Paquete datos

Clase Lote:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?	Х	
¿Hay métodos estáticos que no debieran serlo, o viceversa?	1	Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	1	Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?	1	Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	X	
¿Hay comparaciones entre variables de tipos inconsistentes?	†	Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?	†	X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?	\top	Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	X	

Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? X nterfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Ayudan los comentarios a entender el código? X Hay sufficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Diseño Si No Se utiliza una indentación estándar?	\(\frac{\lambda}{\lambda}\)
Poseen fin todos los bucles? Poseen todas las estructuras switch un default case? Son correctos todos los breaks dentro de una estructura switch? Resultan las anidaciones de bucles correctas y necesarias? Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X Dodos los métodos terminan? Kantrada/Salida Si Na Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? X son todas las excepciones de entrada/salida manejadas de la manera correcta? X terfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? X coinciden las unidades de los valores? X i un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X constructores y los métodos sensibles de una clase están expuestos al acceso? X correctora no bjetos nulos? Son correctas el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? X constructores y los métodos sensibles de una clase están expuestos al acceso? X correctora no bjetos nulos? Son expresados correctamente los comportamientos de cada método? X posee cada declaración de atributo y constante un comentario? X posee cada declaración de atributo y constante un comentario? X pose comentarios de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X pon todos los comentarios consistentes con el código? X puduan los comentarios a entender el código? X puduan los comentarios conentarios en el código? X puduan los comentarios comentarios en el código? X puduan l	\(\frac{\lambda}{\lambda}\)
Poseen todas las estructuras switch un default case? Son correctos todos los breaks dentro de una estructura switch? Resultan las anidaciones de bucles correctas y necesarias? X Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X Todos los métodos terminan? X Intrada/Salidia Si Nal Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? X Son todas las excepciones de entrada/salida manejadas de la manera correcta? X Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X iun objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Son experonan objetos nulos? Son cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Son expresados correctamente los comportamientos de cada método? X Son expresados correctamente los comportamientos de cada método? X Son expresados correctamente los comportamientos de cada método? X Son todos los comentarios a entender el código? X Ayudan los comentarios a entender el código? X Ayudan los comentarios a entender el código? X Ayudan los comentarios en el código? X Son todos los comentarios en el código?	\(\frac{\lambda}{\lambda}\)
Resultan las anidaciones de bucles correctas y necesarias? Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X intrada/Salida Si N Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? X son todas las excepciones de entrada/salida manejadas de la manera correcta? X neterfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X iu no objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? Los constructores y los métodos sensibles de una clase están expuestos al acceso? X se retornan objetos nulos? Comentarios Son expresados correctamente los comportamiento y apropiado comentario en la cabecera? X pose cada declaración de atributo y constante un comentario? X pose cada declaración de atributo y constante un comentario? X es el comentario de la cabecera de cada calase o método correspondiente con el comportamiento de la misma? X es el comentarios de la cabecera de cada case o método correspondiente con el comportamiento de la misma? X pose comentarios a entender el código? X yudan los comentarios a entender el código? X yudan los comentarios a entender el código? X yudana los comentarios en el código? X yudana los una indentación estándar? X yudana los una indentación estándar?	\(\frac{\lambda}{\lambda}\)
Resultan las anidaciones de bucles correctas y necesarias? Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X intrada/Salida Si N Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? X son todas las excepciones de entrada/salida manejadas de la manera correcta? X neterfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X iu no objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? Los constructores y los métodos sensibles de una clase están expuestos al acceso? X se retornan objetos nulos? Comentarios Son expresados correctamente los comportamiento y apropiado comentario en la cabecera? X pose cada declaración de atributo y constante un comentario? X pose cada declaración de atributo y constante un comentario? X es el comentario de la cabecera de cada calase o método correspondiente con el comportamiento de la misma? X es el comentarios de la cabecera de cada case o método correspondiente con el comportamiento de la misma? X pose comentarios a entender el código? X yudan los comentarios a entender el código? X yudan los comentarios a entender el código? X yudana los comentarios en el código? X yudana los una indentación estándar? X yudana los una indentación estándar?	\(\frac{\lambda}{\lambda}\)
Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? Todos los métodos terminan? Kantirada/Salida Si Na Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? Atterfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la teclaración del mismo? Coinciden las unidades de los valores? X iu no objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X los constructores y los métodos sensibles de una clase están expuestos al acceso? X se retornan objetos nulos? Comentarios Son expresados correctamente los comportamientos de cada método? X se sel comentario de la tributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? X se el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X son todos los comentarios a entender el código? Ayudan los comentarios a entender el código? Hay demasiados comentarios en el código? Hay demasiados comentarios en el código? Si Na Si Si Na Si	\(\frac{\lambda}{\lambda}\)
Son todas las excepciones manejadas correctamente? Todos los métodos terminan? X Intrada/Salida Si N Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? X Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X iu no objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Son expresados correctamente los comportamientos de cada método? X Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? X Ayudan los comentarios a entender el código? Hay demasiados comentarios en el código? Kayudan los comentarios en el código? Kayudan los demasiados comentarios en el código? Kayudan los demasiados comentarios en el código? Kayudan los una entender el código? Kayudan los comentarios en el código? Kayudan los comentarios en el código? Kayudan los una indentación estándar? Kayudan los una indentación estándar?	\(\frac{\lambda}{\lambda}\)
Todos los métodos terminan? Intrada/Salida Si N Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Non todas las excepciones de entrada/salida manejadas de la manera correcta? Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X i un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X So retornan objetos nulos? Son dad clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? X Ayudan los comentarios a entender el código? X Ayudan los comentarios a entender el código? Hay seficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Sieño Se utiliza una indentación estándar?	No (
Intrada/Salida Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? I un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Si Nodada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Ayudan los comentarios consistentes con el código? X Ayudan los comentarios e nel código? X Hay demasiados comentarios en el código? X Sieño Si Notician la mantarios en el código? X Sieño Si Notician la mantarios en el código? X Sieño Si Notician la mantarios en el código? X Sieño Si Notician la mantarios en el código? X Sieño Si Notician la mantarios en el código? X Sieño Si Notician la mantarios en el código? X Sientiliza una indentación estándar?	<
Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? Xon todas las excepciones de entrada/salida manejadas de la manera correcta? Xon todas las excepciones de entrada/salida manejadas de la manera correcta? X Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Ayudan los comentarios a entender el código? X Hay sufficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Si on todas los indentación estándar? X Si on todas los comentarios en el código? X Bay demasiados comentarios en el código? X Si on todas los comentarios en el código? X Si on todas los comentarios en el código? X Bay utiliza una indentación estándar? X	(
Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? Xon todas las excepciones de entrada/salida manejadas de la manera correcta? Xon correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? X un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? X comentarios X ada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X posee cada declaración de atributo y constante un comentario? X posee cada declaración de atributo y constante un comentario? X son expresados correctamente los comportamientos de cada método? X Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Ayudan los comentarios consistentes con el código? X Ayudan los comentarios a entender el código? X Hay suficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X biseño Si Nose utiliza una indentación estándar?	
Las salidas están siendo corroboradas que sean las correctas? Son todas las excepciones de entrada/salida manejadas de la manera correcta? Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? Los constructores y los métodos sensibles de una clase están expuestos al acceso? Son exterornan objetos nulos? Somentarios Si Nocada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentarios de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Ayudan los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay demasiados comentarios en el código? X Biseño Si Noca expresados correctamentarios en el código? X Si Noca expresados correctamentarios en el código? X Si Noca expresados comentarios en el código? X Ayudan los comentarios a entender el código? X Si Noca expresados comentarios en el código? X Si Noca expresados entender el c	
Son todas las excepciones de entrada/salida manejadas de la manera correcta? **Neterfaz de módulos** **Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? **Coinciden las unidades de los valores?** **I i un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?** **Los constructores y los métodos sensibles de una clase están expuestos al acceso?** **Se retornan objetos nulos?** **Sonentarios** **Sonentarios** **Posee cada declaración de atributo y constante un comentario?** **Son expresados correctamente los comportamientos de cada método?** **Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?** **Son todos los comentarios consistentes con el código?** **Ayudan los comentarios a entender el código?** **Hay suficiente cantidad de comentarios en el código?** **Hay demasiados comentarios en el código?** **Hay demasiados comentarios en el código?** **Sin Notados los comentarios en el código?** **Son todos los comentarios en el código?** **Hay demasiados comentarios en el código?** **Hay demasiados comentarios en el código?** **Son todos los comentarios en el código?** **Son todos los comentarios en el código?** **Hay demasiados comentarios en el código?** **Hay demasiados comentarios en el código?** **Son todos los c	
As a constructores y los métodos sensibles de una clase están expuestos al acceso? Son expresados correctamente los comportamientos de cada método? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios a entender el código? Ay demasiados comentarios en el código? Si Na Se de tutiliza una indentación estándar?	
Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la leclaración del mismo? Coinciden las unidades de los valores? ii un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? Los constructores y los métodos sensibles de una clase están expuestos al acceso? Se retornan objetos nulos? Si Nacida clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Ayudemasiados comentarios en el código? X Si Nacidada comentarios en el código? X Si Nacida clase es modificado correctamente? X	
leclaración del mismo? Coinciden las unidades de los valores? Is un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? Los constructores y los métodos sensibles de una clase están expuestos al acceso? Se retornan objetos nulos? Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? X Si No Se utiliza una indentación estándar?	
Coinciden las unidades de los valores? ii un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Ayudemasiados comentarios en el código? X Sieño Se utiliza una indentación estándar? X X X X X X X X X X	
iun objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Comentarios Si N Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? X Diseño Se utiliza una indentación estándar? X X X X X X X X X X X	
Los constructores y los métodos sensibles de una clase están expuestos al acceso? Se retornan objetos nulos? Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? Si Nose utiliza una indentación estándar?	
Se retornan objetos nulos? Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? Si Noseño Se utiliza una indentación estándar?	
Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? Si Noseño Se utiliza una indentación estándar?	
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? Si Noseño Se utiliza una indentación estándar?	No
Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Siseño Se utiliza una indentación estándar?	
Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? X Diseño Se utiliza una indentación estándar? X	
Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Diseño Si N Se utiliza una indentación estándar?	<u> </u>
Son todos los comentarios consistentes con el código? Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? X Diseño Se utiliza una indentación estándar? X X	
Ayudan los comentarios a entender el código? Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? Si No Se utiliza una indentación estándar?	<u>`</u>
Hay suficiente cantidad de comentarios en el código? Hay demasiados comentarios en el código? Si N Se utiliza una indentación estándar?	
Hay demasiados comentarios en el código? Si N Se utiliza una indentación estándar? X	
Diseño Si N Se utiliza una indentación estándar? X	
Se utiliza una indentación estándar?	No.
Para cada método: ¿No supera las 60 líneas de largo?	
· •	No
Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) X	
Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	
Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	
lel código repetitivo?	
Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X	
	No
Son los arreglos suficientemente largos? X	
Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	<u> </u>
Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?	No
Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	
Son todos los valores calculados y almacenados verdaderamente utilizados? X	<

