**LISTA DE CHEQUEO PARA EL DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS**

| **Revisado por** | Carlos Challanca | **Cargo o rol** | | | | **Analista** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **26/06/2022** | **Sitio** | | | |  | |
| **Nombre del proyecto** | **SuperShop** | | | | | | |
| **Descripción:** | | | | | | | |
| **REVISIÓN DEL CONTENIDO GENERAL DEL DOCUMENTO** | | | | | | | |
| **ASPECTO O ELEMENTO A VERIFICAR** | | | **SI** | **NO** | **NO APLICA** | | **OBSERVACIONES** |
| 1. **Introducción** | | |  |  |  | |  |
| 1.1. Propósito | | | **x** |  |  | |  |
| 1.2. Ámbito o alcance | | | **x** |  |  | |  |
| 1.3. Definiciones, siglas y abreviaturas | | | **x** |  |  | |  |
| 1.4. Referencias | | |  | **x** |  | | **No hay información** |
| 1.5. Apreciación Global del documento | | |  | **x** |  | | **No hay información** |
|  | | | | | | | |
| **2. Descripción general** | | |  |  |  | |  |
| 2.1. Perspectiva del producto | | | **x** |  |  | |  |
| 2.2. Funciones del producto | | | **x** |  |  | |  |
| 2.3. Características de los usuarios | | | **x** |  |  | |  |
| 2.4. Restricciones | | | **x** |  |  | |  |
| 2.5. Suposiciones y dependencias | | | **x** |  |  | |  |
| 2.6. Requisitos futuros | | |  | **x** |  | | **No hay información** |
|  | | | | | | | |
| **3. Requisitos Específicos** | | |  |  |  | |  |
| 3.1. Interfaces Externas | | |  |  |  | |  |
| 3.1.1 Interfaces del Sistema | | | **x** |  |  | |  |
| 3.1.2 Interfaces del Usuario | | | **x** |  |  | |  |
| 3.1.3 Interfaces del Hardware | | |  |  | **x** | | **No aplica** |
| 3.1.4. Interfaces del Software | | | **x** |  |  | |  |
| 3.1.5. Interfaces de Comunicaciones | | | **x** |  |  | |  |
| 3.2. Funciones | | |  |  |  | |  |
| 3.3. Requisitos de rendimiento | | |  |  | **x** | | **No hay información** |
| 3.4. Requisitos de datos | | | **x** |  |  | |  |
| 3.5 Restricciones de diseño | | |  |  | **x** | | **No hay documento de diseños** |
| 3.6. Atributos del sistema | | |  |  |  | |  |
| 3.6.1. Funcionalidad | | | **x** |  |  | |  |
| 3.6.1.1. Adecuación | | | **x** |  |  | |  |
| 3.6.1.2. Exactitud | | | **x** |  |  | |  |
| 3.6.1.3. Interoperabilidad | | | **x** |  |  | |  |
| 3.6.1.4. Seguridad de acceso | | | **x** |  |  | |  |
| 3.6.1.5. Cumplimiento funcional | | | **x** |  |  | |  |
| 3.6.2. Fiabilidad | | |  |  |  | |  |
| 3.6.2.1. Madurez | | | **x** |  |  | |  |
| 3.6.2.2. Tolerancia a fallos | | | **x** |  |  | |  |
| 3.6.2.3. Capacidad de recuperación | | |  |  | **x** | | **No hay información** |
| 3.6.2.4. Cumplimiento de la fiabilidad | | | **x** |  |  | |  |
| 3.6.3. Usabilidad | | |  |  |  | |  |
| 3.6.3.1. Capacidad para ser entendido | | | **x** |  |  | |  |
| 3.6.3.2. Capacidad para ser aprendido | | | **x** |  |  | |  |
| 3.6.3.3. Capacidad para ser operado | | | **x** |  |  | |  |
| 3.6.3.4. Capacidad de atracción | | | **x** |  |  | |  |
| 3.6.3.5. Cumplimiento de la usabilidad | | | **x** |  |  | |  |
| 3.6.4. Eficiencia | | |  |  |  | |  |
| 3.6.4.1. Comportamiento temporal | | | **x** |  |  | |  |
| 3.6.4.2. Utilización de recursos | | | **x** |  |  | |  |
| 3.6.4.3. Cumplimiento de la eficiencia | | | **x** |  |  | |  |
| 3.6.5. Mantenibilidad | | |  |  |  | |  |
| 3.6.5.1. Capacidad de ser analizado | | | **x** |  |  | |  |
| 3.6.5.2. Capacidad de ser cambiado | | | **x** |  |  | |  |
| 3.6.5.3. Estabilidad | | | **x** |  |  | |  |
| 3.6.5.4. Capacidad de ser probado | | | **x** |  |  | |  |
| 3.6.5.5 Cumplimiento de la mantenibilidad | | | **x** |  |  | |  |
| 3.6.6. Portabilidad | | |  |  |  | |  |
| 3.6.6.1. Adaptabilidad | | | **x** |  |  | |  |
| 3.6.6.2. Instalabilidad | | | **x** |  |  | |  |
| 3.6.6.3. Coexistencia | | | **x** |  |  | |  |
| 3.6.6.4. Capacidad para reemplazar | | | **x** |  |  | |  |
| 3.6.6.5. Cumplimiento de la portabilidad | | | **x** |  |  | |  |
| 3.6.7. Calidad de uso | | |  |  |  | |  |
| 3.6.7.1. Efectividad | | | **x** |  |  | |  |
| 3.6.7.2. Productividad | | | **x** |  |  | |  |
| 3.6.7.3. Seguridad Física | | |  |  | **x** | | **El sistema no aplica para seguridad física** |
| 3.6.7.4 Satisfacción | | | **x** |  |  | |  |
| 3.7. Otros requisitos | | |  |  | **x** | |  |
|  | | | | | | | |
| **4. Apéndices** | | |  |  |  | |  |
| Observaciones Generales: | | | | | | | |
| FIRMA DEL REVISOR : | | | | | | | |

