Documento de Diseño

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaborado por:** | Click IT | **Fecha de Elaboración:** | 28-03-2014 |
| **Fecha de entrega** |  |

Control de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 28/03/2014 | 1.0 | Versión Inicial | Equipo Click IT |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla de contenido

[1 Introducción 3](#_Toc385191158)

[1.4 Propósito del documento 3](#_Toc385191159)

[2 Diseño de Alto Nivel 3](#_Toc385191160)

[2.1 Patrón de Diseño Fachada 4](#_Toc385191161)

[2.2 DTO 4](#_Toc385191162)

[2.3 Propiedades 5](#_Toc385191163)

[2.4 Mensajes de Error 5](#_Toc385191164)

[2.5 Mensajes del Sistema 5](#_Toc385191165)

[2.6 ComparadorException 5](#_Toc385191166)

[3 Diseño detallado 6](#_Toc385191167)

[3.4 Estándar de defectos 6](#_Toc385191168)

[3.5 Estándar para contar LOC 7](#_Toc385191169)

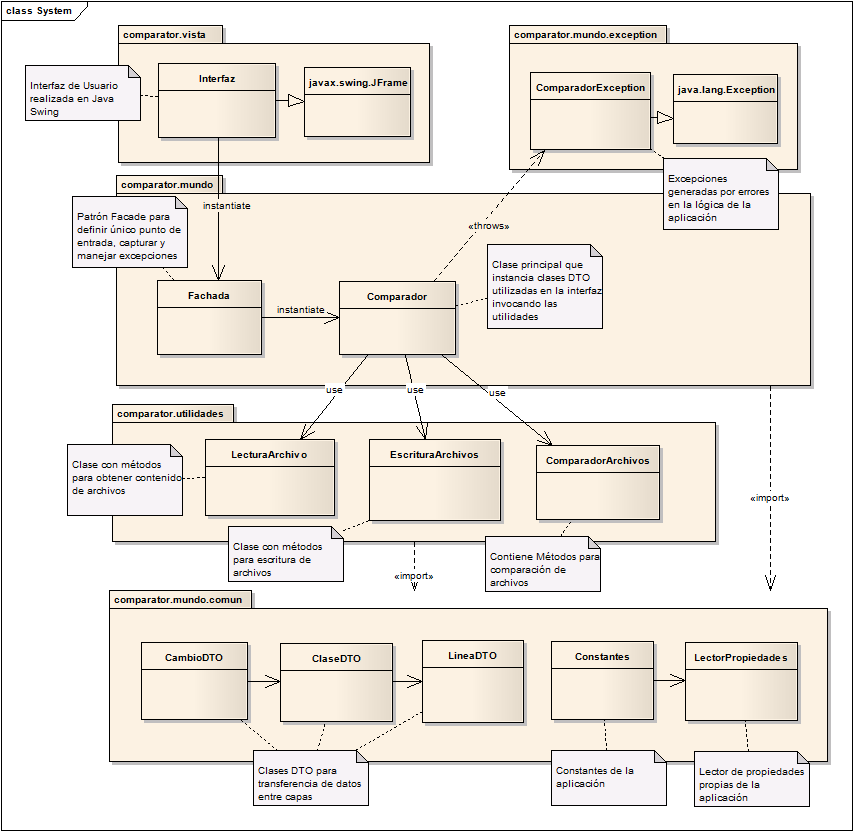
# Introducción

## Propósito del documento

El presente documento tiene como objetivo realizar diseño de alto nivel y detallado que permita al equipo de trabajo comprender e identificar los componentes necesarios a desarrollar, que satisfagan los requerimientos descritos en el documento SRS.

# Diseño de Alto Nivel

A continuación se observa un diseño de alto nivel donde se especifican los componentes a desarrollar y sus relaciones.



## Patrón de Diseño Fachada

Como se observa en el diagrama anterior de alto nivel es necesario utilizar el patrón de diseño Fachada que nos permitirá definir un único punto de entrada entre la interfaz y la lógica de la aplicación. Esta clase será encargada de invocar los responsables para procesar las peticiones provenientes de la interfaz y dar manejo a posibles errores provenientes de las capas responsables, con el fin de que no se generen errores que no se puedan controlar en la interfaz.

## DTO

Se considera conveniente la creación de clases DTO (Objeto de transferencia de datos) para transportar los datos entre capas. Esto se realiza debido a que reduce el costo entre llamados y a que puede mejorar el rendimiento de la aplicación si en algún momento se deba realizar una modificación al programa en la que los datos deban viajar entre interfaces remotas.

