

PARTE 2: Ejercicios de aplicación

Duración: 60 minutos

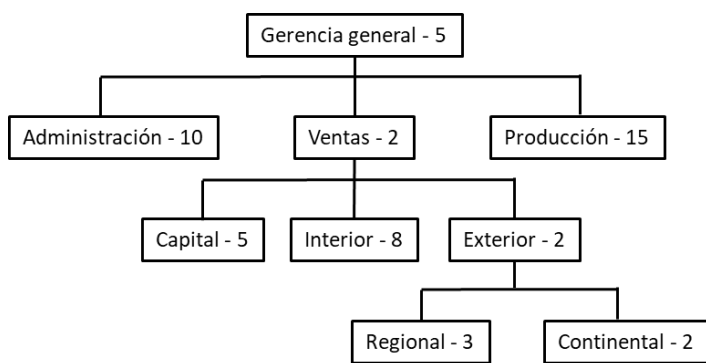
Puedes tener **únicamente** material propio

EJERCICIO 1

Escenario

Una cierta organización almacena su organigrama en un árbol genérico. Entre la variada información que se mantiene para cada componente del organigrama, está la cantidad de empleados que trabajan en ese lugar, y entre la información que se debe obtener del árbol, es la cantidad total de empleados que trabajan en cada componente y en todos los componentes que dependen de él.

Por ejemplo, si el organigrama es:



La cantidad total de empleados que trabajan en “Ventas” y todas sus dependencias es 22.

Consigna

Sabiendo que el organigrama está representado por un árbol genérico implementado con la técnica “primer hijo – siguiente hermano”, desarrollar un método que, aplicado sobre un nodo del árbol, devuelva la cantidad total de empleados que trabajan en el componente asociado y en todos los componentes dependientes de él, respetando las siguientes firmas:

En la clase TNodeArbolGenérico:

TEtiqueta etiqueta
TNodeArbolGenérico primerHijo
TNodeArbolGenérico siguienteHermano
TComponente datos

Entero cantidadEmpleados() // método a desarrollar

En la clase TComponente

TEtiqueta nombre

Entero empleadosComponente() // método disponible para usar

EJERCICIO 2

Escenario

Se han recibido una serie de mensajes encriptados en forma de Código binario (los caracteres son ceros y unos), y se desea comenzar a descifrarlos. Para ello se ha de comenzar identificando la ocurrencia de un cierto patrón en estos mensajes.

Se desea entonces utilizar un árbol de sufijos para **reconocer patrones en una secuencia de caracteres que representan el mensaje**.

Ejemplo sencillo de mensaje: **"01100011"** (los mensajes reales tienen cientos o miles de caracteres)

En este caso, el "alfabeto" está integrado por los caracteres **"0"** y **"1"**.

Consigna

Se debe:

- 1- **Diseñar** una estructura de datos apropiada para representar un nodo de este trie, para este alfabeto (**NO BASADA EN COLECCIONES**)
- 2- Construir un **árbol de sufijos** para el siguiente mensaje:
"01100110"
- 3- Escribir en pseudocódigo un algoritmo que permita indicar si un cierto patrón (una string de caracteres del alfabeto definido) se encuentra contenido en el mensaje, y en caso que esté, indique todas las posiciones del mensaje en que comienza dicho patrón.
- 4- Demostrar la ejecución del algoritmo manualmente, utilizando el trie de sufijos creado en **2.** , para el patrón **"110"**
- 5- ¿Cuál es el orden del tiempo de ejecución de este algoritmo? (justifícalo)
- 6- Desarrollar y dibujar el TRIE COMPRIMIDO correspondiente.

De TArbolTrie

buscarPatron(tipo cadena elPatron) devuelve una "colección" de elementos de tipo **"entero"** que contiene todas las posiciones del mensaje en que aparece el patrón, o -1 si no existe.

De TNodeTrie: "buscarPatron"

Definir una firma apropiada para el método en TNodeTrie y desarrollarlo.