

Instituto Tecnológico de Costa Rica

Algoritmos y estructuras de datos

Proyecto I

LinkedDB

Andrés Artavia

Carnet: 2017075875

Profesor: Antonio González

II Semestre

Índice

Introducción.....	3
Descripción del problema.....	3
Historias de usuario y características.....	3
Ordenamiento historias de usuario.....	3
Especificaciones del programa.....	4
Requerimientos del sistema.....	4
Bitácora.....	5
Diagrama componentes.....	6
Diagrama de clases.....	7
Bibliotecas utilizadas.....	7
Estructuras de datos utilizadas.....	8
Algoritmos desarrollados.....	8
Problemas encontrados.....	8
Bibliografía.....	9

Introducción

El presente es la documentación del proyecto desarrollado para el curso: Algoritmos y Estructuras de datos I. Consiste en una aplicación que lee archivos, los despliega y los modifica. Además, que deja buscar el contenido de estos archivos y eliminar.

Descripción del problema

Se debe crear una aplicación para crear, leer y modificar, archivos que contengan objetos en formato “.Json”. El programa se debe crear en java, utilizando javafx, para la creación de la interfaz.

Historias de usuario y característica

1. El programa crea y lee archivos “.Json”, que contienen una lista con archivos del mismo formato.
2. El programa deja que el usuario acceda a cualquiera de estos archivos y lo modifique.
3. El programa puede agregar más archivos y también puede agregar más características a los archivos existentes o nuevos.
4. El programa permite buscar el archivo que contiene una característica.
5. El programa permite eliminar cualquier archivo o característica creada.
6. El programa presenta una lista con los cambios realizados.

Ordenamiento historias de usuario por importancia y secuencia de uso

1. Leer los documentos “.Json”.
2. Leer el contenido de los mismos.
3. Crear un árbol con los documentos y su contenido.
4. Desplegar en una tabla un documento o una característica para poder modificarlo.
5. Poder guardar los cambios.
6. Poder descartar los cambios al no guardar.
7. Eliminar algún documento o característica.
8. Presentar una lista con los cambios realizados

Especificaciones del programa

El programa crea archivos “.Json” en los que almacena otros objetos del mismo formato.

Debe de modificar los objetos anteriores y añadir más si se necesita, además de eliminarlos.

El programa debe de tener un árbol con todos los documentos y sus características.

Para aplicar los cambios se debe realizar un botón de “commit”, sino se descartan.

Requerimientos del sistema

El sistema debe de tener al menos la versión 1.8 de java, de lo contrario podría dar problemas. Para ejecutar el programa desde la consola, se debe de tener configurado java, como una variable del sistema. Se necesita tener configurada la biblioteca de json-simple-1.1, incluida en el paquete del proyecto.

Bitácora

Semana	Actividad	Duración
2	Investigación sobre como implementar la lectura y escritura de Json en java. Creación clases para crear y leer Json	6 Horas
3	Implementación de escritura en memoria de archivos. Diseño interfáz. Desarrollo ventana de la interfáz	2 Horas
5	Creación árbol de documentos. Creación de las funciones miselaneas de la interfáz, crear etiquetas, ícono, tamaño, etc.	10 Horas
6	Creación de context menú para el árbol, definición de funciones de crear nuevo, eliminar y modificar; también sus respectivas verificaciones. Implementación de tabla para modificar los documentos. Intentos agregar valores a la tabla.	15 Horas
7	Modificación de las funciones para crear y leer archivos. Crear cellFactory para tabla de modificar, y lo demás necesario para agregar los valores a la tabla para modificar. Implementación de la eliminación y creación de achivos, en memoria. Creación de la ventana para los cambios, e implementación de la misma. Creación de botón de commit. Inicio documentación.	20 Horas
8	Implementación de botón commit y deshacer y rehacer. Toques finales en el código. Término de la domjdocumentación.	10 Horas

Diagrama componentes

Manejo de Archivos “.Json”.

