Instituto Tecnológico de Costa Rica

Algoritmos y Estructuras de Datos 1

Proyecto #1

Estudiante:

Esteban Campos Granados

Carnet:

2017097066

Profesor:

Antonio González Torres

Grupo 3

20 de Septiembre del 2017

Indice

Introducción

En el presente proyecto se realizó el diseño e implementación motor de bases de datos sencillo basado en listas enlazadas.

Este motor de bases de datos es noSQL, y 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.

Éste se empaqueta como un único JAR. El usuario ejecuta dicho JAR y se presenta una interfaz gráfica desarrollada con Java FX.

El trabajo fue realizado en el lenguaje de programación Java, requeriendo de conocimiento de Programación Orientada a Objetos (POO) y conceptos básicos de bases de datos.

Desarrollo

Descripción del Problema

Crear LinkedDB, un motor de bases de datos noSQL que permite definir, insertar, eliminar, actualizar y buscar documentos JSON. Un documento JSON contiene objetos JSON que representan una entidad específica.

Éste 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, entre otros.

Cuando se crea un documento, se crea un archivo JSON con el nombre del documento dentro de la carpeta correspondiente al store seleccionado.

La interfaz gráfica, tiene un botón "Commit" que escribe en disco los cambios realizados. Es importante notar, que cuando la interfaz gráfica carga, se deben cargar los archivos a memoria. Si la aplicación se cierra sin hacer commit, los cambios se descartan.

Planificación y Administración del Proyecto

Historias de Usuario.

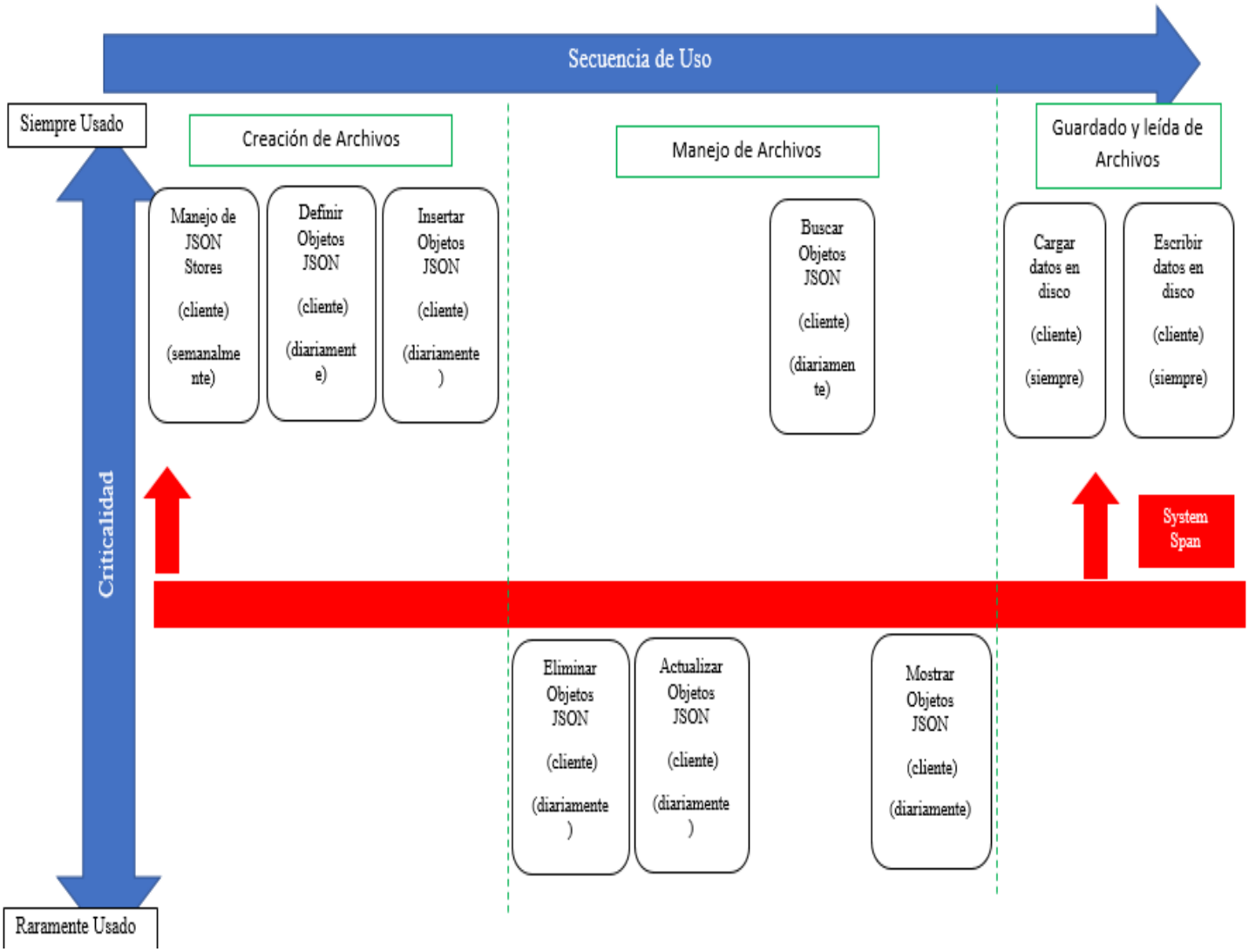
1. Como estudiante quiero crear documentos para poder almacenar información de mis proyectos de computación
2. Como bibliotecario quiero manejar documentos para tener control de la información que éstos contengan y sus caracteristicas
3. Como profesor quiero crear carpetas para organizar los archivos que creo.
4. Como usuario quiero leer y escribir documentos JSON en memoria para poder recurrir a su información cuando la necesite

