UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERIA
DEPARTAMENTO DE COMPUTACIÓN
75.06 - ORGANIZACIÓN DE DATOS
Primer Cuatrimestre - 2011

Booquerio

Tutor: Maximiliano Stibel
Grupo: IV
Correo: datos201101@yahoogroups.com
Integrantes:

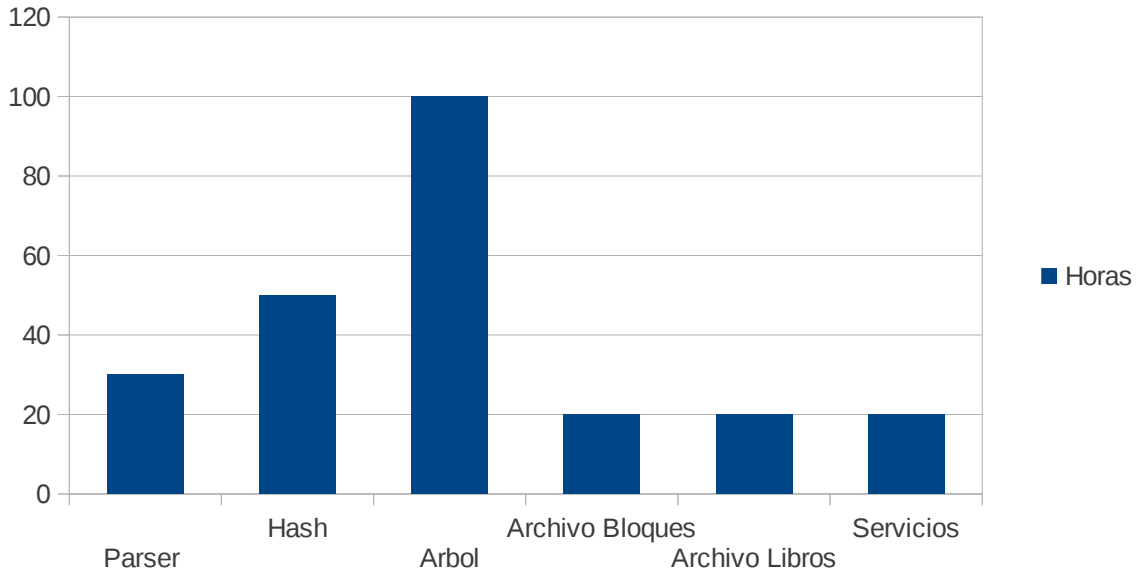
- Ciruzzi Martin - 90983
- Cruz Rodolfo - 89510
- Marasco Hernan - 89333
- Notari Pablo - 88548

Funcionalidad

La funcionalidad requerida para esta entrega se detalla en “tp_2011_1C.pdf”.