- Clase que crea la estructura de los documentos.
- Clase que recibe la estructura anterior y escribe en memoria.
- Clase que lee la carpeta donde se guardan los documentos y lee todos los documentos disponibles.
- Clase que recibe una lista con los archivos disponibles de la función anterior y lee un archivo específico y su contenido.

Interfaz

- Creación de la ventana principal
- Función que crea el árbol de documentos.
- Funciones que crean componentes secundarios, botones, menús, etiquetas.
- Función que crea el espacio que despliega los cambios.
- Función que crea la tabla.

Manejo elementos de interfaz y datos.

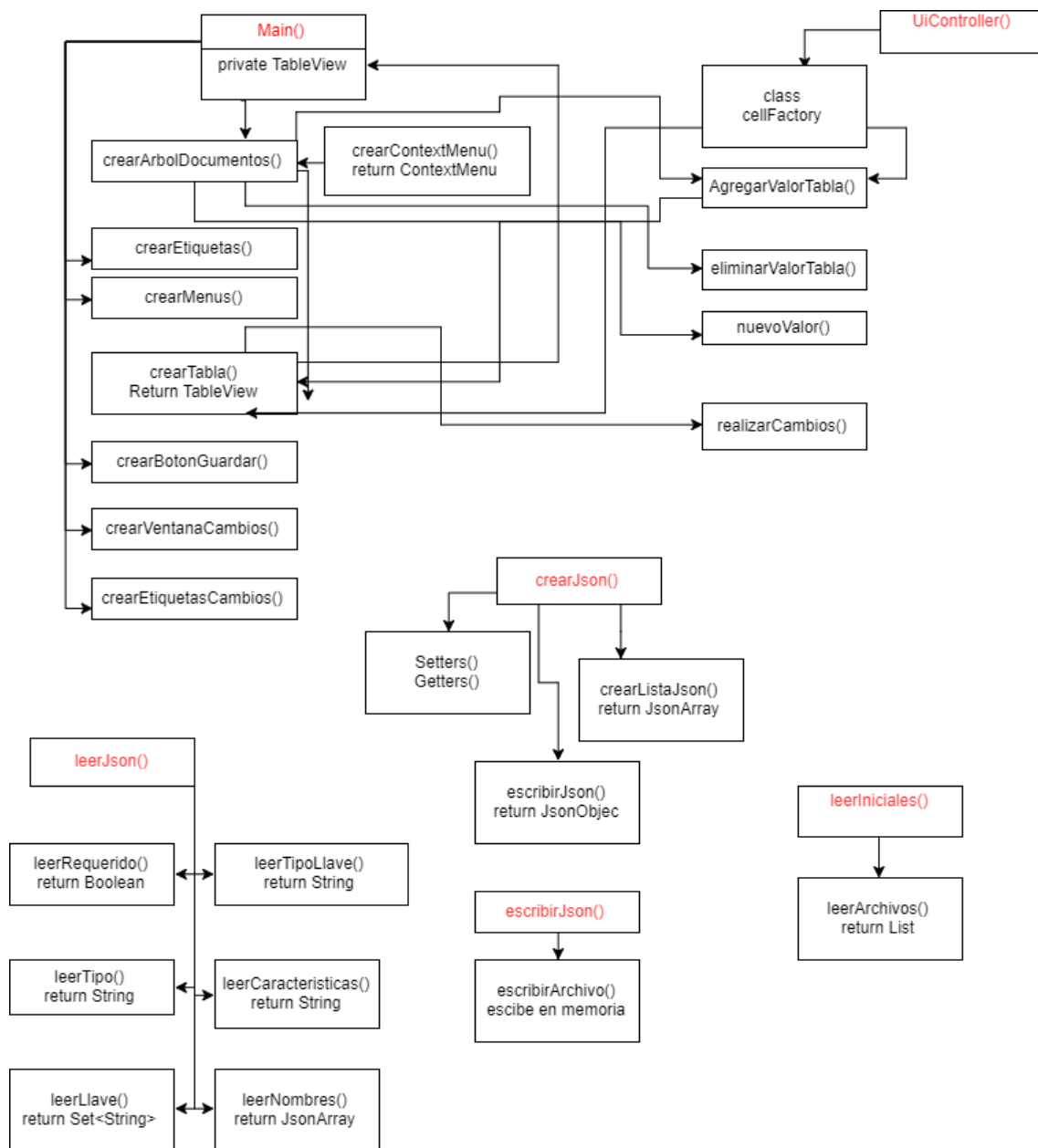
De forma general:

