# Software Requirement Specification

|  |  |  |
| --- | --- | --- |
| Evento | Autor | Fecha |
| Creación del documento | Jorge Sarmiento | 29/03/2014 |
| Revisión y aprobación | Jorge Luis Sarmiento Herrera  Nestor Andres Cabezas  Gilberto Mangones  Gustavo Andres Suarez  Miguel Angel Dominguez  Diego Felipe Sanchez | 31/03/2014 |

Contenido

[Software Requirement Specification 1](#_Toc383979608)

[1. Introducción 3](#_Toc383979609)

[1.1. Propósito del documento 3](#_Toc383979610)

[1.2. Descripción del problema 3](#_Toc383979611)

[2. Descripción de requerimiento funcionales 3](#_Toc383979612)

[2.1. Descripción de los requerimientos de funciones del sistema 3](#_Toc383979613)

[2.2. Requerimientos del ciclo 1 4](#_Toc383979614)

[2.3. Requerimientos del ciclo 2 4](#_Toc383979615)

[3. Definición de reglas usadas en los requerimientos 4](#_Toc383979616)

[3.1. Conteo 4](#_Toc383979617)

[3.2. Formateo 5](#_Toc383979618)

[3.3. Diferenciación 5](#_Toc383979619)

[4. Requerimientos de interfaz externa 5](#_Toc383979620)

[4.1. Interfaz de usuario 5](#_Toc383979621)

[5. Restricciones de diseño e implementación 6](#_Toc383979622)

[5.1. Estándares de conformidad 6](#_Toc383979623)

[5.2. Restricciones de desarrollo 6](#_Toc383979624)

[6. Requerimientos especiales de sistema 6](#_Toc383979625)

[6.1. Documentación 6](#_Toc383979626)

[6.2. Compatibilidad 6](#_Toc383979627)

[7. Referencias y fuentes de información 7](#_Toc383979628)

# Introducción

# Propósito del documento

Este documento provee las descripciones de las necesidades funcionales para el desarrollo del sistema contador de cambios.

# Descripción del problema

Un contador de cambio es una herramienta de software para contar el tamaño de un programa. Cuando se modifican o desarrollan programas en varios ciclos, es importante conocer cuantas líneas de código (LOC) fueron adicionadas, eliminadas, o modificadas entre las versiones del programa. Las funciones principales de un contador de cambio son las siguientes:

* Conteo, consiste en contar las LOC totales, modificadas, adicionadas y eliminadas de un programa.
* Etiquetado de encabezado, consiste en insertar comentarios en el encabezado de cada programa o clase donde resume la lista de cambios realizados en cada versión.
* Formateo, consiste en dar formato al etiquetado tanto en encabezados como en LOC.
* Diferenciación, cosiste en utilizar algoritmos de comparación que identifican diferencias de LOC entre las diferentes versiones del programa o clase.

# Descripción de requerimiento funcionales

# Descripción de los requerimientos de funciones del sistema

Los requerimientos seleccionados para el desarrollo son los siguientes:

|  |  |
| --- | --- |
| **ID** | **DESCRIPCION** |
| S1 | Diseñe el programa para funcionar inicialmente con Java |
| S2 | Comparar un programa modificado con su versión anterior |
| S3 | Identificar los LOC adicionadas y eliminadas en el programa |
| S4 | Contar los LOC adicionadas y eliminadas en el programa. |
| S5 | Contar el LOC total en el programa modificado |
| S6 | Agregar una etiqueta a la línea, para cada línea modificada o eliminada para indicar el número del cambio |

# Requerimientos del ciclo 1

|  |  |
| --- | --- |
| **ID** | **DESCRIPCION** |
| S1 | Diseñe el programa para funcionar inicialmente con Java |
| S2 | Comparar un programa modificado con su versión anterior |
| S3 | Identificar los LOC adicionadas y eliminadas en el programa |

