Documento de análisis de requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaborado por:** | Click IT | **Fecha de Elaboración:** | 10-05-2014 |
| **Fecha de entrega** |  |

Control de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 10/05/2014 | 1.0 | Versión Inicial | Equipo Click IT |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla de Contenido

[1 Introducción 3](#_Toc386318442)

[1.1 Propósito del documento 3](#_Toc386318443)

[1.2 Descripción 3](#_Toc386318444)

[1.3 Información del Equipo 3](#_Toc386318445)

[2 Análisis Funcional 3](#_Toc386318446)

[2.1 Caso de uso 1: Identificar los LOC adicionados y modificados en el programa. 3](#_Toc386318447)

[2.2 Caso de uso 2: Contar los LOC adicionados y borrados del programa 4](#_Toc386318448)

[3 Requerimientos No funcionales 4](#_Toc386318449)

[3.1 Usabilidad 4](#_Toc386318450)

[3.2 Portabilidad 4](#_Toc386318451)

[3.3 Mantenibilidad 4](#_Toc386318452)

[3.4 Modularidad 5](#_Toc386318453)

[3.5 Validación de información 5](#_Toc386318454)

[3.6 Desempeño 5](#_Toc386318455)

[3.7 Escalabilidad 5](#_Toc386318456)

[3.8 Interfaz de Usuario 5](#_Toc386318457)

[3.9 Facibilidad para las pruebas 5](#_Toc386318458)

[3.10 Estándares de calidad 5](#_Toc386318459)

[4 Interfaz de Usuario 6](#_Toc386318460)

[5 Restricciones 6](#_Toc386318461)

# 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, para el segundo ciclo.

## 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, agregar una etiqueta en el comentario de encabezado del programa, indicando el número del cambio, la fecha en que fue hecho el cambio, quién hizo el cambio, porqué lo hizo y los LOC adicionados, borrados y totales, funciona con el lenguaje con Java y se imprime un listado con el número de línea del programa modificado al principio de cada línea de código.

## 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: Contar el LOC total en el programa modificado.

|  |  |
| --- | --- |
| **Nombre**:  Contar el LOC total en el programa modificado. | |
| **Descripción**:  *Se debe realizar el conteo de los LOC del programa modificado y mostrarlo.* | |
| **Complejidad:** Media | **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 después de ingresar la nueva versión mostrara los LOC del programa nuevo, es decir el modificado. | |

## Caso de uso 2: Agregar una etiqueta en el comentario de encabezado del programa, indicando el número del cambio, la fecha en que fue hecho el cambio, quién hizo el cambio, porqué lo hizo y los LOC adicionados, borrados y totales.

|  |  |
| --- | --- |
| **Nombre**:  Agregar una etiqueta en el comentario de encabezado del programa, indicando el número del cambio, la fecha en que fue hecho el cambio, quién hizo el cambio, porqué lo hizo y los LOC adicionados, borrados y totales. | |
| **Descripción**:  *El sistema debe mostrar la etiqueta según corresponda al cambio realizado, si es modificada, si es nueva, o si fue eliminada con respecto a la versión anterior.* | |
| **Complejidad:** ALTA | **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. | |

## Caso de uso 3: El programa debe funcionar con Java.

|  |  |
| --- | --- |
| **Nombre**:  El programa debe funcionar con Java. | |
| **Descripción**:  *El sistema debe debe soportar la comparación de archivos Java.* | |
| **Complejidad:** BAJA | **Documentos asociados:** |
| **Actores:**  Usuario Final. | |
| **Precondiciones:**  Se debe haber seleccionado un archivo .java para su analisis. | |
| **Flujo básico:**  El usuario seleccionas desde el browser el archivo .java donde se encuentra el proyecto en el árbol de carpetas para su respectivo análisis. | |

## Caso de uso 4: Imprima un listado con el numero de línea del programa modificado al principio de cada línea de código.

|  |  |
| --- | --- |
| **Nombre**:  Imprima un listado con el numero de línea del programa modificado al principio de cada línea de código | |
| **Descripción**:  *El sistema detalla el cambio de las líneas en el histórico de los cambios subidos y de las nuevas versiones agregadas* | |
| **Complejidad:** ALTA | **Documentos asociados:** |
| **Actores:**  Usuario Final. | |
| **Precondiciones:**  Debe existir al menos una versión. | |
| **Flujo básico:**  El usuario entra al histórico de los cambios, selecciona la fecha del cambio y selecciona el detalle del cambio. | |

# 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 tanto en código como en documentos y manuales, la modularidad de la aplicación, metodología y documentos de diseño 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.

## Validación de información

El aplicativo debe validar la información ingresada por el usuario (ruta del proyecto), verificando que el directorio exista y evitar de esta manera errores en su ejecución

## Desempeño

El aplicativo deberá tener un buen desempeño en las plataformas aprobadas para su ejecución, sin importar el tamaño del aplicativo, el tamaño de los archivos o la cantidad de proyectos que se estén versionando por los usuarios. EL aplicativo deberá requerir un máximo de 2 seg en tiempos de respuesta, no obstante; este tiempo puede variar de acuerdo al computador donde se ejecute.

## Escalabilidad

El aplicativo debe ser desarrollado y diseñado de tal manera, que permita eliminar, modificar y/o incorporar nuevas funcionalidades y requerimientos afectando en el menor grado posible el código desarrollado. Para ello, el código deberá contar con funciones y/o componentes reutilizables.

## Interfaz de Usuario

La interfaz del aplicativo debe funcionar correctamente en las platafomas definidas (Windows, Mac), permitir autoajustar su tamaño de acuerdo a la preferencia del usuario y debe visualizarse correctamente en su tamaño por defecto sin importar la resolución de pantalla del usuario.

## Facibilidad para las pruebas

El aplicativo debe facilitar la realización de pruebas para detección de errores. Para ello, el aplicativo debe contar con pruebas realizadas en código mediante la implementación de JUnit para evaluar el funcionamiento de los métodos y clases desarrollados.

## Estándares de calidad

El aplicativo debe cumplir con los estándares de calidad establecidos para codificación, diseño, manejo de errores, documentación, integración de código y defectos. Para más información, consultar el documento “Estándares de calidad.docx”.

# Interfaz de Usuario para el Cliclo 2

**Pantalla definitiva de Comparación**:



# Restricciones

* Restricciones tecnológicas:
  + Lenguaje de desarrollo:java
  + Librerías gráficas a implementar: Swing
  + El software debe poder ejecutarse en cualquier terminal, que tenga instalado JRE.
  + El programa podrá ejecutarse en Windows y en Mac
* 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.