

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores
(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores
(Licentiate Degree Program in Computer Engineering)

Curso: CE-1103 Algoritmos y Estructuras de Datos I
(Course: CE-1103 Algoritmos y Estructuras de Datos I)



Proyecto #1 - LinkedDB
(Project #1 - LinkedDB)

Realizado por:
Made by:

Deiber Granados Vega, 2017159397

Profesor:
(Professor)

Jose I. Ramirez Herrera

Fecha: Cartago, Septiembre 20, 2017

II Semestre, 2017

Índice

Introducción	3
Objetivo General y Objetivos Específicos	4
Descripción del Problema	5
Planificación y administraron del proyecto	6
Diseño	9
Implementación	12
Problemas encontrados	13
Conclusiones y Recomendaciones del proyecto	13
Bibliografía	14

Introducción

En el presente documento se presentará un motor de base de Datos.

El cual permite: definir, insertar, eliminar, actualizar e incluso buscar documentos JSON, esto por medio del uso de listas y archivos JSON que llevarán el control de la información suministrada. Además, cabe destacar que se presenta una interfaz gráfica desarrollada con Java FX. Dicha interfaz es similar a una herramienta de bases de datos, como DBVisualizer, DataGrip, MySQL Workbench y otras.

Objetivo General

- Diseñar e implementar un motor de bases de datos sencillo basado en listas enlazadas.

Objetivos Específicos

- Implementar listas enlazadas y algunas variaciones.
- Aprender sobre algunos conceptos básicos de bases de datos.
- Investigar y desarrollar una aplicación en el lenguaje de programación Java.
- Investigar acerca de programación orientada a objetos en Java.
- Aplicar patrones de diseño en la solución de un problema.

