Tabla de contenido

[Software Requirements Specification 2](#_Toc384208353)

[Introducción 2](#_Toc384208354)

[Propósito del documento 2](#_Toc384208355)

[Objetivo General 2](#_Toc384208356)

[Objetivos específicos 2](#_Toc384208357)

[Planteamiento del problema 2](#_Toc384208358)

[Información del equipo 3](#_Toc384208359)

[Requerimientos Funcionales 3](#_Toc384208360)

[Requerimientos funcionales ciclo 1 3](#_Toc384208361)

[Requerimientos funcionales para el ciclo 2 5](#_Toc384208362)

[Diagrama de casos de uso 7](#_Toc384208363)

[Interfaz de Usuario 8](#_Toc384208364)

[Ventana principal 8](#_Toc384208365)

[Mensaje comparación exitosa. 8](#_Toc384208366)

[Mensaje de error. 8](#_Toc384208367)

[Restricciones 9](#_Toc384208368)

[Rc1 9](#_Toc384208369)

[Rc2 9](#_Toc384208370)

[Rc3 9](#_Toc384208371)

[Referencias y Contexto 9](#_Toc384208372)

[Tema 9](#_Toc384208373)

[Uso 9](#_Toc384208374)

[Desarrollo 9](#_Toc384208375)

[Tecnologías 9](#_Toc384208376)

# Software Requirements Specification

## Introducción

La Ingeniería de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades para actualización o creación de un producto software, estas tareas deben ser documentadas y ser compartidas tanto para el cliente como para el equipo de trabajo y así poder conocer el alcance del proyecto que se está desarrollando; para el caso del proyecto académico en la implementación de TSP no es la excepción, el cual se basa en los requerimientos planteados en el ítem: The Change Counter Functional Need Statement[[1]](#footnote-1) de la literatura asignada para el curso.

### Propósito del documento

Siguiendo el proceso de la metodología TSP sobre la definición de los requerimientos (capitulo 6), en el presente documento se encuentra plasmado los requerimientos determinados por el equipo Enterprise Gear Set para el curso actual.

### Objetivo General

Definir el alcance del proyecto de desarrollo de software por medio de los requerimientos (funcionales, operacionales, interfaz externa) que el sistema va a cubrir.

### Objetivos específicos

* Definir el problema a solucionar.
* Definir las características de los miembros del equipo de trabajo.
* Definir los estándares de diseño y desarrollo del proyecto.
* Definir los requerimientos funcionales del proyecto.
* Separar los requerimientos funcionales a desarrollar en cada ciclo.
* Definir la interfaz de usuario del sistema.

### Planteamiento del problema

El contador de cambio es una herramienta de software diseñada para mantener información cuantitativa de un proyecto de software que se desarrolla por medio de varios ciclos de desarrollo. Por medio de este software se desea contar de manera automática las diferencias entre versiones correspondientes a: total LOC, total LOC agregadas, y total LOC borradas, durante las fases de desarrollo.[[2]](#footnote-2)

### Información del equipo

El equipo que hace parte de Enterprise Gear Set se describe en el siguiente catalogo:

|  |  |  |
| --- | --- | --- |
| Nombre | Rol | Descripción |
| Paula Castellanos | Administradora de procesos | Ingeniera de sistemas, desarrolladora backend en lenguajes POO como Java y .Net |
| Ingrid Echavarría | Administradora de planeación | Ingeniera de sistemas, desarrolladora en lenguaje PHP y analista de sistemas |
| Carlos Felipe Roa | Líder técnico | Ingeniero Electrónico, administrador de proyectos informáticos |
| Felipe Fagua | Administrador de desarrollo | Ingeniero de sistemas, Docente en la Universidad Konrad Lorenz y desarrollador de video Juegos |
| Gabriel Martinez | Administrador de soporte | Ingeniero de Sistemas, desarrollador en lenguaje PHP |
| David Abril | Administrador de Calidad | Ingeniero de Sistemas, desarrollador front end |