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

| **Revisado por** | Carlos Challanca | **Cargo o rol** | | | | Tester | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **26/06/2022** | **Sitio** | | | |  | | |
| **Nombre del proyecto** | | **SuperShop** | | | | | | |
| **Nombre de la función o procedimiento** | | **Login** | | | | | | |
| **Descripción:**  **Login del usuario** | | | | | | | |
| **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? | | |  | x |  | | Algunos nombres poseen mas de 8 caractertes |
| 2. ¿El código cumple con los estándares?  El código debe cumplir con los estándares de programación como modularidad, interrelación entre módulos, secuencia lógica, etc. | | | x |  |  | | El código cumple con la modularizacion |
| 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. | | |  | x |  | | No existe documentacion |
| 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. | | | x |  |  | |  |
| 5. ¿Se usan todas las variables, constantes y parámetros? Todo lo creado debe ser utilizado, lo que sobre debe ser eliminado. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| 8. ¿Faltan validaciones?  Verificar cuales validaciones hacen falta: condicionales, estructuras compuestas, procedimientos, funciones… | | |  | x |  | |  |
| 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. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| Observaciones Generales:  En líneas generales se han cumplido con los estándares del uso de funciones y parámetros | | | | | | | |
| FIRMA DEL REVISOR :  Carloschallanca | | | | | | | |

**LISTA DE CHEQUEO PARA EL DO**

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

| **Revisado por** | Carlos Challanca | **Cargo o rol** | | | | Tester | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **26/06/2022** | **Sitio** | | | |  | |
| **Nombre del proyecto** | | **SuperShop** | | | | | |
| **Nombre de la función o procedimiento** | | **Login** | | | | | |
| **Descripción:**  **Login del usuario** | | | | | | | |
| **REVISIÓN DE IMPLEMENTACIÓN DEL LENGUAJE DE PROGRAMACIÓN** | | | | | | | |
| **ASPECTO O ELEMENTO A REVISAR** | | | **SI** | **NO** | **NO APLICA** | | **OBSERVACIONES** |
| 1. ¿Utiliza el comparador === en lugar de ==? | | |  | x |  | | No hace uso del comprador |
| 2.¿Evita usar la función eval ? | | | x |  |  | | No hace uso de eval |
| 3.¿Abusa del uso del condicinoal for? | | |  | x |  | | No abusa del condicional for |
| 4.¿Realiza los comentarios necesarios? | | |  | x |  | | Falta el uso de mas comentarios |
| 5.¿Hace uso de {} en lugar de new Object() al crear un objeto? | | | x |  |  | | No hace uso la palabra clave new Object |
| 6. ¿.Hace uso de[ ] en lugar de new Array() al crear un nuevo array? | | | x |  |  | | No hace de la la sentencia new Array() |
| 7.¿Evita usar la palabra clave var en lugar de usar comas a la hora de declarar? | | |  | x |  | | Usa la palabra clave var en cada línea en lugar de una sola |
| 8.¿Hace uso de notaciones cortas? | | | x |  |  | | Si hace uso de notacinoes cortas |
| 9.¿Hace uso del formato estándar de nombre de variables y funciones? | | | x |  |  | | Usa los estándares de nombre de variables y funciones |
| 10.¿Evita el uso de la palabra clave any? | | |  | x |  | | Abusa del uso de la palabra clave any |
| 11.¿Hace uso de la palabra clave let en lugar de var? | | | x |  |  | | Usa la palabra clave let en lugar de var en las partes necesarias del codigo |
| Observaciones Generales:  El código usa los estándares de las buenas prácticas de desarrollo. | | | | | | | |
| FIRMA DEL REVISOR: Carloschallanca | | | | | | | |

