**SRC CICLO # 1**

Tabla de Contenido

[1. Introducción 2](#_Toc415817323)

[1.1. Propósito de Especificación de requerimiento 2](#_Toc415817324)

[1.2. Planteamiento del problema 2](#_Toc415817325)

[1.3. Información del equipo del proyecto 2](#_Toc415817326)

[2. Especificación de requerimientos funcionales 2](#_Toc415817327)

[2.1. Descripción de los requerimientos funcionales del sistema 2](#_Toc415817328)

[2.2. Requisitos Ciclo 1 2](#_Toc415817329)

[3. Definición de las reglas usadas en los requerimientos 5](#_Toc415817330)

[4. Requerimientos de interfaces externas 5](#_Toc415817331)

[4.1. Interfaces de usuario 5](#_Toc415817332)

[4.2. Diseños de pantalla 5](#_Toc415817333)

[5. Restricciones de diseño / implementación 5](#_Toc415817334)

[5.1. Estándares de cumplimiento 5](#_Toc415817335)

[5.2 Restricciones de desarrollo 5](#_Toc415817336)

[6. Requerimientos especiales del sistema 6](#_Toc415817337)

[6.1 Documentación 6](#_Toc415817338)

[6.2 Compatibilidad 6](#_Toc415817339)

[7. Referencias y fuentes de información 6](#_Toc415817340)

# 1. Introducción

## 1.1. Propósito de Especificación de requerimiento

El propósito del SRS (Especificación de Requerimientos de Software) es producir un listado detallado de los requerimientos funcionales y operacionales del software “analizador De Funciones de Código”.

## 1.2. Planteamiento del problema

Determinar la complejidad de los programas que se han desarrollado previamente para determinar factores de calidad, se ha convertido una herramienta útil y necesaria para los desarrolladores; por lo que se requiere de un software que dé solución al problema planteado.

## 1.3. Información del equipo del proyecto

El equipo Making Solutions esta conformado por:

* Jorge Beltran (Líder Grupo)
* Angela Suarez (Líder Calidad/Proceso)
* Camilo Marroquín (Líder Desarrollo)
* Johans Gonzalez (Líder Planeación)
* John Osorio (Líder Desarrollo)
* Sergio Forero (Líder Soporte)

# 2. Especificación de requerimientos funcionales

## 2.1. Descripción de los requerimientos funcionales del sistema

1. Desarrollar una herramienta de software que ayude a los desarrolladores a determinar la complejidad de los programas que han desarrollado.

## 2.2. Requisitos Ciclo 1

1.1 Analizar un programa para determinar sus características. *(TSP 1.1 Pag 427.)*

1.2 Cuente el LOC lógico del programa y el LOC de cada una de sus funciones, procedimientos o rutinas. *(TSP 1.2 Pag 427.)*

1.3 Calcular los data bindings con el ambiente externo y los data binding de cada una de sus funciones. *(TSP 1.4 Pag 427.)*

1.4 Definición de estándar de Conteo *(TSP 2 Pag 428.)*

|  |  |  |
| --- | --- | --- |
| ***Tipo de conteo*** | ***Tipo*** | ***Comentarios*** |
| *Físico / Lógico* | *Físico* |  |
| ***Tipo de sentencia*** | ***Incluida*** | ***Comentarios*** |
| *Declaraciones* | *SI* |  |
| *Directiva de compilación* | *SI* | *Import, Package* |
| *Comentarios* | *NO* | *(\*) Los asteriscos no son contados* |
| *Líneas en blanco* | *NO* |  |
| *Other elements* |  |  |
| *Anotaciones* | *NO* |  |
|  |  |  |
| ***Aclaraciones*** |  | ***Ejemplos*** |
| *Sentencias vacías* | *NO* | *if (condición){}, ; , {}* |
| *Inicio /fin* | *SI* | *Corchetes de inicio y su correspondencia* |
| *Expresión de asignación o incremento* | *SI* |  |
| *Sentencia válida* | *SI* | *for{int x=0; x…)...}, do{...},else if* |
| *Palabras clave* | *SI* | *Class, Public,* |
| *Etiquetas* | *SI* |  |

1.5 Casos de uso.

|  |  |
| --- | --- |
| **Caso de Uso 001** | |
| **Nombre de caso de uso:** | Presentar resultado de conteo de líneas de código. |
| **Precondición**: | El usuario ha ubicado en el directorio XXXX el proyecto con el código fuente sobre el cual el sistema debe aplicar el conteo de líneas de código por clase y cantidad de líneas de código por método. |
| **Flujo funcional CU001 Básico:** | 1. El sistema valida la existencia del directorio de fuentes. 2. El sistema por medio de la lectora de archivos y el iterador, inicia la identificación de las clases, el conteo de líneas por clase y el conteo de líneas por método. 3. El sistema presenta en pantalla el resultado:  * Clase: Nombre de la clase * Cantidad de líneas en la clase: X * Cantidad de métodos: X * Método: Nombre de método * Cantidad de líneas en método: X |
| **Flujo funcional CU001 Alterno:** | 1. El sistema valida la existencia del directorio de fuentes. 2. En el directorio de fuentes no se encuentra un archivo sobre el cual aplicar el conteo. 3. El sistema presenta en pantalla el mensaje:    * “No existe archivo para aplicar el conteo, el directorio de fuentes está vacío”. |

# 3. Definición de las reglas usadas en los requerimientos

* Se analizará solo código fuente del lenguaje java.
* El estándar de conteo del LOC para el lenguaje se definirá en el numeral 1.4 de este documento.
* La complejidad se basará en la complejidad ciclomática de McCabe

# 4. Requerimientos de interfaces externas

## 4.1. Interfaces de usuario

* Interfaz de resultados “Consola”

## 4.2. Diseños de pantalla

* No aplica para el ciclo 1.

# 5. Restricciones de diseño / implementación

## 5.1. Estándares de cumplimiento

* El desarrollo del software se debe dividir en dos ciclos, cada una debe contener un entregable con los requisitos funcionales establecidos previamente.
* Se debe llevar un control de cambios para cualquier modificación realizada.
* Completar los formatos semanales para medir el avance del proyecto “Grupales e Individuales”
* Completar el notebook del proyecto.
* Realizar reuniones semanales con el equipo.
* Realizar un producto de calidad (cumplir con las revisiones e inspecciones de código según los estándares definidos)

## 5.2 Restricciones de desarrollo

* Lenguaje implementación Java
* Versionamiento en GIT
* Despliegue en Heroku
* Maven

# 6. Requerimientos especiales del sistema

## 6.1 Documentación

* No aplica para este ciclo.

## 6.2 Compatibilidad

* Java 1.8
* JDK 8

# 7. Referencias y fuentes de información

* *Introduction to the Team Software Process TSPi “Watts S. Humphrey”*