Clase 02/10/2025

En la clase anterior se habló de Objetos Compuestos. Ya está habilitado el TP8.

Estructura compleja/compuesta 🡪 Realizar búsquedas compuestas

# Búsquedas

(Se vieron cosas similares con Filtros 🡪 son abstractos, llamados Condición)

* Permitir búsquedas y filtros de más de un tipo distinto de elementos.
* Filtrar con mayor profundidad que lineal.
* En las condiciones/filtros no hay que cambiar la asignatura del método abstracto.

## Ejercicio de clase similar a TP9 Ej1

* Hay diferencia entre CondicionNombreIgual y CondicionNombreContiene.
  + Herencia por atributo es una mala práctica.
* CondicionTamanioMayor
* CondicionFechaPosterior
* Condición combinada 🡪 todas extends Condicion
  + CondicionAND
  + Condición 1 y 2 🡪 se guardan en el contructor.
  + cumple(Archivo archivo){

return cond1.cumple(archivo) && cond2.cumple(archivo);

}

* + CondicionOR
  + Condición 1 y 2 🡪 se guardan en el contructor.
  + cumple(Archivo archivo){

return cond1.cumple(archivo) || cond2.cumple(archivo);

}

* + CondicionNOT
  + Condición 🡪 se guarda en el contructor.
  + cumple(Archivo archivo){

return

}

* Está bueno meter las condiciones en un paquete si uno prefiere mantener orden.
* Ahora hay que crear el método Buscar, en donde podamos poner todas las condiciones y que nos encuentre los archivos. Lo ponemos en la clase “Directorio” que hereda de “ElementosSA”.
* Se crea el método buscar tanto en la clase ElementoSA, que es abstracta, como en sus hijos.
  + Retorna una Lista
  + ArrayList<Archivo> buscar(Condicion condicion){

//Creamos lista vacía que devolvemos a la visual.

ArrayList resultado = new ArrayList<>();

//Chequeo si el mismo directorio cumple con la condición

If(condición.cumple(this))

resultado.add(this);

//O en vez del siguiente FOR, podría hacerse

resultado.addAll(elemento.buscar(condicion))

for (ElementoSA elemento : elementos){

//No se puede consultar el tipo del elemento. No se pregunta si es Archivo.

ArrayList<Archivo>resultadoParcial = elemento.buscar(condición);

resultado //concatenar la búsqueda del elemento. Unir listas.

for (Archivo archivoQueCumple : resultadoParcial){

Resultado.add(archivoQueCumple);

}

}

Return resultado;

}

* En la clase Archivo
  + Agregar el método buscar(Condicion condicion)
    - Retorna una lista. 🡪 Una lista vacía no es lo mismo que un null.
* Si hago “Carpeta oculta” hay que devolver una lista vacía, así no aparece en resultados de búsqueda.
* Un “Comprimido” es hijo de “Directorio”
* Un “Link” es hijo de “ElementoA”
* Al final el código del método buscar(Condicion) y si se cumple la condición se repite en todas las subclases. Por lo tanto, es mejor poner el método en la clase abstracta “ElementoA”
* Patrones de métodos

# TP9

## TP9 Sistema de Archivos