceptación*

1. Si se elige *Modificar un estudiante* debe aparecer una lista de los estudiantes que tiene el semestre y se debe poder elegir de la lista el estudiante que se quiere modificar. Cada estudiante debe aparecer con un número que lo identifique para seleccionarlo y una opción final de *Volver al menú anterior*.

- e. Si el número del estudiante no aparece en la lista, debe aparecer un mensaje en pantalla diciendo *El estudiante no existe* y debe mostrar la lista de estudiantes de nuevo.
- f. Si el número del estudiante existe, debe aparecer en pantalla una opción que pregunta al usuario si quiere modificar el nombre del estudiante.
  - i. Si desea modificar el nombre del estudiante debe solicitar el nombre nuevo y realizar la misma validación del menú agregar curso punto iv y solicitar el nuevo nombre.
  - ii. Si el usuario escribe cancelar en mayúscula o minúscula debe volver al menú principal.
  - iii. Si el nombre nuevo no existe, lo modifica y aparece en pantalla un mensaje diciendo *Nombre del estudiante modificado correctamente* y regresa al menú principal.

#### **Historia 11:** Borrar un estudiante

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Estudiante – Borrar un estudiante de la aplicación **para que** se pueda retirar un curso de la lista de cursos disponibles.

#### *Criterios de aceptación*

- 1. Si se elige *Borrar un estudiante*, se muestra la lista de cursos disponibles cada uno con un número para identificarlo. Al final de la lista, debe aparecer la opción *Volver al menú anterior*.
  - a. Si se elige la opción de *Volver al menú anterior*, regresa al menú anterior.
  - b. Si el estudiante elegido está matriculado en un curso, debe aparecer en pantalla en una línea independiente el mensaje *No se puede borrar un estudiante que está matriculado en un curso* y vuelve al menú de estudiantes.
- 2. Si el estudiante no está matriculado, en un curso, se retira el estudiante y se vuelve al menú principal.

#### **Historia 12:** Mostrar los estudiantes existentes

**Yo** como profesor, **requiero** que se creen las acciones del submenú de Estudiante – Mostrar los estudiantes existentes de la aplicación **para que** se pueda consultar y mirar en pantalla la lista de los estudiantes existentes.

#### *Criterios de aceptación*

- 1. Si se elige *Mostrar los estudiantes existentes* y no existe ningún estudiante aún creado durante el semestre, se muestra en pantalla un mensaje diciendo que *El semestre no tiene estudiantes matriculados aún*. Se debe volver al menú principal.
- 2. Si se elige *Mostrar los estudiantes existentes* y tiene estudiantes, se muestra la lista de estudiantes disponibles y al final de la lista, debe aparecer el mensaje *Para volver al menú anterior presione A*.



- a. Si se elige una opción diferente a la letra A, debe mostrar en pantalla un mensaje diciendo *Opción inválida* y debe mostrar nuevamente el mensaje con la opción válida.
- b. Si se elige la opción de A de *Volver al menú anterior*, regresa al menú anterior.

### **Historia 13:** Terminar el programa

**Yo** como profesor, **requiero** que se creen las acciones para terminar la ejecución de la aplicación **para que** se pueda finalizar el programa correctamente.

#### *Criterios de aceptación*

1. Si se elige *Terminar el programa*, el programa debe guardar en archivos la información de los objetos Semestre, Cursos, Estudiantes y Matriculados en los archivos Semestre.txt, Cursos.txt, Estudiantes.txt y Matriculados.txt donde cada línea es la información de un objeto. Luego, el programa debe terminar.
2. Si se presenta algún error durante la grabación del archivo, se debe mostrar un mensaje en pantalla notificándolo.

### **ACTIVIDADES**

Construir un programa en Java que implemente el funcionamiento de la Universidad. Debe tener como mínimo:

1. Implementación de las clases Semestre, Curso y Estudiante.
2. Implementar una clase Semestre con métodos setter y getter de sus atributos y con algo que permita almacenar los cursos que se tienen en la Universidad.
3. Implementar una clase Curso con los métodos setter y getter de sus atributos y con algo que permita almacenar cada uno de los estudiantes que tenga matriculados en el curso.
4. Implementar una clase Estudiante con los métodos getter y setter de sus atributos que permita almacenar el nombre del estudiante.
5. Implementar un método que permita buscar en los cursos si el estudiante está matriculado.
6. El programa principal debe tener una interfaz básica mediante un menú como el antes descrito en la historia 1.