# Requerimientos del ciclo 2

|  |  |
| --- | --- |
| **ID** | **DESCRIPCION** |
| S4 | Contar los LOC adicionadas y eliminadas en el programa. |
| S5 | Contar el LOC total en el programa modificado |
| S6 | Agregar una etiqueta a la línea, para cada línea modificada o eliminada para indicar el número del cambio |

# Definición de reglas usadas en los requerimientos

# Conteo

|  |  |
| --- | --- |
| **ID** | **DESCRIPCION** |
| R1 | El contador de líneas contará líneas de texto teniendo en cuenta un estándar de codificación el cual requiere que cada línea de código (LOC) ocupe una línea de texto física. |
| R2 | Analizar completamente la sintaxis de todos los archivos fuente de un programa escrito en lenguaje Java y contar las LOC lógicas independiente del formato del programa. |
| R3 | Al comparar un programa con su versión anterior, las comparaciones se realizan de la siguiente manera:   * Si una línea de código está contenida en ambas versiones (original y nueva), a pesar de ser posteriormente modificada, se considera como una línea sin modificar. * Si una línea se encuentra únicamente en la nueva versión y no en la versión original, se considera como línea adicionada. * Si una línea se encuentra únicamente en la versión original y no en la versión nueva, se considera como una línea eliminada. * Inicialmente cuente cada línea cambiada como adicionada o eliminada, no tenga en cuenta las modificadas. * Cuando una sección de código es movida o copiada de un lugar a otro del programa, el programa debe mostrarlas como eliminadas en la ubicación de origen y como adicionadas en la ubicación de destino. Esto mismo se debe aplicar para archivos de código fuente. Intentar rastrear tales segmentos y archivos copiados o movidos está fuera del alcance del requerimiento. |
| R3 | El programa debe analizar archivos de código fuente del lenguaje Java, los cuales tienen las extensión .java. |

# Formateo

|  |  |
| --- | --- |
| **ID** | **DESCRIPCION** |
| R4 | Etiquetar cada línea adicionada o eliminada con el número del cambio en un comentario en el costado derecho de la línea. |
| R5 | En el lugar donde la línea fue eliminada, incluya la línea eliminada en un comentario. |
| R6 | Incluir los números de las líneas al principio de cada una. |
| R7 | Conservar la identación original del código fuente en lo posible. |

# Diferenciación

|  |  |
| --- | --- |
| **ID** | **DESCRIPCION** |
| R8 | El contador de cambios debe ejecutarse independientemente del sistema operativo y no utilizar implementaciones de funciones de diferenciación existentes en el sistema operativo. |
| R9 | La documentación del contador de cambios debe describir la instalación y operación del programa. |

# Requerimientos de interfaz externa

# Interfaz de usuario

La interfaz gráfica de usuario debe ser basada en texto, propia de una aplicación de consola. Se presentará un menú de opciones donde el usuario digita la letra o número correspondiente a la opción de la función que quiere ejecutar, cuando se trate de suministrar las rutas a los archivos o carpetas raíz de los fuentes para comparación el programa pedirá al usuario que digite tales rutas.

Todas las salidas de la interfaz de usuario son basadas en texto.

# Restricciones de diseño e implementación

# Estándares de conformidad

* El programa debe cumplir 100% con los requerimientos establecidos y cada una de las especificaciones descritas anteriormente.

# Restricciones de desarrollo

* Se debe escribir el programa en lenguaje Java versión 7
* Para el versionamiento del código fuente se usará GIT y el repositorio estará alojado en GitHub.

# Requerimientos especiales de sistema

# Documentación

Aparte de la documentación requerida por TSPi, se debe generar la siguiente documentación:

* Javadocs del programa.
* Instrucciones de instalación.
* Instrucciones de uso para usuario final.

# Compatibilidad

* El programa debe poder ejecutarse independiente del sistema operativo.
* Java VM versión 7.

# Referencias y fuentes de información

* Introduction to the Team Software Process, Defining the requirement, capítulo 6. Watts S. Humphrey.
* Introduction to the Team Software Process, The Change Counter Functional Need Statement, anexo A. Watts S. Humphrey.