¿Puede un cálculo ser realizado fuera de un ciclo?		Х
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el método public Pedido getPedido() se utiliza el atributo pedido sin verificar que no sea null.

En el método **public String detalles()** se realiza el catch de ArgumentollegalException pero no se realiza nada con ella.

Faltan comentarios y JavaDoc en:

- Cabecera de la clase
- Atributos de la clase
- Constructor de la clase
- Método public String toString()
- Método public String detalles()
- Método public Pedido getPedido()

Clase material:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?	Х	
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	X	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	X	
Flujo de control	Si	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	X	
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	X	
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	Х	1
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	
¿Se retornan objetos nulos?		X
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?	1	X
¿Posee cada declaración de atributo y constante un comentario?	1	X
¿Son expresados correctamente los comportamientos de cada método?	1	X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?	1	X
¿Hay suficiente cantidad de comentarios en el código?	<u> </u>	X
¿Hay demasiados comentarios en el código?	1	X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	110
Para cada método: ¿No supera las 60 líneas de largo?	1	X
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	<u> </u>	X
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	X	+~-
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	+~	X
del código repetitivo?		^
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	+
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	1
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	+	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+^-
¿Puede un cálculo ser realizado fuera de un ciclo?	+^-	X
Ci dede un calculo ser realizado faera de all ciclo;		

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el método **private boolean verificaCodigo(String código)** no verifica que el parámetro **codigo** sea distinto de null antes de acceder al método length() correspondiente a la clase String.

En el método **private boolean verificaDescripcion(String descripcion)** no verifica que el parámetro **descripcion** sea distinto de null antes de acceder al método length() correspondiente a la clase String.

Ni la clase ni ningún método ni declaraciones de atributos ni variables poseen sus correspondientes comentarios.

Comentarios:

- Solamente los constructores están abiertos al acceso para que el objeto pueda ser creado, los métodos sensibles no.
- Los atributos nunca pueden ser retornados como null mediante un método getter debido a que si estos son establecidos de manera incorrecta anteriormente, el programa finalizara mediante una excepción.
- Los métodos necesarios poseen el acoplamiento correcto con los demás métodos, aquellos que no lo necesitan no.

Clase pedido:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		X
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		X
¿Puede ser la jerarquía de herencia simplificada?		X
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?	<u> </u>	X
¿Existe posibilidad de overflow o underflow durante un cálculo?		X
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	1
correctas?	,	
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	X	
¿Hay comparaciones entre variables de tipos inconsistentes?	1	Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No
rujo de control) JI	INU

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	<u> </u>	X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	X	
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?	X	
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?	X	
¿Son todos los comentarios consistentes con el código?	X	
¿Ayudan los comentarios a entender el código?	X	
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?		X
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?	1	X
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)		Х
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	X	1
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	+	X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	110
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	†	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+
¿Puede un cálculo ser realizado fuera de un ciclo?	+	X
C. wowe an outout of realizado racia de an oldio.		

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		X
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		X
¿Existe algún objeto que sea innecesario o redundante?		X
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		X

El atributo estadoActual no posee inicialización en null.

En el método **private boolean verificaNumeroPedido(String numeroPedido)** el parámetro numeroPedido no se verifica que sea distino de null ni que sea de la longitud correcta. La longitud de **numeroPedido** debería ser verificada en este método y no en el método **private boolean verifica(String str)**.

El atributo estadoActual no posee ningún valor antes del constructor.

Métodos que no poseen comentarios:

- evaluarPedido()
- detalles()

Faltan comentarios a las declaraciones de variables, atributos y/o constantes.

Comentarios:

- Solamente los constructores están abiertos al acceso para que el objeto pueda ser creado, los métodos sensibles no.
- Los atributos nunca pueden ser retornados como null mediante un método getter debido a que si estos son establecidos de manera incorrecta anteriormente, el programa finalizara mediante una excepción.
- Los métodos necesarios poseen el acoplamiento correcto con los demás métodos, aquellos que no lo necesitan no.

Clase Observación:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?	1	Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	X	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	X	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	

¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	!
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	X	
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	Х	
declaración del mismo?		
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?	Х	
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?	Х	
¿Son todos los comentarios consistentes con el código?	Х	
¿Ayudan los comentarios a entender el código?	Х	
¿Hay suficiente cantidad de comentarios en el código?		Х
¿Hay demasiados comentarios en el código?		Х
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?	X	
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	Х	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	1	Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	1	X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	+	Х
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	1	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+**
¿Puede un cálculo ser realizado fuera de un ciclo?	+	X
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?	+	X
C. acad accadinate an pacific conto de tai forma de certai la danzación de un pacific:		

¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		X
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		X
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

Falta aclarar que los parámetros de la clase se suponen distintos de null en todo momento luego de su creación, ya que son verificados e inicializados en el constructor.

