

Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS1205 – Algorítmica y Programación 2
Hoja de trabajo teórica Nivel 9

Para el desarrollo del examen usted debe trabajar sobre el modelo del ejercicio del nivel 9. Complete las signaturas de clase y métodos donde sea necesario.

1. (35%) La administración de CupieMart quiere poder acceder con mayor rapidez a las sucursales que manejan un número de líneas de productos mayor que un valor tope dado. Por lo tanto quiere colocar las sucursales con mayor número de líneas de producto al principio de la lista de sucursales. Escriba un método de la clase CupieMart que coloque al principio de la lista de sucursales aquellas que tienen más de tope líneas de productos.

```
Public class CupieMart
{ ...
    /**
     * Método que organiza las sucursales, dejando al comienzo de la lista todas las
     * que tienen más de tope número de líneas de producto
     * @param tope cantidad de líneas de producto mínima para estar al comienzo
     * de la lista de sucursales
     */
    void organizarSucursales (int tope)
    {

    }
}
```

2. (35%) Se quiere que cuando

las líneas de producto ingresan a la sucursal queden organizadas alfabéticamente por código de línea de producto. Modifique el método que en una sucursal añade un nuevo producto para que ahora queden en el orden indicado.

```
public class Sucursal
{
    /**
     * Método que agrega una nueva línea de producto en la sucursal, en orden
     * ascendente de código de producto <br>
     * <b>post:</b> Se agrega una nueva línea de producto a la sucursal. En caso que ya
     * exista una línea de producto con el mismo código se lanza una excepción<br>
     * @param nombreProducto Nombre que se da al producto. Diferente de null
     * @param codigoProducto Código que se le asignara al producto. Diferente de null
     * @param descripcion Descripción de la línea de producto. Diferente de null
     * @param precio Precio que tendrán los productos de esa línea. Mayor que cero
     * @param fecha Fecha cuando fue lanzada la línea del producto. Diferente de null
     * @throws LineaProductoExisteException Cuando ya se encuentra un producto con el
     * mismo código
     */
    public void lanzarLineaProducto( String nombreProducto, String
    codigoProducto, String descripcion, int precio, Fecha fecha ) throws
    LineaProductoExisteException
    {

    }
}
```

3. (30%) Desarrolle el método `quitarSucursalesSinProductos` que elimina de `CupEMart` aquellas sucursales que no tienen líneas de productos registradas actualmente.

```
public class CupEMart
{
    ...
    /**
     * Elimina las sucursales que no tienen líneas de productos registradas.
     * <b>Post:</b> Todas las sucursales registradas tienen al menos una línea de
     * producto</br>
     */
    public void quitarSucursalesSinProductos()
    {

    }
}
```