**LISTAS DE CHEQUEO DE ESTÁNDARES DE FUNCIONES Y PROCEDIMIENTOS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Revisado por** | Gutierrez Tineo Cesar | **Cargo o rol** | | | |  | |
| **Fecha** |  | **Sitio** | | | |  | |
| **Nombre del proyecto** | |  | | | | | |
| **Nombre de la función o procedimiento:** | |  | | | | | |
| **Descripción:** | | | | | | | |
| **REVISIÓN DE FUNCIONES Y PROCEDIMIENTOS** | | | | | | | |
| **ACTIVIDAD O ELEMENTO A REVISAR** | | | **SI** | **NO** | **NO APLICA** | | **OBSERVACIONES** |
| 1. ¿El nombre cumple con los estándares?  El nombre debe ser mnemotécnico, no poseer más de 8 caracteres, comenzar por letra, ¿otros? | | |  |  |  | |  |
| 2. ¿El código cumple con los estándares?  El código debe cumplir con los estándares de programación como modularidad, interrelación entre módulos, secuencia lógica, etc. | | |  |  |  | |  |
| 3. ¿Está la función/ procedimiento documentada?  Cada función, procedimiento, objeto o en general módulo debe especificar que hace y cuál es su función dentro de la aplicación. | | |  |  |  | |  |
| 4. ¿Se ha realizado el proceso de afinamiento de las búsquedas?  Las búsquedas deben ser las necesarias, deben cumplir con su objetivo: deben encontrar lo buscado y deben generar la información requerida. | | |  |  |  | |  |
| 5. ¿Se usan todas las variables, constantes y parámetros? Todo lo creado debe ser utilizado, lo que sobre debe ser eliminado. | | |  |  |  | |  |
| 6. ¿La asignación de valores a las variables, constantes y parámetros tienen un propósito?  Verificar que los procesos sean rápidos, correctos, funcionales y con un alto rendimiento. | | |  |  |  | |  |
| 7. ¿Son correctas las validaciones de condiciones? (ciclos infinitos, división por cero, verificación de rangos, etc.)  Revisar todos los ciclos en el límite, dentro del límite y fuera de este, verificar que no existan divisiones por cero y que los valores tengan una magnitud aceptada por el tipo de dato definido. | | |  |  |  | |  |
| 8. ¿Faltan validaciones?  Verificar cuales validaciones hacen falta: condicionales, estructuras compuestas, procedimientos, funciones… | | |  |  |  | |  |
| 9. ¿Se tienen en cuenta todas las excepciones posibles?  Dentro de cada proceso o procedimiento es necesario preguntarse qué pasaría si ocurriera un caso diferente y verificar si la aplicación está en capacidad de asumir dicho caso sin descontrolarse. | | |  |  |  | |  |
| 10. ¿Las variables que guardan datos de columnas de tablas se han definido de acuerdo con ellas?  Para identificar cada variable y su función, esta debe ser apropiadamente nombrada, para ser fácilmente identificada. | | |  |  |  | |  |
| 11. Si se llaman otras funciones y/o procedimientos, ¿tienen el número de parámetros y el tipo de datos adecuado?  El número y la clase de parámetros deben ser compatibles para no crear errores o conflictos dentro de la aplicación. | | |  |  |  | |  |
| Observaciones Generales: | | | | | | | |
| FIRMA DEL REVISOR : | | | | | | | |

**LISTAS DE CHEQUEO DE ESTÁNDARES DE PROGRAMACIÓN - CÓDIGO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Revisado por** | Maverick Poma Rosales | **Cargo o rol** | | | |  | |
| **Fecha** |  | **Sitio** | | | |  | |
| **Nombre del proyecto** | |  | | | | | |
| **Descripción:** | | | | | | | |
| **REVISIÓN DE LA PROGRAMACIÓN – CÓDIGO** | | | | | | | |
| **ASPECTO O ELEMENTO A REVISAR** | | | **SI** | **NO** | **NO APLICA** | | **OBSERVACIONES** |
| **1. Código en general** | | | | | | | |
| * 1. ¿Está el código con los espacios establecidos?   Verificar que el código se entienda, que esté bien dividido, que en cada módulo o estructura se pueda reconocer fácilmente en el comienzo y en la terminación. | | |  |  |  | |  |
| * 1. ¿Están ordenadas alfabéticamente las constantes, las variables y los cursores?   En la documentación es importante que las constantes y las variables estén ordenadas alfabéticamente. | | |  |  |  | |  |
| * 1. ¿Están alineados a la izquierda la definición del tipo de dato de las constantes y de las variables?   Este es un estándar de programación que debe cumplirse. | | |  |  |  | |  |
| **2. Documentación** | | | | | | | |
| 2.1 ¿Está toda la documentación en una línea diferente al código que se está documentando?  Verificar este aspecto en todo el código. | | |  |  |  | |  |
| 2.2 ¿Comprende la documentación de funciones/procedimientos tres partes: una descripción general de los que hace la función o procedimiento (módulo), la descripción de los parámetros de entrada y la descripción de los posibles valores y/o parámetros de salida? | | |  |  |  | |  |
