

# Examen de prácticas

## Prácticas de Programación (Ingeniería Informática)

Junio 2006

Se desea hacer parte del programa de gestión del inventario de los artículos que se venden en una tienda. Para ello, se dispone de la siguiente clase ya realizada, cuya interfaz es:

```
/**
 * Clase que representa un artículo que se vende en una tienda
 */
public class Articulo
{
    /**
     * Constructor al que se le pasan los datos del artículo
     */
    public Articulo(String nombre, int cantidad, double precioUnidad) {...}

    /**
     * Retorna el nombre
     */
    public String nombre() {...}

    /**
     * Retorna el número de unidades disponibles
     */
    public int cantidad() {...}

    /**
     * Retorna el precio de una unidad
     */
    public double precioUnidad() {...}

    /**
     * Vende num artículos (restándolos de la cantidad disponible), si los
     * hay. En caso contrario, no modifica la cantidad y
     * lanza NoDisponible.
     */
    public void vende(int num) throws NoDisponible {...}
}
```

Y se dispone de una clase, parcialmente implementada, que almacena la lista de artículos y que responde a la siguiente interfaz

```
/**
 * Clase que contiene una lista con los artículos en venta en una tienda
 */
public class Tienda
{
    /**
     * Constructor al que se le pasa el nombre de la tienda
     * y que deja la lista vacía
     */
    public Tienda(String nombre) {...}

    /**
     * Añade un artículo a la lista
     */
    public void inserta (Articulo a) {...}

    /**
     * Busca y retorna el artículo cuyo nombre se indica
     * Lanza NoExiste si no se encuentra
     */
    public Articulo busca (String nombre) throws NoExiste {...}
}
```

```

/**
 * Vende num unidades del artículo del nombre indicado. Para ello busca
 * el artículo en la lista y usa su método vende. Si no encuentra
 * el artículo lanza NoExiste. Si la cantidad queda a cero o si se ha
 * lanzado NoDisponible, elimina el artículo de la lista. Retorna el
 * valor total de las unidades vendidas.
 */
public double vende (String nombre, int num) throws NoExiste {...}

/**
 * Escribe en el fichero de texto cuyo nombre se indica
 * aquellos artículos de la lista de los que queden menos
 * de min unidades. Se escriben uno por línea, cada uno
 * con sus tres datos. En la primera línea se escribirá el
 * nombre de la tienda
 */
public void escribePedido(String nombreFichero, int min) {...}
}

```

La clase tiene como atributos privados una lista de artículos (implementada mediante la clase `LinkedList`), y el nombre de la tienda (texto). Las excepciones están declaradas en clases aparte en la forma:

```

public class NoDisponible extends Exception {}
public class NoExiste extends Exception {}

```

El código de las clases descritas puede descargarse de la página Web de la asignatura.

Lo que se pide es

- 1) Añadir a la clase `Tienda` el método estático `leeDeFichero()` al que se le pasa como parámetro el nombre de un fichero de texto y retorna un objeto de la clase `Tienda` con los artículos leídos del fichero. Antes de meter en la lista cada artículo leído, comprobar si está en la lista, y en ese caso no meterlo. El formato del fichero es el mismo utilizado por `escribePedido`. Si el fichero no existe, se pide por teclado el nombre de la tienda y se retorna una tienda con ese nombre y ningún artículo. Si se detecta algún fallo de formato de los datos del fichero, se lanzará `FormatoIncorrecto`.
- 2) Escribir un programa principal con un menú que permita gestionar la tienda con las opciones de
  - Insertar un artículo nuevo, cuyos datos se leen de teclado. Evitar meter artículos de nombre repetido.
  - Mostrar los datos de un artículo cuyo nombre se pide por teclado
  - Vender un cierto número de unidades de un artículo. El nombre y las unidades se piden por teclado. Al acabar se muestran los datos del artículo, o un mensaje diciendo que ese artículo desaparece en caso de que ya se hayan vendido todas sus unidades.
  - Salir del programa.

Al entrar en el programa leer todos los artículos de un fichero de texto. Al salir del programa, escribirlos en ese fichero llamando a `escribePedido` con un mínimo igual al mayor número entero, para que se escriban todos los artículos.