

EXAMEN DEL PRIMER BIMESTRE

PROGRAMACIÓN ORIENTADA A OBJETOS

Sistema de Inscripción Académica – 'EPN-PLUS'

- Nombre del estudiante: Antony Ariel Cisneros Benavides
- Curso o asignatura: Programación Orientada a Objetos
- Fecha: 12/6/2025

INTRODUCCIÓN

La institución educativa ficticia "EPN-PLUS" desea automatizar su sistema de inscripción de estudiantes en cursos académicos. Este examen tiene como objetivo aplicar los conceptos fundamentales de la Programación Orientada a Objetos (POO) mediante el desarrollo de un sistema en Java. Deberás demostrar el uso adecuado de herencia, creación de clases, métodos personalizados, validaciones, manipulación de objetos y buenas prácticas de codificación.

ESCENARIO GENERAL

EPN-PLUS ofrece dos tipos de cursos:

1. Curso Regular:

- Tiene una duración fija en semanas.
- Se evalúa de forma continua.
- No entrega certificado.

2. Curso Certificado (hereda de Curso Regular):

- Incluye evaluación final.
- Entrega certificado válido a nivel nacional.
- Tiene un recargo del 25% sobre el precio regular.

Cada curso debe estar asignado a un docente, y los estudiantes se inscriben a través de un proceso de matrícula. Cada inscripción puede estar asociada a una evaluación final para registrar el desempeño del estudiante.

ACTIVIDADES A REALIZAR

1. Diagrama UML

Diseña un diagrama UML que muestre todas las clases necesarias, sus atributos y métodos.

Representa claramente las relaciones de herencia (por ejemplo, de Curso hacia CursoCertificado) y asociaciones (como entre Curso y Docente).

2. Estructura del Proyecto (Package y Clases)

Crea un paquete en Java para organizar todas las clases. Cada clase debe estar en su propio archivo.

3. Desarrollo de Clases y Relaciones

Implementa las siguientes clases, según el diseño de tu UML:

- Curso (clase base general)
- CursoRegular y CursoCertificado (herencia multinivel)
- Estudiante (datos personales y matrícula)
- Docente (dicta uno o varios cursos)
- Matricula (gestiona inscripción)
- Evaluacion (calificación por curso)

Incluye constructores, setters, getters y métodos personalizados.

4. Herencia Multinivel

Aplica herencia, implementando diferencias entre tipos de curso.

5. Métodos Personalizados

Cada clase debe tener varios métodos propios. No basta con getters/setters.

Métodos:

- Curso: mostrarResumen(), asignarDocente(String docente)
- CursoCertificado: calcularPrecioFinal(), mostrarResumen() (sobrescrito)
- Estudiante: mostrarDatos(), validarEdad()
- Docente: mostrarPerfil()
- Matricula: registrarEvaluacion(), mostrarDetalle()
- Evaluacion: getResultado()

Nota: Si requiere puede agregar más métodos.

Validaciones:

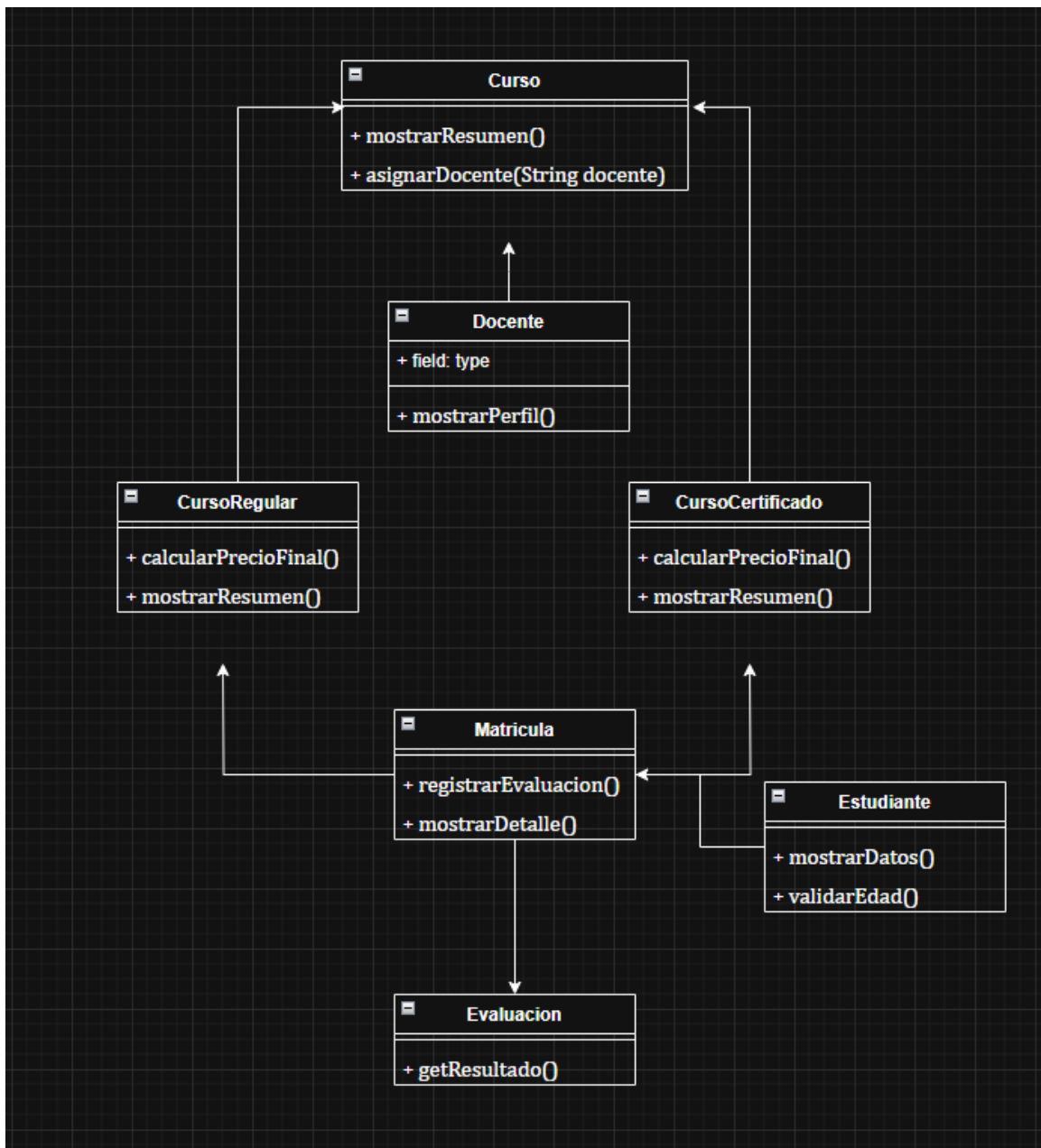
- Nota entre 0 y 10
- Edad entre 13 años y 60 años
- Curso debe tener docente asignado

6. Creación y Manejo de Objetos

Crea 5 objetos con valores nulos y luego usa setters o entrada por consola para asignar datos. También crea 5 objetos con valores desde el constructor e imprime su información.

7. Uso de GitHub

Crea un repositorio, haz 2 commits con avances y adjunta capturas del sistema funcionando en formato PDF.



