

Programación 2  
Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas  
Práctica N° 8 – 2020

## 1. 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, una fecha de última modificación 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 (**sencillos y compuestos**), por ejemplo:

- 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 ascendente por fecha de creación y luego por nombre, ordenados descendente por tamaño, luego por fecha de creación y finalmente por nombre descendente, entre otros posibles criterios de ordenamiento.

## 2. Sistema Electoral

Un sistema de recuento automático de votos para una elección registra los votos recibidos por los diferentes candidatos. Los candidatos poseen un nombre, un partido político y una agrupación.

**Programación 2**  
**Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas**  
**Práctica N° 8 – 2020**

Cada votante registra su voto en una mesa. Cada mesa tiene asociado un conjunto de personas que pueden votar en la misma (padrón), identificadas únicamente por su DNI. Las mesas se encuentran distribuidas en diferentes establecimientos educativos, que a su vez se agrupan por barrio y localidad. Adicionalmente, puede haber mesas especiales que no pertenezcan a un establecimiento educativo sino al barrio o localidad directamente.

Cuando un votante se presenta a emitir su voto, se debe controlar que el mismo se encuentre en el padrón correspondiente. Si se cumple esta condición, se registra el voto de la persona, que puede contener una referencia a un candidato o ser un voto en blanco (sin referencia a ningún candidato). Adicionalmente el voto guarda la hora en que se emitió el mismo (el momento en el que se crea el voto).

**Nota:** Todos los votos emitidos se guardan en la mesa, aunque sean en blanco (es decir, siempre existe el voto, es lo mismo que sucede con un sobre en una elección real)

Implementar los siguientes servicios:

1. Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer el porcentaje de votos recibidos por un candidato particular.
2. Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer el porcentaje de votos en blanco recibidos.
3. Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer el porcentaje de votos que se emitieron entre las 10:00 y las 11:00. (La clase LocalDate implementa la interface Comparable)
4. Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer la cantidad total de votos que se emitieron.
5. Obtener un listado de todos los candidatos que se presentan a la elección, por partido político, agrupación y nombre, en ese orden.

**Aclaración,** los porcentajes se calculan como la cantidad de votos que cumplen la condición dividido el total de votos emitidos en la mesa, establecimiento, barrio o localidad multiplicado por 100.

Programación 2  
Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas  
Práctica N° 8 – 2020

### **3. Navidad**

En esta Navidad Papá Noel quiere informatizar su sistema de pedido y distribución de regalos. Para ello instaló un buzón en cada pueblo o ciudad, para que los niños del mundo puedan depositar sus cartas. Cada carta posee un remitente y una lista de regalos solicitados (un regalo es simplemente un String).

Cada niño deposita su carta en el buzón correspondiente a su barrio. Cada buzón tiene asociado un conjunto de “niños buenos” que pueden dejar su carta en el mismo, identificados únicamente por su DNI. Los niños que no se encuentran en el listado de niños buenos de un buzón, se los considera “niños malos”. Si un “niño malo” intenta depositar una carta en el buzón, la lista de regalos correspondiente se vacía y se reemplaza el regalo por un “Trozo de carbón”. Si un niño deposita más de una carta en el buzón, solo se considera la primera.

Los buzones se encuentran distribuidos en diferentes barrios, que a su vez se agrupan por localidad, provincia o estado y país. Adicionalmente, puede haber buzones especiales que no pertenezcan a un barrio, sino a la localidad, provincia o país directamente (por ejemplo en la embajada de un país en otro).

Cuando un niño deposita su carta en un buzón, se debe controlar que el mismo se encuentre en el listado de niños buenos asociado al buzón. Si se cumple esta condición, se guarda la carta para que luego sea recolectada por Barney y entregada a Papá Noel.

**Nota:** Todas las cartas recibidas se guardan en el buzón, aunque sean de “niños malos” (en este caso se debe realizar el reemplazo de regalos como se mencionó anteriormente).

Implementar los siguientes servicios:

1. Dado un buzón, provincia o país, conocer el porcentaje de cartas recibidas que piden un determinado regalo.
2. Dado un buzón, provincia o país, conocer la cantidad de cartas recibidas que piden un determinado regalo.
3. Dado un buzón, provincia o país, conocer la cantidad de niños malos que enviaron una carta.
4. Dado un buzón, provincia o país, conocer la cantidad de total de cartas recibidas.

**Aclaración**, los porcentajes se calculan como la cantidad de cartas que cumple la condición dividido el total de cartas recibidas en el buzón, provincia o país, multiplicado por 100.

## Programación 2

Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas  
Práctica N° 8 – 2020

### 4. 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.

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, sub-secciones) 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

Programación 2  
Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas  
Práctica N° 8 – 2020

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**”.

## **5. Control de Gastos**

Se desea implementar un sistema para el control de gastos de una organización. El sistema debe poder permitir almacenar el gasto con su descripción, la fecha, su monto y demás características que se le deseen incorporar. Asimismo, es posible incluir un detalle pormenorizado sobre un gasto. De esta forma el usuario puede detallar cómo se compone originalmente el gasto. Por ejemplo, el gasto “construcción edificio” se basa en gastos en cimientos, gastos en edificación y gastos en pintura. El **monto** total del gasto es la suma de cada una de las partes que lo componen.

Siempre se puede **caracterizar** **cualquiera** de los tipos de gasto acorde a lo deseado por el **usuario**, por ejemplo para gastos de pintura, color rojo, calidad media, metros 35.

# Programación 2

## Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas

### Práctica N° 8 – 2020

Se deben proveer servicios que permitan obtener:

1. El **total de dinero** gastado en una fecha dada.
2. El **total de dinero** que se gastó en el rubro: Alimentos.
3. Un **listado con todos los gastos** de una fecha dada.
4. Un **listado con todos los gastos** de calidad media que se hicieron.
5. El **total de dinero** que se gastó en pintura, en una fecha dada.
6. Un **listado de todos los gastos** que superen los 1556 pesos.
7. Estos son solo algunos ejemplos de listados y sumas de gastos que se pueden solicitar. Es posible que se combinen distintas formas de pedir sumas o listados.

**Aclaración sobre los listados de gastos:** Cuando se trabaja con un gasto que se encuentra **detallado** se debe proceder de la siguiente forma. Por ejemplo, si se solicitan los productos de calidad alta, y el gasto (que está detallado) posee una calidad alta, este gasto es incluido en el listado, ahora sí el mismo NO posee calidad alta pero dentro del detalle hay gastos que sí poseen calidad alta, estos últimos son los que deben ser incluidos en el listado.

Proveer la funcionalidad necesaria para el **siguiente tipo de gasto**: la organización administra un caso de **gasto pendiente de pago**. Dicho gasto puede ser cualquier tipo existente (un gasto simple o un gasto detallado) o cualquier tipo de gasto que se agregue en un futuro, tiene un porcentaje de actualización anual, la cantidad de días pendiente, si el mismo ya se pagó o está pendiente, y el monto se calcula en base al gasto contenido más el porcentaje acorde a los días pendiente de pago. Si ya se pagó el monto es un valor fijo, calculado al momento que se informó el pago. Implementar la funcionalidad que permita informar el pago del gasto y fijar el valor de dicho monto.

## 6. 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

## Programación 2

### Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas

### Práctica N° 8 – 2020

- 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:

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 suma 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).