**SRC CICLO # 1**

Tabla de Contenido

[1. Introducción 2](#_Toc417740374)

[1.1. Propósito de Especificación de requerimiento 2](#_Toc417740375)

[1.2. Planteamiento del problema 2](#_Toc417740376)

[1.3. Información del equipo del proyecto 2](#_Toc417740377)

[2. Especificación de requerimientos funcionales 2](#_Toc417740378)

[2.1. Descripción de los requerimientos funcionales del sistema 2](#_Toc417740379)

[2.2. Requisitos Ciclo 2 2](#_Toc417740380)

[3. Definición de las reglas usadas en los requerimientos 4](#_Toc417740381)

[4. Requerimientos de interfaces externas 4](#_Toc417740382)

[4.1. Interfaces de usuario 4](#_Toc417740383)

[4.2. Diseños de pantalla 4](#_Toc417740384)

[5. Restricciones de diseño / implementación 5](#_Toc417740385)

[5.1. Estándares de cumplimiento 5](#_Toc417740386)

[5.2 Restricciones de desarrollo 5](#_Toc417740387)

[6. Requerimientos especiales del sistema 5](#_Toc417740388)

[6.1 Documentación 5](#_Toc417740389)

[6.2 Compatibilidad 5](#_Toc417740390)

[7. Referencias y fuentes de información 5](#_Toc417740391)

# 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 Planeación)
* Angela Suarez (Líder Desarrollo)
* Camilo Marroquín (Líder Calidad)
* Johans Gonzalez (Líder Grupo)
* John Osorio (Líder Soporte)
* Sergio Forero (Líder Desarollo)

# 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 2

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

1.2 Calcular los data bindings con el entorno externo de cada procedimiento *(TSP 2 Pag 428.)*

*1.3* Calcular el acoplamiento del programa y sus funciones (*TSP 2 Pag 428.)*

1.4 Calcular la complejidad ciclomatica de McCabe del programa (*TSP 2 Pag 428.)*

1.5 Calcular la complejidad ciclomatica de McCabe de los procedimientos (*TSP 2 Pag 428.)*

1.6 Calcular Fuerza de los data bindings para el programa y sus funciones (*TSP 2 Pag 428.)*

1.7 Calcular el índice de empaquetamiento/robustes (*TSP 2 Pag 428.)*

1.8 Casos de uso.

|  |  |
| --- | --- |
| **Caso de Uso 002** | |
| **Nombre de caso de uso:** | Presentar resultado de análisis de un programa. |
| **Precondición**: | El usuario ha cargado el programa desde el navegador en un archivo “.rar” el cual contiene 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 para realizar las operaciones descritas en el punto 2.2 |
| **Flujo funcional CU002 Básico:** | 1. El sistema valida que el archivo halla cargado correctamente y lo descomprime. 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, el conteo de líneas por método y demás operaciones para retornar un análisis coherente del programa. 3. El sistema presenta en un reporte el siguiente 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 * Data bindings con el entorno externo del programa. * Data bindings con el entorno externo de cada procedimiento. * Acoplamiento del programa y sus funciones. * Complejidad ciclomatica de McCabe del programa. * Complejidad ciclomatica de McCabe de los procedimientos. * Fuerza de los data bindings para el programa y sus funciones. * Índice de empaquetamiento/robustes. |
| **Flujo funcional CU002 Alterno:** | 1. El sistema valida que el archivo halla cargado correctamente y lo descomprime. 2. En el directorio de fuentes no se cargo correctamente. 3. El sistema presenta en pantalla el mensaje:    * “No se cargó el archivo correcto y no se puede generar el análisis.” |

# 3. Definición de las reglas usadas en los requerimientos

* Se analizará solo código fuente del lenguaje java.
* 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”*