Documento de análisis de requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaborado por:** | Click IT | **Fecha de Elaboración:** | 08-01-2014 |
| **Fecha de entrega** |  |

Control de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 08/01/2014 | 1.0 | Versión Inicial | Equipo Click IT |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla de contenido

[1 Introducción 3](#_Toc384212175)

[1.1 Propósito del documento 3](#_Toc384212176)

[1.2 Descripción 3](#_Toc384212177)

[1.3 Información del Equipo 3](#_Toc384212178)

[2 Análisis Funcional 3](#_Toc384212179)

[2.1 Caso de uso 1: Identificar los LOC adicionados y modificados en el programa. 3](#_Toc384212180)

[2.2 Caso de uso 2: Contar los LOC adicionados y borrados del programa 4](#_Toc384212181)

[3 Requerimientos No funcionales 4](#_Toc384212182)

[3.1 Usabilidad 4](#_Toc384212183)

[3.2 Portabilidad 4](#_Toc384212184)

[3.3 Mantenibilidad 4](#_Toc384212185)

[3.4 Modularidad 5](#_Toc384212186)

[4 Interfaz de Usuario 5](#_Toc384212187)

# Introducción

## Propósito del documento

El presente documento tiene como objetivo realizar un análisis de los requisitos exigidos por el proyecto Contador de Cambios y las implicaciones y condiciones necesarias para su desarrollo. Se especifica en el mayor grado posible de análisis las funciones a realizar en los ciclos de desarrollo del proyecto.

## Descripción

El proyecto tiene como objetivo principal la creación de un aplicativo que permite identificar las modificaciones realizadas a un programa teniendo en cuenta las líneas de código adicionadas y eliminadas.

## Información del Equipo

|  |  |
| --- | --- |
| **ROL** | **INTEGRANTE** |
| **LIDER DE PROYECTO** | Rafael Ortega |
| **LIDER DE DESARROLLO** | Layne Granados |
| **LIDER DE CALIDAD** | Claudia Vargas |
| **LIDER DE PLANEACIÓN** | Fabián Becerra |
| **LIDER DE SOPORTE** | Oscar Velásquez |

# Análisis Funcional

## Caso de uso 1: Identificar los LOC adicionados y modificados en el programa.

|  |  |
| --- | --- |
| **Nombre**:  Identificar los LOC adicionados y modificados en el programa. | |
| **Descripción**:  *Se debe realizar una comparación entre un programa base y el programa modificado para identificar las líneas de código agregadas y eliminadas del programa modificado.* | |
| **Complejidad:** ALTA | **Documentos asociados:** |
| **Actores:**  Usuario Final. | |
| **Precondiciones:**  Se debe tener permisos de acceso a los archivos que se van a leer y analizar. | |
| **Flujo básico:**  El usuario ingresa a la aplicación selecciona mediante la interfaz el programa modificado, el sistema realiza la comparación de este archivo con el archivo base almacenado e identifica las líneas de código que fueron agregadas y eliminadas en el programa modificado. | |

## Caso de uso 2: Contar los LOC adicionados y borrados del programa

|  |  |
| --- | --- |
| **Nombre**:  Contar los LOC adicionados y borrados del programa | |
| **Descripción**:  *El sistema debe contar el número de líneas de código identificadas en la comparación que fueron adicionadas y eliminadas del programa modificado.* | |
| **Complejidad:** MEDIA | **Documentos asociados:** |
| **Actores:**  Usuario Final. | |
| **Precondiciones:**  Se debe haber realizado la comparación para poder contar las líneas identificadas. | |
| **Flujo básico:**  El usuario ingresa a la aplicación selecciona mediante la interfaz el programa modificado, el sistema realiza la comparación y cuenta las líneas de código agregadas y eliminadas que han sido identificadas. | |

# Requerimientos No funcionales

## Usabilidad

El aplicativo tendrá una interfaz amigable y fácil de utilizar para el usuario final que lo guiará a realizar las funciones que dispone. Colores, textos y guías en la interfaz deben ser muy legibles y entendibles para que usuarios sin conocimientos en el aplicativo comprendan las funciones que posee y que puede utilizar.

## Portabilidad

El aplicativo podrá ser ejecutado en cualquier terminal siempre que este tenga instalado el JRE (Java Runtime Environment), de esta forma el terminal cumple con las características básicas de hardware necesarias para ejecutar el aplicativo.

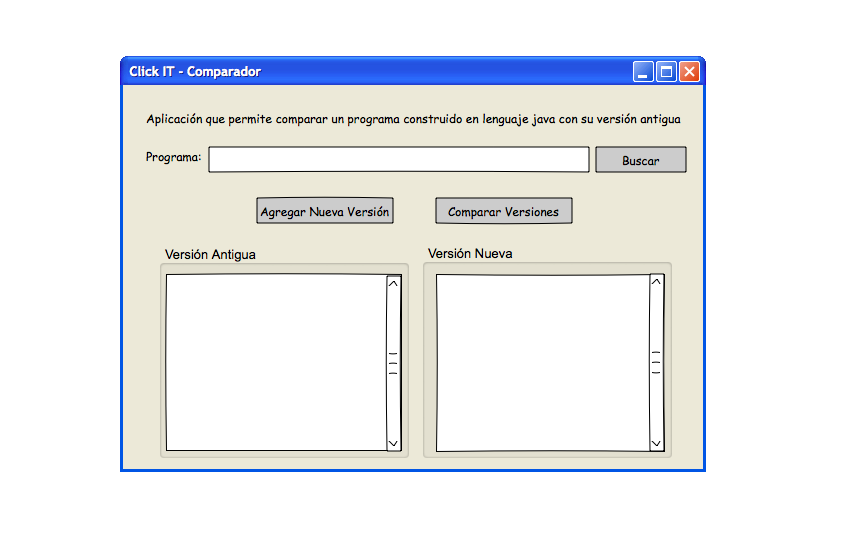
## Mantenibilidad

El sistema debe ser diseñado y desarrollado de tal forma que permita realizar mantenimientos y mejoras sin requerir esfuerzos adicionales. Para ello es importante la documentación de código, la modularidad de la aplicación y la parametrización de propiedades en archivos que permitan realizar cambios sin afectar código fuente.

## Modularidad

El sistema se desarrollará en clases y funciones que cumplirán con labores específicas, esto con el fin de no mezclar lógica de negocio y poder realizar cambios o adición de nuevas funcionalidades, sin requerir esfuerzos adicionales ni afectar la calidad de funcionalidades antiguas.

# Interfaz de Usuario

**Prototipo Pantalla de Comparación**:

# 5. Restricciones

* Restricciones tecnológicas:
  + Lenguaje de desarrollo:java
  + Librerías gráficas a implementar: Swing
  + El software debe poder ejecutarse en cualquier terminal, que tenga instalado JRE.
* Restricciones Organizacionales:
  + Los miembros del equipo podrán trabajar un total de 6 horas semanales.
  + El proyecto se debe desarrollar implementando la metodología TSP y durante su desarrollo se debe diligenciar todos los formatos correspondientes.
  + El proyecto debe realizarse en 2 ciclos y finalizarse el 9 de mayo del 2014
  + El proyecto debe cumplir con los estándares de calidad previamente definidos.