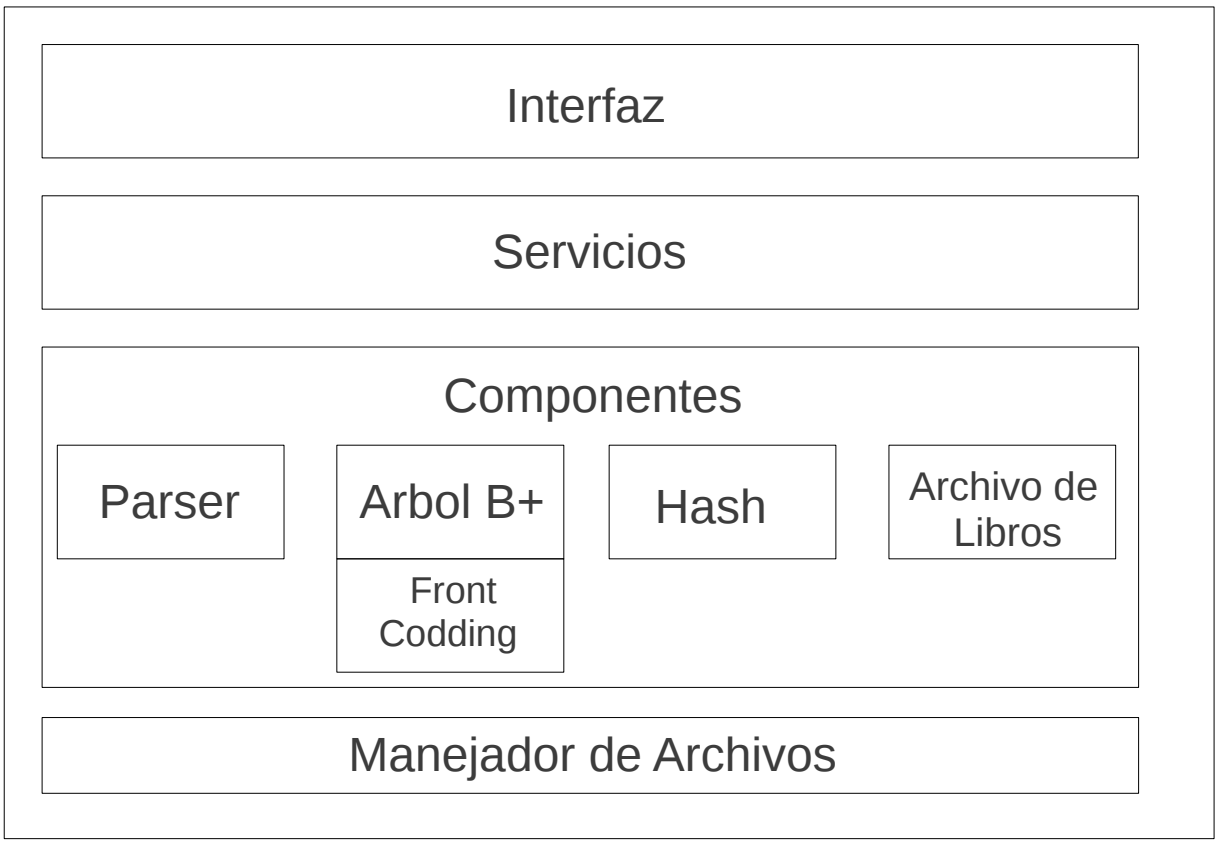
Plan

Componente	**Horas**	**Integrante**
Parser	30	Pablo
Hash	50	Hernan
Arbol	100	Martin – Rodolfo
Archivo Bloques	20	Martin – Rodolfo
Archivo Libros	20	Hernan
Servicios	20	Pablo
Total	240	
Total por integrante	60	



Diseño de la aplicación

Se dividió el diseño en cuatro capas según el siguiente diagrama:

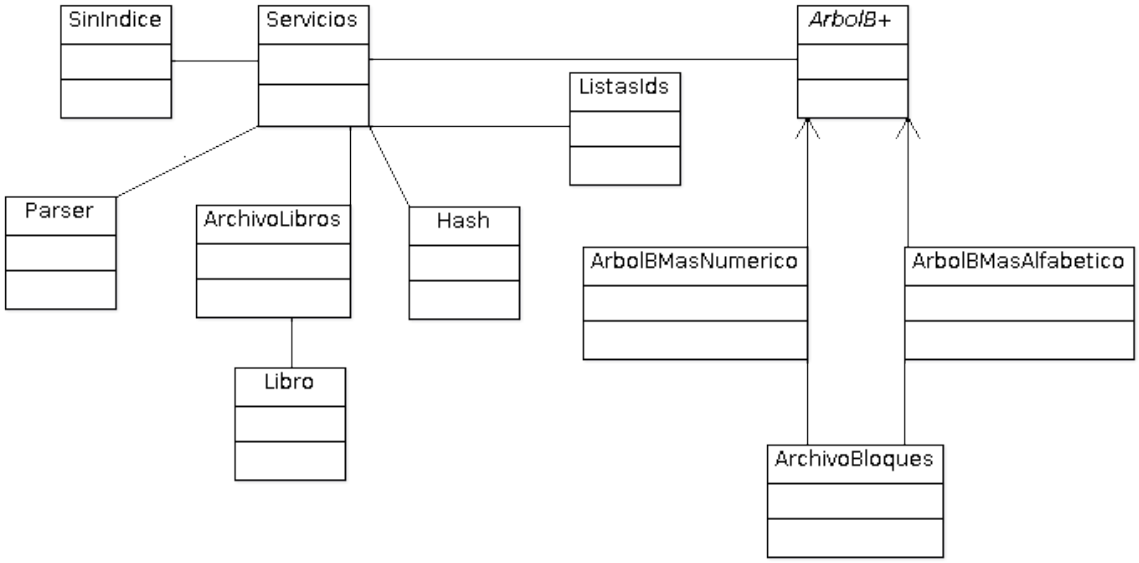


Interfaz

Interacción entre el usuario y la capa de servicios. Tiene como funcionalidad el procesamiento de comandos y la llamada correspondiente al servicio que resuelve la petición.

