**TSP Design Review Checklist**

|  |  |  |
| --- | --- | --- |
| **Evento** | **Autor** | **Fecha** |
| Creación del documento | Gilberto Mangones | 25/04/2014 |
| Revisión | Gilberto Mangones y Nestor Andres Cabezas | 27/04/2014 |
| Revisión | Diego Felipe Sanchez | 03/05/2014 |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Andres Cabezas – Gilberto Mangones | Date | 25/04/2014 |
| Team | EAIT | Language | Java |
| Instructor | Daniel Benavides | Cycle | 2 |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective design 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. |

|  |  |
| --- | --- |
| **ID** | **Descripción** |
| **1** | Revisión del 27/04/2014 |
| **2** | Revisión del 03/05/2014 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Categroría** | **Descripción** | **1** | **2** |
| 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 | **√** |
| 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. | **√** | **√** |
| 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. | **√** | **√** |
| Lógica proceso | * Verificar que la lógica diseñada como solución es la correcta. | **√** | **√** |
| Fórmulas Matemáticas | Si el programa lleva fórmulas matemáticas verificar que estas se estén usando correctamente. | N/A | N/A |
| 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 |
| Casos especiales | Protección para casos fuera de limites | N/A | N/A |