En el método **private boolean verificaTexto(String texto)** se utiliza el parametro **texto** sin verificar si es o no null.

En el método **public int CompareTo(Object object)** se utiliza el parámetro **object** sin verificar si es o no null.

Faltan comentarios y JavaDoc en:

- Atributos de la clase
- Método public boolean verificacion()
- Método String toString()
- Método compareTo(Object object)

Clase tipoProducto:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		X
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	X	
¿Están todas las variables correctamente inicializadas?	X	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		X
¿Hay atributos que debieran ser variables locales?		X
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		X
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	X	
¿Son los operadores de las comparaciones correctos?	X	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		Х
¿Son todas las excepciones manejadas correctamente?	Х	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		Х
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	Х	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Х	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	Х	
declaración del mismo?		
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	
¿Se retornan objetos nulos?	X	
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?		Х
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		Χ
¿Son todos los comentarios consistentes con el código?		Χ
¿Ayudan los comentarios a entender el código?		Х
¿Hay suficiente cantidad de comentarios en el código?		Х
¿Hay demasiados comentarios en el código?		Х
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?		Х
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)		Х
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	Х	
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		Х
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	Х	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	Х	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	1	Х
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	
-	1.00	1

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Χ
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

El método **public void setListaMat(ListaMateriales listaMat)** no verifica que el parámetro **listaMat** sea distinto de null antes de asignarlo al atributo **this.listaMat.**

Los parámetros pasados en el constructor no son verificados que sean distintos de null.

Los parámetros no son verificados de ser distintos de null, por lo tanto mediante los getters se puede obtener un objeto null indeseadamente.

No hay comentarios ni en la clase, ni en ningún método, ni en ningún parámetro o constante.

En el método **public void generarTipoProd()** la variable **aux** no es setada en null luego de ser utilizada.

Comentarios:

• El constructor posee un modificador de acceso público para que sea posible crear el objeto.

Paquete datos.estadosPedido

<u>Clase Estado:</u> falta agregar comentarios

Clase Aceptado:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	X	
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х

¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	X	
Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	\bot
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	X	
declaración del mismo?		
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?		Х
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		Х
		Х
¿Son todos los comentarios consistentes con el código?	+	Х
¿Ayudan los comentarios a entender el código?		Х
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código?		
¿Ayudan los comentarios a entender el código?	Si	X X No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño		Х
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar?	Х	Х
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo?	X	X No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad	X X Si	Х
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	X	X No No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	X X Si	X No No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos) ¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	X X Si	X No No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos) ¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo?	X X Si X	X No No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos) ¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? ¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X X Si X	No No X
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos) ¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? ¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento	X X Si X	X No No
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos) ¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? ¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento ¿Son los arreglos suficientemente largos?	X X Si X	X No No X X
¿Ayudan los comentarios a entender el código? ¿Hay suficiente cantidad de comentarios en el código? ¿Hay demasiados comentarios en el código? Diseño ¿Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) ¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos) ¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? ¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento	X X Si X	No No X

	Х
X	
	Х
	Х
	Х
	Х
Х	
	Х
	Х
	Х
	Х
	Х

En el constructor de la clase no se verifica que el parámetro **pedido** sea distinto de null.

En el método **public void agregarObservacion(Observacion obs)** no se verifica que el parámetro **obs** sea distinto de null, aunque no es utilizado.

Faltan comentarios en:

- Cabecera de la clase
- Constructor de la clase
- Todos los métodos

Clase evaluación:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?	<u> </u>	X
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	X	
¿Están todas las variables correctamente tipeadas?	X	
¿Están todas las variables correctamente inicializadas?	X	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	X	
¿Hay variables que debieran ser constantes?		X
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	\top
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	X	
Siempre que se utiliza toString: ¿Es sobrescrito?	X	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	X	
¿Posee cada clase un constructor apropiado?	X	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		X
¿Puede ser la jerarquía de herencia simplificada?		X
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	X	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	1	Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?	J	X
¿Existe posibilidad de overflow o underflow durante un cálculo?		X
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	X	+
correctas?	^	
¿Se usan paréntesis para evitar ambigüedad?	X	-
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	X	140
¿Son los operadores de las comparaciones correctos?	X	+
¿Hay comparaciones entre variables de tipos inconsistentes?	+^	X
¿Existen efectos secundarios inesperados resultantes de las comparaciones?	+	X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?	+	X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	X	+^-
	Si	No
Flujo de control	اد	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		Х
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		Х
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		Х
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	+
	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	 ^	- V
¿Se retornan objetos nulos? Comentarios	Si	X
	31	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		X
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?		X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		X
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?		X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	
Para cada método: ¿No supera las 60 líneas de largo?		Х
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	X	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)		Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo?		X
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	1.0
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?	J	X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	+	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	1
¿Puede un cálculo ser realizado fuera de un ciclo?		X
C. dede di. Galedio dei Tedifeddo Taera de difficio.		

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		X
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		X
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el constructor de la clase no se verifica que el parámetro **pedido** sea distinto de null.

Clase EstadoBase:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		<u> </u>
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?	T	Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	Х	
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?	†	Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?	Ī	Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No

		_
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	
¿Poseen fin todos los bucles?	Х	
¿Poseen todas las estructuras switch un default case?	Х	
¿Son correctos todos los breaks dentro de una estructura switch?	Х	
¿Resultan las anidaciones de bucles correctas y necesarias?	Х	
¿Puede modificarse alguna anidación de if por una estructura switch?		Х
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	Х	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Χ	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?	JI	X
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?		X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		X
¿Hay suficiente cantidad de comentarios en el código?		X
		X
¿Hay demasiados comentarios en el código?	C:	
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	
Para cada método: ¿No supera las 60 líneas de largo?	X	A1.
Modularidad (1.1.2.48)	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	X	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	<u> </u>	X
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?	ļ.,	
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	0:	X
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		Х
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		Х
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	1
¿Puede un cálculo ser realizado fuera de un ciclo?		Х

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el constructor de la clase no se verifica que el parámetro **pedido** sea distinto de null.

Faltan comentarios en:

- Cabecera de la clase
- Constructor de la clase
- Atributos de la clase
- Todos los métodos

Clase Iniciado:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	X	
¿Están todas las variables correctamente tipeadas?	X	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	X	
Siempre que se utiliza toString: ¿Es sobrescrito?	X	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	X	
¿Posee cada clase un constructor apropiado?	X	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?	1	X
¿Puede ser la jerarquía de herencia simplificada?	+	X
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	X	110
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	X	
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?	J.	X
¿Existe posibilidad de overflow o underflow durante un cálculo?	+	X
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	X	 ^
correctas?	^	
¿Se usan paréntesis para evitar ambigüedad?	X	+
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	X	
¿Son los operadores de las comparaciones correctos?	X	+
¿Hay comparaciones entre variables de tipos inconsistentes?	+^-	X
¿Existen efectos secundarios inesperados resultantes de las comparaciones?	+	X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?	+	X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	X	
Flujo de control	Si	No
Trujo de Control	اد	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	Х	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		Х
¿Son todas las excepciones manejadas correctamente?	Х	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		Х
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		Х
¿Las salidas están siendo corroboradas que sean las correctas?	Х	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Х	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	Х	
declaración del mismo?		
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo, variable y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?		Х
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		Х
¿Son todos los comentarios consistentes con el código?		Х
¿Ayudan los comentarios a entender el código?		Х
¿Hay suficiente cantidad de comentarios en el código?	1	Х
¿Hay demasiados comentarios en el código?	1	X
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?	X	1
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	Х	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	1	Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	1	X
del código repetitivo?		``
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	†	X
L CONTIDO FEIETENCIDO A ANTERAS A DIFERIAS ESTANIERIADO EN UNIO ANA ARE ESTAS NA SE HERESTADA MAS.	Si	No
Performance	31	Х
Performance ¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?	31	X
Performance	X	X

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		X
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		X

En el constructor de la clase no se verifica que el parámetro **pedido** sea distinto de null.

