

|  |
| --- |
| Documento de Análisis y Diseño |
| Nivel 16: [LlamaTech Decoder]  [Edgar Andrés Margffoy] |
| [Camila García] |
| Estructuras de Datos  Universidad de Los Andes  [2015-1] |
|  |

Documento de Análisis y Diseño

[Nombre del ejercicio]

|  |  |  |
| --- | --- | --- |
| **Versión del documento** | **Fecha de modificación** | **Responsables** |
| 1.0 | 15/04/15 | Edgar Andrés Margffoy  Camila García |
|  |  |  |
|  |  |  |

Contenido

[1 Requerimientos Funcionales 3](#_Toc409507633)

[2 Análisis del modelo del mundo 4](#_Toc409507634)

[3 Diseño de Interfaces 5](#_Toc409507635)

[3.1 Interfaces del mundo de la aplicación 5](#_Toc409507636)

[3.2 Interfaces de las estructuras de datos 5](#_Toc409507637)

[4 Diseño de estructuras de datos 6](#_Toc409507638)

[5 Justificación de decisiones de diseño de las estructuras de datos 7](#_Toc409507639)

[6 Diseño final de la aplicación 8](#_Toc409507640)

[7 Justificación de decisiones de diseño del mundo 9](#_Toc409507641)

[8 Análisis de complejidad de operaciones del mundo 10](#_Toc409507642)

[9 Mapa de Navegación (GUI) 11](#_Toc409507643)

# Requerimientos Funcionales

A continuación se describen los requerimientos funcionales de la aplicación. La definición de estos es independiente de la tecnología que será usada para implementarlos.

|  |  |
| --- | --- |
| **ID** | R1 |
| **Nombre** | Cargar un diccionario de palabras |
| **Resumen** | Carga la información de un diccionario dado por el usuario. |
| **Entradas** | Diccionario. Puede ser texto o archivo de texto |
| **Resultados** | El sistema tiene ahora el diccionario y puede reconocer palabras que pertenecen al mismo. |

|  |  |
| --- | --- |
| **ID** | R2 |
| **Nombre** | Sistema virtual de carpetas |
| **Resumen** | La aplicación web depsliega un sistema de carperas que pueden crearse o borrarse donde se almacenan los archivos. Puede importarse y exportarse. |
| **Entradas** |  |
| **Resultados** |  |

|  |  |
| --- | --- |
| **ID** | R3 |
| **Nombre** | Cargar documento para reconocimiento |
| **Resumen** | Cargar documento desde texto o archivo de texto |
| **Entradas** | Texto o archivo de texto |
| **Resultados** | El texto se encuentra guardado en el sistema |

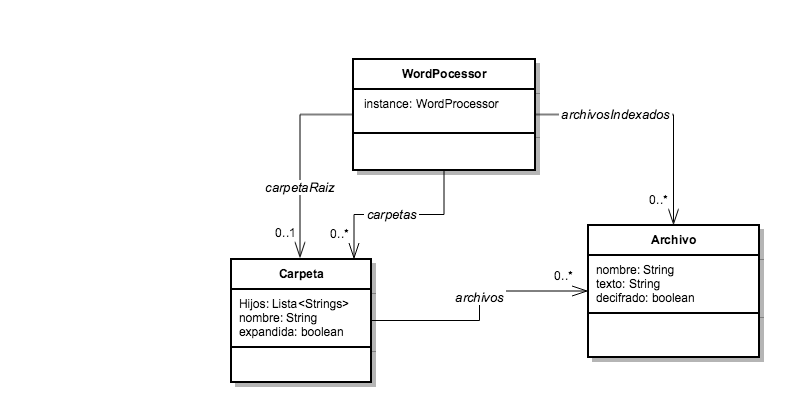
|  |  |
| --- | --- |
| **ID** | R4 |
| **Nombre** | Reconocimiento de documentos |
| **Resumen** | Reconoce el archivo dañado y devuelve el texto reconocido |
| **Entradas** | Escoge archivo a decodificar |
| **Resultados** | Se le muestra al usuario el texto reconocido |

