

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

Para cada uno de los siguientes problemas Implementarlos en Java.

1. Fútbol 5

Un club de fútbol 5 necesita un sistema que le permita llevar control de sus socios y alquileres de canchas. Cada socio posee un nombre y apellido, una edad, si tiene paga la última cuota y una lista que contiene información de los alquileres de canchas que ha realizado. Cuando un socio alquila una cancha, se registra la fecha, el ID de la cancha que alquiló y cuánto pagó por el alquiler.

Cuando el dueño del club arriba al predio, le pide a los que administran el sistema, diferentes listados, para comprobar cómo está funcionando el club:

- a) Todas los socios con cuota impaga, **ordenados alfabéticamente (primero por apellido, y si hay varios con el mismo apellido por nombre)**
- b) Todos los socios menores de edad, **ordenados por edad**
- c) Todos los socios que alquilaron alguna vez la cancha N°13, **ordenados cantidad total de alquileres de esa cancha.**
- d) Todos los socios que pagaron más de \$500 por algún alquiler de cancha, listando primero los que tienen las cuotas pagas, **ordenados alfabéticamente y luego los morosos, también ordenados alfabéticamente.**
- e) Cualquier combinación lógica de búsquedas.

2. Sistema de Archivos

Un Sistema de Archivos debe organizar y manipular: archivos, directorios, links y archivos comprimidos. Los archivos se definen por un nombre, una fecha de creación y un tamaño. Los directorios tienen un nombre, una fecha de creación y además contienen un conjunto de archivos y un conjunto de subdirectorios. El tamaño de un directorio está dado por el tamaño de sus sub-directorios, sus archivos, sus links y sus archivos comprimidos. Los links son vínculos a otro archivo o directorio y tienen un nombre, una fecha de creación y su tamaño en disco es siempre igual a 1Kb. Los archivos comprimidos son un tipo particular de archivo el cual contiene otros archivos y/o directorios en formato comprimido según una tasa de compresión dada (El tamaño de un archivo comprimido es el tamaño de todo lo contenido dividido el factor de compresión), tiene un nombre y una fecha de creación.

El sistema debe permitir realizar búsquedas de **cualquiera de sus elementos** según diferentes criterios (**simples y compuestos**¹), por ejemplo:

¹ ojo con eso que cambia de como se vio en la clase

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

- Todos los elementos cuyo nombre contenga la palabra “parcial”.
- Todos los elementos que se hayan modificado luego del 15-10-2020.
- Todos los elementos que fueron creados antes del 15-03-2020 y cuyo nombre sea igual a “parcial”.
- Cualquier combinación lógica (**and**, **or** y **not**) de búsquedas simples.

Aclaración sobre las búsquedas. Un *archivo comprimido* será incluido en el resultado de la búsqueda, si alguno de los elementos que contiene en su interior cumple con el criterio de la búsqueda (se agrega el archivo comprimido al resultado, pero no los elementos que contiene). Un *link cumple con el criterio* de búsqueda por sí mismo, sin tener en cuenta el elemento que apunta.

A la hora de realizar una búsqueda, el usuario puede especificar también un criterio de **orden** en el que quiere que se retornen los elementos resultantes. Por ejemplo, ordenados ascendentemente por fecha de creación y luego por nombre, ordenados descendentemente por tamaño, luego por fecha de creación y finalmente por nombre descendente, entre otros posibles criterios de ordenamiento.

3 - Sitio Web de Noticias

Se desea modelar un administrador de contenidos de un sitio web de noticias. Los contenidos del sitio están agrupados en **categorías** y dentro de cada categoría se dividen en **secciones**. Asimismo, es posible que una sección se sub-divida nuevamente en otras **sub secciones**, y así sucesivamente. Cada **categoría**, posee una descripción y una imagen asociada, lo mismo para las **secciones**.

Una noticia puede estar asociada a una **categoría**, **sección** o **sub-sección**, posee un título, un listado de palabras claves, una introducción, el texto, el autor, y un link asociado.

El modelo presentado debe proveer servicios que permitan obtener la cantidad de noticias que contiene una **categoría**, y la cantidad de noticias que contiene una **sección**, **sub-sección**, etc.

Se debe proveer un mecanismo mediante el cual el administrador de contenidos pueda restringirse a un tópico determinado, es decir *ingresando una palabra clave se genera una nueva* interfaz en donde **sólo** las noticias que contienen esta palabra clave se muestran (es importante que se **respete la organización de las mismas**, es decir categoría, sección, sub-sección y así sucesivamente).