Lista de Features (Tareas).

1. Manejo de JSON Stores (cliente) (Semanalmente)
2. Definir Objetos JSON (cliente) (diariamente)
3. Insertar Objetos JSON (cliente) (diariamente)
4. Eliminar Objetos JSON (cliente) (diariamente)
5. Actualizar Objetos JSON (cliente) (diariamente)
6. Buscar Objetos JSON (cliente) (diariamente)
7. Mostrar Objetos JSON (cliente) (diariamente)

8. Escribir datos en disco (cliente) (siempre)
9. Cargar datos en disco (cliente) (siempre)

Distribución de features



Bítacora.

10/8/2017.

Creacion de las listas

Tiempo: 2 horas

5/9/2017.

Creacion del repositorio en git, modificaciones de listas simples e instalacion de javafx.

Tiempo: 2 horas

6/9/2017.

Practica con Javafx, instalacion y practica con json-simple, nodo doble y lista doble.

Tiempo: 5 horas

7/9/2017.

Avance con la implementacion de JSON, Lista Doble Circular, arreglos en listas y nodos.

Tiempo: 3.5 horas

8/9/2017.

Creacion y lectura de JSON

Tiempo: 3 horas

11/9/2017.

Creacion de la lista simple implementada con la creacion y manejo de JSON

Tiempo: 2 horas

12/9/2017.

Modificaciones en clase simple

Tiempo: 0.5 hora

13/9/2017.

Creacion de json stores y planteamiento de los documentos.

Tiempo: 3 horas

14/9/2017.

Manejo de los tipos de JSON.

Tiempo: 4 horas

15/9/2017.

Manejo de los documentos JSON

Tiempo: 1 hora

16/9/2017.

Mas comandos de manejo de datos

Tiempo: 3 horas

17/9/2017.

Lectura JSON

Tiempo: 4 horas

18/9/2017.

Manejo de Reads

Tiempo: 4 horas

18/9/2017.

Documentación e Implementacion de la interfaz y codigo

Tiempo: 6 horas

Diseño

Diagrama de arquitectura de la solución.

