

Rúbrica de programación (segunda evaluación)

Categoría	Sobresaliente	Notable	Bien	Regular-Suficiente	No aceptable
Funcionamiento 40%	El programa funciona correctamente y todas las especificaciones de requisitos se cumplen. (Validaciones de variables, uso de librerías y métodos según los requisitos)	En muy pocos casos particulares no funciona pero cumple con las especificaciones del enunciado. Las variables de entrada se validan. No hay excepciones no controladas	En algunos casos no funciona el programa. alguna especificación del enunciado no está soportada en el código y no se validan todas las entradas. No hay excepciones no controladas	Sólo funciona en casos concretos, las variables no están validadas. Al menos la mitad de las especificaciones están operativas correctamente. No hay excepciones no controladas	No funciona en ningún caso. Tiene errores de compilación. Se producen excepciones no controladas.
Razonamiento lógico 45%	El código utiliza las estructuras adecuadas arrays, herencia, polimorfismo, colecciones, ficheros y está muy depurado. Las clases están bien definidas, los métodos están en la clase que les corresponde. Sigue POO.	El código utiliza las estructuras adecuadas arrays, herencia, polimorfismo, colecciones y ficheros. Las clases están bien definidas, los métodos están en la clase que les corresponde. Sigue POO.	El código no utiliza las estructuras adecuadas arrays, herencia, polimorfismo, colecciones y ficheros. Las clases están bien definidas, los métodos están en la clase que les corresponde. Sigue POO.	El código no utiliza las estructuras adecuadas arrays, herencia, polimorfismo, colecciones y ficheros. No todas las clases están bien definidas y hay métodos que no están en la clase que les corresponde. Sigue POO.	No hay razonamiento lógico. No sigue paradigmas de la POO. Estructuras no adecuadas y clases y métodos no bien definidos.
Estructuración del código 5%	El código tiene la estructura que facilita su lectura y seguimiento: sangrado (indentado), espacios e interlineado y comentarios. Nombres autoexplicativos de variables y procesos según las normas de estilo propuestas. Clases en sus archivos fuente adecuados.	El código tiene la estructura que facilita su lectura y seguimiento: sangrado (indentado), espacios e interlineado y comentarios. No todos los nombres de variables son autoexplicativos. Sigue las normas de estilo propuestas. Clases en sus archivos fuente adecuados.	El código no siempre tiene la estructura que facilita su lectura y seguimiento: sangrado (indentado), espacios e interlineado. No todos los nombres de variables son autoexplicativos. Sigue las normas de estilo propuestas. Clases en sus archivos fuente adecuados.	No tiene sangría sólo en algunos casos, no hay saltos de línea que clarifiquen la separación de funciones. Los nombres de variables no son autoexplicativos y hay muy pocos errores en el uso de las normas de estilo propuestas. Clases en sus archivos fuente adecuados.	El código está escrito en forma plana sin sangrías y sin saltos de línea. Los nombres de variables no son autoexplicativos y no sigue las normas de estilo propuestas. Clases mezcladas sin criterios de encapsulamiento.
Documentación 10%	La cabecera incluye el autor, enfoque de la solución del problema y los objetivos bien definidos. Las variables están descritas y las partes del código bien identificadas y documentadas.	Le faltan algunos datos en cabecera, los objetivos no están completos y alguna variable sin definir y alguna parte del código no está documentada.	La cabecera está incompleta, faltan objetivos, las variables sin definir y las partes del código no tienen comentarios apropiados.	No tiene casi cabecera, no tiene enunciado ni objetivos, no define la documentación de las partes del código.	No hay documentación.
Total	9 a 10	7 a 8,9	6 a 6,9	5 a 5,9	0 a 4,9