**Revisión**:

**TSP Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Andres Cabezas – Gilberto Mangones | Date | 27/04/2014 |
| Team | EAIT | Language | Java |
| Instructor | Daniel Benavides | Cycle | 2 |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective develop review |
| General | * Review the entire program for each checklist category; do not attempt to review for more than one category at a time! * As you complete each review step, check off that item in the box at the right. * Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Completar | Verifique que el diseño cubre todos los requerimientos:   * Descripción del problema. * Diseño conceptual. * Diagrama UML de clases. * Relaciones entre clases. * Describe la responsabilidad de cada clase. * Todas las salidas son producidas. * Todas las entradas son proporcionadas y completamente especificadas. | **X** | **√** | X |  |
| Uso Funcional | * Verificar que todos los métodos son completamente comprendidos y apropiadamente utilizados, incluir parámetros de entrada y de retorno. * Todas las abstracciones referenciadas externamente son definidas precisamente. * Que el objetivo de cada método sea real y este alineado con el objetivo principal de la clase. * Confirmar si el tipo de dato asignado a las variables se ajusta apropiadamente de acuerdo a su uso. Tener en cuenta el desbordamiento de los diferentes tipos de variables. * Verificar que todas las interfaces de error están incluidas. | **X** | **√** | **√** |  |
| Variables | * Verificar que las variables tengan nombres nemotécnicos al propósito. * Verificar si las variables están claramente definidas. | **√** | **√** | **√** |  |
| Nombres | Verificar que:   * Todos los nombres (objetos, métodos y parámetros) son claros y acordes con su funcionalidad. * No existen clases anónimas * El alcance de todas las variables y parámetros están definidos. | **X** | **√** | **√** |  |
| Lógica proceso | * Verificar que la lógica diseñada como solución es la correcta. | **X** | **√** | **√** |  |
| Fórmulas Matemáticas | Si el programa lleva fórmulas matemáticas verificar que estas se estén usando correctamente. | N/A | N/A | NA |  |
| Librerías Java | Si se está utilizando una librería desconocida de java, realizar una prueba de concepto y ajustar el diseño de ser necesario. | N/A | N/A | NA |  |
| Casos especiales | Protección para casos fuera de limites | N/A | N/A | NA |  |

**INSPECCIONES**

**Listado de inspecciones a realizar:**

* Inspección de SRS (Plantilla especificación de requerimientos)
* Inspección listas de chequeos de diseño
* Inspección listas de chequeos de desarrollo
* Inspeccionar diagrama de clases
* Inspeccionar diagramas de secuencias
* Inspeccionar casos de uso al final del desarrollo
* Redactar acta
* Subir SumQ
* Organizar la plantillas de inspección

De acuerdo al Plan de Calidad inicial y plantilla de Inspección, se hará inspección:

Plantilla: Inspección

|  |  |
| --- | --- |
| Equipo | EAIT |
| Proyecto | Desarrollo Comparador de Programas |
| Ciclo | No. 2 |
| Documento | Inspección de diseño y artefactos |
| Inspector | Nestor Andres Cabezas y Gilberto Mangones |
| Fecha | 27 de abril de 2014 |
| Duración |  |
| Tamaño | 2 página |
| Autor Documento | Diego Sánchez y Gustavo Suarez |

Hallazgos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Hallazgo** | **Mayor** | **Menor** | **Notas del Autor** |
| 1 | Explicación de cada clase o entidad presentada en el diagrama de clases, describiendo su función y salida |  | x |  |
| 2 | Método Contar Lineas de código, que devuelve? Tipo Entero o impresión. | x |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| TOTAL | | **1** | **1** |  |