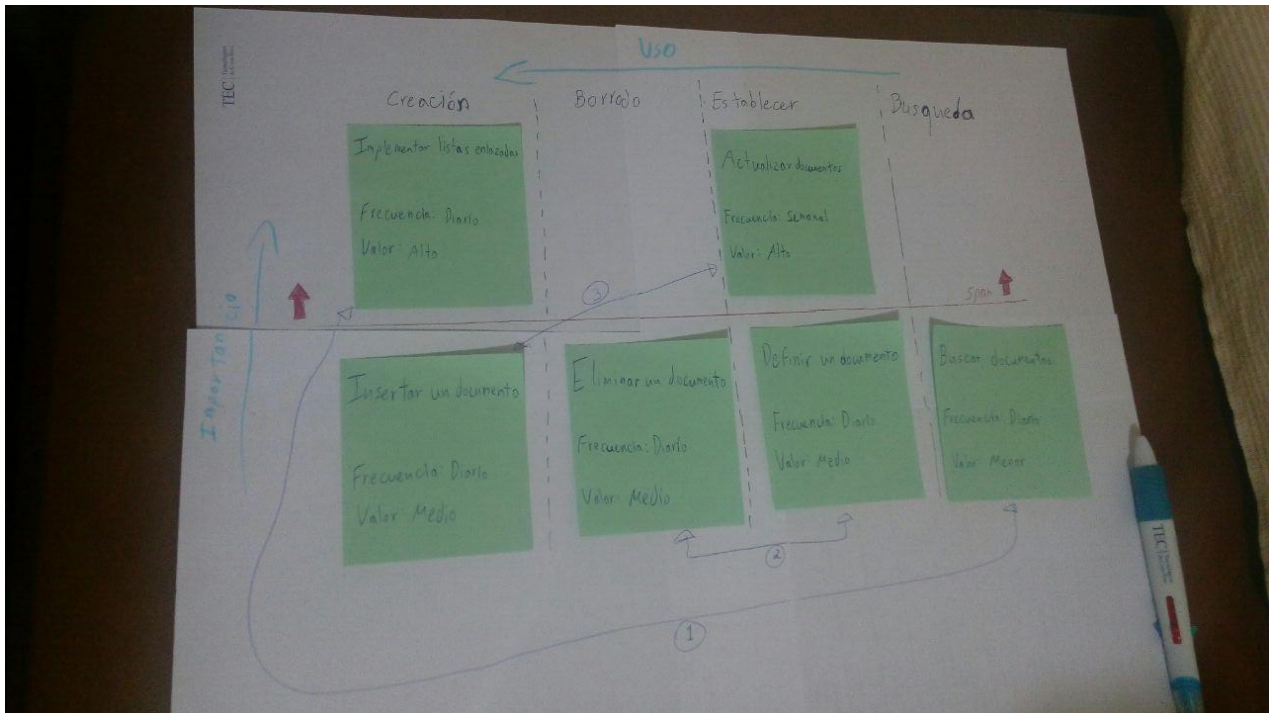
Descripción del Problema

LinkedDB es un motor de bases de datos noSQL. Permite definir, insertar, eliminar, actualizar y buscar documentos JSON.

Un documento JSON contiene objetos JSON que representan una entidad específica, por ejemplo un estudiante específico, una persona, un vehículo, entre otras.

LinkedDB se empaqueta como un único JAR. El usuario ejecuta dicho JAR y se presenta una interfaz gráfica desarrollada con Java FX. Dicha interfaz es similar a una herramienta de bases de datos, como DBVisualizer, DataGrip, MySQL Workbench y otros.

Planificación y administración del proyecto



Minimal system span:

Sistema operativo: Windows, Linux o Mac.

Java.

Apertura, información y guardado de datos mediante botón.

Bitácora de trabajo:

22/7/17, 5pm - 6pm

Lectura y análisis del Proyecto.

2/8/17, 8pm - 11pm

Investigación sobre git, netbeans, eclipse y otras herramientas necesarias en el proyecto.

3/8/17, 5pm – 10pm

Investigación relacionada con principales necesidades del proyecto relacionadas con listas.

5/8/17, 7pm – 9pm

Investigación sobre Json.

8/8/17, 9pm – 11pm

Investigación sobre JavaFX.

10/8/17, 8pm – 11pm

Investigación y pruebas sobre Json.

13/8/17, 7pm – 10pm

Pruebas base de JavaFX.

15/8/17, 8pm – 10pm

Pruebas base de listas.

16/8/17, 6pm – 11pm

Investigación relacionada con los métodos necesarios.

18/8/17, 8am – 10am

Base de historias de usuario.

20/8/17, 8pm – 11pm

Avance en Diagramas.

21/8/17, 9pm – 11pm

Investigación sobre interfaz.

23/8/17, 8pm – 24/8/17, 3am

Archivos Json, lectura y escritura.

25/8/17, 8pm – 11pm

Solución a problema con las listas.

28/8/17, 11pm - 29/8/17, 4am

Avance en interfaz.

30/8/17, 8pm – 31/8/17, 3am

Solución a problema en interfaz, implementación de algunos métodos para Json.

31/8/17, 8pm - 11pm

Investigación sobre funciones de JavaFX.

1/9/17, 7pm - 11pm

Investigación y pruebas en interfaz.

4/9/17, 9am - 7pm

Interfaz e investigación sobre Json.

5/9/17, 8am - 11am

Implementación de menú de click derecho en interfaz.

7/9/17, 7:30pm - 9pm

Relectura de proyecto, aclaración dudas.

8/9/17, 10pm – 9/9/17, 2am

Pruebas y solución a errores en listas.

11/9/17, 10am – 12/9/17, 2am

Documentación, interfaz y Json.

15/9/17, 8pm – 16/9/17, 6am

Mejoras en listas, avance en Json y solución a problemas en interfaz.

16/9/17, 12pm – 9pm

Avances en métodos de menú de la interfaz.

17/9/17, 5am – 18/9/17, 3am

Documentación, avances en botón commit.

18/9/17, 10am – 3pm

Avances en la lectura de archivos.

18/9/17, 6pm – 11:30pm

Avances en documentación y recopilación de información.

19/9/17, 6am – 12pm

Mejoramiento de uso de Json.

19/9/17, 4pm – 20/9/17, 3am

Avances en el uso de archivos y creación de nodos con nombres específicos.