## Requerimientos Funcionales

### Requerimientos funcionales ciclo 1

|  |  |
| --- | --- |
| Nombre | R1.1 - Comparar un programa modificado con su versión anterior. |
| Resumen | Dado que se tiene un programa se quiere comparar esta carpeta con su versión anterior para identificar las modificaciones que se han realizaron en la nueva versión. |
| Entradas | |
|  | Programa de la versión anterior. |
|  | Programa de la versión actual |
| Resultados | |
|  | Programa resultado donde se consolidan los cambios. Este programa contiene marcas con las modificaciones. |

|  |  |
| --- | --- |
| Nombre | R2.1 - Identificar las líneas que fueron agregadas en el programa actual |
| Resumen | Se requiere identificar cuales líneas de la versión actual del programa son nuevas respecto a una versión anterior del mismo programa. | |
| Entradas | | |
|  | Líneas del programa actual | |
|  | Líneas de una versión anterior del programa | |
| Resultados | | |
|  | Líneas nuevas del programa respecto a una versión anterior. | |

|  |  |
| --- | --- |
| Nombre | R2.2 - Identificar las líneas que fueron eliminadas en el programa actual |
| Resumen | Se requiere identificar cuales líneas de una versión anterior del programa fueron eliminadas respecto a la versión actual del mismo programa. |
| Entradas | |
|  | Líneas del programa actual |
|  | Líneas de una versión anterior del programa |
| Resultados | |
|  | Líneas eliminadas de una versión anterior del programa respecto a la versión actual. |

|  |  |
| --- | --- |
| Nombre | R3.1 - Determinar el número de líneas que fueron agregadas en el programa actual |
| Resumen | Se requiere determinar el número de líneas de la versión actual del programa que fueron agregadas respecto a una versión anterior del mismo programa |
| Entradas | |
|  | Líneas del programa actual |
|  | Líneas de una versión anterior del programa |
| Resultados | |
|  | Número de líneas nuevas del programa respecto a una versión anterior. |

|  |  |
| --- | --- |
| Nombre | R3.2 - Determinar el numero las líneas que fueron eliminadas en el programa actual |
| Resumen | Se requiere determinar el número de líneas de una versión anterior del programa que fueron eliminadas respecto a la versión actual del programa |
| Entradas | |
|  | Líneas del programa actual |
|  | Líneas de una versión anterior del programa |
| Resultados | |
|  | numero de líneas eliminadas del programa anterior respecto a la versión actual |

|  |  |
| --- | --- |
| Nombre | R4 - Contar el total de LOC en el programa modificado |
| Resumen | De acuerdo al estándar de conteo establecido en el proyecto, deben contarse las líneas de código (lógicas), del programa modificado (versión más reciente). |
| Entradas | Ruta del programa modificado |
| Resultados | Número total de LOC (entero) |