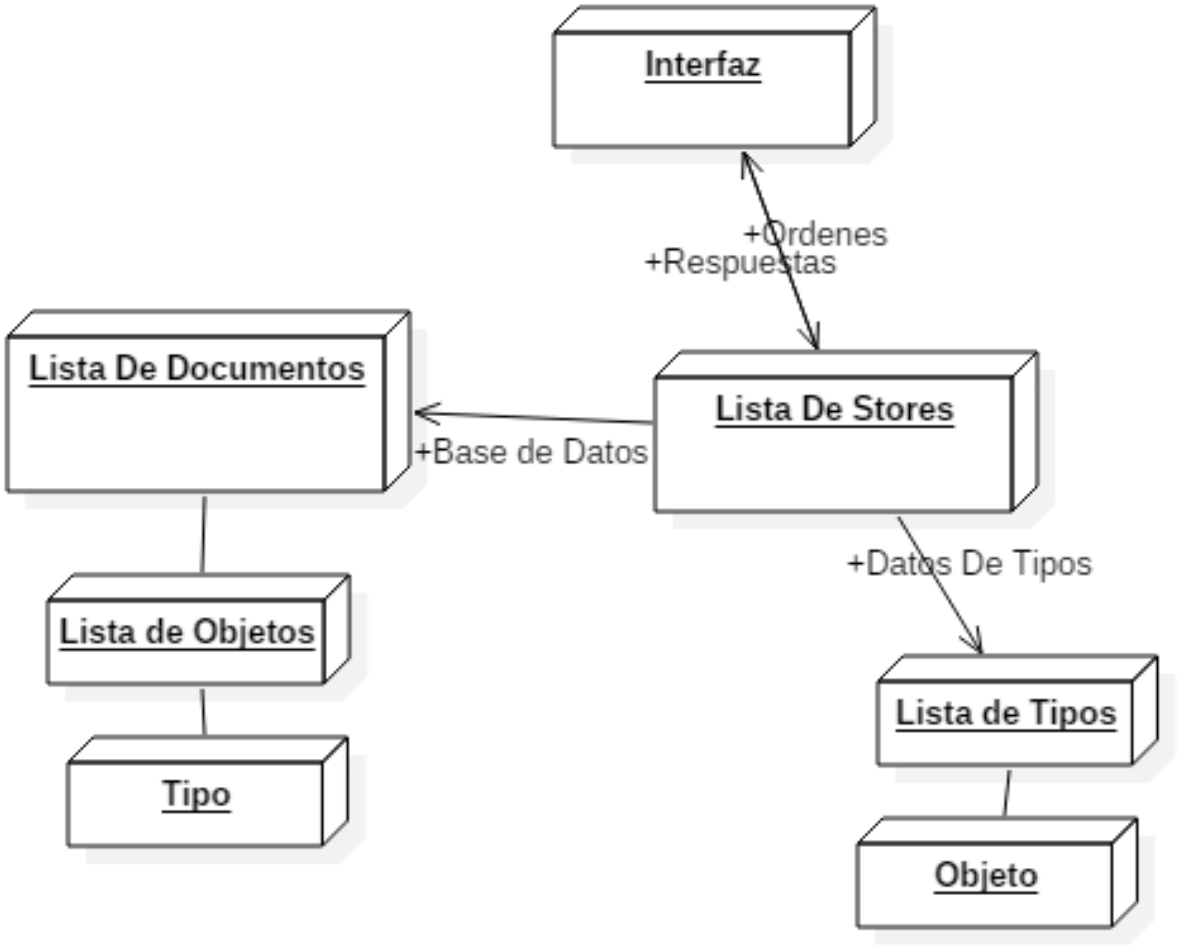
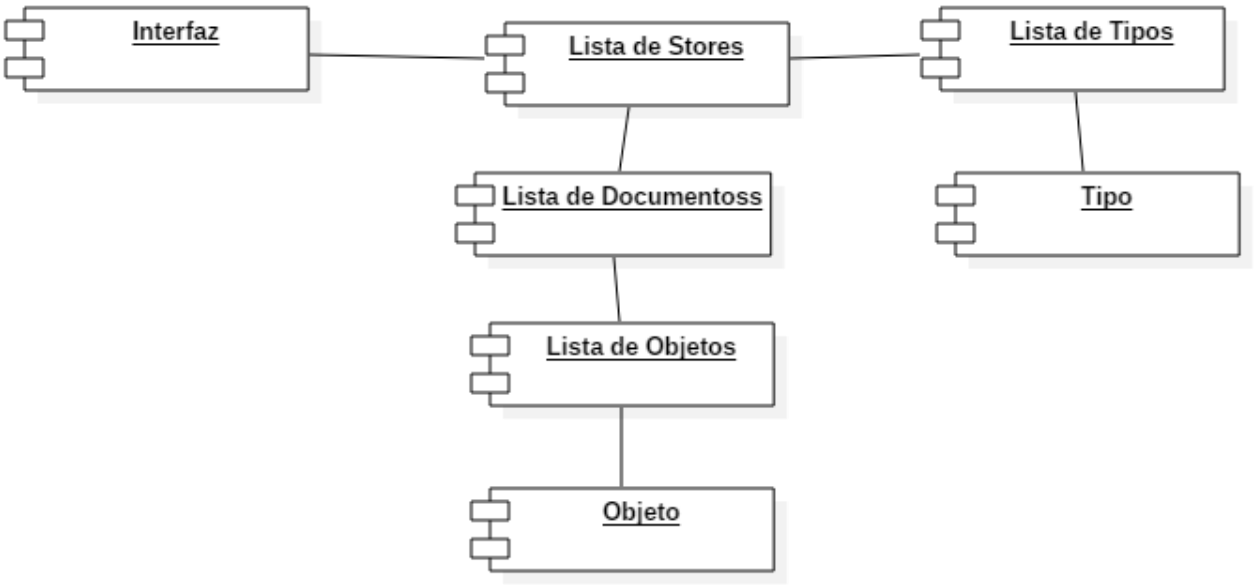