Primer Commit

```
Windows PowerShell
create mode 100644 clases/Estudiante.java
create mode 100644 clases/Evaluacion.java
create mode 100644 clases/Matricula.java
create mode 100644 test/TestCurso.java
PS C:\Users\P00\IdeaProjects\Programacion1\src\examen> remote add origin https://github.com/AntonyCis/Examen1.git
remote : El término 'remote' no se reconoce como nombre de un cmdlet, función, archivo de script o programa
ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta
es correcta e inténtelo de nuevo.
En línea: 1 Carácter: 1
+ remote add origin https://github.com/AntonyCis/Examen1.git
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (remote:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

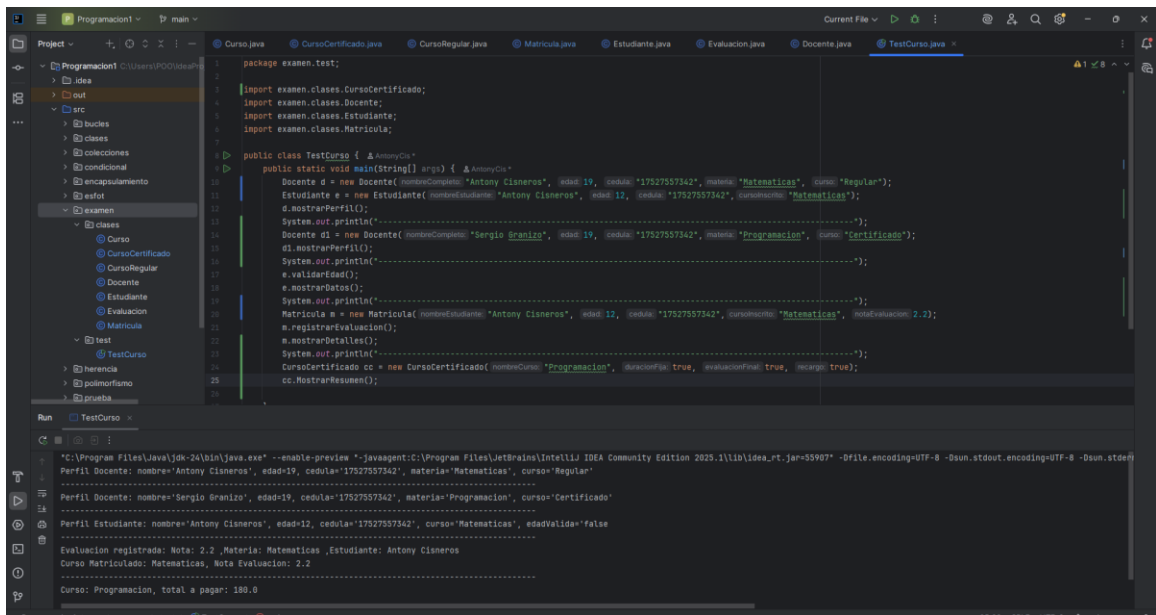
PS C:\Users\P00\IdeaProjects\Programacion1\src\examen> git remote add origin https://github.com/AntonyCis/Examen1.git
PS C:\Users\P00\IdeaProjects\Programacion1\src\examen> git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/AntonyCis/Examen1.git'
PS C:\Users\P00\IdeaProjects\Programacion1\src\examen> git branch -M main
PS C:\Users\P00\IdeaProjects\Programacion1\src\examen> git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 16 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 2.46 KiB | 631.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AntonyCis/Examen1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\P00\IdeaProjects\Programacion1\src\examen> |
```

Link de Github: <https://github.com/AntonyCis/Examen1.git>

CAPTURAS SISTEMA FUNCIONAL

Ejecución de 5 objetos

```
"C:\Program Files\Java\jdk-24\bin\java.exe" --enable-preview "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2\lib\idea_rt.jar=1273.0:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2\bin" -Dfile.encoding=UTF-8 -jar C:\Users\P00\IdeaProjects\Programacion1\src\examen\Examen1.jar
Perfil Docente: nombre='Antony Cisneros', edad=19, cedula='17527557342', materia='Matematicas', curso='Regular'
-----
Perfil Docente: nombre='Sergio Granizo', edad=19, cedula='17527557342', materia='Programacion', curso='Certificado'
-----
Perfil Estudiante: nombre='Antony Cisneros', edad=12, cedula='17527557342', curso='Matematicas', edadValida='false'
-----
Evaluacion registrada: Nota: 2.2 ,Materia: Matematicas ,Estudiante: Antony Cisneros
Curso Matriculado: Matematicas, Nota Evaluacion: 2.2
-----
Curso: Programacion, total a pagar: 180.0
```



RECOMENDACIONES

- Subir un archivo PDF con capturas y el enlace al repositorio de GitHub.
- Mantén una copia de respaldo del trabajo.
- Aplica buenas prácticas de programación.
- Comprende y participa en todo el desarrollo.

RÚBRICA DE EVALUACIÓN

Criterio	Puntos
Diagrama UML claro y coherente	0.5
Organización en paquete y clases	0.5
Implementación completa de clases	0.5
Herencia correcta	0.5
Métodos funcionales y útiles	1
Instanciación y pruebas con objetos	0.5
Uso y evidencia en GitHub	0.5

Recuerda que se evaluará el conocimiento y participación individual. ¡Mucho éxito!