En el método **public void agregarObservacion(Observacion obs)** no se verifica que el parámetro **obs** sea distinto de null, aunque no es utilizado.

En el método **public void evaluarPedido()** no se verifica que el atributo de clase **pedido** sea distinto de null antes de utilizarlo.

Faltan comentarios en:

- Cabecera de la clase
- Constructor de la clase
- Todos los métodos

Paquete exceptions

<u>Clase ArgumentollegalException:</u>

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		X
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	

Por cada bucte: £8 Is mejor opción de construcciones de bucles elegida? Poseen fin todos los bucles? Poseen fin todos los bucles? Poseen todos las estructuras switch un default case? Poseen todos las estructuras switch un default case? Pose correctos todos los breaks dentro de una estructura switch? Puede modificarse alguna anidación de if por una estructura switch? Puede modificarse alguna anidación de if por una estructura switch? Puede modificarse alguna anidación de if por una estructura switch? Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de los parámetros switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Puede se número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la valuación de la mismo? X Puede se número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la valuación de la mismo? X Puede se número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la valuación de la mismo al valuació	Flujo de control	Si	No
Poseen fin todos los bucles? Poseen todos las estructuras switch un default case? Poseen todos las estructuras switch un default case? Son correctos todos los breaks dentro de una estructura switch? Resultan las anidaciones de bucles correctas y necesarias? Resultan las anidaciones de bucles correctas y necesarias? Resultan las anidaciones de bucles correctas y necesarias? Y 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		Х	
Poseen todas las estructuras switch un default case? Son correctos todos los breaks dentro de una estructura switch? Son correctos todos los breaks dentro de una estructura switch? Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X Puede modificarse alguna anidación de if por una estructura switch? X Todos los métodos terminan? X Las salidas están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las salidas están siendo corroboradas que sean las correctas? X Las	¿Poseen fin todos los bucles?		
Resultan las anidaciones de bucles correctas y necesarias? Puede modificarse alguna anidación de if por una estructura switch? X Puede modificarse alguna anidación de if por una estructura switch? X Poson todas las excepciones manejadas correctamente? X Poson todas las excepciones manejadas correctamente? X Poson todas las excepciones manejadas correctamente? X Poson todas las excepciones anidación corroboradas por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Poson todas las excepciones de entrada/salida manejadas de la manera correcta? X Poson todas las excepciones de entrada/salida manejadas de la manera correcta? X Poson correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Poson correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Poson correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración el su nuidades de los valores? X Poson correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración el mismo? Poson con arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Poso constructores y los métodos sensibles de una clase están expuestos al acceso? X Poso es cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declaración de atributo y constante un comentario? X Pose e cada declarac	¿Poseen todas las estructuras switch un default case?	Х	
Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X Controdas las excepciones manejadas correctamente? X Controdas las excepciones manejadas correctamente? X Controdas las manejadas correctamente? X Controdas las manejadas en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de la manera correctamente? X Controdas las excepciones de la manera correcta? X Controdas las excepciones de la manera correctamente? X Controdas las excepciones de la manera correcta? X Controdas las excepciones de la manera correcta? X Controdas las excepciones de la manera correctamente? X Controdas la	¿Son correctos todos los breaks dentro de una estructura switch?	Х	
Puede modificarse alguna anidación de if por una estructura switch? Son todas las excepciones manejadas correctamente? X Controdas las excepciones manejadas correctamente? X Controdas las excepciones manejadas correctamente? X Controdas las manejadas correctamente? X Controdas las manejadas en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de entrada/salida manejadas de la manera correcta? X Controdas las excepciones de la manera correctamente? X Controdas las excepciones de la manera correcta? X Controdas las excepciones de la manera correctamente? X Controdas las excepciones de la manera correcta? X Controdas las excepciones de la manera correcta? X Controdas las excepciones de la manera correctamente? X Controdas la	¿Resultan las anidaciones de bucles correctas y necesarias?	Х	
Son todas las excepciones manejadas correctamente? X			X
Endos los métodos terminan? Intrada/Salida Intrada/Salida Isi No Intrada/Salida Isi No Italy arrores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Están siendo corroboradas que sean las correctas? X Ison todas las execpciones de entrada/salida manejadas de la manera correcta? X Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Coinciden las unidades de los valores? Isi un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Ison controctores y los métodos sensibles de una clase están expuestos al acceso? X Este retornan objetos nulos? Si no dada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? X Son oxpresados correctamente los comportamientos de cada método? X Son todos los comentarios cansistentes con el código? X Hay demasiados comentarios a entender el código? X Hay demasiados comentarios a entender el código? X Hay demasiados comentarios en el código? X Hay demasiados comentarios en el código? X Hay un alto nivel de acoplamiento entre módulos? (Clases o métodos) X Esta et comentarios X X Son os de almacenamiento Si No Ses utiliza una indentación estándar? X Son de almacenamiento Si No Si No Si No Rodularidad Hay un hajo nivel de acoplamiento entre módulos? (Clases o métodos) X Ses utiliza una indentación estándar? X Son de almacenamiento Si No Si No Rodularidad Si No Si No Rodularidad Si No Si No Rodularidad Si No Rodularida	·	Х	
Entrada/Salida Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Están siendo corroboradas que sean las correctas? X Estón todas las excepciones de entrada/salida manejadas de la manera correcta? X Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Coinciden las unidades de los valores? Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los declaración de atributo y constante un comentario? X Los de accesarios de la cabecera de cada clase o métodos comentarios en el código? X Los de accesarios de la	, ·		
Hay errores gramaticales en el texto mostrado por pantalla? Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? X Están siendo corroboradas sa entradas con respecto a los valores que pueden tomar? X			No
Están siendo corroboradas las entradas con respecto a los valores que pueden tomar? Las salidas están siendo corroboradas que sean las correctas? X Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Coinciden las unidades de los valores? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Los constructores y los métodos sensibles de una clase están expuestos al acceso? X Son expresados correctamente los correspondiente y apropiado comentario en la cabecera? X Los constructores y los método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Los constructores y los método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Los comentarios Si No Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Los comentarios de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Los comentarios de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Lay suficiente cantidad de comentarios en el código? X Lay suficiente cantidad de comentarios en el código? X Lay demasiados comentarios en el código? X Lay demasiados comentarios en el código? X Lay una into nivel de acoplamiento entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohesión entre módulos? (Clases o métodos) Lay un alto nivel de cohe			X
Las salidas están siendo corroboradas que sean las correctas? X Son todas las excepciones de entrada/salida manejadas de la manera correcta? X Interfaz de módulos Inter		Х	+
Son todas las excepciones de entrada/salida manejadas de la manera correcta? Interfaz de módulos Scon correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Coinciden las unidades de los valores? Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Coinciden las unidades de los valores? X Son entructores y los métodos sensibles de una clase están expuestos al acceso? X Comentarios Comentarios Coada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? X Ayudan los comentarios a entender el código? X Calyudan los comentarios en el código?	, , ,		+
Interfaz de módulos Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?			+
Eson correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo? Cioniciden las unidades de los valores? Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		7.	
declaración del mismo? Coinciden las unidades de los valores? Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X Elos constructores y los métodos sensibles de una clase están expuestos al acceso? X Si No Comentarios Si No Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Ayudan los comentarios a entender el código? X Hay sufficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Coincida de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Es el comentario so antender el código? X Coincida de la comentarios en el código? X Hay sufficiente cantidad de comentarios en el código? X Hay un la indentación estándar? X Para cada método: ¿No supera las 60 líneas de largo? X Modularidad Si No Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) X Estáta siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X Si No Son los arreglos suficientemente largos? X Si No Poreformance Si No Porefor		X	
Coinciden las unidades de los valores? Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? X 2 Los constructores y los métodos sensibles de una clase están expuestos al acceso? X 2 Comentarios Si No Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X 2 Posee cada declaración de atributo y constante un comentario? X 2 Eson expresados correctamente los comportamientos de cada método? Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X 2 Son todos los comentarios consistentes con el código? X 2 L'Ayudan los comentarios a entender el código? X 3 L'Hay sufficiente cantidad de comentarios en el código? X 4 L'Bay demasiados comentarios en el código? X 5 L'Bay demasiados comentarios en el código? X 6 L'Bay demasiados comentarios en el código? X 7 Albay un latina indentación estándar? X 8 Albay un latina indentación estándar? X 9 Albay un alto nivel de acoplamiento entre módulos? (Clases o métodos) X 9 L'Existe código repetitivo? L'Exista siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X 9 L'Están siendo utilizadas correctamente largos? X 9 L'Están siendo utilizadas correctamente		``	
si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente? ¿Los constructores y los métodos sensibles de una clase están expuestos al acceso? X ¿Se retornan objetos nulos? X Comentarios Si No Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X ¿Posee cada declaración de atributo y constante un comentario? X ¿Son expresados correctamente los comportamientos de cada método? ¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X ¿Ayudan los comentarios a entender el código? X ¿Hay sufficiente cantidad de comentarios en el código? X ¿Hay demasiados comentarios en el código? X ¿Bo utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad ¿Hay un alto nivel de acoplamiento entre módulos? (Clases o métodos) ¿Existe código repetitivo? ¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X ¿Jos de almacenamiento Si No ¿Son los arreglos suficientemente largos? X ¿Jos de almacenamiento Si No ¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? ¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X ¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		Х	+