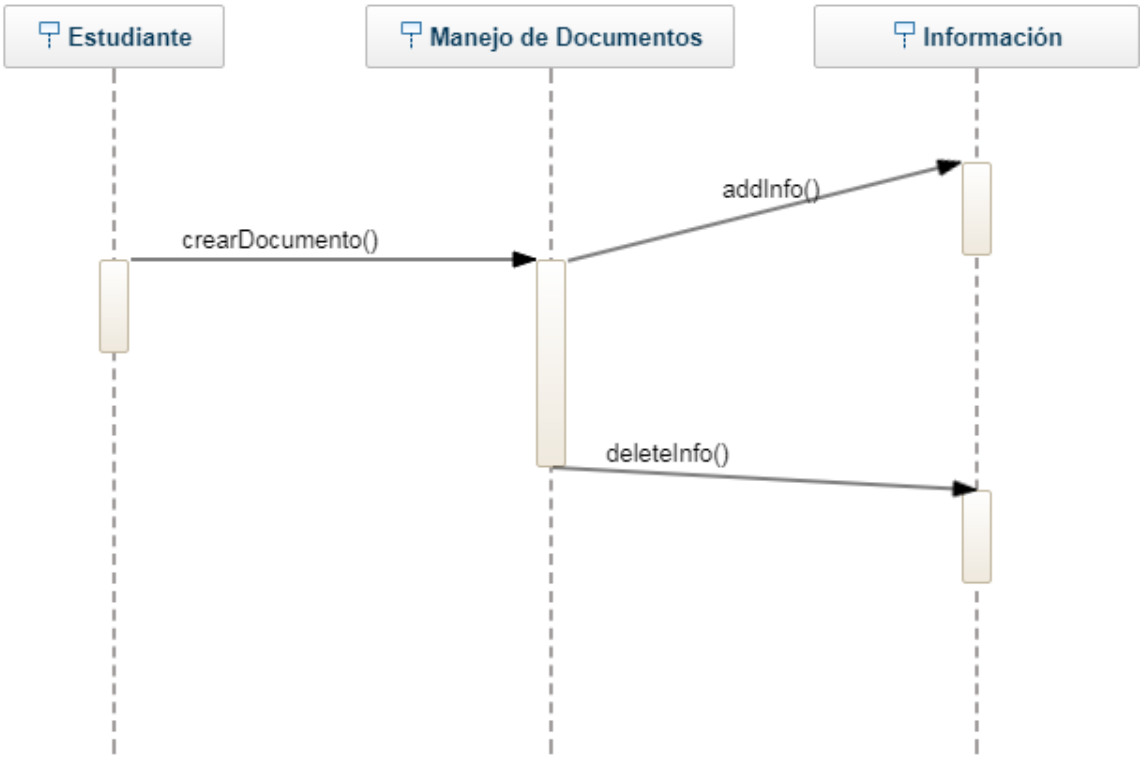
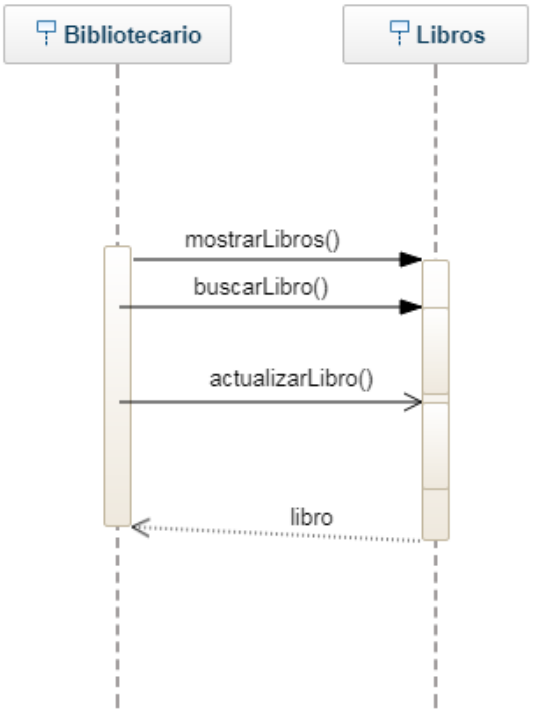