Servicios

Interacción entre la interfaz y los distintos componentes de la aplicación, conoce el funcionamiento del árbol, hash, parser, listas de libros sin procesar y archivos de libros.



Arbol B+

Para la implementación del árbol se opto por construir una clase abstracta con métodos comunes a cualquier tipo de árbol B+ y dos clases que heredan, una para árboles con clave numerica como el árbol primario en el cual las claves son identificadores(Ids) y uno para clave de texto(alfabeticas) para los indices secundarios en los cuales las claves son palabras, titulos, editoriales y autores(todos strings).

En si lo que se intenta es definir una interfaz de arbol comun para todo tipo de registro. Para la implementacion de una arbol B+ en particular se deben "indicar" por cual de todos los campos del registro se indexa, es decir cual es la clave. Para ello se deben requiere definir metodos para el manejo de claves, como puede ser: forma en que se comparan, o creacion de un registro con una clave dada(el arbol generico no sabe que atributo es la clave).

Los arboles poseen como atributo un "Archivo de Bloques" quien es responsable de toda la interfaz del arbol con archivos. Siendo este el encargado de indicarle al arbol casos en los cuales persistir un bloque resulte no posible(bloques que se encuentren por encima/debajo del tamaño de trabajo).
Cabe destacar que para la lectura/persistencia se trabaja siempre nivel de bloque/nodo.

Para el front codding de los árboles secundarios(claves alfabeticas) previo a cualquier escritura o posterior a cualquier lectura, se pasa el bloque de trabajo por una funcion traductora que comprime/descomprime el bloque indicado acorde al algoritmo de front codding.

En cuanto a los casos de ramificacion del arbol:

El split de bloques se da en el caso que al querer grabar el bloque la capa de manejo de archivos informe que el nodo no entra en el bloque que se indica. Se procede entonces a separar este bloque en 2 para su posterior escritura.

Ante una situacion opuesta, es decir, se intenta persistir un nodo en underflow(con menor capacidad a la minima permitida) se procede a intentar balancear dicho nodo con el hermano derecho, en caso de no existir, con el izquierdo. En caso de no poder balancear con dicho hermano(tambien se encuentre sobre el limite de underflow) se procede a la fusion/concatenacion de los mismos.

Hash Extensible

Para la implementación del hash se utilizó una función que acumula la conversión de los caracteres a ASCII multiplicandolos por 37 para luego realizar el resto de la división por el tamaño de la tabla.

Ejemplo

Tamaño de la tabla : 8

h(palindromo) = 1
h(stibel) = 7

Parser

En el parseo se levantan los datos del libro y además se realiza el procesamiento de palabras que se guardan en el libro como una lista separada por comas para su posterior procesamiento.
Por la caracteristica de los archivos de libros se hizo compleja la recuperación de las editoriales por lo que se determinó tener un archivo de editoriales (editoriales.csv) en el cual figura un listado basico de editoriales que se asignan al libro dependiendo de

la primer letra del nombre del autor, en el archivo de configuración config.propiedades se puede parametrizar el factor de asignación de editoriales de modo que una editorial quede asignada a libros de distintos aurtores.