La nueva estructura obtenida es una **copia de la existente**, si se introduce un cambio en esta, el mismo **NO** se refleja en la estructura original.

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

Adicionalmente se debe proveer servicios que permitan la **búsqueda de noticias**, por ejemplo:

1. Buscar todas las **noticias** donde el largo del texto sea superior a 200.
2. Buscar todas las **noticias** del autor “Juan Pérez”.
3. Buscar todas las **noticias** que posean la palabra clave “Partido” y la palabra Clave “Goleada”.
4. Buscar las **noticias** que el título sea “Ultimo momento”.

Los anteriores son sólo algunos ejemplos de búsqueda. Los distintos tipos de búsqueda también pueden combinarse.

Debido a la dinamicidad del sitio (se crean constantemente nuevas categorías, secciones, subsecciones) es necesario proveer un mecanismo que permita armar el mapa del sitio, es decir, obtener un listado de los distintos niveles del sitio y los links asociados a cada una de las noticias, por ejemplo a partir del contenido del sitio actual obtener una **lista** de la siguiente forma:

Generales

Generales\Espectáculos

Generales\Espectáculos\Link 1

Generales\Espectáculos\Divorcios

Generales\Espectáculos\Divorcios\Link 2

Generales\Espectáculos\Divorcios\Link 3

Generales\Deportes

Generales\Deportes\Fútbol

Generales\Deportes\Fútbol\Link 4

Generales\Deportes\Link 5

Generales\Link 6

Solo se devuelven los nombres de las **categorías/secciones** y en el caso de las **noticias** el link asociado. La lista resultado concatena la información de manera tal de contar con el path de la estructura.

En el ejemplo anterior, el **Link5** está dentro de la sección “**Deportes**”, la cual está dentro de la categoría “**Generales**”.

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

4 - Work Breakdown Structure

Una Work Breakdown Structure (Wbs) es una estructura de árbol completa y jerárquica (desde lo más general a lo más específico) de tareas que necesitan ser realizadas para completar un proyecto.

Un ejemplo de una estructura Wbs para las actividades de pintar una habitación sería:

- +Preparar Materiales
 - Comprar pintura
 - Comprar Pinceles/Rodillos
 - Comprar Lijas
- +Preparar la habitación
- +Preparar paredes
 - Remover empapelado
 - Sacar toda la decoración desmontable
 - Sacar apliques de luz
- +Proteger elementos
 - Cubrir el piso con diarios
 - Cubrir enchufes/bordes con cinta de papel
 - Cubrir los muebles con sábanas viejas
- Pintar el cuarto
- +Limpiar el cuarto
 - Guardar la pintura que sobró
 - Limpiar los pinceles y/o rodillos
 - Desechar los diarios
 - Remover los cobertores de los muebles

A cada **tarea terminal** del Wbs se le pueden asignar recursos. Los recursos pueden ser materiales (computadora, mesa, etc) o personas (Juan Perez, Roberto Giord, etc). Estos recursos pueden ser de dos tipos: **exclusivos** (es decir que pueden estar asignados a una única tarea en un tiempo dado, por ejemplo una persona) o **compartidos** (recursos que se pueden compartir en tareas, por ejemplo una impresora).

Las **tareas terminales** poseen una *fecha de inicio* y una *fecha de finalización* estimada, así como también *fecha de inicio* y *finalización real*. También se guarda si una tarea ya se *completó*, *está en espera* o *en procesamiento*. La *fecha de inicio* de una tarea **compuesta** queda determinada por la *fecha de inicio más temprana de sus subtareas*. De manera análoga, la *fecha de finalización* de la tarea compuesta queda determinada por la fecha de finalización más tardía de sus subtareas.

Servicios que deben ser provistos por el administrador de Wbs:

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

1. Devolver la **lista de tareas** que están atrasadas (fecha de finalización estimada es posterior a la fecha actual y su estado es “en espera”).
2. Dado un recurso, devolver la **lista de tareas** que lo utilizan.
3. Dada una **tarea**, devolver los **recursos involucrados** (en caso de no ser una tarea terminal es la unión de todos los recursos de las tareas que lo componen, SIN REPETIDOS).
4. **Duración** estimada de una tarea.
5. Consultar si es posible asignar el recurso X a la tarea YY (tener en cuenta las fechas de utilización de los recursos y si los mismos son o no exclusivos).