20/9/17, 5am – 10am

Detallado de documentación, arreglo de problema con la tabla.

Diseño

Diagrama de arquitectura de la solución: (asumiendo que tuviera también Aplicación)

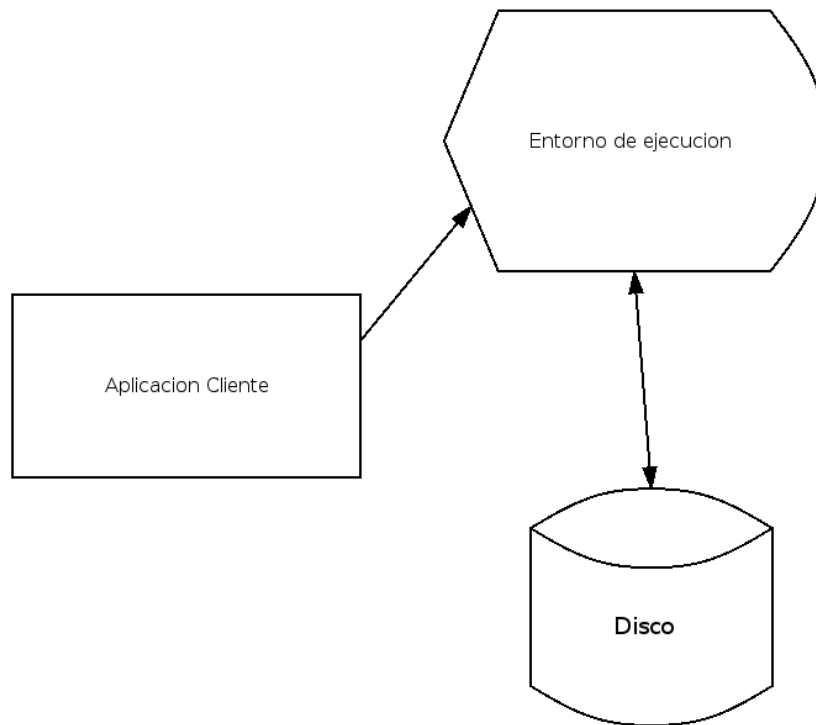


Diagrama de componentes:

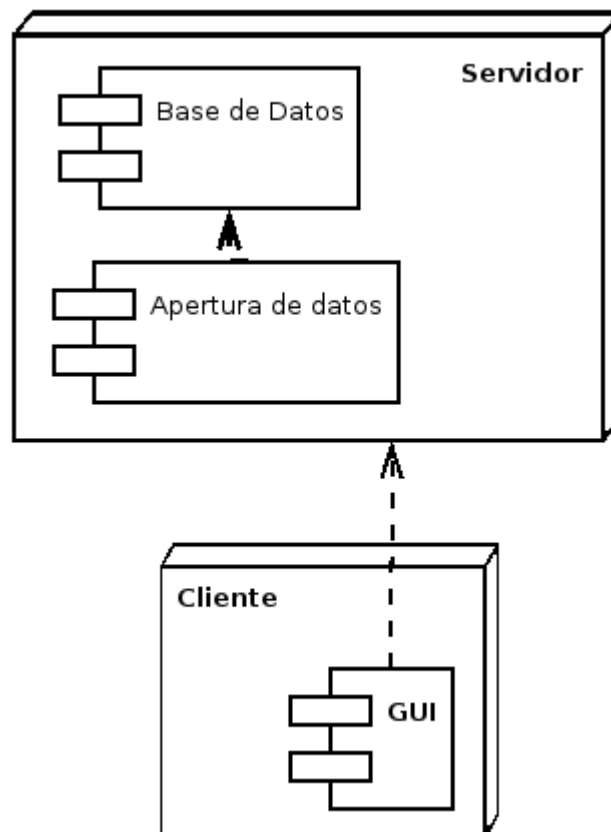


Diagrama de Secuencia:

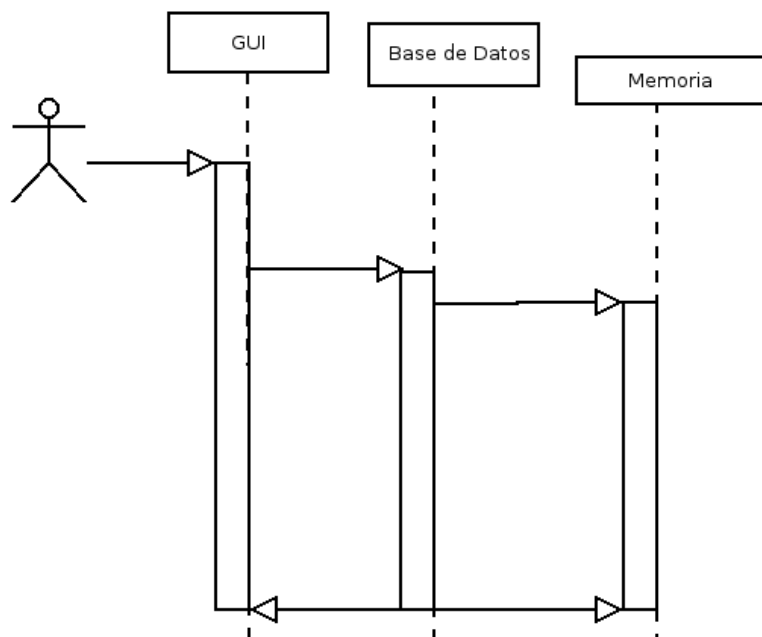
