**DOCUMENTO DE REQUERIMIENTOS**

**ITERACIÓN 2 TSP**

**STATUS QUO**

**ZAMIR ANDREI GARCÍA ROMERO**

**ÁLVARO DAVID LÓPEZ PINILLA**

**DIEGO ANDRÉS LOZANO ROLDÁN**

**WILMAN RINCON BAUTISTA**

**DEIVIS ENRIQUE VERGEL ARENAS**

**CONCEPTOS AVANZADOS DE INGENIERÍA DE SOFTWARE**

**UNIVERSIDAD DE LOS ANDES**

**2.015**

Tabla de contenido

[Introducción 3](#_Toc416611324)

[Propósito del documento 3](#_Toc416611325)

[Planteamiento del problema 3](#_Toc416611326)

[Información del equipo 3](#_Toc416611327)

[Requerimientos Funcionales (Objetivos) 3](#_Toc416611328)

[Iteración 1 3](#_Toc416611329)

[Iteración 2 3](#_Toc416611330)

[Iteración 3 4](#_Toc416611331)

[Interfaz de Usuario 4](#_Toc416611332)

[Interfaz con otros sistemas 4](#_Toc416611333)

[Atributos de calidad 4](#_Toc416611334)

[Restricciones 5](#_Toc416611335)

[Referencias y Contexto 5](#_Toc416611336)

[Uso 5](#_Toc416611337)

[Tema 5](#_Toc416611338)

[Desarrollo 5](#_Toc416611339)

[Tecnologías 5](#_Toc416611340)

[Anexo A: Estándar de conteo de líneas 6](#_Toc416611341)

# Introducción

## Propósito del documento

El propósito de este documento es hacer la lista de requerimientos que serán contemplados para la primera iteración de la construcción del analizador de complejidad ciclomática de McGabe.

## Planteamiento del problema

El analizador programa es una herramienta de software para ayudar a los programadores a determinar la complejidad de los programas que han desarrollado. Hay evidencia, por ejemplo, que la complejidad McCabe, el acoplamiento / relación de resistencia, y la accesibilidad son útiles predictores de la calidad del programa. Por lo tanto, estos parámetros pueden ser de valor para los desarrolladores de software cuando diseñan programas y más tarde cuando tratan de modificar, ampliar, o mejorarlos. Con la ayuda de estos datos de complejidad, los equipos de desarrollo pueden identificar los componentes que probablemente tienen problemas de diseño, y se pueden solucionar estos problemas antes de las pruebas.

## Información del equipo

El equipo de trabajo debe contar con 5 miembros. Cada integrante deberá de representar un rol durante el proceso de construcción y los posibles son: Lider, Planeacion, Desarrollo, Soporte y Calidad. Las funciones por cada rol están especificadas en el libro de TSP, sin embargo todos los integrantes deberán de participar en las diferentes fases.

# Requerimientos Funcionales (Objetivos)

Los siguientes son los requerimientos que realizaran para las deferentes iteraciones:

## Iteración 1

1. Crear una funcionalidad para determinar las características de un programa.
2. Proveer las funciones de conteo de LOC para el lenguaje java.
3. Contar LOC lógicas y con LOC por cada procedimiento, función o rutina de un programa.
4. Los conteos de LOC deben corresponder a un documento de estándares de codificación para el lenguaje Java.

## Iteración 2

1. Calcular complejidad ciclomática para cada método de un programa.
2. Determinar la accesibilidad de cada procedimiento, función o rutina de un programa.

## Iteración 3

1. Calcular las relaciones con entornos externos y las relaciones por cada procedimiento, función o rutina de un programa.
2. Calcular las relaciones con entornos internos y las relaciones por cada procedimiento, función o rutina de un programa.
3. Determinar el acoplamiento de todo el programa y por cada procedimiento, función o rutina.
4. Producir reporte de análisis con todos los datos recopilados.
5. Proveer las funciones de conteos LOC para otros lenguajes como PHP y C++.

# Interfaz de Usuario

No se especifica una interfaz en particular para cada iteración, pero la interfaz resultado después de las 3 iteraciones debe contener los siguientes componentes:

1. Selección del proyecto (Para subir un proyecto en algún lenguaje de programación)
2. Información general del proyecto subido.
3. Generador del reporte de complejidad.
4. Generador del reporte de características.