Ejemplo:
factor_editoriales=1;

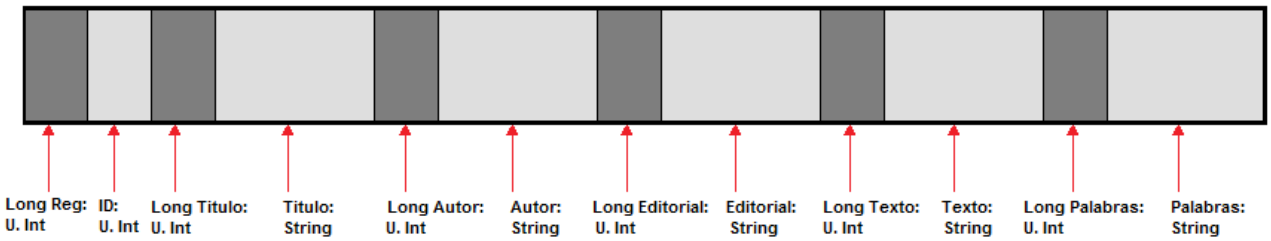
Alejando Pérez Visa → 1er editorial de la lista
Christopher Paolini → 3da editorial de la lista
Dan Brown → 4ta editorial de la lista

factor_editoriales=2;

Alejando Pérez Visa → 1er editorial de la lista
Christopher Paolini → 3da editorial de la lista
Dan Brown → 3ta editorial de la lista

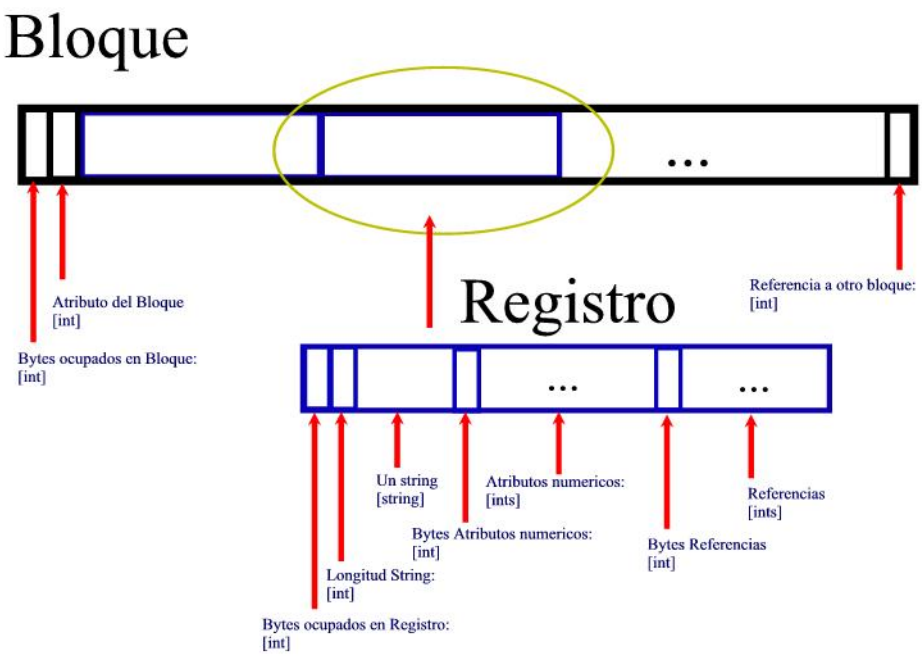
Archivos de Registros Variables (Archivo de Libros)

Estructura de datos:



Manejador de Archivos

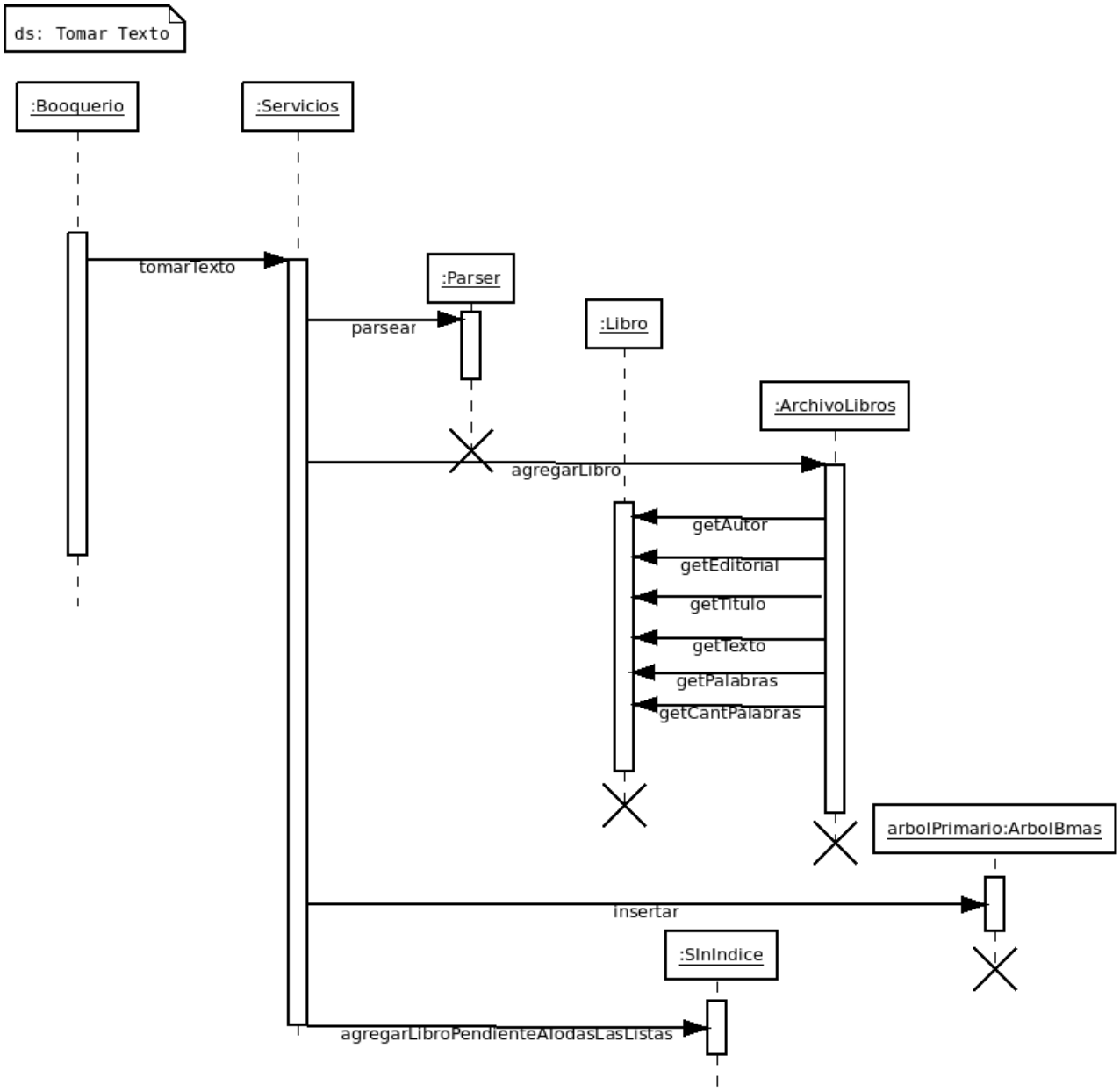
Estructura de datos:



Diagramas de Secuencia

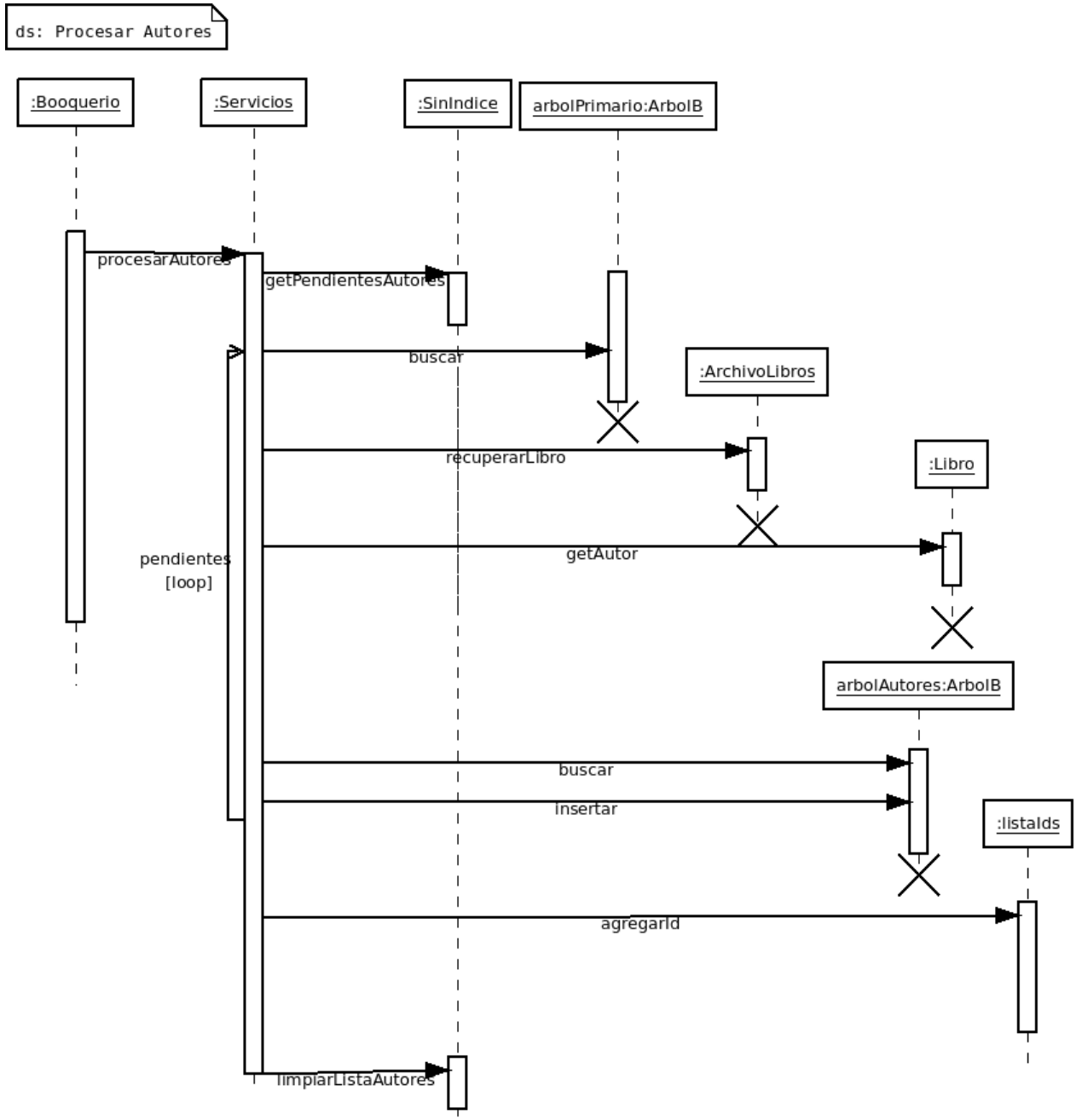
Agregar texto:

El usuario agrega un libro, se parsea, se agrega a archivo de registros variables y se agrega a las cuatro listas de libros sin procesar (autores, editoriales, palabras, titulos).