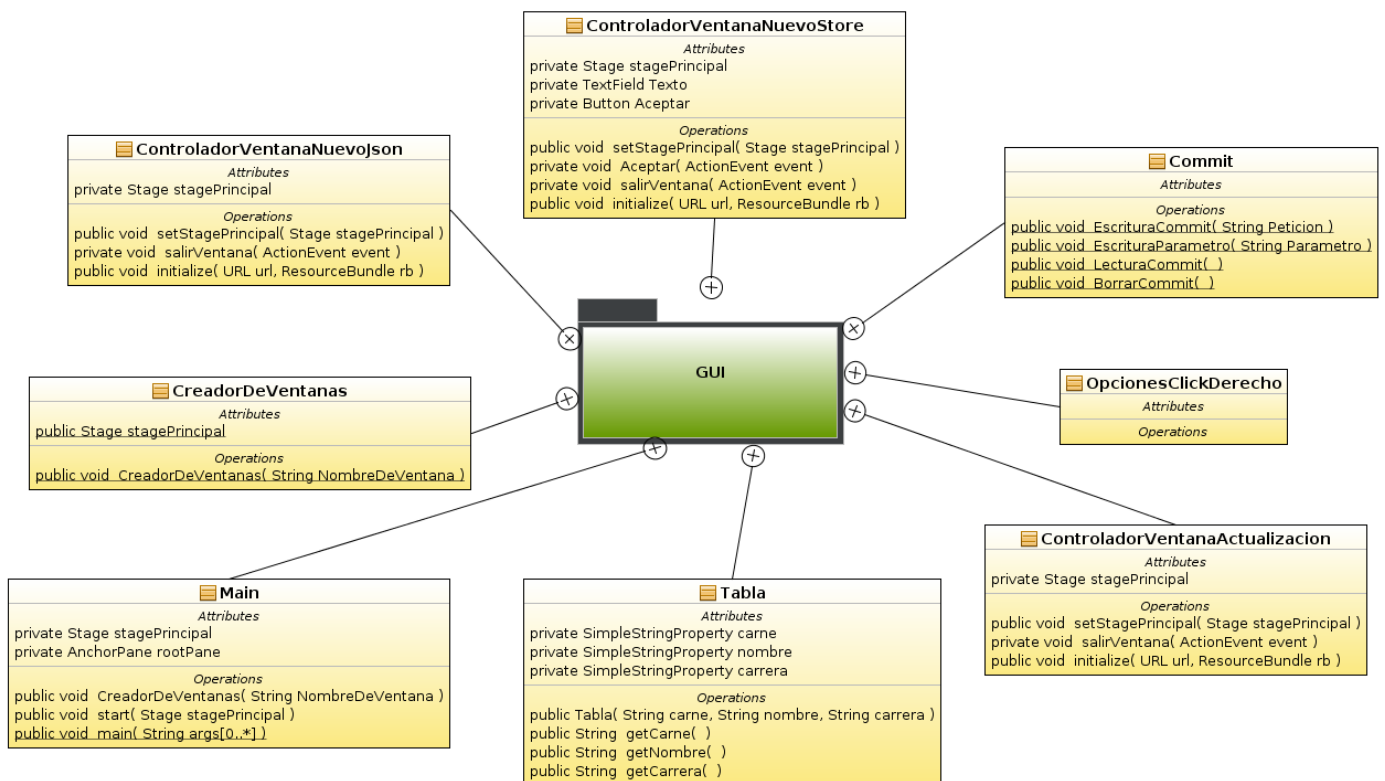
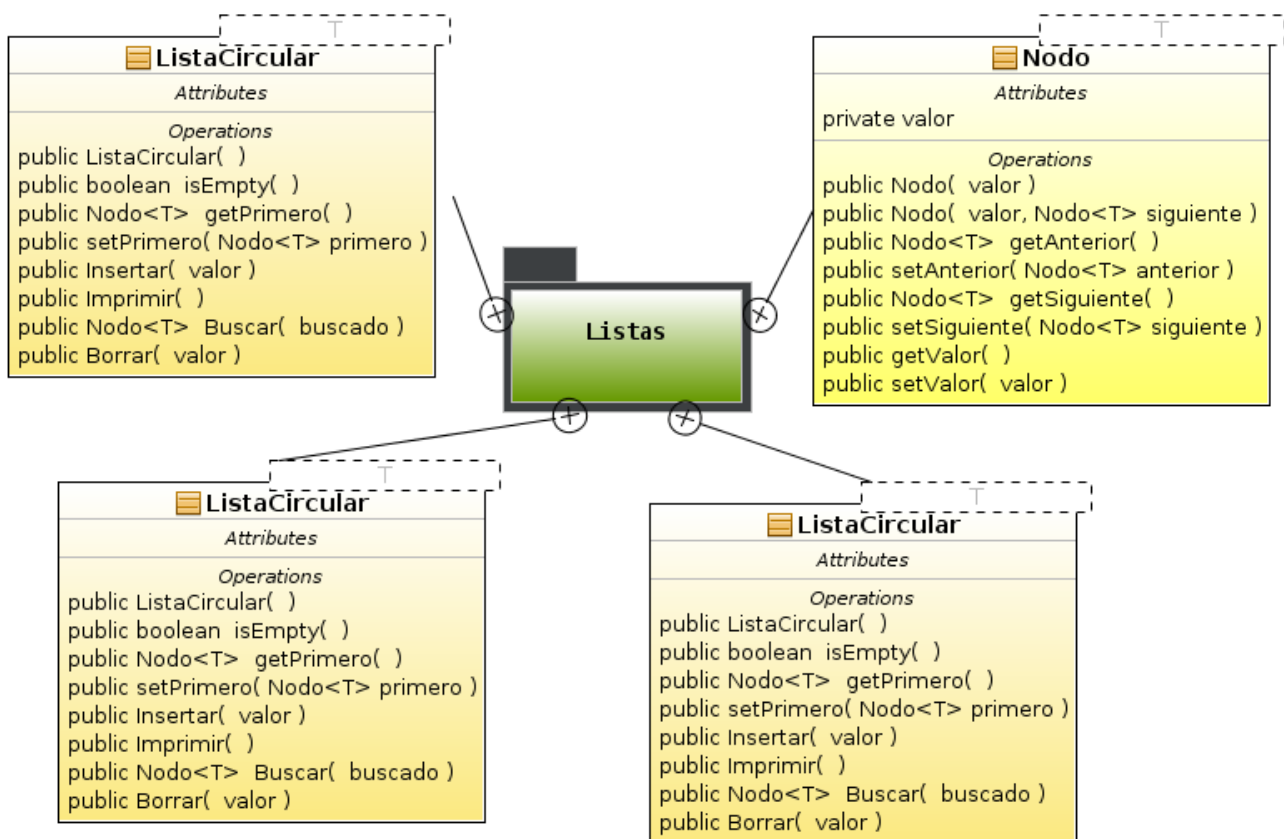
