

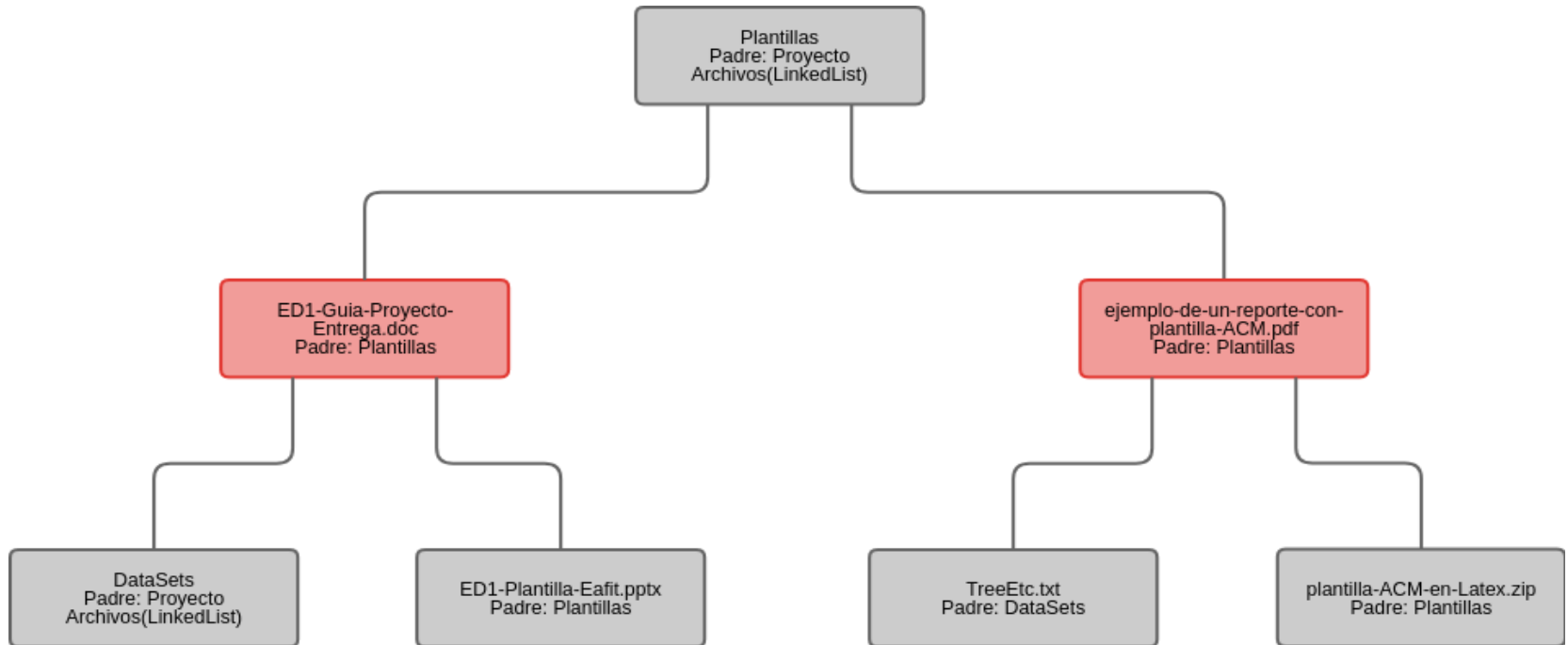
ESTRUCTURA DE DATOS PARA BUSCAR CONTENIDOS EN UN DIRECTORIO

Kevin Arley Parra

Daniel Alejandro Mesa Arango

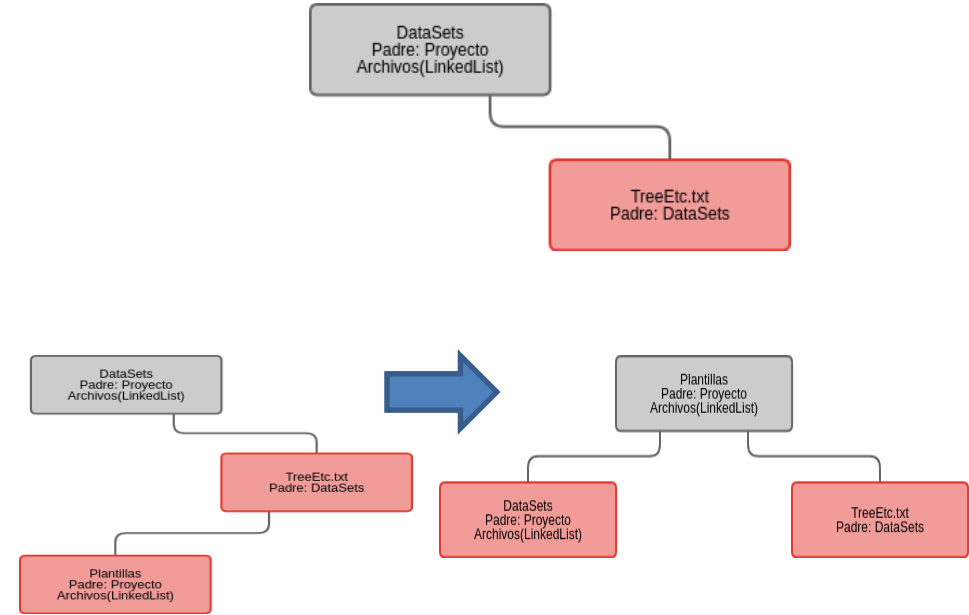
Medellín, 30 de octubre de 2017

Estructuras de Datos Diseñada



Gráfica 1: Árbol rojo negro de archivos y carpetas, un archivo y carpeta son clases que contienen un padre y un nombre o si hay iguales será una LinkedList con todos los iguales.

Operaciones de la Estructura de Datos



Método	Complejidad
Creación	$O(n \log n)$
Búsqueda	$O(\log n)$
Obtener ruta	$O(n*m)$
Listar todos los elementos	$O(n)$
Imprimir elementos repetidos	$O(m \log n)$

Gráficos uno, dos y tres: Creación, inserción y balanceo de un elemento en el árbol.

Tabla 1: Complejidad de las operaciones de la estructura de datos.

Criterios de Diseño de la Estructura de Datos