|  |  |
| --- | --- |
| **ID** | R5 |
| **Nombre** | Indexar documentos |
| **Resumen** | Cada vez que se hace el reconocimiento el archivo reconocido se guarda en la misma carpeta que el archivo sin reconocer |
| **Entradas** |  |
| **Resultados** | El archivo está ahora en el sistema de carpetas |

|  |  |
| --- | --- |
| **ID** | R6 |
| **Nombre** | Busqueda de documentos indexados |
| **Resumen** | Dada una palabra o un prefijo se muestran todos los archivos que la contienen |
| **Entradas** | Palabra o prefijo |
| **Resultados** | Se muestran en una lista todos los archivos que contienen la palabra o el prefijo |

# Análisis del modelo del mundo

Diagrama de clases (UML) de la abstracción del problema que se pretende solucionar. Solo se tienen en cuenta entidades, características y relaciones relevantes para el mismo. Aún no se toman decisiones de diseño.



# Diseño de Interfaces

## Interfaces del mundo de la aplicación

A partir de los requerimientos funcionales, defina las operaciones (a través de métodos) que debe proveer el mundo del problema. Incluya parámetros, tipos de retorno y documentación completa de cada operación (incluyendo precondiciones).

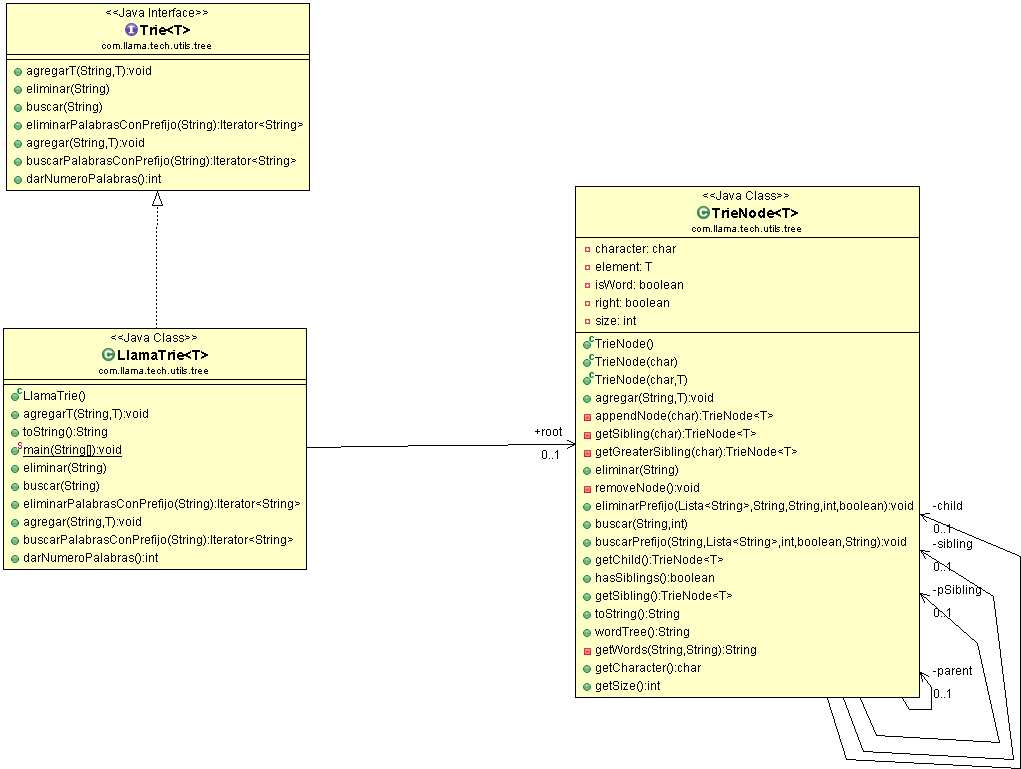
## Interfaces de las estructuras de datos

Seleccione cuáles son las estructuras de datos genéricas que debe utilizar y las operaciones que estas deben proveer para minimizar la complejidad temporal de las operaciones definidas en el numeral 3.1.