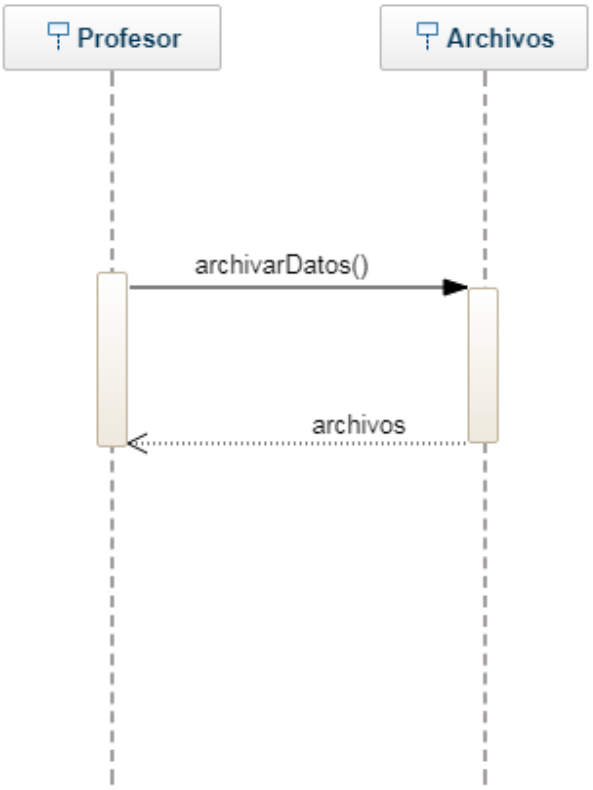
Diagrama de componentes.