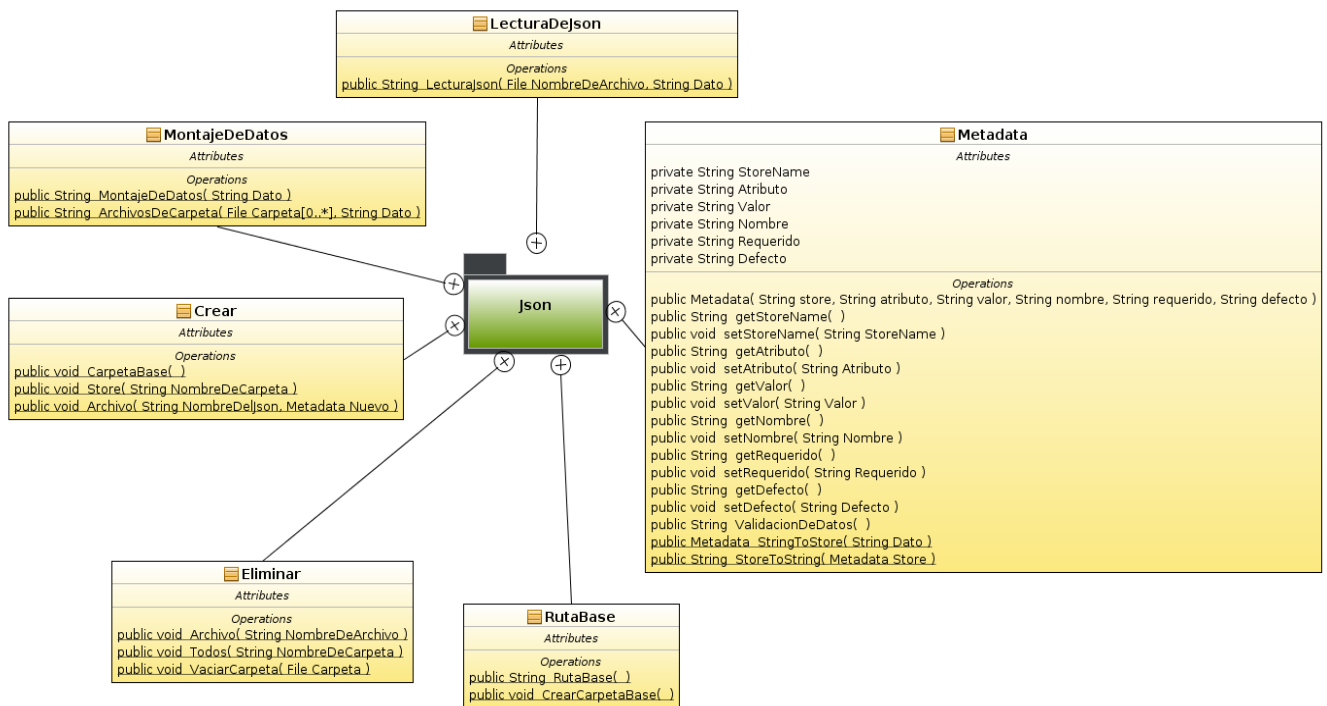