Se creó una nueva interfaz para definir el Trie. Basada en la estructura que se envió para las pruebas, nos permite encontrar, añadir y eliminar valores del Trie basados en las palabras que definen su estructura. Es apropiado para el problema, pues permite buscar, por ejemplo, todos los archivos que contengan una palabra.

# Diseño de estructuras de datos

Describa el detalle de las Estructuras de Datos que va a utilizar en su solución, a través de un diagrama UML de las mismas. No olvide incluir invariantes si las requiere.



# Justificación de decisiones de diseño de las estructuras de datos

Describa en detalle los aspectos que tuvo en cuenta para seleccionar y diseñar las estructuras de datos descritas en el punto anterior.

El Trie es extremadamente útil para éste problema, pues se tratan diccionarios y búsquedas de archivos por palabras. Por lo tanto, el Trie permite que a cada palabra se anexen todos los documentos que la contienen, y que además sea eficiente buscar palabras para la búsqueda del reconocimiento.

# Diseño final de la aplicación

Describa en detalle el diseño final de su aplicación a través de un diagrama UML, incluya atributos y métodos. Cuando se refiera a una estructura de datos no es necesario que incluya toda su definición dentro del diagrama. Basta con referenciar su interfaz.

# Justificación de decisiones de diseño del mundo

Describa en detalle los aspectos que guiaron el diseño final de su aplicación.

Para eñ diseño del mundo tuvimos en cuenta que la aplicaicón debía guardar y manejar tanto archivos como carpetas, razón por la cual se crearon las respectivas clases. Por otro lado, se necesitaba la clase principal que manejara todo esto.

# Análisis de complejidad de operaciones del mundo

Basado en los diseños de las estructuras de datos y en el mundo de la aplicación, defina la complejidad de cada una de las operaciones. Utilice la notación O(f(n)). Justifique por qué considera que esta es la mínima posible para el problema que está resolviendo.

R1: O(n\*#de letras\*anchura del arbol\*mayor longitud palabra)=O(n) donde n es el número de palabras a cargar. Cada letra debe añadirse, en el peor de los casos, hasta el lugar más lejano, es decir, más a la derecha del árbol. Sin embargo, como tanto el número de letras como la longitud del árbol y su anchura son independientes a el número de palabras, se toman como constantes.

R2: Borrar y crear carpeta: O(n) en peor caso, O(1) en caso promedio, donde n es el numero de carpetas. La idea es tener una lista de nombres de carpetas que pertenecen a otra carpeta, y los nombres dan pie a un diccionario donde el nombre es la llave y la carpeta el valor. Teniendo el nombre de la carpeta, basta con buscarlo en el diccionario.

R3 y R5: O(n\*#de letras\*anchura del arbol\*mayor longitud palabra)=O(n). Esto sucede pues es muy parecido a agregar una nueva palabra. Debe seguir el mismo proceso, hasta que alcanza el final y añade el elemento (archivo). Por otro lado, debe agregarlo también a la carpeta seleccionada. Ya que se usa un diccionario, es O(n) en el peor caso y O(1) en el promedio.

R4: O(n\*n). Donde n es el número de letras del texto, pues debe buscar letra por letra si la palabra existe, y en caso de fallar, devolverse, lo que causará que en el peor de los casos, deba volver n veces hasta el principio.

R6: O(n\*#de letras\*anchura del arbol\*mayor longitud palabra)=O(n) pues buscar sigue un procedimiento parecido a agregar o indexar. Busca la palabra en el Trie, y devuelve todos los elementos asociados. Es prudente notar que existe una pequeña mejora en tiempo de ejecución, aunque no complejidad, pues al estar ordenado el árbol, se puede saber si no se debe buscar más a la derecha cuando se llega a un carácter mayor lexicograficamente. O(n\*n) cuando es por perfijo, pues primero debe buscar todas las palabras y luego, palabra por palabra, buscar los archivos indexados.

# Mapa de Navegación (GUI)

Basado en los requerimientos funcionales, defina un bosquejo general con los elementos que definen la interfaz gráfica (GUI) de la aplicación. Puede usar la herramienta *Balsamiq Mockups* (http://balsamiq.com/products/mockups/) u otra que permita una funcionalidad similar (NO



