



## Ejercicio 7: FileSystem

Un File System es un componente que forma parte del sistema operativo. Este está estructurado jerárquicamente en forma de árbol, comenzando con un directorio raíz.

Los elementos del file system pueden ser directorios o archivos. Los archivos contienen datos y los directorios contienen archivos u otros directorios. De cada archivo se conoce el nombre, fecha de creación y tamaño en bytes. De cada directorio se conoce el nombre, fecha de creación y contenido (el tamaño es siempre la cantidad inicial de 32kb más la suma del tamaño de su contenido). Modele el file system y provea la siguiente funcionalidad:

```
public class FileSystem {  
    /**  
     * Retorna el espacio total ocupado, incluyendo todo su contenido.  
     */  
    public int tamanoTotalOcupado()  
  
    /**  
     * Retorna el archivo con mayor cantidad de bytes en cualquier nivel del  
     * filesystem  
     */  
    public Archivo archivoMasGrande()  
  
    /**  
     * Retorna el archivo con fecha de creación más reciente en cualquier nivel  
     * del filesystem  
     */  
    public Archivo archivoMasNuevo()  
  
    /**  
     * Retorna el primer elemento con el nombre solicitado contenido en  
     * cualquier  
     * nivel del filesystem  
     */  
    public ?? buscar(String nombre)  
  
    /**  
     * Retorna la lista con los elementos que coinciden con el nombre solicitado  
     * contenido en cualquier nivel del filesystem  
     */  
    public List<??> buscarTodos(String nombre)  
  
    /**  
     * Retorna un String con los nombres de los elementos contenidos en todos  
los  
     * niveles del filesystem. De cada elemento debe retornar el path completo  
     * (similar al comando pwd de linux) siguiendo el modelo presentado a  
     * continuación
```



```
/Directorio A  
/Directorio A/Directorio A.1  
/Directorio A/Directorio A.1/Directorio A.1.1  
/Directorio A/Directorio A.1/Directorio A.1.2  
/Directorio A/Directorio A.2  
/Directorio B
```

```
*/
```

```
public String listadoDeContenido()
```

```
}
```

## Tareas:

1. Diseñe y represente un modelo UML de clases de su aplicación, identifique el patrón de diseño empleado (utilice estereotipos UML para indicar los roles de cada una de las clases en ese patrón).
2. Diseñe, implemente y ejecute test cases para verificar el funcionamiento de su aplicación.
3. Implemente completamente en Java.