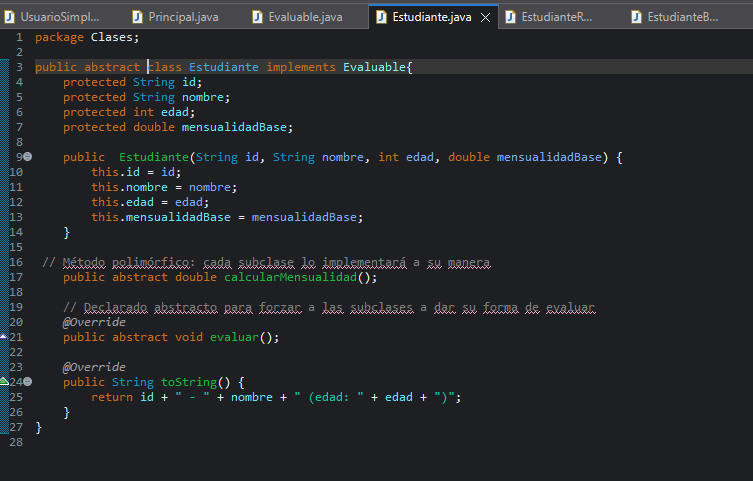
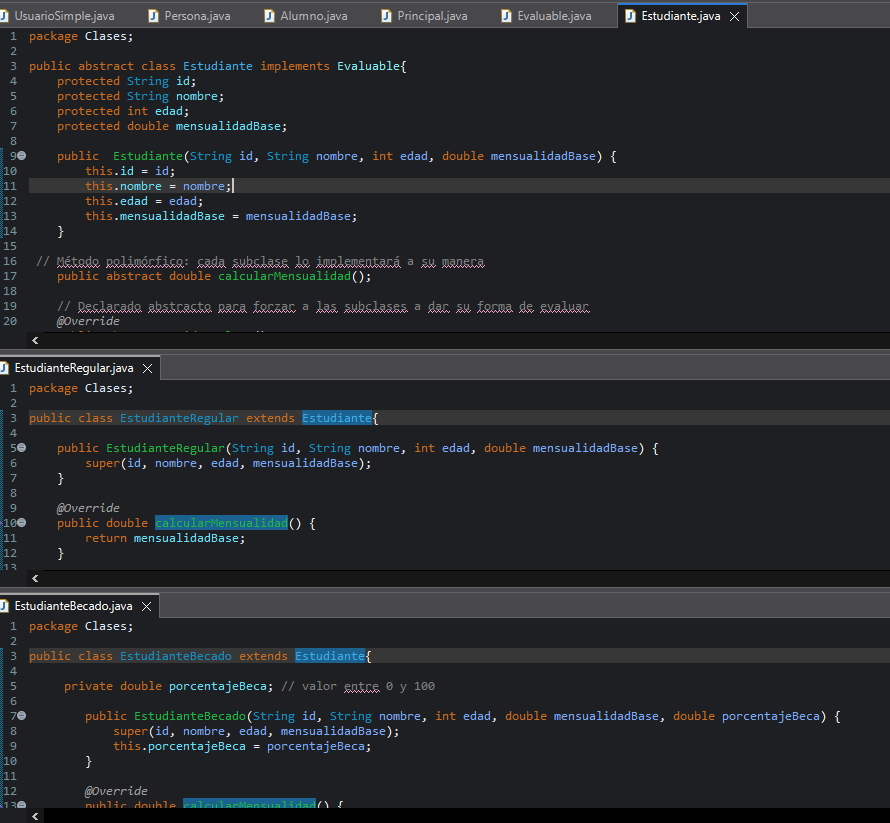
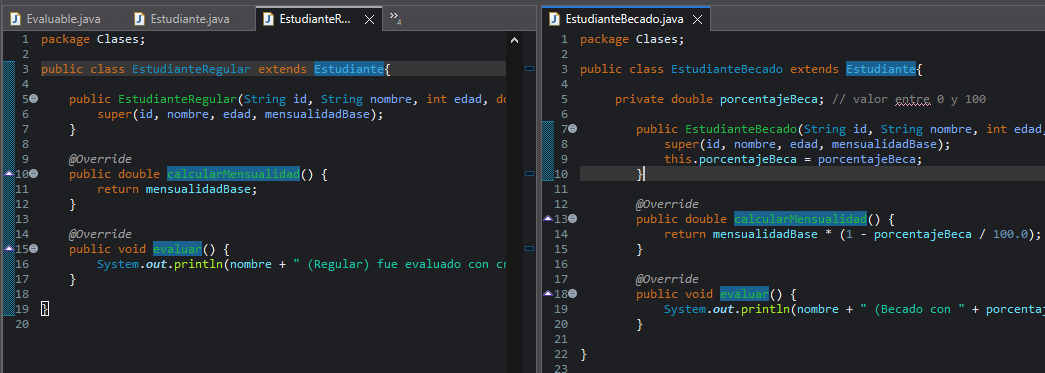
**Caso 2 – Gestión de Estudiantes**