### Requerimientos funcionales para el ciclo 2

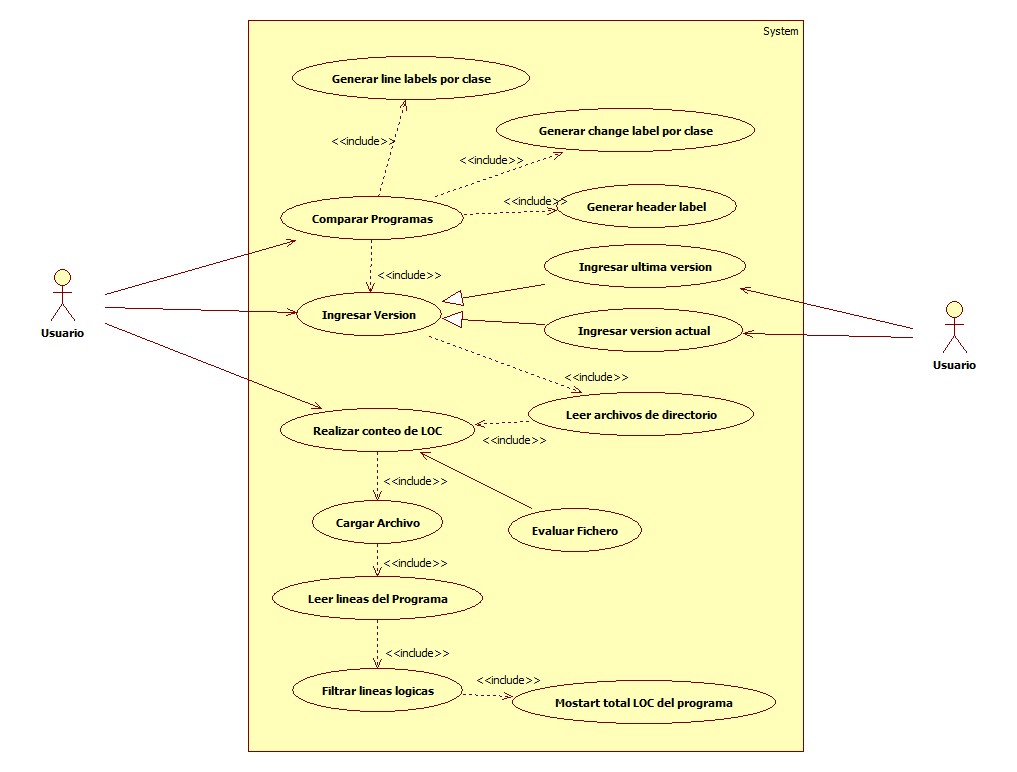
|  |  |
| --- | --- |
| Nombre | R5 - Adjuntar una referencia de línea de etiqueta por cada línea adicionada o borrada para indicar el número de cambio |
| Resumen | Inserta el tipo de modificación y el número de cambio en el nuevo programa |
| Entradas | |
|  | Carácter de comentario. |
|  | Tipo de modificación. |
|  | Numero de cambio. |
|  | Posición de modificación. |
| Resultados | |
|  | Insertar las etiquetas de linea en las clases de programa del nuevo programa. |

|  |  |
| --- | --- |
| Nombre | R6 - Crear un header en las clases del programa que contenga la información del cambio realizado. |
| Resumen | Se requiere plasmar en el archivo fuente de la clase principal un header con la información pertinente al diff de la clase con su versión anterior. En particular se sobre escribe en el archivo el número de la versión, la fecha del cambio, quien lo hizo, porque se hizo, y LOC agregadas, LOC borradas, y LOC total del programa. |
| Entradas | |
|  | Ruta de la clase contenedora del punto de entrada del programa. |
|  | Número del cambio |
|  | Nombre del autor del cambio |
|  | Descripción del cambio |
|  | Número de líneas agregadas |
|  | Número de líneas eliminadas |
|  | Total de líneas lógicas |
| Resultados | |
|  | En el archivo contenedor de la clase con el punto de entrada del programa se escribe el nuevo header en la clase. |

|  |  |
| --- | --- |
| Nombre | R9 - Modificar los archivos fuente del proyecto con los changle-label y log-label correspondientes. |
| Resumen | Se requiere plasmar en los archivos fuente de las clases un header con la información pertinente al diff de la clase con su versión anterior. En adición se quiere escribir comentarios en las líneas que describan los cambios que ha habido en cada archivo durante la comparación. |
| Entradas | |
|  | Ruta de la raíz de los archivos fuente del programa. |
|  | Nombre del autor del cambio |
|  | Descripción del cambio |
|  | Número de líneas agregadas por clase |
|  | Número de líneas eliminadas por clase |
|  | Total de líneas lógicas por clase |
|  | Listado de cambios en cada línea |
| Resultados | |
|  | Todos los archivos fuente del proyecto son modificados con la información del change-label y cada uno de los line-label. |