| **3. Parámetros** | | | | | | | |
| 3.1 ¿El nombre de los parámetros es significativo?  Debe ser representativo y fácilmente identificable | | |  |  |  | |  |
| **4. Constantes** | | | | | | | |
| 4.1 ¿El nombre de las constantes es significativo?  Debe ser representativo y cumplir con los estándares de programación. | | |  |  |  | |  |
| **5. Variables** | | | | | | | |
| 5.1 ¿El nombre de las variables es significativo?  Debe ser representativo y cumplir con los estándares de programación. | | |  |  |  | |  |
| 5.2 ¿Están los nombres de las constantes y de las variables alineados a la izquierda junto a la definición del tipo? | | |  |  |  | |  |
| Observaciones Generales: | | | | | | | |
| FIRMA DEL REVISOR: | | | | | | | |

**LISTA DE CHEQUEO DE INSPECCIÓN DE CÓDIGO PARA EL LENGUAJE DE PROGRAMACIÓN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Revisado por** | Rogger Antonio Maita Asmat | **Cargo o rol** | | | |  | |
| **Fecha** | **09/12/2019** | **Sitio** | | | | **Oficinas de Debra’s Solution** | |
| **Nombre del proyecto** | |  | | | | | |
| **Nombre de la función o procedimiento** | | **Código Fuente** | | | | | |
| **Descripción:** | | | | | | | |
| **REVISIÓN DE IMPLEMENTACIÓN DEL LENGUAJE DE PROGRAMACIÓN** | | | | | | | |
| **ASPECTO O ELEMENTO A REVISAR** | | | **SI** | **NO** | **NO APLICA** | | **OBSERVACIONES** |
| 1. ¿El código está escrito siguiendo el estándar de codificación definido para el lenguaje usado? | | | X |  |  | |  |
| 2. El código implementa completa y correctamente el diseño detallado correspondiente | | | | | | | |
| 2.1 ¿Hay funciones/procedimientos faltantes? | | | X |  |  | | Falta la funcionalidad para la creación de tickets. |
| * 1. ¿El código de cada función/procedimiento es consistente con su especificación de diseño detallado? | | |  | X |  | | No todas las funciones son consistentes |
| 3. ¿Se tiene en cuenta condiciones de frontera (valores nulos o negativos)? | | | X |  |  | |  |
| 4. ¿Se tienen en cuenta todas las posibles condiciones de error? | | |  | X |  | | En algunos casos solo captura el error en general sin considerar diversas condiciones. |
| 5. Los procedimientos para cada invocación a función/procedimiento son correctos | | | | | | | |
| 5.1 ¿Los parámetros de entrada contienen los valores correctos? | | | X |  |  | |  |
| 5.2 ¿Los parámetros de salida tienen el tamaño y tipo correcto? | | | X |  |  | |  |
| 6. ¿Cada sentencia de selección es válida (¿la cláusula “else” está asociada a la sentencia “if*”* correcta?) | | | X |  |  | |  |
| 7. Para cada sentencia de selección múltiple (“switch” , “case of” o una sentencia con el mismo propósito) | | | | | | | |
| 7.1 ¿Los casos considerados están completos? | | | X |  |  | |  |
| 7.2 ¿El caso “default” se considera y se utiliza correctamente? | | | X |  |  | |  |
| 8. Para cada ciclo (while, for, repeat o similares) | | | | | | | |
| 8.1 ¿El ciclo ejecuta el número correcto de iteraciones? | | | X |  |  | |  |
| 8.2 ¿El ciclo no se queda en “loop” infinito? | | | X |  |  | |  |
| 8.3 ¿La lógica de la última iteración del ciclo está completa y correcta? | | | X |  |  | |  |
| 8.4 ¿Hay algún error en la condición que haga que se ejecute una vez más o una vez menos de lo debido? | | |  | X |  | |  |
| 9. Para cada expresión matemática o booleana: | | | | | | | |
| * 1. ¿La precedencia de operadores y de paréntesis son utilizados correctamente? | | | X |  |  | |  |
| * 1. ¿Las conversiones no causan pérdida de información? | | | X |  |  | |  |
| 10. Todos los arreglos y cadenas: | | | | | | | |
| 10.1 ¿Tienen el tamaño correcto? | | | X |  |  | |  |
| 10.2 ¿Están inicializados correctamente? | | | X |  |  | |  |
| 11. Todos los apuntadores usados: | | | | | | | |
| 11.1 ¿Son inicializados antes de hacer referencia a lo apuntado por ellos? | | |  |  | X | |  |
| 11.2 ¿Apuntan a memoria ya liberada? | | |  |  | X | |  |
| 12. ¿Todos los archivos abiertos son cerrados? | | |  |  | X | |  |
| 13. ¿Todos los objetos creados en memoria dinámica son liberados? | | |  |  | X | |  |
| Observaciones Generales:  Debido a la cantidad de código para las 3 partes del sistema: front web, backend y móvil, es normal que algunas funciones realicen la misma operación (operaciones repetidas) y existen tanttos tipos de casos de error que fue más conveniente el manejar una menor cantidad de excepciones de error. | | | | | | | |
| FIRMA DEL REVISOR:  Rogger Antonio Maita Asmat 254615465689768 | | | | | | | |