- Se requieren unas complejidades de inserción y búsqueda eficientes.
- La operación de búsqueda en un árbol rojo negro (TreeMap en java) tiene una complejidad de $O(\log n)$.
- La operación de inserción tiene una complejidad de $O(\log n)$, solo se utiliza en la creación siendo $O(n \log n)$.
- Las carpetas y archivos guardan cada uno de sus datos, por lo tanto no perdemos información y se conserva la jerarquía.
- Las listas enlazadas nos permiten guardar elementos repetidos, su inserción es $O(1)$ por lo cual no afecta la complejidad del árbol.

Consumo de Tiempo y Memoria

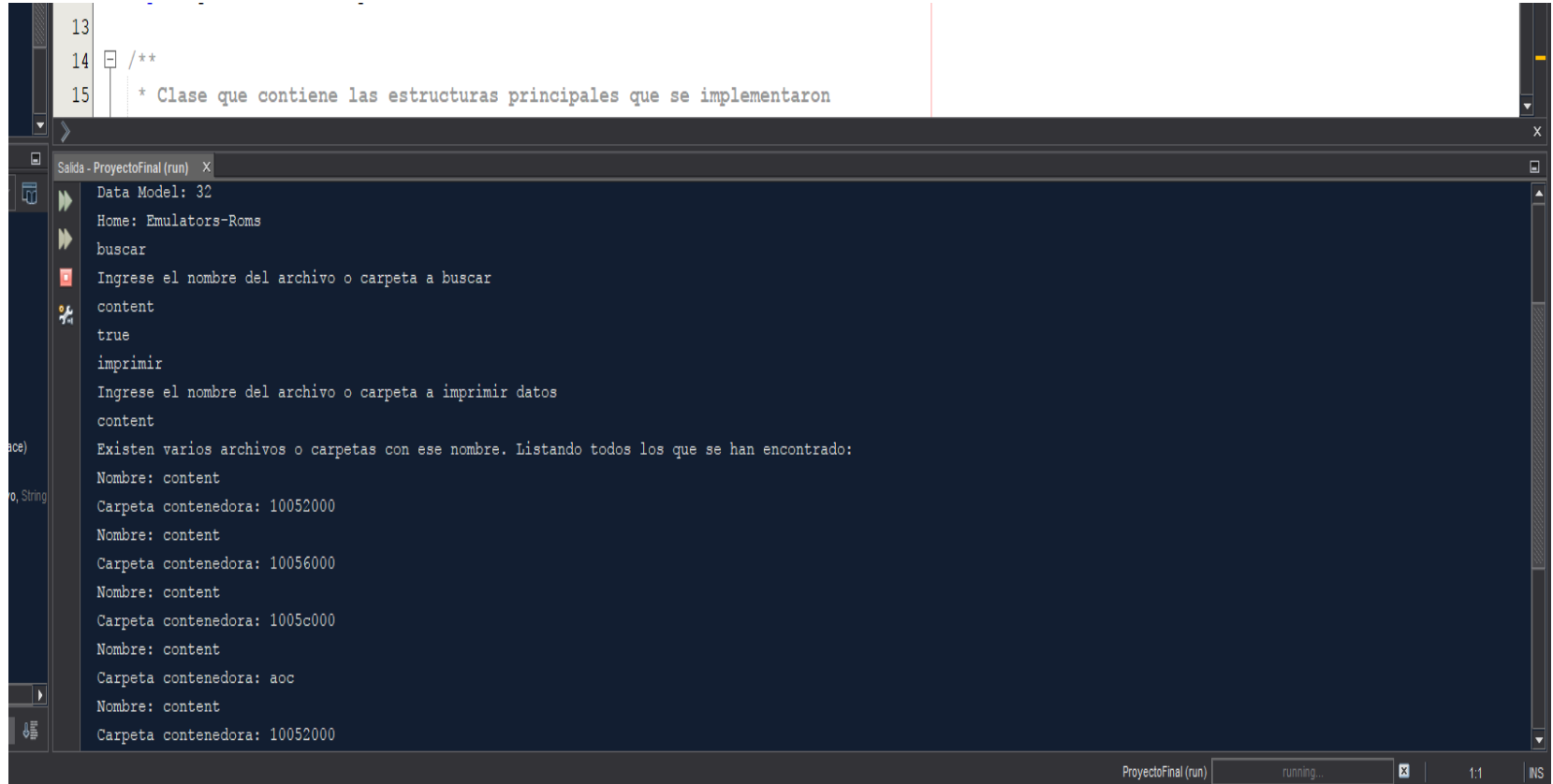
	juegos.txt	treeEtc.txt	ejemplito.txt
Creación	155.22ms	44.02ms	35.12ms
Búsqueda	0.0168ms	0.0051ms	0.0122ms
Obtener ruta	0.8ms	0.06ms	0.40ms
Listar todos los elementos	12582ms	483.81ms	0.86ms
Imprimir elementos repetidos	0.466ms	0.061ms	0.616ms