La descripción del comando es la siguiente:

#### **NOMBRE**

Universidad – Simulación de operación de una Universidad muy simple.

#### **SINOPSIS**

**java** Universidad

#### **DESCRIPCION**

Aplicación para gestionar una universidad simple.



**Estado de salida del proceso:**

- |   |                                   |
|---|-----------------------------------|
| 0 | Si el proceso termina OK          |
| 1 | Si el proceso termina con errores |

**CONSIDERACIONES GENERALES**

1. El desarrollo de la práctica es *individual*.
2. El código en Java se debe subir al momento de la entrega por el buzón recepción de trabajos de Eafit Interactiva (cualquier otro medio no será admitido).
3. La práctica se debe realizar en Java.
4. Se debe crear el programa usando programación orientada a objetos.
5. Criterios de evaluación (ver Anexo 1)

**CÓDIGO DE HONOR Y ÉTICA**

1. Cada estudiante declarará que el código fuente es 100% realizado por él, en caso tal que se encuentre piezas de código en Internet, DEBERA DECLARARLO explícitamente y colocar las direcciones en Internet de donde las saco.
2. Si alguien utiliza ChatGPT o herramienta similar, deberá declararlo explícitamente, y decir que utilizó y como de dichas herramientas.
3. Debe declarar explícitamente que no copio o mando a realizar ninguna parte de este proyecto.
4. El código se sustentará y cada alumno deberá mostrar solvencia, apropiación y alto dominio del código.

Muchas de las anteriores aclaraciones, ya están consignadas en el Reglamento Académico Estudiantil, o se adicionan al enunciado de este proyecto, si se incurre en alguna falta de los 4 anteriores puntos, la nota de esta práctica será 0.0 y se comunicará a la oficina respecto de asuntos estudiantes (Integridad Académica).

**FECHA DE ENTREGA**

Viernes 8 de noviembre hasta las 11:59 pm a través de Eafit Interactiva.

Nombre de la asignatura: Fundamentos de programación

Competencia a la que aporta la asignatura: Practicar el diseño de soluciones y la escritura de programas en Java para solucionar mediante la simulación un problema dado.

Resultado de asignatura evaluado: Construcción de una aplicación de universidad simple.

Evento evaluativo: Práctica 1

Porcentaje del evento evaluativo: 10%

<b>Criterios (que contributen al RA de asignatura)</b>	<b>Cumple con altos estándares (4.5-5)</b>	<b>Cumple a satisfacción (4-4.4)</b>	<b>Cumple parcialmente (3.5-3.9)</b>	<b>Incumple parcialmente (2.5-3.4)</b>	<b>Incumple totalmente (0-2.4)</b>	<b>Peso asignado al criterio sobre la calificación.</b>
Análisis de la necesidad – Claridad en el concepto	Presenta un pseudocódigo que funciona correctamente para cada función implementada en Java.	Presenta pseudocódigo para todas las funciones pero algunas no funcionan correctamente.	Presenta un pseudocódigo para algunas de las funciones y no se probó.	No presenta pseudocódigo pero el código implementado demuestra entendimiento del problema.	Se entrega la solución parcialmente o no se entrega ninguna solución.	30%
Diseño del algoritmo para la solución con objetos Claridad en el concepto	Entiende completamente la necesidad y plantea varias opciones de solución. Utiliza conceptos de objetos de forma adecuada para la solución.	Entiende completamente la necesidad y plantea una opción de solución.	Omitió un elemento clave para el entendimiento de la necesidad	Omitió varios elementos para el entendimiento de la necesidad.	Demuestra poco o nulo entendimiento del problema.	30%
Documentación del código fuente	La documentación excede los criterios de código limpio, bien indentado, buena definición de variables, buen estilo de programación (Clean Code).	La documentación cumple con todos los criterios de Clean Code.	La documentación no cumple con uno de los criterios de Clean Code	La documentación no cumple con dos o tres de los criterios de Clean Code.	La documentación no cumple con ninguno de los criterios de CleanCode.	10%
Implementación de la solución	El programa funciona correctamente y tiene una solución óptima.	La solución elegida no es la óptima pero el programa funciona correctamente	El programa con los conceptos vistos en clase no funciona correctamente.	El programa no tiene en cuenta ningún concepto visto en clase y la solución funciona correcta o parcialmente.	Se entrega la solución parcialmente o no se entrega ninguna solución.	30%