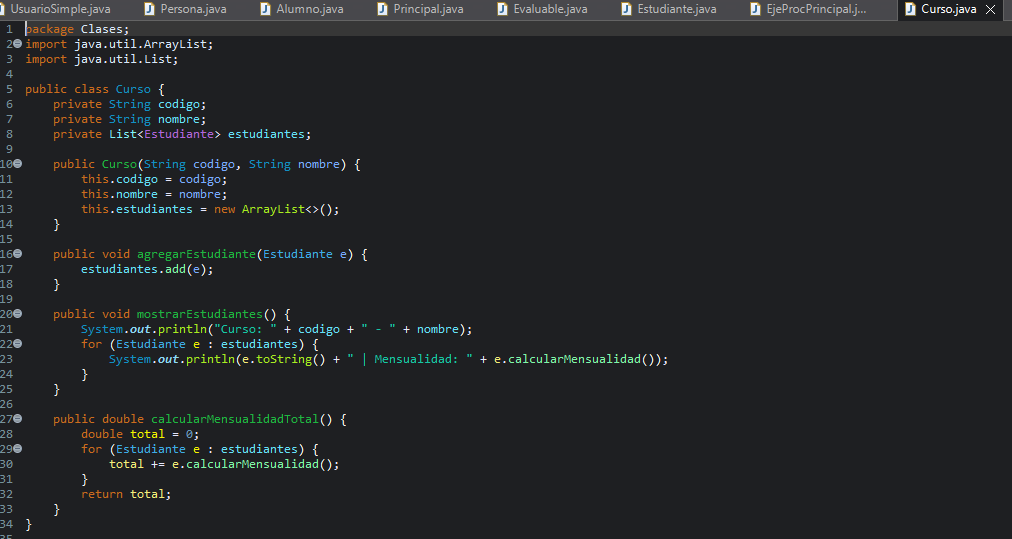
Un colegio requiere un sistema para gestionar estudiantes y cursos.

Requerimientos:

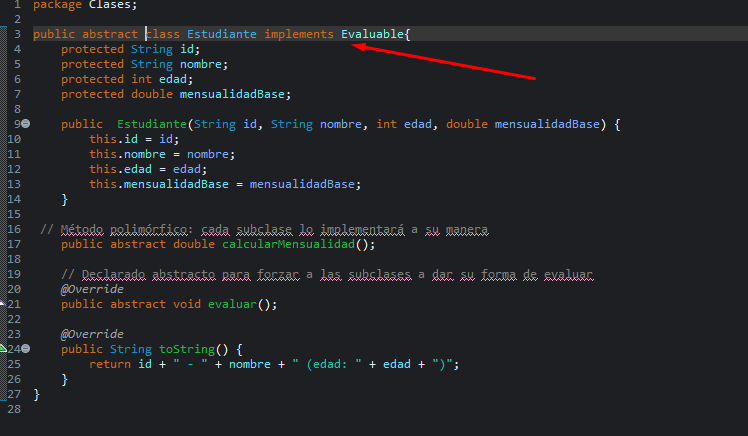
1. Crear la clase Estudiante y la clase Curso.
2. Establecer una relación entre ellas usando herencia y polimorfismo: - EstudianteRegular y EstudianteBecado deben heredar de Estudiante. - Cada uno debe implementar un método calcularMensualidad() (sobreescrito).