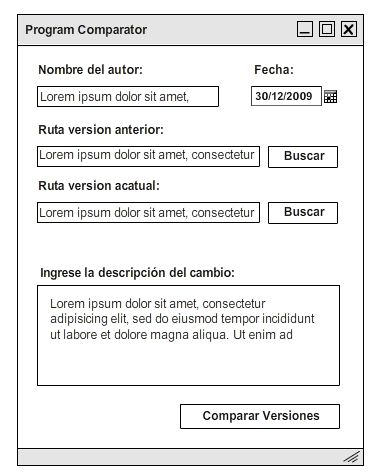
|  |  |
| --- | --- |
| Nombre | R10 - Revisar e identificar la sintaxis de un programa bajo lenguaje de programación Java |
| Resumen | Teniendo en cuenta el estandard de programación grupal de Java de Enterprise Gear Set se evalúa la sintaxis para determinar si es una línea lógica la lectura de ficheros que conforman clases/interfaces. |
| Entradas | |
|  | Directorio raíz de proyecto java |
|  | Reglas de validación de sintaxis lógica de Java |
| Resultados | |
|  | Conteo total de LOC por clase/interface leída |
|  | Cifra total de LOC del programa |

## Diagrama de casos de uso

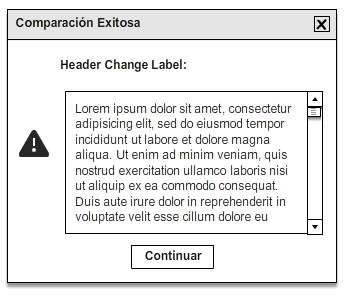


## Interfaz de Usuario

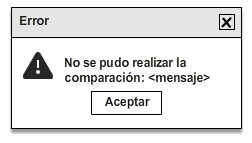
### Ventana principal



### Mensaje comparación exitosa.



### Mensaje de error.



## Restricciones

### Rc1

El producto de software a se debe desarrollar usando la metodología de TSP en 2 ciclos de 4 semanas cada uno.

### Rc2

Los diagramas de representación del sistema deben ser realizados bajo el estándar UML, con la herramienta StarUML.

### Rc3

El ingreso de las rutas de proyectos debe ser manual y es responsabilidad del usuario.

## Referencias y Contexto

### Tema

Los objetos principales del dominio del problema son los archivos de texto donde se encuentran las implementaciones de todas las partes del programa. Este programa se encarga de cargar las líneas de texto que componen el archivo y depurarlas para obtener solo las líneas lógicas, de esta forma compara las líneas lógicas y obtiene el total de LOC agregadas, LOC eliminadas, LOC total de la versión.

### Uso

#### Usuario desarrollador

El usuario final único de la aplicación, es el desarrollador del proyecto de software que desea comparar las versiones del código fuente durante los ciclos de desarrollo.  
El desarrollador deberá seleccionar manualmente las rutas raíz de las dos versiones del proyecto, y procede a hacer la solicitud manual de la comparación, como salida se imprime el reporte del resultado en el header del programa y en cada clase el resultado de la comparación.

### Desarrollo

Todo el proyecto se realizará aplicando la metodología de desarrollo del proyecto es TSP.

### Tecnologías

#### Eclipse Standard 4.3.2

Es el IDE de desarrollo seleccionado para realizar la implementación del programa.

#### GIT

Es la herramienta seleccionada para la administración del repositorio de versiones del proyecto. En el repositorio se guardarán los archivos de gestión del proyecto, y el código fuente del programa.

#### Google Hangout

Es la herramienta usada para la comunicación del equipo durante las reuniones no presenciales.

#### Java SDK 6

Es la versión del development kit de java seleccionada por el equipo que servirá de base para la programación del proyecto.

#### Process Dashboard

Es la herramienta personal para la gestión de tiempo y tareas durante todas las fases de desarrollo del proyecto.

#### Team Dashboard

Es la herramienta de grupo usada para la generación del cronograma general del equipo. El team dashboard centraliza la información de los reportes generados en la herramienta personal de los miembros del grupo. Esta herramienta provee las funcionalidades necesarias para la generación de los formatos de soporte de la metodología seleccionada.

1. Introduction to the Team Software - Humphrey, Watts S, The Change Counter Functional Need Statement, pag 422 [↑](#footnote-ref-1)
2. Introduction to the Team Software - Humphrey, Watts S, The Change Counter Functional Need Statement, pag 422 [↑](#footnote-ref-2)