6. Coleccionables

Collectors 3.0 es un sistema que se utiliza para la organización de figuritas coleccionables. Una figurita coleccionable posee un nombre, un álbum al que pertenece, un número y un precio. Las figuritas se agrupan en colecciones temáticas las cuales pueden contener figuritas o incluso otras colecciones menores. Cada colección temática posee un nombre, un precio que se calcula como la suma de los precios de cada uno de sus elementos menos un 5% de descuento (hay colecciones temáticas discontinuas que posee un 10% o 13%). Además de los servicios que se desprenden del enunciado anterior, se deben proveer mecanismos que permitan:

- Buscar todas las figuritas que:
 - sean del Album Copa America 2024
 - el precio sea menor a 250
 - el número es 19.
 - Combinaciones lógicas de las anteriores

Nota: en el caso de las colecciones, las mismas **no se incluye en la búsqueda**, pero sí los *elementos* que contiene que si cumplan con lo solicitado (en cualquier nivel de profundidad). Los listados se ordenan **siempre** por el precio de la figurita y en caso de dos figuritas con el mismo precio, se ordenan alfabéticamente por su nombre.

Asimismo hay coleccionistas que poseen colecciones más restrictivas que no permiten realizar búsquedas sobre las mismas: Son colecciones que pueden tener figuritas o incluso otras colecciones pero cuando se busca un elemento siempre retorna una lista vacía, ya que evitan que se conozca su contenido.

Con el auge de las figuritas se desea incorporar una figurita edición limitada que además del nombre, álbum, número y precio incluye un número de unidades disponibles en el mundo. En estas figuritas, el precio se puede calcular de diferentes maneras, en algunos casos tiene un precio fijo, en otros es el precio fijo multiplicado por 1000 - la cantidad de unidades disponibles, en otros casos es 100 - la cantidad de unidades disponibles. Esta forma de

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

cálculo puede cambiar. **Nota importante**, el cálculo especial del precio nunca debe ser menor al precio que ya posee el Coleccionable.

7. Electronica 1.1

Un sistema de gestión para una cadena de tiendas de electrónica permite el registro de productos tales como televisores, teléfonos móviles, laptops, etc. Cada producto posee nombre, un valor, y una lista de palabras claves. Los productos físicos además poseen una cantidad en stock, mientras que los productos digitales cuando le preguntan por el stock siempre retornan el mismo valor de 2.500.000 (el cual es fijo y no va a variar nunca para este tipo de producto). La tienda agrupa los productos en diversas secciones y categorías que se identifican por un nombre. Las categorías y secciones pueden contener productos así como también otras subcategorías y subsecciones. El valor y la cantidad de productos de una categoría o sección se calcula como la suma de los valores/cantidades en stock de los elementos que contienen. Las palabras claves de una categoría quedan determinadas por, como máximo, las 5 (cinco) últimas palabras de la unión sin repetidos de todas las palabras claves de sus elementos. Las palabras claves de una sección, sin embargo, quedan determinadas por, como máximo, las 3 (tres) últimas palabras claves de la unión sin repetidos de todas las palabras claves de sus elementos.

Las secciones y categorías, no se forman al azar, sino que los elementos que se agregan a las mismas deben respetar ciertas reglas, por ejemplo:

- Sección “tecnología”: sólo acepta elementos en cuyas palabras claves se encuentre “tecnología”.
- Categoría “ofertas”: sólo contiene elementos cuyo valor no supere los \$20.000
- Sección “escasos”: contiene únicamente elementos con menos de 10 productos en stock
- Sección “hogar”: sólo contiene elementos cuyo nombre incluya la frase “para el hogar”
- Sección “exclusivos”: sólo contiene elementos cuyo valor supera \$1.000.000
- Categoría “eco-friendly”: contiene sólo elementos entre cuyas palabras claves se encuentre “reciclado”

El sistema debe permitir al usuario, la búsqueda de productos por cada uno de sus atributos (nombre, valor, cantidad en stock y/o palabras clave) así como por combinaciones lógicas de ellos. Los listados deben devolverse ordenados, según el criterio especificado por el usuario (por ejemplo, los más baratos primero, y para productos de igual valor, por nombre ascendente, o por la primera palabra clave, o por valor). Tener en cuenta que los listados anteriores solo incluyen productos, y no categorías o secciones ni subsecciones.