Estructura de datos para búsqueda en un directorio	Árbol rojo negro o TreeMap en java
Espacio en el Heap	20,78MB
Búsqueda de "content"	0.027ms
Búsqueda de "rules.txt"	0.016ms
Búsqueda de "title"	0.014ms
Búsqueda general	0.0168ms

	juegos.txt	treeEtc.txt	ejemplito.txt
Consumo de memoria	28,8MB	14,61MB	10,72MB

Como se puede observar en las tablas anteriores los tiempos son excelentes, siendo eficientes para cada uno de los casos y dependiendo de si existen o no elementos iguales dentro del .txt claro que esto no afecta mucho la complejidad y solo aumenta en una mínima parte el tiempo de ejecución.

Software Desarrollado



```
13
14 /**
15  * Clase que contiene las estructuras principales que se implementaron

```

Salida - ProyectoFinal (run) X

```
>> Data Model: 32
>> Home: Emulators-Roms
>> buscar
Ingrese el nombre del archivo o carpeta a buscar
content
true
imprimir
Ingrese el nombre del archivo o carpeta a imprimir datos
content
Existen varios archivos o carpetas con ese nombre. Listando todos los que se han encontrado:
Nombre: content
Carpeta contenedora: 10052000
Nombre: content
Carpeta contenedora: 10056000
Nombre: content
Carpeta contenedora: 1005c000
Nombre: content
Carpeta contenedora: aoc
Nombre: content
Carpeta contenedora: 10052000

```

ProyectoFinal (run) running... 1:1 INS

Gráfico 4: Búsqueda y obtención de carpetas contenedoras.

```
14  / **
15  * Clase que contiene las estructuras principales que se implementaron
```



```
Salida - ProyectoFinal (run) X
```

```
Listando elementos del directorio: Emulators-Roms
-007 - GoldenEye (USA).n64
-bios
--SCPH-10000_BIOS_V1_JAP_100.BIN
--SCPH-10000_BIOS_V1_JAP_100.MEC
--SCPH-10000_BIOS_V1_JAP_100.NVM
--SCPH-30000_BIOS_V4_JAP_150.BIN
--SCPH-30000_BIOS_V4_JAP_150.EROM
--SCPH-30000_BIOS_V4_JAP_150.MEC
--SCPH-30000_BIOS_V4_JAP_150.NVM
--SCPH-30000_BIOS_V4_JAP_150.ROM1
--SCPH-30000_BIOS_V4_JAP_150.ROM2
--SCPH-30003_BIOS_V3_UK_120.BIN
--SCPH-30004R_BIOS_V6_EUR_160.BIN
--SCPH-30004R_BIOS_V6_EUR_160.MEC
--SCPH-30004R_BIOS_V6_EUR_160.NVM
--SCPH-39001_BIOS_V7_USA_160.BIN
--SCPH-39001_BIOS_V7_USA_160.MEC
```

ProyectoFinal (run) running... 1:1 INS

Gráfico 5: Listar todos los elementos de un directorio.