

I – Desviación de los Objetivos				
#	I.1 Desviación	Si	No	N/A
1	El código implementa correctamente el diseño ?	x		
2	El código implementa justo lo que establece el diseño ?	x		
3	El mecanismo de envío (valor o referencia) de todos los parámetros de cada método es apropiado ?	x		
4	Cada método retorna el valor correcto en cada punto de retorno ?	x		
II – Omisión de Objetivos				
#	II.1 Omisión	Si	No	N/A
5	El código implementa completamente el diseño ?	x		
	No hay restos de código innecesario o test de prueba en el código ?	x		
III – Defectos en los Objetivos				
#	III.1 Declaración de Variables y Constantes	Si	No	N/A
7	Los nombres de las variables y constantes son descriptivos y cumplen con las convenciones de nombres ?	x		
8	Los tipos de las variables son correctos ?	x		
9	Cada variables esta inicializada apropiadamente ?	x		
10	Todas las variables que controlan ciclos (ciclos for) están declaradas en la cabecera del ciclo ?			x
11	no hay variables que deberían se constantes ?	x		
12	no hay atributos que deberían ser variables locales ?	x		
13	Todos los atributos tienen un indicador de acceso apropiado (private, protected, public)?	x		
14	no Hay atributos estáticos (static) que no deberían serlo o viceversa ?	x		
#	III.2 Definición de Métodos	Si	No	N/A
15	Los nombres de los método son descriptivo y cumplen con las convenciones de nombres ?	X		
16	Todos los métodos tienen un indicador de acceso apropiado (private, protected, public) ?	X		
17	El valor de los parámetros de cada método es chequeado antes de usarlo ?	X		
18	Hay métodos estáticos (static) que no debieran serlo p viceversa ?	X		
#	III.3 Definición de Clases	Si	No	N/A
19	Cada clase tiene un constructor adecuado ?	X		
20	No Existe algunas subclases con miembros comunes que deberían estar en una superclase ?	X		
21	No Puede simplificarse la jerarquía de herencia de la clase ?		X	
#	III.4 Referencia a los Datos	Si	No	N/A

22	Para referencia a un arreglo los valores de los subíndices está dentro del rango permitido ?	X		
23	Se verifica que toda referencia a un objeto o arreglo no sea nula ?	X		
#	III.5 Expresiones y Tipos de Datos	Si	No	N/A

24	No hay ningún cálculo con tipos de datos mezclados ?	x		
25	Es imposible el overflow o el underflow, durante un cálculo ?	x		
26	Por cada expresión se respeta el orden de evaluación y precedencia correcta ?	x		
27	Se usan paréntesis para evitar ambigüedades ?			x
28	El código previene los errores por redondeo en forma sistemática			x
29	El código evita sumas y restas sobre números con magnitudes muy diferentes ?	x		
30	Se chequea la división por cero o el ruido ?			x
#	III.6 Comparación y Relaciones	Si	No	N/A
31	Las expresiones booleanas han sido simplificadas, usando "driving negations inward" (expresión negada) ?	X		
32	Cada prueba booleana chequea la condición correcta ?	X		
33	No hay comparaciones entre variables de tipos inconsistentes ?	X		
34	Son correctos los operadores de comparación ?	X		
35	Todas las expresiones booleanas son correctas ?	X		
36	No Existen efectos colaterales inapropiados de una comparación ?	X		
37	Se intercambiado un "&" por un "&&" ó un " " por un " " ?	X		
38	El código evita la comparación de igualdad en números de punto flotante ?	X		
39	Están cubiertas las tres ramas de los if (menor, igual, mayor)	X		
#	III.7 Control de Flujo	Si	No	N/A
40	Por cada ciclo se usa la mejor elección de construcción de ciclos ?	x		
41	Todos los ciclos terminan ?	x		
42	Cuando un ciclo tiene múltiples condiciones de salida todas están manejadas apropiadamente ?			x
43	Todas las sentencias SWITCH tienen un caso por defecto ?			X
44	Las salidas de un Switch no manejadas están debidamente comentadas y con una sentencia break ?			X
45	Es correcta la profundidad en el anidamiento de ciclos ?	x		
46	no es necesario convertir algún if anidado en sentencias SWITCH ?		x	
47	Los cuerpos nulos en las estructuras de control están marcados con llaves, marcados y comentados correctamente?	x		
48	Todos los métodos terminan ?	x		
49	Todas las excepciones son manipuladas apropiadamente ?	x		
50	Las sentencias break con con etiqueta derivan el control al lugar correcto ?			x
#	III.8 Entrada/Salida	Si	No	N/A

51	Todos los archivos se abren antes de usarlos ?			X
52	Los atributos de las sentencias de apertura de los archivos son consistente con el uso de los mismos ?			X
53	Todos los archivos se cierran cuando dejan de usarse ?			X
54	Los datos en el buffer se envían al disco ?			X