Elos constructores y los métodos sensibles de una clase están expuestos al acceso? X Se retornan objetos nulos? Si No Comentarios Si No Comentarios Si No Comentarios Si No Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Son expresados correctamente los comportamientos de cada método? El se l comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? X El se l comentarios en entender el código? X El se y comentarios en el código? X Comentarios en el código? X El se y comentarios en el código? X El se y comentarios en el código? X Comentarios en el códig		<u> </u>	
Ese retornan objetos nulos? Comentarios Si No Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Eson expresados correctamente los comportamientos de cada método? Eson expresados correctamente los comportamientos de cada método? Eson et comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? X Ayudan los comentarios a entender el código? X Hay suficiente cantidad de comentarios en el código? X Belay suficiente cantidad de comentarios en el código? X Deseño Si No Ese utiliza una indentación estándar? X Para cada método: ¿No supera las 60 líneas de largo? Modularidad Si No Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) Hay un alto nivel de cohesión entre módulos? (Clases o métodos) Existe código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X Jos de almacenamiento Si No Son los arreglos suficientemente largos? X Performance Si No Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			+
Comentarios Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? X Posee cada declaración de atributo y constante un comentario? X Eson expresados correctamente los comportamientos de cada método? X Eson etodos los comentarios consistentes con el código? X Eson todos los comentarios consistentes con el código? X Esulay suficiente cantidad de comentarios en el código? X Estay sufficiente cantidad de comentarios en el código? X Estay sufficiente cantidad de comentarios en el código? X Estay sufficiente cantidad de comentarios en el código? X Estay sufficiente cantidad de comentarios en el código? X Esta y sufficiente cantidad de comentarios en el c	,		X
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera? Z Posee cada declaración de atributo y constante un comentario? Son expresados correctamente los comportamientos de cada método? El el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X Son todos los comentarios consistentes con el código? X Ayudan los comentarios a entender el código? X Hay suficiente cantidad de comentarios en el código? X Hay demasiados comentarios en el código? X Hay demasiados comentarios en el código? X Diseño Si No Se utiliza una indentación estándar? X Para cada método: ¿No supera las 60 líneas de largo? X Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) X Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) El Hay un alto nivel de cohesión entre módulos? (Clases o métodos) El Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X Jos de almacenamiento Si No Si Son los arreglos suficientemente largos? X Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		Si	
Posee cada declaración de atributo y constante un comentario? ESOn expresados correctamente los comportamientos de cada método? ES el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X ESON todos los comentarios consistentes con el código? EA yudan los comentarios a entender el código? EN Ay udan los comentarios a entender el código? EN Ay usificiente cantidad de comentarios en el código? EN Ay usificiente cantidad de comentarios en el código? EN AY USON COMENTA SE AN AY USON COMENTA SE		<u> </u>	
Son expresados correctamente los comportamientos de cada método? El el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? X El el comentarios consistentes con el código? A yudan los comentarios a entender el código? El Hay suficiente cantidad de comentarios en el código? El Hay demasiados comentarios en el código? X Bellay una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? X Bellay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) El Hay un alto nivel de acoplamiento entre módulos? (Clases o métodos) El Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? El Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) X Boon los arreglos suficientemente largos? X Son los arreglos suficientemente largos? X Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X El puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			
Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma? Son todos los comentarios consistentes con el código? Avudan los comentarios a entender el código? El Ayudan los comentarios a entender el código? El Ayudan los comentarios en el código? El Ayudan los comentarios en el código? El Ayudan los comentarios en el código? X Diseño Si No Se utiliza una indentación estándar? X Para cada método: ¿No supera las 60 líneas de largo? X Modularidad Si No El Ayu n alto nivel de acoplamiento entre módulos? (Clases o métodos) El Ayu n alto nivel de cohesión entre módulos? (Clases o métodos) El Ayudan los correctamente las clases o métodos) El Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? El Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No Son los arreglos suficientemente largos? X Dispon los arreglos sufic			+
Eson todos los comentarios consistentes con el código? EAyudan los comentarios a entender el código? EAyudan los comentarios a entender el código? El Hay suficiente cantidad de comentarios en el código? El Hay demasiados comentarios en el código? El Hay demasiados comentarios en el código? Si No Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad Si No El Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) El Hay un alto nivel de cohesión entre módulos? (Clases o métodos) Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? El Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No Son los arreglos suficientemente largos? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			+
Ayudan los comentarios a entender el código? El Hay suficiente cantidad de comentarios en el código? El Hay suficiente cantidad de comentarios en el código? El Hay demasiados comentarios en el código? Si No El Beseño Si No El Hay demasiados comentarios en el código? X Para cada método: ¿No supera las 60 líneas de largo? Modularidad Si No El Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) El Hay un alto nivel de cohesión entre módulos? (Clases o métodos) El Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? El Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento El Son los arreglos suficientemente largos? El Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No El Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X El Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			_
Hay suficiente cantidad de comentarios en el código? L'Hay demasiados comentarios en el código? Si No Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad Si No Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) Hay un alto nivel de cohesión entre módulos? (Clases o métodos) Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Jos de almacenamiento Si No Si Son los arreglos suficientemente largos? Zon las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No Equeden ser usadas mejores estructuras de datos o algoritmos más eficientes? X Equede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	-		
Hay demasiados comentarios en el código? Si No Useño Si No Use utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad E Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) E Hay un alto nivel de cohesión entre módulos? (Clases o métodos) E Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? E Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No E Son los arreglos suficientemente largos? E Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X			+
Diseño Si No Se utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad E Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) E Hay un alto nivel de cohesión entre módulos? (Clases o métodos) E Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? E Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Si No Son los arreglos suficientemente largos? Si No E Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			
Ese utiliza una indentación estándar? Para cada método: ¿No supera las 60 líneas de largo? Modularidad El Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) El Hay un alto nivel de cohesión entre módulos? (Clases o métodos) El Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No Eson los arreglos suficientemente largos? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No Equeden ser usadas mejores estructuras de datos o algoritmos más eficientes? Equede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		Si	
Para cada método: ¿No supera las 60 líneas de largo? Modularidad EHay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) EHay un alto nivel de cohesión entre módulos? (Clases o métodos) Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No Si No Son los arreglos suficientemente largos? Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Yerformance Si No Equeden ser usadas mejores estructuras de datos o algoritmos más eficientes? Equede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			110
Modularidad E Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) E Hay un alto nivel de cohesión entre módulos? (Clases o métodos) E Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? E Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No E Son los arreglos suficientemente largos? E Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? E Performance E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X		-	
EHay un bajo nivel de acoplamiento entre módulos? (Clases o métodos) EHay un alto nivel de cohesión entre módulos? (Clases o métodos) Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No ESon los arreglos suficientemente largos? ESon las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? E Performance E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X			No
Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No Si Son los arreglos suficientemente largos? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Performance Si No Equeden ser usadas mejores estructuras de datos o algoritmos más eficientes? X Equede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X Equede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?			110
Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Si No Eson los arreglos suficientemente largos? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Esperformance Si No Espueden ser usadas mejores estructuras de datos o algoritmos más eficientes? Espuede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X		1	X
del código repetitivo? Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Eson los arreglos suficientemente largos? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Performance Equeden ser usadas mejores estructuras de datos o algoritmos más eficientes? Equede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X	·		
Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar) Uso de almacenamiento Eson los arreglos suficientemente largos? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Eson las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Esperformance Espueden ser usadas mejores estructuras de datos o algoritmos más eficientes? Espuede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X			
Uso de almacenamiento Si No Son los arreglos suficientemente largos? Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance Si No E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X		Х	
ESon los arreglos suficientemente largos? ESon las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? X Performance EPueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X EPuede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X	Uso de almacenamiento		No
ESon las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más? Performance E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X			
Performance E Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X E Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X			X
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes? X ¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X	Performance	Si	
Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor? X			
			_
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	¿Son todos los valores calculados y almacenados verdaderamente utilizados?	Х	1