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

| **Revisado por** | Carlos Challanca | **Cargo o rol** | | | | Tester | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **26/06/2022** | **Sitio** | | | |  | | |
| **Nombre del proyecto** | | **SuperShop** | | | | | | |
| **Nombre de la función o procedimiento** | | **CRUD usuarios** | | | | | | |
| **Descripción:**  **Operaciones basicas del usuario** | | | | | | | |
| **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? | | |  | x |  | | Algunos nombres poseen mas de 8 caractertes |
| 2. ¿El código cumple con los estándares?  El código debe cumplir con los estándares de programación como modularidad, interrelación entre módulos, secuencia lógica, etc. | | | x |  |  | | El código cumple con la modularizacion |
| 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. | | |  | x |  | | No existe documentacion |
| 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. | | | x |  |  | |  |
| 5. ¿Se usan todas las variables, constantes y parámetros? Todo lo creado debe ser utilizado, lo que sobre debe ser eliminado. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| 8. ¿Faltan validaciones?  Verificar cuales validaciones hacen falta: condicionales, estructuras compuestas, procedimientos, funciones… | | | x |  |  | | Muchas validaciones de tipo de dato ingresado por el usuario |
| 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. | | |  | x |  | | Algunos errores no llegan al usuario |
| 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. | | | x |  |  | |  |
| 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. | | | x |  |  | |  |
| Observaciones Generales:  En líneas generales se han cumplido con los estándares del uso de funciones y parámetros, solo faltarian validar las operaciones necesarias | | | | | | | |
| FIRMA DEL REVISOR :  Carloschallanca | | | | | | | |

**LISTA DE CHEQUEO PARA EL DO**

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

| **Revisado por** | Carlos Challanca | **Cargo o rol** | | | | Tester | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **26/06/2022** | **Sitio** | | | |  | |
| **Nombre del proyecto** | | **SuperShop** | | | | | |
| **Nombre de la función o procedimiento** | | **CRUD usuarios** | | | | | |
| **Descripción:**  **CRUD usuarios** | | | | | | | |
| **REVISIÓN DE IMPLEMENTACIÓN DEL LENGUAJE DE PROGRAMACIÓN** | | | | | | | |
| **ASPECTO O ELEMENTO A REVISAR** | | | **SI** | **NO** | **NO APLICA** | | **OBSERVACIONES** |
| 1. ¿Utiliza el comparador === en lugar de ==? | | |  | x |  | | No hace uso del comprador |
| 2.¿Evita usar la función eval ? | | | x |  |  | | No hace uso de eval |
| 3.¿Abusa del uso del condicinoal for? | | |  | x |  | | No abusa del condicional for |
| 4.¿Realiza los comentarios necesarios? | | |  | x |  | | Falta el uso de mas comentarios |
| 5.¿Hace uso de {} en lugar de new Object() al crear un objeto? | | | x |  |  | | No hace uso la palabra clave new Object |
| 6. ¿.Hace uso de[ ] en lugar de new Array() al crear un nuevo array? | | | x |  |  | | No hace de la la sentencia new Array() |
| 7.¿Evita usar la palabra clave var en lugar de usar comas a la hora de declarar? | | |  | x |  | | Usa la palabra clave var en cada línea en lugar de una sola |
| 8.¿Hace uso de notaciones cortas? | | | x |  |  | | Si hace uso de notacinoes cortas |
| 9.¿Hace uso del formato estándar de nombre de variables y funciones? | | | x |  |  | | Usa los estándares de nombre de variables y funciones |
| 10.¿Evita el uso de la palabra clave any? | | |  | x |  | | Abusa del uso de la palabra clave any |
| 11.¿Hace uso de la palabra clave let en lugar de var? | | | x |  |  | | Usa la palabra clave let en lugar de var en las partes necesarias del codigo |
| Observaciones Generales:  El código usa los estándares de las buenas practicas de desarrollo, sin embargo algo que destacar es el uso de la palabra clave any en muchas partes del código , el desarrollador declara muchas modelos de datos y sin embargo usa la palabra clave any en demasiadas oportunidades. | | | | | | | |
| FIRMA DEL REVISOR: Carloschallanca | | | | | | | |

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

| **Revisado por** | Carlos Challanca | **Cargo o rol** | | | | Tester | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **26/11/2021** | **Sitio** | | | |  | |
| **Nombre del proyecto** | | **SuperShop** | | | | | |
| **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. | | | x |  |  | |  |
| * 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. | | |  | x |  | | No se encuentra en orden alfabético |
| * 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. | | | x |  |  | |  |
| **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. | | |  | x |  | |  |
| 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? | | |  | x |  | |  |
| **3. Parámetros** | | | | | | | |
| 3.1 ¿El nombre de los parámetros es significativo?  Debe ser representativo y fácilmente identificable | | | x |  |  | |  |
| **4. Constantes** | | | | | | | |
| 4.1 ¿El nombre de las constantes es significativo?  Debe ser representativo y cumplir con los estándares de programación. | | | x |  |  | | Los nombres de constantes indican su funcionamiento |
| **5. Variables** | | | | | | | |
| 5.1 ¿El nombre de las variables es significativo?  Debe ser representativo y cumplir con los estándares de programación. | | | x |  |  | | Los nombres de las vaiables indican su funcionamiento |
| 5.2 ¿Están los nombres de las constantes y de las variables alineados a la izquierda junto a la definición del tipo? | | | x |  |  | |  |
| Observaciones Generales: | | | | | | | |
| FIRMA DEL REVISOR: | | | | | | | |