Diagrama de Clases:





Implementación:

Descripción de las bibliotecas utilizadas:

JavaFX Scene Builder: Utilizada para el desarrollo de una interfaz gráfica de manera mas cómoda y versátil.

Gson: Utilizada para la importación y exportación de información a JSON.

Estructuras y algoritmos:

Listas: Creadas a partir de nodos y usadas para la recopilación de la información y el manejo de la misma.

Crear: Generación de carpetas, documentos JSON y base de los Store a partir de indicaciones guardadas en un .txt activado al dar al boton.

Eliminar: Borrado de carpetas, documentos JSON y vaciado de carpetas según su respectivo llamado.

Lectura De JSON: Lectura de datos de cada archivo JSON mediante Gson y las propiedades del mismo que permiten el sacar un dato especifico según su formato de escritura pre establecido.

Commit: Creación y borrado de documento “commit.txt” que guarda y aplica cada función solicitada.

Creador De Ventanas: Generación de Ventanas Hijas.

Ruta Base: Generación de Carpeta Base para guardado de archivos (en Documentos).

Montaje De Datos: Generación de tabla y árbol con información de los JSON realizada mediante lectura de archivos .txt y datos en las listas esperando su respectivo uso.

Validación: realizada mediante la comparación de datos pre establecidos.

Problemas encontrados:

Guardado y cargado de datos... Se busco intentar de muchas maneras, desde librerías para java, hasta métodos por videos y otras entidades (paginas), pero no se logró ya que surgía algún problema en los métodos o los resultados de los mismos.

Creación especifica de datos... Problemas con uso y falta de conocimiento en la elaboración de metada para cada archivo u Store, donde debe seleccionarse y darse información clave para la determinación de la siguiente información a proporcionar para la realización de los JSON, luego de muchos intentos por distintos caminos (videos y lectura de paginas y libros), no se lo logro realizar con éxito.

Actualización de datos de forma colectiva... Dado que debe hacerse por llaves y selección de información conectada a la tabla, no logré conectar los datos de los mismo para llevar a cabo cambios en ella por medio de selección.

Conclusiones y Recomendaciones del proyecto.

Para la realización de Json y lo que esto conlleva se recomienda el uso de la libreria GSON, ya que facilita mucho la escritura y lectura de los mismos.

Java es versátil, cómodo y no muy difícil de entender, además, tiene una gran comunidad de apoyo y aportes.

Siempre se deben utilizar patrones de diseño para simplificar el trabajo y evitar el gasto de tiempo innecesario.

La programación orientada a objetos en Java es cómoda y natural, ya que nuestro pensamiento se da en objetos.

Se logró observar que existen muchas maneras de crear bases de datos y de manejar las mismas, pero que deben tener ciertos fundamentos base para su funcionamiento

básico, tales como crear nuevos archivos en memoria, llenar los mismos y poder hacer una lectura de lo escrito en ellos.

Bibliografía consultada en todo el proyecto:

Oracle Legal, Copyright © 1994-2014, Oracle and/or its affiliates. All rights reserved. http://docs.oracle.com/javafx/2/ui_controls/tree-view.htm

Csharls Code, Octubre 18, 2013, JavaFX en Netbeans.:
<https://charlescode.wordpress.com/2013/10/18/multiples-ventanas-en-javafx/>

OpenAalfa, © 2013 Blog OpenAlfa.: <https://blog.openalfa.com/como-leer-y-escribir-ficheros-json-en-java>

Víctor Cuervo, 2001-2017 © Línea de Código.:
<http://lineadecodigo.com/java/borrar-un-fichero-con-java/>

Rose India, ©2001 - 2017 Rose India Technologies Pvt. Ltd. All Rights are Reserved.:
<https://www.roseindia.net/java/example/java/swing/ShowMessageDialog.shtml>

Juan Carlos Lafuente – Eworks, © 2013 Juan Lafuente . All Rights Reserved.:
<https://jclafuente.blogspot.com/2013/01/verificar-y-crear-directorio-con-java.html>

Stack Overflow en español, © 2017 Stack Exchange Inc;
<https://es.stackoverflow.com/questions/39503/borrar-fichero-en-java>

Java™ Platform Standard Ed. 7, Copyright © 1993, 2016, Oracle and/or its affiliates.:
<https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html>

Stack Overflow, © 2017 Stack Exchange Inc;
<https://stackoverflow.com/questions/40707131/how-to-enable-or-disable-the-buttons-in-javafx-form>

Chuwiki chuidiang, 16 jul 2015 a las 20:17.:
http://chuwiki.chuidiang.org/index.php?title=Lectura_de_teclado_en_java#Ejemplo_de_lectura_de_un_fichero_con_Scanner