¿Puede un cálculo ser realizado fuera de un ciclo?		Х
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		X
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el constructor no se verifica que el parámetro **argumentoInvalido** sea distinto de null. Tampoco se verifica que el parámetro String **texto** sea distinto de null.

Faltan comentarios en:

- Cabecera de la clase
- Atributos de la clase
- Todos los métodos

<u>Clase FaltantesException:</u>

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	
¿Poseen fin todos los bucles?	Х	
¿Poseen todas las estructuras switch un default case?	Х	
¿Son correctos todos los breaks dentro de una estructura switch?	Х	
¿Resultan las anidaciones de bucles correctas y necesarias?	Х	
¿Puede modificarse alguna anidación de if por una estructura switch?		Х
¿Son todas las excepciones manejadas correctamente?	Х	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	Х	
¿Las salidas están siendo corroboradas que sean las correctas?	Х	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Х	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?		Х
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		Х
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		Х
¿Hay suficiente cantidad de comentarios en el código?		Х
¿Hay demasiados comentarios en el código?		Х
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?		X
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	Х	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)		Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	Х	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	Х	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	Х	
¿Puede un cálculo ser realizado fuera de un ciclo?		Х

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		X

No se verifica que en el constructor los parámetros **mensaje** y **faltantes** sean distintos de null.

Clase StateException

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	
¿Poseen fin todos los bucles?	Х	
¿Poseen todas las estructuras switch un default case?	Х	
¿Son correctos todos los breaks dentro de una estructura switch?	Х	
¿Resultan las anidaciones de bucles correctas y necesarias?	Х	
¿Puede modificarse alguna anidación de if por una estructura switch?		Х
¿Son todas las excepciones manejadas correctamente?	Х	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	Х	
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?		Х
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?		Х
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		Х
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		Х
¿Hay suficiente cantidad de comentarios en el código?		Х
¿Hay demasiados comentarios en el código?		Х
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?		X
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	Х	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)		Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	Х	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	Х	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	Х	
¿Puede un cálculo ser realizado fuera de un ciclo?		Х

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		X

No se verifica que en el constructor el parámetro **mensaje** sea distinto de null.

Clase LengthException:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Χ	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Χ	
¿Hay variables que debieran ser constantes?		X
¿Hay atributos que debieran ser variables locales?		X
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	X	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	

¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?	+~	X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	31	X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	X	
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos		
	X	
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	^	
¿Coinciden las unidades de los valores?	X	
• • • • • • • • • • • • • • • • • • • •		
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
¿Se retornan objetos nulos?	C:	X
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		X
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?		X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		X
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?		X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	
Para cada método: ¿No supera las 60 líneas de largo?	X	
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	X	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)		Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		Х
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?		Х
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		Х
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	1	Х
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	
¿Puede un cálculo ser realizado fuera de un ciclo?	+	X

¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el constructor no se verifica que ambos parámetros sean distintos de null.

Faltan comentarios en:

- Cabecera de la clase
- Atributos de la clase
- Todos los métodos

Paquete listas

<u>Clase ListaLotes:</u>

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Χ
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	X	
¿Están todas las variables correctamente tipeadas?	X	
¿Están todas las variables correctamente inicializadas?	X	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	X	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?	Х	
¿Hay variables o atributos con nombres similares confusos?	T	Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	

Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	
¿Poseen fin todos los bucles?	х	
¿Poseen todas las estructuras switch un default case?	х	
¿Son correctos todos los breaks dentro de una estructura switch?	х	+
¿Resultan las anidaciones de bucles correctas y necesarias?	х	+
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	х	1
¿Todos los métodos terminan?	х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		Х
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	Х	+
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	+
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	X	110
declaración del mismo?	^	
¿Coinciden las unidades de los valores?	Х	+
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	+
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	^	X
¿Se retornan objetos nulos?		X
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?	J.	X
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?		X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		X
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?		X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	110
Para cada método: ¿No supera las 60 líneas de largo?	^	X
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	X	110
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	 	X
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?		^
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	Х	+
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	+
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	 	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+**
Coon todos los valores calculados y annacentados verdaderamente atinizados:	1 ^	

¿Puede un cálculo ser realizado fuera de un ciclo?		Х
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		X
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el método public void borrarLote(Lote lot) no se verifica que lot sea distinto de null.

Faltan comentarios de la clase y del método borrarLote(Lote lot).

Clase ListaPedidos:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	X	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?	1	Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	X	1
¿Son todos los valores de los parámetros chequeados antes de ser usados?		X
¿Hay métodos estáticos que no debieran serlo, o viceversa?		X
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	X	+
Siempre que se utiliza toString: ¿Es sobrescrito?	X	_
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	X	110
¿Posee cada clase un constructor apropiado?	X	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		X
¿Puede ser la jerarquía de herencia simplificada?	+	X
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	X	110
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	X	
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?	J.	X
¿Existe posibilidad de overflow o underflow durante un cálculo?	+	X
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	X	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
correctas?	^	
¿Se usan paréntesis para evitar ambigüedad?	X	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	X	
¿Son los operadores de las comparaciones correctos?	X	+
¿Hay comparaciones entre variables de tipos inconsistentes?	+^	X
¿Existen efectos secundarios inesperados resultantes de las comparaciones?	+	X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?	+	X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	X	
Flujo de control	Si	No
•		INU
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	

¿Poseen fin todos los bucles?	Х	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?	+	X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	31	X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	X	- -
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	X	
declaración del mismo?	^	
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	
¿Se retornan objetos nulos?	 ^ -	X
Comentarios	Si	No
	31	X
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		X
¿Posee cada declaración de atributo y constante un comentario?	X	^
¿Son expresados correctamente los comportamientos de cada método?	 ^	V
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?	X	
¿Ayudan los comentarios a entender el código?	X	
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?	6:	X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	
Para cada método: ¿No supera las 60 líneas de largo?	X	
Modularidad (1.1.1.2.4.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	X	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)		X
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?		X
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	
¿Puede un cálculo ser realizado fuera de un ciclo?		X
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		X

¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el método public **void agregarNuevo(Pedido nuevo)** no se verifica que el parámetro sea distinto de null antes de utilizarlo.