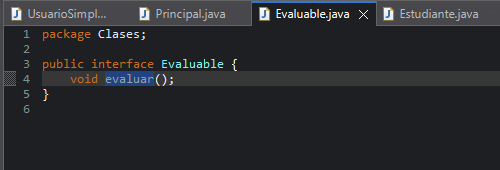
En la imagen se evidencia la herencia al observar que la clase EstudianteBecado y la clase EstudianteRegular heredan de la clase Estudiante.

Y el polimorfismo se evidencia cuando se usa el método calcularMensualidad y para ambos casos tienen diferentes comportamientos.

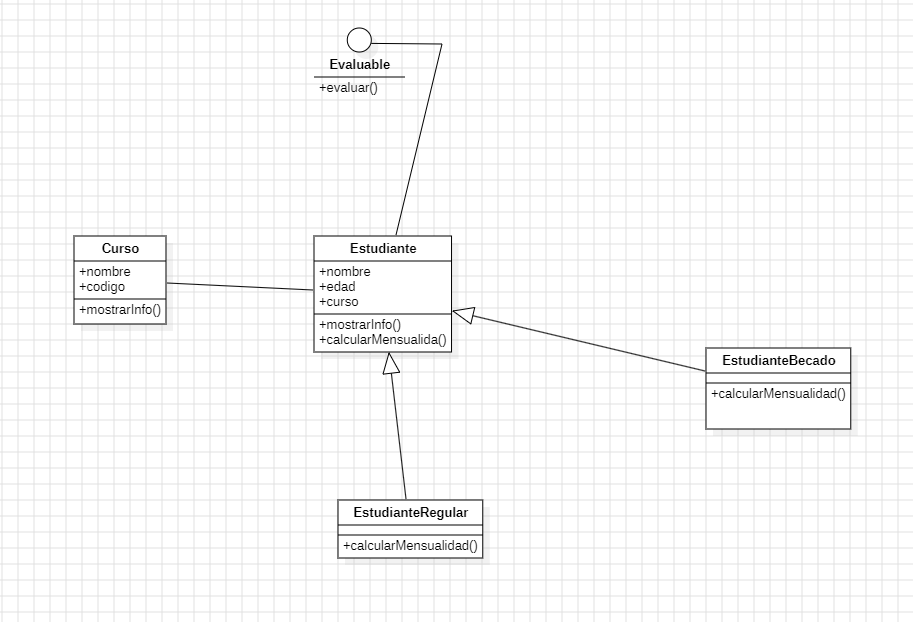


1. Usar interfaces para definir el contrato Evaluable, que incluya el método evaluar().





1. Representar las relaciones en un diagrama UML.



Preguntas de análisis: - ¿Por qué es útil aplicar herencia y polimorfismo en este sistema?

Es útil porque no ayuda a trabajar reutilizando codigo y creando jerarquías lógicas y en cuanto al polimorfismo podemos de alguna manera también ahorrar líneas de codigo ya que usamos mismos métodos que se comportan de diferente manera dependiendo que objeto los use.

- ¿Qué problemas podrían surgir si no se usaran interfaces en el diseño?

El codigo sería más rígido, habría lógica duplicada.

- ¿Qué diferencia hay entre atributos/métodos estáticos y no estáticos en este contexto?

Los atributos son las caracteristicas propias que tienen cada clase y los métodos no estáticos como **calcularMensualidad**() son aquellos que dependen de que objeto es llamado como en el caso de si es un estudiante regular o becado quien invoca al método mientras que un método estático no depende de un objeto en especifico estos métodos solo trabajan con atributos estáticos.

Ejecución final