## Propiedades

Debe existir un archivo de propiedades llamado **“comparator-config.properties”**. Con esto se puede mejorar la flexibilidad del programa al permitir realizar cambios en los valores sin tener que afectar el código de la aplicación. Algunas propiedades que deben existir en el archivo son las etiquetas que se colocan para líneas agregadas o eliminadas, rutas de archivos históricos o modificados y valores que el programador considere que puedan cambiar a futuro.

## Mensajes de Error

Los mensajes de error se definen en el archivo de propiedades para permitir su actualización en cualquier momento. Los mensajes de error de la aplicación deben ser:

ECT000=Error no manejado.

ECT001=Error al cargar las propiedades del Sistema.

ECT002=No se han encontrado archivos a comparar.

ECT003=Error en el sistema al actualizar la nueva versión.

## Mensajes del Sistema

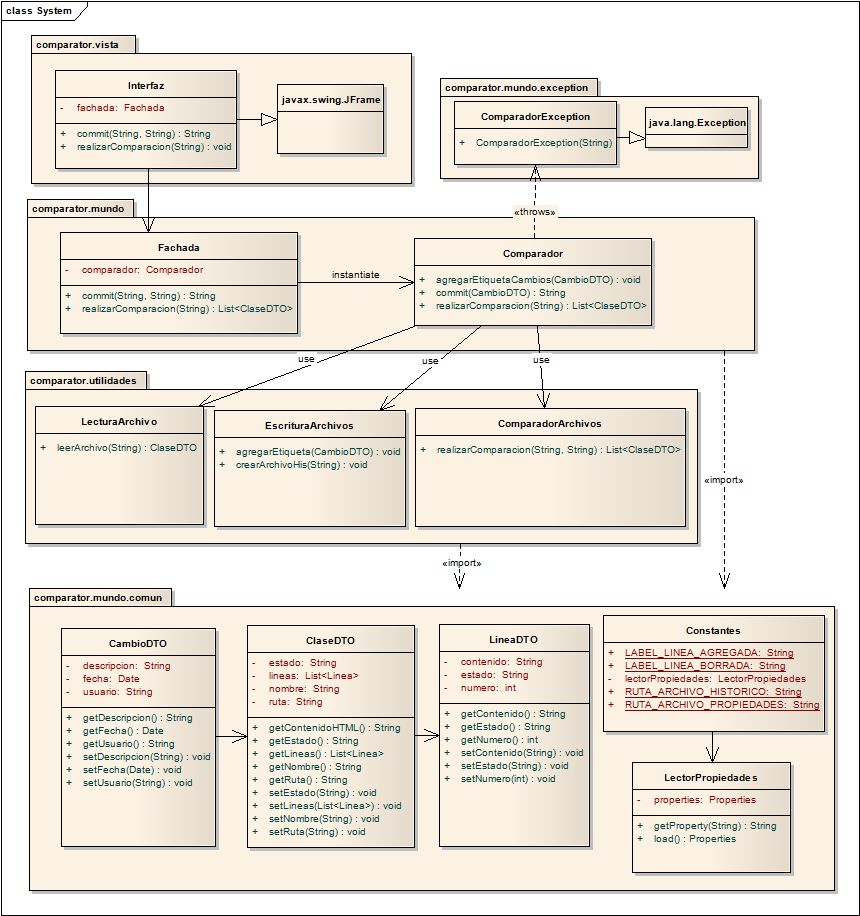
SCT000=Comparación realizada correctamente.

SCT001=Actualización realizada correctamente.

## ComparadorException

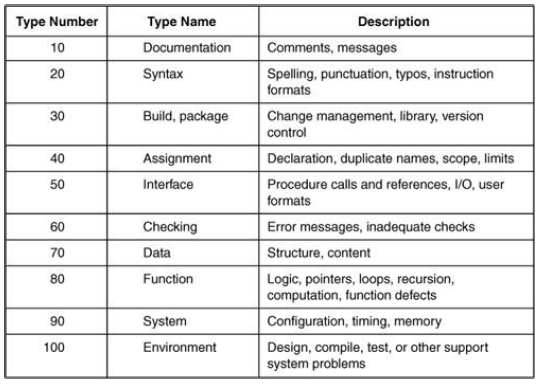
Se define una clase Exception llamada ComparadorException propia de la aplicación para el manejo de errores que se puedan presentar en la lógica de negocio.

# Diseño detallado



## Estándar de defectos

Para el estándar de defectos se utilizarán los mismos vistos en PSP:



## Estándar para contar LOC

El estándar para contar LOC está definido en el documento **“Estandares de Calidad.docx”.**