Faltan comentarios en:

- Atributos de clase
- Constructor de clase
- Método borrarPedido
- Método uptade

Clase Empleado:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?	X	
¿Hay métodos estáticos que no debieran serlo, o viceversa?		X
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	X	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	X	110
¿Posee cada clase un constructor apropiado?	X	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		X
¿Puede ser la jerarquía de herencia simplificada?		X
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	X	110
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		X
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?	31	X
¿Existe posibilidad de overflow o underflow durante un cálculo?		X
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
correctas?	^	
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	X	1.10
¿Son los operadores de las comparaciones correctos?	X	
¿Hay comparaciones entre variables de tipos inconsistentes?		X
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	1
Flujo de control	Si	No
Trujo de control	J1	IVO

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?	<u> </u>	X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	 	Х
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	 	Х
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Х	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	Х	
¿Se retornan objetos nulos?	Х	
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?	Х	
¿Posee cada declaración de atributo y constante un comentario?		Х
¿Son expresados correctamente los comportamientos de cada método?	Х	
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?	Х	
¿Son todos los comentarios consistentes con el código?	Х	
¿Ayudan los comentarios a entender el código?	Х	
¿Hay suficiente cantidad de comentarios en el código?		Х
¿Hay demasiados comentarios en el código?		Х
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?		Х
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)		Х
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	Х	
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		Х
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	Х	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	Х	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		Х
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		Х
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	Х	
¿Puede un cálculo ser realizado fuera de un ciclo?		Х
		·

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

El método **private boolean verificaLegajo(String legajo)** no verifica que el parámetro **legajo** sea distinto de null.

El método **private boolean verificaNombreyApellido(String nya)** no verifica que el parámetro **nya** sea distinto de null.

Paquete visual

Ventana Producción

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		X
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	

Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	Х	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	Х	+
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		Х
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	Х	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	X	110
declaración del mismo?	^	
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	1
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	 	X
¿Se retornan objetos nulos?	+	X
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?	<u> </u>	X
¿Posee cada declaración de atributo y constante un comentario?	 	X
¿Son expresados correctamente los comportamientos de cada método?	+	X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?	+	X
¿Son todos los comentarios consistentes con el código?	+	X
¿Ayudan los comentarios a entender el código?	 	X
¿Hay suficiente cantidad de comentarios en el código?	 	X
¿Hay demasiados comentarios en el código?	 	X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	110
Para cada método: ¿No supera las 60 líneas de largo?	X	+
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	<u> </u>	X
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	X	+
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	 	X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	+
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	+	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+
coon reads to tailores edicaledos y annacendados veradacidinente atilizados.		

¿Puede un cálculo ser realizado fuera de un ciclo?		Х
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		X
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el constructor el parámetro **control** no se verifica que sea distinto de null.

Comentarios:

- El constructor es accesible pero no es un método sensible.
- El método **protected void IniciarComponentes()** supera las 60 líneas, pero es la única manera de ser codificado debido a que debe iniciar todos los componentes pertenecientes a la ventana.

Clase VentanaVentas:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Х	\top
¿Poseen fin todos los bucles?	X	+
¿Poseen todas las estructuras switch un default case?	X	+
¿Son correctos todos los breaks dentro de una estructura switch?	X	+
¿Resultan las anidaciones de bucles correctas y necesarias?	X	_
¿Puede modificarse alguna anidación de if por una estructura switch?	+^-	X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	_
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	31	X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	+	X
¿Las salidas están siendo corroboradas que sean las correctas?	X	
·		
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta? Interfaz de módulos	X Si	No
		No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la declaración del mismo?	X	
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	Х	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?		Х
¿Se retornan objetos nulos?		X
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?	<u> </u>	X
¿Posee cada declaración de atributo y constante un comentario?	1	X
¿Son expresados correctamente los comportamientos de cada método?	1	X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?	1	X
¿Son todos los comentarios consistentes con el código?	1	X
¿Ayudan los comentarios a entender el código?	1	X
¿Hay suficiente cantidad de comentarios en el código?	+	X
¿Hay demasiados comentarios en el código?	1	X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	110
Para cada método: ¿No supera las 60 líneas de largo?	X	+
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	J.	X
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	X	+~
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento	+~	X
del código repetitivo?		^
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	+
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	+	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+
¿Puede un cálculo ser realizado fuera de un ciclo?	+**	X
C. add an entario ser realizado racia de arroldo.		

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el constructor el parámetro control no se verifica que sea distinto de null.

El método **protected void IniciarComponentes()** supera las 60 líneas, pero es la única manera de ser codificado debido a que debe iniciar todos los componentes pertenecientes a la ventana.

Faltan comentarios:

- Cabecera de la clase
- Atributos de la clase
- Métodos de la clase

Clase PanelFechas:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		Х
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	X	
¿Están todas las variables correctamente tipeadas?	Χ	
¿Están todas las variables correctamente inicializadas?	X	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?		Х
¿Hay variables que debieran ser constantes?		Χ
¿Hay atributos que debieran ser variables locales?		X
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	X	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Χ
¿Hay variables o atributos con nombres similares confusos?		Χ
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	X	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	X	
¿Son todos los valores de los parámetros chequeados antes de ser usados?	X	
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	Х	
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	X	
Comparaciones y relaciones	Si	No
Por cada condición booleana: ¿Es correcta la condición chequeada?	Χ	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		Х
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	Χ	

¿Poseen fin todos los bucles?	Х	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?	+~	X
¿Son todas las excepciones manejadas correctamente?	X	
¿Todos los métodos terminan?	X	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?	- 0.	X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	X	
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	X	
Interfaz de módulos	Si	No
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	X	110
declaración del mismo?	^	
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?		X
¿Se retornan objetos nulos?		X
Comentarios	Si	
	31	No X
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		X
¿Posee cada declaración de atributo y constante un comentario?	+	X
¿Son expresados correctamente los comportamientos de cada método?	+	
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		X
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?	C:	X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	.,
Para cada método: ¿No supera las 60 líneas de largo?		X
Modularidad (1.1.1.2.4.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)		X
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	X	
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	X	
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		X
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	
¿Puede un cálculo ser realizado fuera de un ciclo?		X
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		X

¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		X
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	Χ	
AMBIGÜEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		X
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el método private void verificaDias(boolean bisiesto) no declara las variables dentro del for.

Clase Controlador:

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	Х	
¿Están todas las variables correctamente tipeadas?	Х	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?	Х	
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?		Х
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Son todos los valores de los parámetros chequeados antes de ser usados?		Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?		Х
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	
Siempre que se utiliza toString: ¿Es sobrescrito?	Х	
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	Х	
¿Posee cada clase un constructor apropiado?	Х	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?		Х
¿Puede ser la jerarquía de herencia simplificada?		Х
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	Х	
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?		Х
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?		Х
¿Existe posibilidad de overflow o underflow durante un cálculo?		Х
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	Х	
correctas?		
¿Se usan paréntesis para evitar ambigüedad?	Х	
Comparaciones y relaciones		
Por cada condición booleana: ¿Es correcta la condición chequeada?	Х	
¿Son los operadores de las comparaciones correctos?	Х	
¿Hay comparaciones entre variables de tipos inconsistentes?		Х
¿Existen efectos secundarios inesperados resultantes de las comparaciones?		Х
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?		X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?	Х	
Flujo de control	Si	No
Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	

¿Poseen fin todos los bucles?	Х	\top
¿Poseen todas las estructuras switch un default case?	X	_
¿Son correctos todos los breaks dentro de una estructura switch?	X	+
¿Resultan las anidaciones de bucles correctas y necesarias?	X	+
¿Puede modificarse alguna anidación de if por una estructura switch?	 ^	X
¿Son todas las excepciones manejadas correctamente?	X	+^-
¿Todos los métodos terminan?	X	_
Entrada/Salida	Si	No
·	31	X
¿Hay errores gramaticales en el texto mostrado por pantalla?		
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?	 	X
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Х	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	X	
declaración del mismo?		
¿Coinciden las unidades de los valores?	Х	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	<u> </u>	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	
¿Se retornan objetos nulos?		
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		Х
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?		X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		Х
¿Son todos los comentarios consistentes con el código?		Х
¿Ayudan los comentarios a entender el código?		Х
¿Hay suficiente cantidad de comentarios en el código?		Х
¿Hay demasiados comentarios en el código?		Х
Diseño	Si	No
¿Se utiliza una indentación estándar?	Х	
Para cada método: ¿No supera las 60 líneas de largo?	Х	
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	Х	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)		Х
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		X
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	X	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	X	110
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?	1.	X
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		113
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?	1	X
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	X	+^-
¿Puede un cálculo ser realizado fuera de un ciclo?	+^-	
¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?	+	X
Cruede desdobiaise un bucie corto de tai forma de evitar la utilización de un bucie?		Λ

¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		X
¿Las acciones de concatenación de strings se realizan correctamente?	X	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		X
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х

En el método getPedidosEvaluacion hay casteos innecesarios.