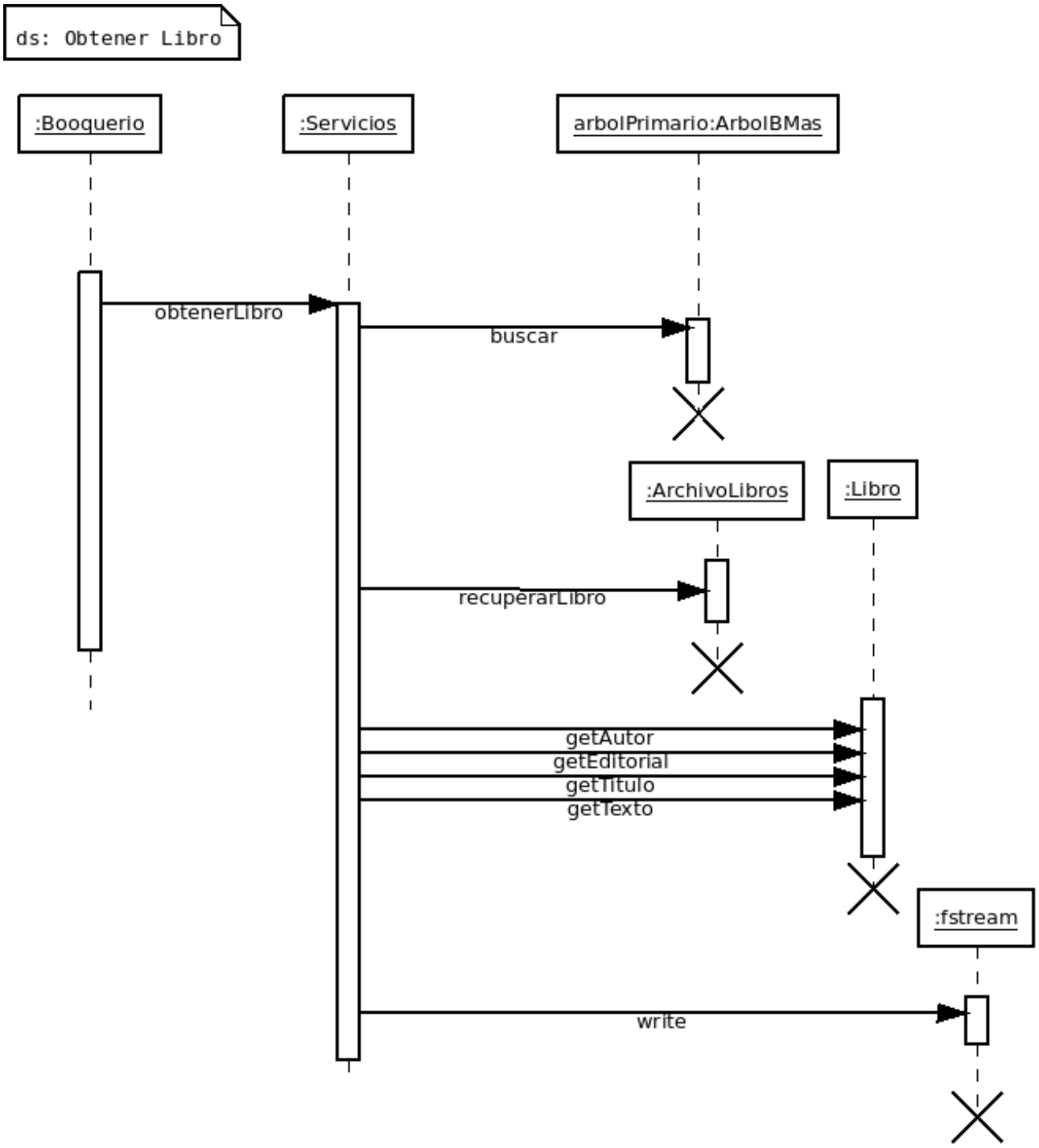

ds: Tomar Texto
:Booquerio
:Servicios
tomarTexto
:Parser
parsear
:Libro
:ArchivoLibros
agregarLibro
getAutor
getEditorial
getTitulo
getTexto
getPalabras
getCantPalabras
arbolPrimario:ArbolBmas
insertar
:SInIndice
agregarLibroPendienteAlodasLasListas

Procesar autores:

Se obtiene la lista de libros pendientes de procesamiento y por cada libro se llama al indice de autores consultando si el autor ya esta en el indice, en caso positivo se inserta el nuevo id en la lista de ese autor, en caso negativo se crea una nueva lista y se inserta ese autor en el arbol referenciando a la lista creada.

