

EXAMEN ENTORNOS DE DESARROLLO

1ª Convocatoria Ordinaria

Segunda Parte

Realizar las siguientes tareas, de forma que para cada una de ellas **quede constancia de que la tarea ha sido realizada**.

1. Se nos ha asignado la tarea de realizar la codificación de una clase que formará parte de un software de gestión para un fabricante de productos lácteos. Para ellos debemos realizar las siguientes tareas:
 - 1.1. Crear un nuevo proyecto con el nombre "Lacteos".
 - 1.2. Crear una nueva clase llamada "Queso" que contendrá la información referente a la fabricación de un queso. La clase contendrá las siguientes variables:
 - *tipo_leche*: indicará el tipo de leche con que se ha fabricado el queso.
 - *cantidad_leche*: indicará la cantidad, en centrilitros, utilizada para su fabricación.
 - *peso*: peso en gramos del queso

El código para la clase será el siguiente:

```
public class Queso {  
    String tipo_leche;  
    int cantidad_leche;  
    double peso;  
}
```

- 1.3. Crear dos métodos constructores (uno sin parámetros y otro con todos los parámetros), además de los métodos getter & setter para todas las variables de clase. Para esta tarea se debe usar la generación de código automática.
- 1.4. Crear un método llamado *imprimir* que muestre los datos de un objeto de esta clase. El código fuente para este método será el siguiente:

```
public void imprimir () {  
    System.out.println("QUESERÍA ARTESANA TALAVERA DE LA REINA");  
    System.out.println("Registro Sanitario Nº 48/38751");  
    System.out.println("Para consultar el lote del producto revise la etiqueta");  
    System.out.println("Peso: " + this.peso);  
    System.out.println("Tipo de leche: " + this.tipo_leche);  
}
```

EXAMEN ENTORNOS DE DESARROLLO

1ª Convocatoria Ordinaria

Segunda Parte

```
        System.out.println("Cantidad de leche: " + this.cantidad_leche);  
    }
```

- 1.5. Generar los comentarios Javadoc para nuestro proyecto, incluyendo comentarios para la clase y todos sus métodos. Comprobar mediante la herramienta "Analyze Javadoc" que no tenemos ningún método sin documentar y ningún error en la documentación.
 - 1.6. Una vez comprobado que nuestra documentación está preparada (sin errores en la salida del Analyzer), generar la documentación Javadoc para el proyecto.
 - 1.7. Inicializar el repositorio Git para nuestro proyecto y generar una primera versión con el comentario "v1 – Creación clase queso". Aseguraos que se incluyen todos los ficheros del proyecto (hay que posicionarse correctamente para llamar a la inicialización del repositorio).
 - 1.8. Cambiar el nombre a la clase *Queso*, asignando el nuevo nombre *Producto*. El cambio de nombre debe aplicarse también a los comentarios de la clase, si los hubiese.
 - 1.9. Extraer un método con las líneas que imprimen los datos del objeto *Queso* (es decir, las tres últimas líneas del método). El nuevo método se llamará *imprimir_detalle*.
 - 1.10. Cambiar el nombre al método imprimir de la clase *Queso* y asignarle el nuevo nombre *imprimir_cabecera*.
 - 1.11. Cambiar el nombre a la variable *tipo_leche* para que pase a llamarse *tipo*, de forma que el cambio aplique también a los métodos creados.
 - 1.12. Encapsular todas las variables de clase de la clases *Queso*.
 - 1.13. Generar una segunda versión con el comentario "v2 – Primera Entrega".
 - 1.14. Mostrar las diferencias de los ficheros entre las versiones v1 y v2.
 - 1.15. Comprobar en el browser de repositorios de Git que la nueva versión ha sido creada.
2. Generar un **diagrama de clases** UML en Visual Paradigm según la siguiente descripción:

A un taller mecánico los clientes llevan sus vehículos para ser reparados. La primera vez que un cliente acude al taller se registrarán sus datos (DNI, nombre, apellidos y teléfono) y se le asignará un número de cliente del taller.

El taller realiza reparaciones de dos tipos de vehículos: comerciales (coches, motos,

EXAMEN ENTORNOS DE DESARROLLO

1ª Convocatoria Ordinaria

Segunda Parte

furgonetas, etc.) e industriales (camiones, tractores, etc.).

Cuando un vehículo acude por primera vez al taller se registrará su marca, el modelo y la matrícula. Para los vehículos comerciales se registrará además el tipo de combustible (diésel, gasolina, híbrido, eléctrico, etc) y la cilindrada. Y para el caso de los vehículos industriales se registrará el sector al que se dedican y el peso del vehículo.

Para cada reparación se registrará el número de reparación y la descripción de la avería detectada. Además, para cada reparación se generará una orden de trabajo donde se anotarán las horas de mano de obra, el coste de las piezas reparadas o sustituidas y se asignará a un mecánico (o mecánicos en caso de que participe más de uno) que realiza la reparación.

De cada reparación se obtendrá una factura. Cada factura tendrá un número de factura y contendrá una serie de líneas con el detalle de la reparación realizada (con el coste individual de cada uno) y un total que indicará el coste total de la reparación.

A su vez, cada factura tendrá asociados un número de pagos donde se indicará el número de pago y la cantidad a pagar en cada uno de ellos.

Una vez realizadas las tareas se comprimirá en un único archivo el proyecto completo de Netbeans, el archivo de Visual Paradigm y el documento conteniendo las imágenes y explicaciones correspondientes a cada apartado de forma que quede constancia de la realización de cada uno de ellos, y se subirá a la plataforma para su corrección.

Criterios de evaluación:

Aptdo.	Descripción	Puntos
1.1, 1.2	Creación del proyecto y la clase	0,25
1.3	Creación métodos constructores	0,5
1.4	Método imprimir	0,25
1.5	Creación comentarios Javadoc	0,5
1.6	Generación documentación Javadoc	0,5
1.7	Inicializar repositorio GIT y generar versión	0,5
1.8	Cambio nombre clase	0,25
1.9, 1.10	Extracción método y cambio de nombre del método	0,25
1.11	Cambio de nombre variable	0,25
1.12	Encapsular variables	0,5
1.13	Generar segunda versión	0,5

EXAMEN ENTORNOS DE DESARROLLO

1ª Convocatoria Ordinaria

Segunda Parte

1.14	Mostrar diferencias entre versiones v1 y v2	0,5
1.15	Comprobar que las versiones han sido creadas	0,25
2	Creación diagrama de clases	4