Faltan comentarios en:

- Atributos de la clase
- Constructor
- Mayoría de los métodos

En ningún método se verifica que los parámetros sean distintos de null.

Tabla resultante de la aplicación

DESVIACIÓN		
Desviación	Si	No
¿Hay algún requerimiento que no se haya implementado?		X
DEFECTOS		
Variables y declaraciones de constantes	Si	No
¿Están nombradas las variables y las constantes con respecto a las convenciones de nomenclatura?	X	
¿Están todas las variables correctamente tipeadas?	X	
¿Están todas las variables correctamente inicializadas?	Х	
¿Están todas las variables del ciclo for declaradas en la cabecera del ciclo?		Х
¿Hay variables que debieran ser constantes?		Х
¿Hay atributos que debieran ser variables locales?		Х
¿Todos los atributos poseen correctos modificadores de acceso (privado, protegido, público)?	Х	
¿Hay atributos estáticos que no debieran serlo, o viceversa?	Х	
¿Hay variables o atributos con nombres similares confusos?		Х
¿Puede cualquier parámetro ser convertido a variable local?		Х
Definiciones de métodos	Si	No
¿Los métodos están nombrados de acuerdo a las convenciones de nomenclatura?	Х	
¿Poseen todos los métodos correctos modificadores de acceso (privado, protegido, público)?	X	
¿Son todos los valores de los parámetros chequeados antes de ser usados?	1	Х
¿Hay métodos estáticos que no debieran serlo, o viceversa?	_	X
¿Devuelven todos los métodos el valor correcto en cada retorno del mismo?	Х	1
Siempre que se utiliza toString: ¿Es sobrescrito?	 	X
Definiciones de clases	Si	No
¿Están nombradas las clases de acuerdo a las convenciones de nomenclatura?	X	110
¿Posee cada clase un constructor apropiado?	X	
¿Poseen algunas sub-clases miembros en común que debieran estar en la super-clase?	 ^ -	X
¿Puede ser la jerarquía de herencia simplificada?		X
Referencia a datos	Si	No
Para cada referencia a un arreglo: ¿Se encuentran todos los valores de subíndices dentro de los límites definidos?	X	110
Para cada referencia a objeto o arreglo: ¿Es el valor distinto de null?	 ^	X
Cálculos	Si	No
¿Existe algún cálculo con tipos de datos mezclados?	31	X
¿Existe algun calculo con tipos de datos mezciados: ¿Existe posibilidad de overflow o underflow durante un cálculo?		X
Para cada operación con más de un operador: ¿Son las asunciones sobre el orden de evaluación y precedencia	\vdash	X
correctas?		^
¿Se usan paréntesis para evitar ambigüedad?	X	
Comparaciones y relaciones		
,	V	
Por cada condición booleana: ¿Es correcta la condición chequeada?	X	
¿Son los operadores de las comparaciones correctos?	X	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
¿Hay comparaciones entre variables de tipos inconsistentes?	<u> </u>	X
¿Existen efectos secundarios inesperados resultantes de las comparaciones?	 	X
¿Han sido confundidos los operadores "&&" o " " con los operadores binarios "&" y " "?	 	X
¿Están siendo cubiertas todas las ramas (igual, menor, mayor)?		Х
Flujo de control	Si	No

Por cada bucle: ¿Es la mejor opción de construcciones de bucles elegida?	X	
¿Poseen fin todos los bucles?	X	
¿Poseen todas las estructuras switch un default case?	X	
¿Son correctos todos los breaks dentro de una estructura switch?	X	
¿Resultan las anidaciones de bucles correctas y necesarias?	X	
¿Puede modificarse alguna anidación de if por una estructura switch?		X
¿Son todas las excepciones manejadas correctamente?	Х	
¿Todos los métodos terminan?	Х	
Entrada/Salida	Si	No
¿Hay errores gramaticales en el texto mostrado por pantalla?		X
¿Están siendo corroboradas las entradas con respecto a los valores que pueden tomar?		X
¿Las salidas están siendo corroboradas que sean las correctas?	X	
¿Son todas las excepciones de entrada/salida manejadas de la manera correcta?	Х	
Interfaz de módulos		
¿Son correctos el número, tipo, orden y valor de los parámetros en cada llamada al método, con respecto a la	Х	
declaración del mismo?		
¿Coinciden las unidades de los valores?	X	
Si un objeto o un arreglo es pasado por parámetro: ¿Este es modificado correctamente?	X	
¿Los constructores y los métodos sensibles de una clase están expuestos al acceso?	X	
¿Se retornan objetos nulos?	X	
Comentarios	Si	No
Cada clase y método, ¿poseen su correspondiente y apropiado comentario en la cabecera?		X
¿Posee cada declaración de atributo y constante un comentario?		X
¿Son expresados correctamente los comportamientos de cada método?		X
¿Es el comentario de la cabecera de cada clase o método correspondiente con el comportamiento de la misma?		X
¿Son todos los comentarios consistentes con el código?		X
¿Ayudan los comentarios a entender el código?		X
¿Hay suficiente cantidad de comentarios en el código?		X
¿Hay demasiados comentarios en el código?		X
Diseño	Si	No
¿Se utiliza una indentación estándar?	X	
Para cada método: ¿No supera las 60 líneas de largo?	X	
Modularidad	Si	No
¿Hay un bajo nivel de acoplamiento entre módulos? (Clases o métodos)	Х	
¿Hay un alto nivel de cohesión entre módulos? (Clases o métodos)	Х	
¿Existe código repetitivo que pueda ser reemplazado por la llamada a un método que posea el comportamiento		Х
del código repetitivo?		
¿Están siendo utilizadas correctamente las clases de librerías de Java? (En caso de utilizar)	Х	
Uso de almacenamiento	Si	No
¿Son los arreglos suficientemente largos?	Х	
¿Son las referencias a objetos o arreglos establecidas en nulo una vez que estos no se necesitan más?		Х
Performance	Si	No
¿Pueden ser usadas mejores estructuras de datos o algoritmos más eficientes?		Х
¿Puede ser disminuido el coste de recalcular un valor calculándolo a este una sola vez y guardando su valor?		Х
¿Son todos los valores calculados y almacenados verdaderamente utilizados?	Х	
¿Puede un cálculo ser realizado fuera de un ciclo?	1	Х

¿Puede desdoblarse un bucle corto de tal forma de evitar la utilización de un bucle?		Х
¿Existen 2 bucles que operan sobre los mismos datos que pueden ser fusionados en uno?		Х
INCONSISTENCIA		
¿Existe alguna parte del código implementada de manera inconsistente?		Х
¿Las acciones de concatenación de strings se realizan correctamente?	Х	
AMBIGUEDAD		
¿Existe algún método excesivamente complejo que deba ser reestructurado o dividido en varios métodos?		Х
REDUNDANCIA		
¿Existe alguna variable o atributo redundante o sin uso?		Х
¿Existe algún método sin llamar o innecesario?		Х
¿Existe algún objeto que sea innecesario o redundante?		Х
¿Hay código que pueda ser reemplazado por alguna llamada a un objeto externo reutilizable?		Х