**FR-0001 - LISTA DE VERIFICACIÓN PARA LA INSPECCIÓN DE CÓDIGO FUENTE EN C #**

|  |  |  |
| --- | --- | --- |
| **Categoría** | **Descripción** | **Chequeado** |
| Declaraciones | 1. Nombramiento. ¿El nombramiento de variables está de acuerdo al estándar de la compañía? |  |
| 2. Nombramiento. ¿Son todos los nombres de variables y funciones claros y significativos? |  |
| 3. Tipos correctos. ¿Son todas las variables y argumentos de función del tamaño y tipo correctos? |  |
| 4. Ámbito de aplicación. ¿No hay variables públicas, propiedades o métodos que podrían ser privados o protegidos? |  |
| 5. Ámbito de aplicación. ¿No hay variables cuyo alcance podría ser más estrecho? |  |
| 6. ¿Son utilizadas todas las variables y funciones? (No se aplica a las bibliotecas). |  |
| 7. Uso adecuado de los espacios de nombres. |  |
| 8. Función adecuada de sobrecarga. |  |
| 9. ¿Uso apropiado de operador de asignación? |  |
| Variables | 1. Inicialización. ¿Están todas las variables inicializadas correctamente? |  |
| 2. Los valores asignados están siempre dentro del rango de valores válidos para el tipo. |  |
| Expresiones | 1. Aritmética. ¿Hay división por cero o desbordamientos? |  |
| 2. Aritmética. ¿Son todas las variables en los cálculos de tipos compatibles? |  |
| 3. Aritmética. No hay conversiones de tipos involuntarias (implícitas)? |  |
| 4. Expresión booleana. No se evalúa nunca el valor verdadero o falso explícitamente. |  |
| 5. Expresión booleana. Utilización del tipo bool para almacenamiento de resultados. |  |
| 6. Precedencia de operadores. ¿Está correctamente usada? ¿Puede utilizar paréntesis para aclarar? |  |
| Control de Flujo | 1. Valor de retorno. ¿Todos los flujos de control en una función conllevan a un resultado correcto, o en caso contrario a una excepción? |  |
| 2. Control de errores. ¿Uso correcto de tratamiento de errores? |  |
| 3. Control de errores. ¿No hay errores no capturados o excepciones? |  |
| 4. Loops. No hay bucles infinitos? ¿Todos los bucles tienen final? |  |
| 5. Loops. ¿Es correcta la indexación en los bucles? |  |
| 6. Terminación del programa. ¿Puede el programa terminar correctamente? |  |
| 7. Sentencias. Las sentencias "Switch" cubren todas las posibilidades necesarias? |  |
| API´s | 1. Documentación. ¿Están todas las propiedades o métodos documentados? |  |
| 2. Documentación. ¿Es la documentación precisa y actualizada? |  |
| 4. Argumentos. ¿Están todos los argumentos de entrada verificados? |  |
| 5. Argumentos. ¿Son correctos los valores para los parámetros por defecto? |  |
| 5. Argumentos. ¿Todos los argumentos de salida tienen los valores correctos al salir? |  |
| Formato | 1. Restricciones. ¿No hay líneas de código que excedan el estándar de la compañía para longitud máxima de línea? |  |
| 2. Restricciones. ¿No hay funciones que excedan el estándar de la compañía para número máximo de líneas? |  |
| 3. Complejidad. ¿No hay funciones o expresiones que se deban dividir en partes más pequeñas? |  |
| 4. Sangría. ¿Sangría adecuada? |  |
| Misceláneos | 1. Recursos. ¿No hay pérdidas de memoria? |  |
| 2. Calidad. Algoritmos, eficiencia, mantenibilidad |  |
| 3. ¿El código fuente está apropiadamente comentado? |  |
| 4. ¿Tiene sentido común? |  |