Diagramas de Secuencia.






Profesor
Archivos
archivarDatos()
archivos

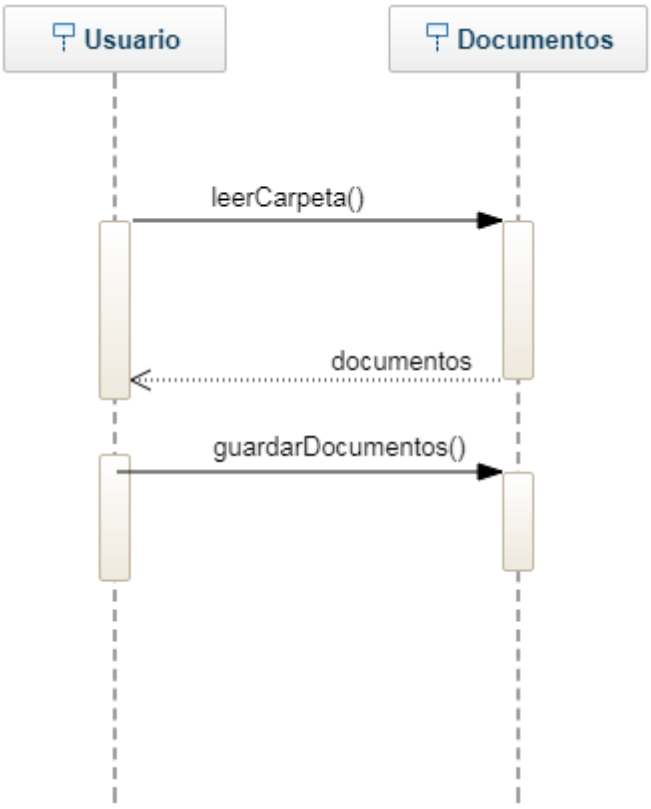
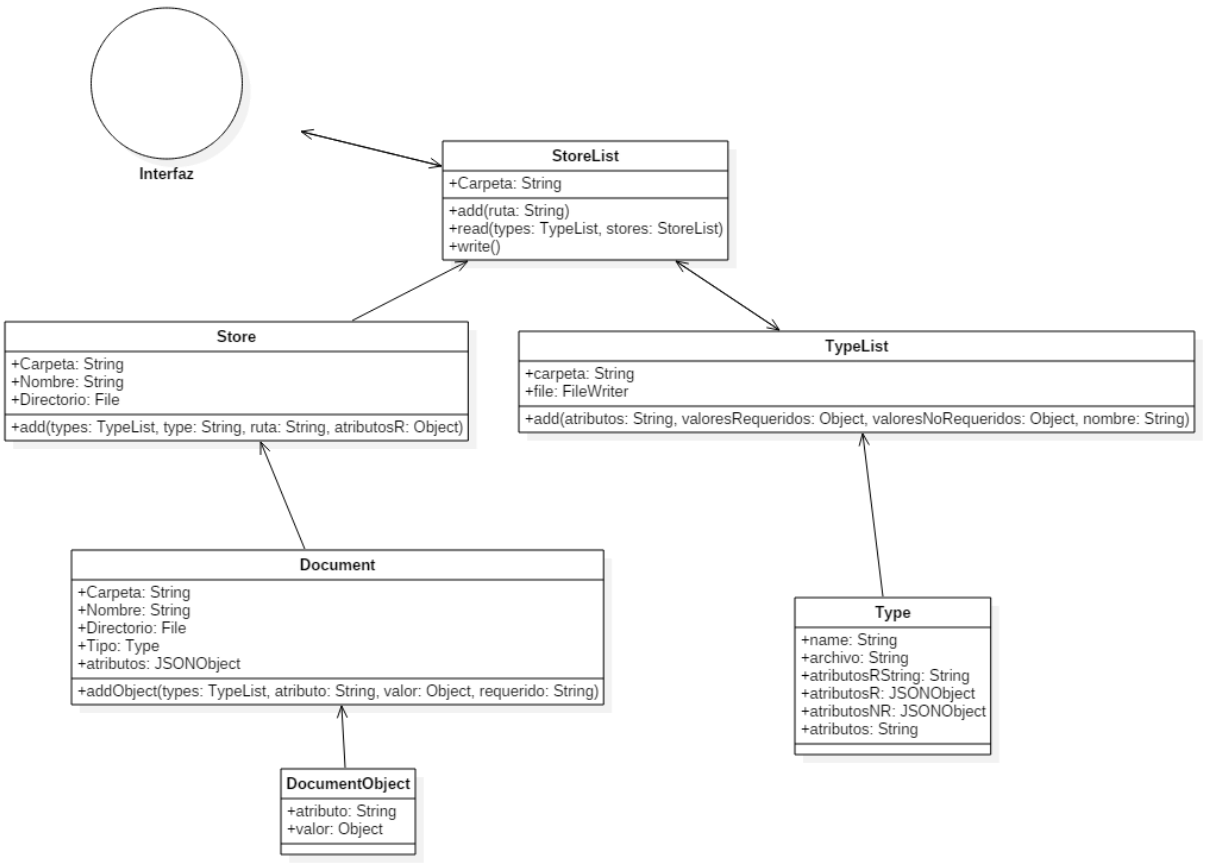

