| **CHECKLIST PARA LA VERIFICACIÓN DEL CÓDIGO** | | | |
| --- | --- | --- | --- |
| **Completitud** | | | |
| **Cualidad** | **Si** | **No** | **N/A** |
| Verificar que se importan solo las clases o archivos necesarios evadiendo el uso de “import \*” | X |  |  |
| Verificar que todas las variables de entrada de un método sean utilizadas. | X |  |  |
| Verificar que los métodos no retornen variables nulas | X |  |  |
| Verificar que los métodos tengan un único return siempre que sea posible y la legibilidad no se vea comprometida. | X |  |  |
| **Lógica** | | | |
| **Cualidad** | **Si** | **No** | **N/A** |
| Verifica la utilización excesiva de sentencias if |  | X |  |
| Verificar que las sentencias de los métodos no sean muy largas. |  | X |  |
| Verificar la creación de instancias no utilizadas en el código. | X |  |  |
| Verificar el uso de concatenación excesiva de variables Strings. | X |  |  |
| Verificar que las variables de ciclos estén iniciadas con el valor correcto | X |  |  |
| Verificar que las variables de ciclos terminen con el valor correcto. (<, <=, >, >=) | X |  |  |
| **Casos especiales** | | | |
| **Cualidad** | **Si** | **No** | **N/A** |
| Verificar que los métodos públicos declarados, se utilicen por otros módulos del sistema, esto a excepción de los getters/setters. | X |  |  |
| Verifica que no haya números mágicos en el código a excepción de 0, 1 o cadena vacía. | X |  |  |
| Verifica que no se apliquen los comentarios autogenerados por javadoc en los métodos autoexplicativos.  Javadoc es opcional para métodos "simples y obvios" como getFoo, en los casos en los que realmente no hay nada más que valga la pena decir que "returns the foo”. | X |  |  |
| Verificar que se emplean excepciones en secciones de código que puedan tener fallos como valores inválidos o index out of bounds. | X |  |  |
| **Uso funcional** | | | |
| **Cualidad** | **Si** | **No** | **N/A** |
| Verificar que los métodos y variables estén nombrados como en el diseño. | X |  |  |
| Verificar que el código escrito sea de acuerdo al diseño establecido. | X |  |  |
| Verificar que los métodos tengan un único propósito o realicen una única tarea. Descomponer métodos que hagan múltiples tareas en distintos métodos que realicen una única tarea siempre que sea posible. | X |  |  |
| **Estándares** | | | |
| **Cualidad** | **Si** | **No** | **N/A** |
| Verifica que las variables tengan nombres descriptivos | X |  |  |
| Verifica que los comentarios son descriptivos y solo se incluyen cuando el código es complejo de entender. | X |  |  |
| Verifica que todas {, (, [, se cierren de manera correcta. | X |  |  |
| Verifica que todas las llamadas a métodos se hacen con los parámetros requeridos y se reciben bien los retornos si es el caso. | X |  |  |
| Verifica que todas las operaciones lógicas sean correctas. Que los operadores <, <=, >, >=, ==, !=, &&, ||, ! , estén siendo utilizados adecuadamente. | X |  |  |
| Verifica que los métodos son nombrados por verbos escritos en minúsculas. Cuando estos estén compuestos por varias palabras se tendrá la primera letra de cada palabra en mayúscula (camelCase). | X |  |  |

| **Observaciones** | Se mantuvo la estructura original del código, no se optimizó ni refactorizó |
| --- | --- |
| **Resultado** | Se realizó de manera correcta el código. |