Además de lo mencionado anteriormente el modelo propuesto debe permitir crear los siguientes elementos en el sistema:

- Productos “todo por 1000 pesos”, que posee un nombre, una cantidad en stock, una lista de palabras claves y un valor de 1000 pesos

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

- Secciones “promocionadoras”, son secciones que además de tener un nombre y una lista de productos, secciones o categorías, tienen un producto en promoción, cuando se buscan productos en dicha sección incorporan como primer elemento al resultado clásico de la búsqueda, el producto en promoción. Las palabras claves de esta categoría quedan conformadas por las primeras 4 palabras claves (como máximo) del producto en promoción.
- Productos “versátiles”, son productos que tiene las palabras claves “cocina” “living” “dormitorio”
- Categorías “amigables”, son categorías que tienen productos de menos de 3000 pesos
- Producto “variable” cuyo stock se calcula como la cantidad de palabras claves que poseen.

8. Kiko

Kiko es un robot didáctico orientado a iniciarse en el arte de programar desde la infancia. Para programar a Kiko, se utilizan tarjetas RFID con diferentes comandos que se cargan secuencialmente en el robot. Algunas tarjetas RFID contienen comandos con una instrucción, como “avanzar”, “retroceder”, “girar a izquierda”, “girar a derecha”, “producir sonido” y “bailar”, aunque futuras versiones de Kiko incorporarán tarjetas con nuevos comandos. Por otro lado, existen tarjetas especiales que permiten la definición de bloques de comandos. Cada bloque agrupa diferentes comandos, y puede a su vez contener otros bloques de comandos. Existen también tarjetas que indican la iteración de otra tarjeta (bloque o comandos) y por lo tanto su contenido se ejecuta una determinada cantidad de veces, de inicio a fin.

Cada comando implica, además de la instrucción, un gasto de batería determinado (medido en mAh) y tarda ciertas unidades de tiempo en ejecutarse que dependen de la complejidad del comando. El tiempo de ejecución de un bloque es la suma de los tiempos de ejecución de sus elementos. En los bloques iterativos, el tiempo de ejecución se calcula como el tiempo de ejecutar el bloque por la cantidad de iteraciones. El gasto de batería de un bloque se calcula como la suma de los gastos de batería de cada elemento del bloque, más un extra fijo **multiplicado** por la cantidad de unidades de tiempo que insume ejecutar el bloque (*por defecto el extra es de 5mAh, y es el mismo para todos los bloques, pero podría tomar otro valor a futuro y debe actualizarse para todos los bloques*). En el caso de las tarjetas iterativas, el gasto de batería de su contenido se multiplica por la cantidad de iteraciones.

La solución propuesta debe proveer los siguientes servicios:

- a. Determinar el tiempo necesario para la ejecución de un programa.
- b. Determinar la cantidad mAh necesarios para ejecutar un programa.
- c. Determinar la cantidad de tarjetas utilizadas para crear el programa.
- d. Retornar una copia de un programa, pero que solo incluya comandos con ciertas características, por ejemplo:
 - que no superen 10 unidades de tiempo

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 11

- que consuman menos de 100mAh y contengan el comando “bailar”
- que contengan la instrucción “girar a izquierda” o “girar a la derecha” y no consuman más de 200mAh de batería

NOTA: Los anteriores son solo ejemplos y la solución propuesta debe ser flexible para la definición de nuevas restricciones. En el caso de las tarjetas de bloques e iterativas, si al realizar la copia la misma queda vacía (sin elementos), la misma NO se incluye en la copia.

e. Retorne una lista de las instrucciones de un bloque. Por ejemplo, si el bloque es:

- Iterar 3 veces :
 - avanzar
 - bailar
- girar a la izquierda
- bailar

Se debe retornar la siguiente lista < “avanzar”, “bailar”, “avanzar”, “bailar”, “avanzar”, “bailar”, “girar a izquierda”, “bailar”>

Incorporar al diseño dos nuevos tipos de tarjetas:

- **Emotiva**, tiene la instrucción “sonreír”, consume 1mAh de batería, y tarda 2 unidades de tiempo.
- **Incógnita**, tiene una instrucción, un consumo de energía y una unidad de tiempo, pero la misma no se incluye en la lista de instrucciones (servicio e), no se cuenta en la cantidad de tarjetas (servicio c) y no se copia nunca (servicio d).

Se debe poder solicitar la ejecución de un programa de cualquier tipo (Comando, Bloque, Iterativo, Emotivo, o Incógnito) sobre un robot Kiko dado. Dicha ejecución se realiza siempre y cuando el nivel de batería del robot sea superior al consumo del programa. En caso de poder ejecutarse, se debe actualizar el valor de la batería disponible del robot y retornar un String con la cantidad de tarjetas ejecutadas concatenado con la secuencia de instrucciones ejecutadas. Si no se puede ejecutar se debe retornar null.