Usuario
Documentos
leerCarpeta()
documentos
guardarDocumentos()

Diagrama de Clases



Implementación

Bibliotecas utilizadas.

Json-simple.

Biblioteca de lectura y manejo de archivos JSON. Se usó para la lectura de éstos y para la recopilación de información en JSONObjects para escribir JSON.

Estructuras de Datos utilizadas.

Lista Simple.

Consiste en una secuencia de nodos, en los que se guardan campos de datos y una referencia al nodo anterior o posterior.

Lista Doble.

Consiste en una secuencia de nodos, en los que se guardan campos de datos y dos referencias: una al nodo anterior y otra al posterior.

Lista Doble Circular.

Mantiene el mismo concepto que la Lista Doble, con la unica diferencia de que el ultimo elemento de la lista posee una referencia al primer elemento; y viceversa

Descripción de algoritmos utilizados.

Para los algoritmos de leida, se usaron ciclos while de busqueda, donde se obtiene los archivos de la carpeta y se agregan como nodos a una lista

Para los algoritmos de escribir en memoria, se recorre las listas escribiendo cada informacion de los nodos en archivos de las carpetas correspondientes

Para agregar, eliminar, buscar y actualizar documentos se recorre las listas en busca de documentos con características similares.

Problemas.

Tiempo: Hubo una muy mala administración del tiempo, lo que provocó la dificultad del trabajo al 100%.

Tipos de JSON: Se intento de manipular los tipos de datos de distintas formas, todas fallidas. La única manera encontrada fue hacer una lista simple de tipos de lista, la cual perfecciona correctamente excepto el read(). No hay manera de como leer los tipos, por lo cual, antes de leer todos los datos de la interfaz, se deben especificar los tipos.

Conclusion

Se concluye la importancia del uso de JSON como almacenamiento de datos por su utilidad y simplicidad.

Se promovió un cambio diferente de trabajo, donde se enfocaba mas al manejo de los datos que al simple hecho de programar.

Enseña de la necesidad de un buen planteamiento y administracion de los proyectos, como el buen analisis de requerimiento y del tiempo necesitado para realizar el proyecto.

Bibliografía

http://www.admfactory.com/how-to-read-and-write-json-with-json-simple/

https://stackoverflow.com/questions/29552921/removing-a-json-object-from-a-json-file-in-java

https://examples.javacodegeeks.com/core-java/json/java-json-parser-example/

http://normasapa.net/2017-edicion-6/

http://alex-public-doc.s3.amazonaws.com/json_simple-1.1/index.html