55	No Hay errores de ortografía o gramática en el texto impreso o en la pantalla ?	X		
56	Están chequeadas las condiciones de error ?	X		
57	Se verifica la existencia de los archivos antes de intentar abrirlos ?			X
58	Todas las excepciones de entrada/salida están razonablemente manejadas ?	X		
#	III.9 Interface del Módulo	Si	No	N/A
59	El número, orden, tipo y valores de parámetros en cada llamada de un método esta de acuerdo con la declaración del método ?	X		
60	Los valores respetan los acuerdos de unidades (por.ej., pulgadas versus yardas) ?	X		
61	Si un objeto o arreglo es pasado a un método que lo altera, esta alteración es realizada correctamente por dicho método ?	X		
#	III.10 Comentarios	Si	No	N/A
62	Todos los métodos, clases y archivos tienen los comentarios de cabecera apropiados ?	X		
63	Cada atributo, variable o declaración de constante ha sido comentada ?	X		
64	El comportamiento de cada método y clase es expresado en lenguaje plano ?	x		
65	Los comentarios en la cabecera de cada método y clase son consistentes con el comportamiento del método o clase ?	x		
66	Todos los comentarios son consistentes con el código ?	x		
67	Los comentarios ayudan a entender el código ?	x		
68	Hay suficientes comentarios en el código ?	x		
69	no hay demasiados comentarios en el código ?	x		
#	III.11 Diseño y Empaquetado	Si	No	N/A
70	El formato standard en el diseño e indentación del código es usado consistentemente ?	X		
71	Ningún método excede las 60 líneas ?	X		
72	Ningún módulo excede las 600 líneas ?	X		
#	III.12 Modularidad	Si	No	N/A
73	Hay un bajo nivel de acoplamiento entre módulos (métodos y clases) ?		X	
74	Hay un alto nivel de cohesión en cada módulo (métodos y clases) ?	X		
75	No hay código repetido que se puede reemplazar por un método que implemente el comportamiento de dicho código ?	X		
76	Se usan las librerías de clase java cuando y donde deben usarse ?	X		
#	III.13 Almacenamiento	Si	No	N/A

77	Los arreglos tienen previsto el tamaño suficiente ?			X
78	Las referencias a los objetos y arreglos son seteados a nulo una vez que dejan de usarse?			x
#	III.14 Perfomance	Si	No	N/A
79	No pueden mejorarse las estructuras de datos o usar algoritmos más eficientes ?		X	
80	Los test lógicos están organizados, de manera que los más frecuentes y caros estén primero ?	X		
81	No puede reducirse el costo de recálculo mediante el almacenamiento de los resultados ?	X		

82	Actualmente, se usa cada resultado calculado y almacenado ?	X		
83	No puede un cálculo sacarse fuera de un ciclo ?	X		
84	No hay test dentro de un ciclo que no necesitan ser realizados ?	X		
85	No puede un ciclo corto ser convertido en una estructura más simple ?	X		
86	No hay Dos ciclos sobre los mismos datos se pueden combinar en uno?	X		

IV – Inconsistencia en los Objetivos

#	IV.1 Perfomance	Si	No	N/A
87	No Hay algún código implementado en modo inconsistente ?	x		

V – Ambigüedad en los Objetivos

#	V.1 Declaración de Variables y Constantes	Si	No	N/A
88	no hay ninguna variable con nombres similares y confusos ?	x		
89	Todas las variables están definidas con nombres claros, consistentes y significativos ?	x		
#	V.2 Perfomance	Si	No	N/A
90	No existen módulos excesivamente confusos que se pueden reestructurar o dividir en varias rutinas ?	X		

VI – Redundancia en los Objetivos

#	VI.1 Variables	Si	No	N/A
91	No Existen variables o atributos redundante o no usados ?	X		
92	No necesario que alguna variable no local convertirse en local ?	X		
#	VI.2 Definición de Métodos	Si	No	N/A
93	Hay algunos metodos que no son llamados o son innecesarios ?			
#	VI.3 Perfomance	Si	No	N/A
94	no es necesario que algún código reemplazarse con llamadas a objetos externos reusables ?	x		
95	No existen bloques de código repetidos que pueden condensarse en un método simple ?	x		
96	No hay restos de código no usado o restos de rutinas de test ?	x		

VII – Efectos Colaterales en los

Objetivos				
#	VII.1 Definición de Métodos	Si	No	N/A
97	Después de cambiar un método se analizan los métodos que lo llaman			X
#	VII.2 Base de Datos	Si	No	N/A
98	El proceso de actualización y migración sigue el cambio de estructuras o contenidos en la base del proyecto ?			x