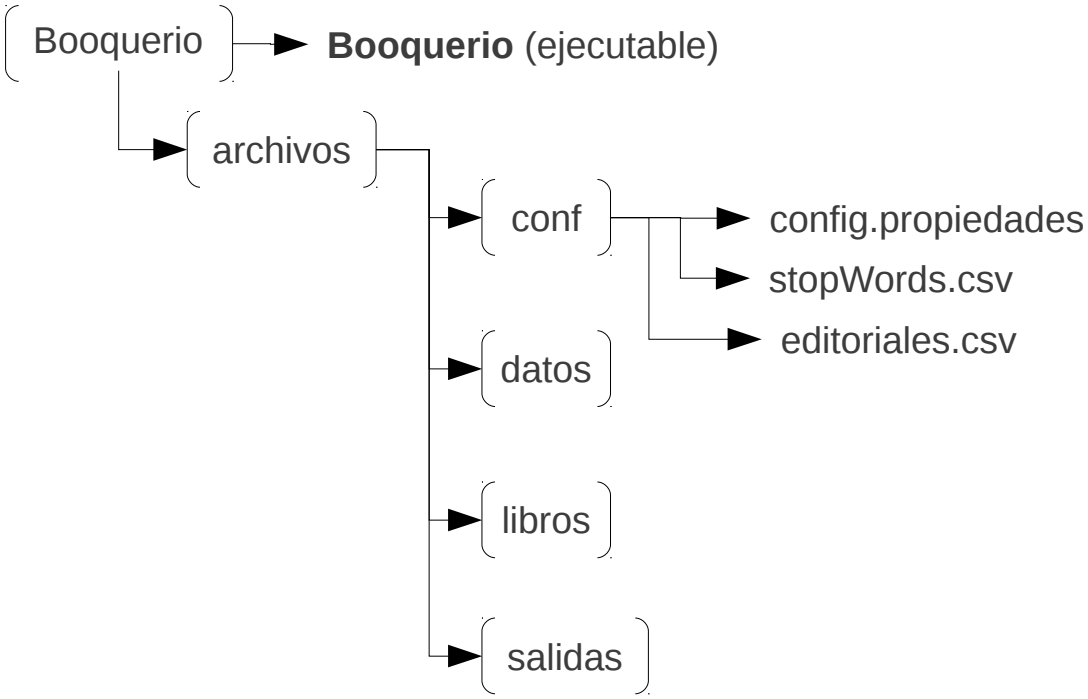


Obtener libro:
Se recupera el offset del libro del indice primario, se recupera el libro con ese offset y se recrea en un nuevo archivo.



Manual de Usuario

Estructura de directorios:



-conf: es el directorio que contiene la información de la configuración basica de la aplicacón. Además contiene los siguientes archivos:

-conf.propiedades: contiene los path de los directorios de trabajo y las parametrizaciones de los archivos de referencias.

-stopWords.csv: contiene las palabras que se ignoran al momento de indexar las palabras.

-editoriales.csv: contine la lista de editoriales posibles. (ver *"Estrategia del Parser"*).

-datos: es el directorio donde se crean y administran todos los archivos de datos de la aplicación.

-libros: es el directorio por defecto desde donde se procesan los libros si se utiliza el path relativo(ver procesamiento de comandos *"Tomar Texto"*).

-salidas: es el directorio donde se almacenan las salidas de programa (ver procesamiento de comandos *"Ver estructuras"*)

Funcionalidad

En todos los casos se supone que la llamada al programa se realiza desde una terminal con el comando: Booquerio -operacion [parametro1] [parametro2]

Procesamiento de comandos

Tomar Texto: ./Booquerio -i "libro"

Toma el libro y lo deja pendiente de procesamiento para los distintos indices, le asigna un identificador único incremental.

Procesar Editorial: ./Booquerio -e

Dispara el proceso de indexación por editoriales.

Procesar Autor: ./Booquerio –a

Dispara el proceso de indexación por autores.

Procesa Título: ./Booquerio –t

Dispara el proceso de indexación por titulos.

Procesa Palabras: ./Booquerio –p

Dispara el proceso de indexación por palabras.
Listar Archivos Tomados: ./Booquerio –l

Muestra identificador, Título, Autor, Editorial y cantidad de palabras registradas para ese libro.

Obtener Archivo: ./Booquerio -o ID_Archivo

Recrea el libro en la ruta por defecto configurada en config.propiedades.

Quita Archivo: ./ Booquerio -q ID_Archivo

Elimina el archivo del sistema.

Ver Estructura: ./Booquerio -v -e | -a | -t | -p | -i “Nombre del archivo”
(-e : arbol de editoriales, -a : Arbol de autores, -t : hash de titulos, -p: hash de palabras, -i arbol de identificadores.)

Genera archivos en forma de texto plano, que describen las estructuras y contenidas de los archivos de almacenamiento y control del sistema.

Carga Masiva: ./Booquerio -z cantidad

Toma la cantidad de libros especificada de la carpeta libros e inicia una carga automatica.