- Creación de diferentes funciones para la comunicación de la interfaz con las clases de manejo de archivos.
- Creación métodos que manejan los eventos de la interfaz

Creación ventanas extra

- Funciones que reciben un dato y crean una ventana de diálogo secundaria.

Diagrama de clases



Bibliotecas externas utilizadas

Para realizar el proyecto se requirió una biblioteca para interpretar el formato “.Json”.

Se utilizan las clases de la biblioteca para definir los objetos con el formato de llave: Característica. Se utiliza la clase para crear arreglos que contienen objetos con este formato.

Estructuras de datos utilizadas

Se utilizan como base de los archivos de guardado, el formato en Json. También se utilizaron arreglos de diferentes tipos, observables para definir las celdas de la tabla, listas para almacenar datos y arreglos de Json, para agrupar los Json en una lista y facilitar su manejo en los documentos.

Algoritmos utilizados

El primer algoritmo creado, es el de creación de los archivos Json, el cuál recibe el nombre del archivo, nombre de la característica, tipo de dato y demás atributos que se solicitaron que el documento contenga. Este crea un primer objeto Json, el cual se le asigna de llave el nombre del archivo y de contenido un arreglo de Json. A este arreglo se le agrega otro objeto Json, el cual contiene un arreglo con los datos de las características y tiene de llave la característica. Este también verifica si es requerido, para definir los valores por defecto y verifica si el tipo de entrada coincide con el tipo especificado.

Otro algoritmo creado, es el que lee cada parte de los documentos Json. Este utiliza la capacidad de las listas creadas anteriormente de utilizar los índices, junto con el orden secuencias de lectura que se utiliza en el árbol de documentos, para ver cuál documento en específico se requiere y devolver todos los datos del mismo. Todo esto utilizando el nombre del documento y el índice en donde va a estar la característica que se desea ver.

Problemas encontrados

- El programa al aplicar los cambios agrega los cambios a las cosas existentes. Al dar guardar o commit, no se modifican las cosas, sino que se le agrega a la lista que contiene las características.
Se debe de modificar la clase que escribe los cambios para recibir los índices que corresponden a las características modificadas y manejar estos mismos en el controlador de la interfaz
Fuentes: Solución propia
- Usar la clase TableView.
La clase anterior requería de muchos métodos para agregar valores a las celdas, y accederlas. Se creó una clase para crear los valores de las líneas, también de una lista observable que contiene las diferentes instancias de la clase anterior.
Fuentes:

Docs.oracle.com. (2017). *TableView (JavaFX 8)*. [online] Available at: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TableView.html> [Accessed 2017].

Conclusión

Java es una herramienta que permite crear muchos programas con eficiencia y con una amplia variedad de bibliotecas integradas. Se logró crear una aplicación con casi todos los requerimientos, ya que se contaba también con una amplia variedad de sitios web e información extra para lograr encontrar soluciones a los problemas encontrados.

Bibliografía

Docs.oracle.com. (2017). *TableView (JavaFX 8)*. [online] Available at: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TableView.html> [Accessed 2017].

Docs.oracle.com. (2017). Home: Java Platform, Standard Edition (Java SE) 8 Release 8. [online] Available at: <http://docs.oracle.com/javase/8/> [Accessed 2017].

Stackoverflow.com. (2017). All Java Language Topics. [online] Available at: <https://stackoverflow.com/documentation/java/topics> [Accessed 2017].