El reporte de complejidad debe contar con los siguientes elementos:

1. Descripción del proyecto.
2. Complejidad ciclomática por cada método, ordenada de forma descendente.

El reporte de características debe contar con los siguientes elementos:

1. Descripción del proyecto.
2. Cantidad de líneas lógicas por proyecto, clase y método.

# Interfaz con otros sistemas

En los requerimientos establecidos con el cliente, y la solución planteada no contemplan interfaces ni comunicaciones con sistemas externos.

# Atributos de calidad

El sistema debe de contar con los siguientes atributos de calidad:

1. Disponibilidad: el sistema debe de estar disponible en horarios de 24 X 7.
2. Confiabilidad: Los datos arrojados por el sistema deben ser fiables respecto a que estos datos se usaran para determinar otros proyectos.
3. Usabilidad: El sistema debe garantizar un fácil uso para cualquier usuario inexperto.
4. Mantenibilidad: El sistema debe contar con estándares de codificación en su elaboración, y debe contar con una arquitectura óptima para el uso que va a tener.
5. Portabilidad: El sistema debe tener la opción de ejecutarse desde múltiples plataformas.
6. Reusabilidad: El código generado para este proyecto debe estar codificado por módulos para reusarse en otros proyectos.
7. Capacidad de ser probado: El sistema debe de contar con una interfaz que permita realizar pruebas sobre el mismo.

# Restricciones

No se tienen restricciones para el desarrollo de este proyecto.

# Referencias y Contexto

Las siguientes son las referencias y el contexto de la aplicación:

## Uso

El sistema será usado por desarrolladores, líderes del proyecto, y analistas para determinar el nivel de calidad y estimación de tiempos como base para otros proyectos. El sistema solo debe soportar los siguientes lenguajes de programación: JAVA, PHP, C++. Los programas que se analicen en la aplicación deben contar con un estándar de codificación establecido y preconfigurado en el software.

## Tema

Los objetos que medirán el proceso serán la complejidad V, y será calculada de acuerdo al número de nodos y al número de arcos. Para los conteos se usaran los LOC como medida de acuerdo a un estándar de líneas válidas.

## Desarrollo

El proceso de desarrollo del proyecto actual se realizara mediante la metodología TSP, y deberá contar con validaciones que garanticen la calidad del producto. Durante el desarrollo de otros proyectos, este proyecto se usara en las fases de análisis, diseño y pruebas con el fin de garantizar una codificación ideal y cumplir con estándares de calidad.

## Tecnologías

Las tecnologías que se deberán usar se dejar a cargo del equipo de trabajo.

# Anexo A: Estándar de conteo de líneas

|  |  |  |  |
| --- | --- | --- | --- |
| Definition Name: | LOC java estandar | Language: | JAVA |
| Author: | Status Quo | Date: | 04/04/2015 |

|  |  |  |
| --- | --- | --- |
| **Count Type** | **Type** | **Comments** |
| Physical/Logical | logical |  |
| **Statement Type** | **Included** | **Comments** |
| Executable | si |  |
| Nonexecutable: |  |  |
| Declarations | Si, Nota 1 | Private String var1; |
| Compiler Directives | No, Nota 3 | Import package |
| Comments | No,Nota 2 | // /\* \*/ |
| Blank lines | no |  |
| **Clarifications** |  | **Examples/Cases** |
| Espacios en blanco | no |  |
| Inicios de deciciones ,ciclos , o metodos | si | Se incluyen los inicios de ciclo CASE, DO, ELSE, ENUM, FOR, IF, SWITCH, WHILE etc, el inicio debe estar en la misma línea “{” en caso contrario se contaran los dos como inicio |
| fin de decisiones o ciclos | no | No se contaran los finales de ciclo que estén solos en una línea así “}” |
| Sentencias multilineas | si | Un métodos o sentencia multilinea será contada individualmente |
| Clases Embebidas | no | Se debe crear una clase por archivo .java |
| Note 1 |  | Se identifica como atributo de una clase una declaración por línea , si son del mismo tipo se deberá hacer en una línea independiente |
| Note 2 |  | Se excluyen los comentarios realizados en una línea completa con los símbolos “//” y los comentarios multilinea con los símbolos “/\*” y “\*/”, y las líneas comentarías intermedias siempre que comiencen por el símbolo “\*” |
| Note 3 |  | No se incluyen las líneas de imports y package |