**Software Requirements Specification**

**Change Counter**

1. Table of contents
2. Introduction
   1. Purpose of the SRS
   2. Problem statement
   3. Team Project information
3. Statement of funcional requirements
   1. Description of the system function requirements
   2. Cycle 1 Requirements
   3. Cycle 2 Requirements
   4. Top-down structure
4. Definition of rules used in the requirements
5. External interface requirements
   1. User interface
   2. Screen layouts
6. Design/Implementation constraints
   1. Standards compliance
   2. Development constraints
7. Special system requirements
   1. Documentation
   2. Compatibility
8. References and sources of information
9. Introduction
   1. Purpose of the SRS

Definir detalladamente los objetivos del proyecto Contador de Cambios, de una forma clara y específica, que sirva como base para el plan de pruebas y el diseño de alto nivel.

* 1. Problem statement

Crear una herramienta para contar el tamaño de un programa, que permita identificar el número de LOC agregados, borrados o modificados en cada versión; que sea aplicable para el desarrollo por ciclos iterativos.

* 1. Team Project information

|  |  |
| --- | --- |
| María Paula Forero | Líder del Equipo |
| Nestor Cruz Hernandez | Líder de Calidad |
| Gladys Castro Muñoz | Líder de Planeación |
| William Felipe Rojas | Líder de Desarrollo |
| Juan Pedro Mendoza | Líder de Soporte |
| Julian Andrés Aguirre | Líder 2 de Desarrollo |

1. Statement of funcional requirements
   1. Description of the system function requirements

|  |  |
| --- | --- |
| **Número** | **Requerimiento** |
| 1 | Comparar un programa modificado con su versión anterior |
| 2 | Identificar los LOC agregados y borrados en el programa modificado |
| 3 | Contar los LOC agregados y borrados en el programa modificado |
| 4 | Contar los LOC totales en el programa modificado |
| 5 | Crear una etiqueta de referencia para las líneas agregadas y borradas que indiquen el número de versión |
| 6 | Proveer una etiqueta de cambio en los comentarios de los encabezados del programa, indicando el número de versión, la fecha en la que se realizó el cambio, los LOC borrados, agregados y totales de todo el programa |
| 7 | Si el programa tuvo varios cambios, mantener un registro de estos cambios en la etiqueta de cambios |
| 8 | Cuando un programa modificado previamente es modificado posteriormente y contado, todos los registros de modificación deben ser mantenidos |
| 9 | Producir nuevo código fuente del programa para la versión modificada con la etiqueta de cambios y las etiquetas de línea |
| 10 | Inicialmente el programa debe funcionar para comparar programas escritos en Java |

* 1. Functional Specifications

Las especificaciones funcionales se pueden encontrar en los archivos

* EF-1 Comparar programa version anterior.docx
* EF-2 Identificar los LOCS agregados y borrados.docx
* EF-3 Contar LOCS programa modificado.docx
* EF-4 Agregar etiqueta numero de cambio.docx
* EF-5 Proporcionar etiqueta de cambio.docx
* EF-6 Mantener registro de cambios.docx
* EF-7 Mantener registro de modificaciones.docx
* EF-8 Producir un nuevo archivo de recursos.docx

Los requerimientos asociados a las especificaciones funcionales se pueden ver descritos en la siguiente matriz.

|  |  |
| --- | --- |
| **Requerimiento** | **Especificación funcional** |
| 1. Comparar un programa modificado con su versión anterior | EF-1 Comparar programa version anterior.docx |
| 1. Identificar los LOC agregados y borrados en el programa modificado | EF-2 Identificar los LOCS agregados y borrados.docx |
| 1. Contar los LOC agregados y borrados en el programa modificado | EF-3 Contar LOCS programa modificado.docx |
| 1. Contar los LOC totales en el programa modificado | EF-3 Contar LOCS programa modificado.docx |
| 1. Crear una etiqueta de referencia para las líneas agregadas y borradas que indiquen el número de versión | EF-4 Agregar etiqueta numero de cambio.docx |
| 1. Proveer una etiqueta de cambio en los comentarios de los encabezados del programa, indicando el número de versión, la fecha en la que se realizó el cambio, los LOC borrados, agregados y totales de todo el programa | EF-5 Proporcionar etiqueta de cambio.docx |
| 1. Si el programa tuvo varios cambios, mantener un registro de estos cambios en la etiqueta de cambios | EF-6 Mantener registro de cambios.docx |
| 1. Cuando un programa modificado previamente es modificado posteriormente y contado, todos los registros de modificación deben ser mantenidos | EF-7 Mantener registro de modificaciones.docx |
| 1. Producir nuevo código fuente del programa para la versión modificada con la etiqueta de cambios y las etiquetas de línea | EF-8 Producir un nuevo archivo de recursos.docx |

* 1. Cycle 1 Requirements

|  |  |
| --- | --- |
| **Número** | **Requerimiento** |
| 1 | Comparar un programa modificado con su versión anterior |
| 2 | Identificar los LOC agregados y borrados en el programa modificado |
| 3 | Contar los LOC agregados y borrados en el programa modificado |
| 4 | Contar los LOC totales en el programa modificado |
| 5 | Crear una etiqueta de referencia para las líneas agregadas y borradas que indiquen el número de versión |
| 6 | Proveer una etiqueta de cambio en los comentarios de los encabezados del programa, indicando el número de versión, la fecha en la que se realizó el cambio, los LOC borrados, agregados y totales de todo el programa |

* 1. Cycle 2 Requirements

|  |  |
| --- | --- |
| **Número** | **Requerimiento** |
| 7 | Si el programa tuvo varios cambios, mantener un registro de estos cambios en la etiqueta de cambios |
| 8 | Cuando un programa modificado previamente es modificado posteriormente y contado, todos los registros de modificación deben ser mantenidos |
| 9 | Producir nuevo código fuente del programa para la versión modificada con la etiqueta de cambios y las etiquetas de línea |

* 1. Top-down structure

1. Definition of rules used in the requirements
2. External interface requirements
   1. User interface
   2. Screen layouts
3. Design/Implementation constraints
   1. Development constraints
      1. El programa debe construirse utilizando Java
      2. El sistema debe comprar programas hechos en Java
      3. El sistema debe ser independiente del sistema operativo
      4. El sistema debe almacenar etiquetas de líneas siguiendo el formato del lenguaje que se este comparando
4. Special system requirements
   1. Documentation
      1. Se deberá incluir documentación para la instalación y operación
   2. Compatibility
      1. El sistema deberá ser diseñado pensando en soportar diversos lenguajes de programación, aunque actualmente solo está pensado para Java
5. References and sources of information

* Apendice A, Introduction to Team Software Process, Watts S. Humphrey