

# Cooperativa Empresarial

Práctica de la Asignatura de Programación Orientada a Objetos  
Escenario para el Curso 2022/2023 – Marzo de 2023 – Versión 1.4

Departamento de Lenguajes y Sistemas Informáticos  
Escuela Técnica Superior de Ingeniería Informática – UNED

## 1.- Introducción

Los objetivos que se plantean en la realización de esta práctica son los siguientes:

- Conseguir que el alumnado se familiarice con la Programación Orientada a Objetos (POO). En concreto, se tratarán aspectos como la definición de clases e instancias, uso de la herencia, definición/uso de métodos estáticos y abstractos y otros.
- Desarrollar un diseño orientado a objetos que plantee una solución al problema que se propondrá.
- Implementar, una vez realizados los pasos anteriores, una aplicación simple en la que se manejen los conceptos relacionados con POO.

La práctica se llevará a cabo mediante el empleo de la plataforma Java SE. El compilador de Java que se usará será BlueJ, tal y como se establece en el programa de la asignatura. Aunque en el espacio virtual de Ágora (OpenLMS) se pueda tratar el tema de la práctica en alguno de los foros existentes (como otro tema más de la asignatura) es importante recalcar que las dudas respecto al desarrollo de la misma deberán tratarse directamente con el profesor – tutor del centro asociado donde el alumno se encuentre matriculado.

## 2.- Programación Orientada a Objetos

El paradigma de programación orientada a objetos (en adelante, POO) define un programa como una colección de entidades que se relacionan para resolver un problema. Estas entidades, que se conocen genéricamente como objetos, están definidos por un conjunto de propiedades y métodos, y están organizadas en torno a una jerarquía de clases. Así, una clase define la existencia de uno o varios objetos que poseen las propiedad y métodos que se establecieron en su definición.

En Java cada objeto puede tener atributos y/o métodos privados y/o públicos. Se puede modificar dicha visibilidad usando modificadores de acceso a miembros. Las dos maneras más habituales de especificar la accesibilidad son:

`private` – la variable o método está disponible solamente para esta clase,

`public` – la variable o método está disponible para todas las clases,

Una clase puede heredar los atributos y/o métodos públicos de otra clase a través del mecanismo de herencia y la palabra clave `extends`. Por ejemplo:

```
//clase base que va a contener información sobre vehículos de nuestra empresa:  
public vehiculo {  
    private int noPuertas;
```

```

        private int noRuedas;
        private String modelo;
        public vehiculo(){}
        public void setNoPuertas(int np) {
            noPuertas = np;
        }
        //etc.
    }

    //una clase para tratar a los coches en general...
    public coche extends vehiculo {
        private boolean airbags;
        public coche(){}
        public void setAirbags(Boolean a) {
            airbags = a;
        }
        //etc.
    }

    //y, por fin, una clase para tratar a los coches deportivos
    public final cocheDeportivo extends vehiculo {
        private String capacidadMotor;
        private int maxVelocidad;
        public cocheDeportivo(){}
        public void setCapacidadMotor(String cm) {
            capacidadMotor = cm;
        }
        //etc.

        //se puede llamar a cualquier método en las superclases como
        //si estuvieran dentro
        //de esta misma clase, p.ej.:
        //setNoPuertas(2);
    }

```

**Nota:** Las clases que extienden otras clases tienen el nombre de subclases y las clases que son extendidas por otras clases tienen el nombre de superclases.

Hay que tener cuidado a la hora de planificar las relaciones de herencia entre clases en Java porque una clase solamente puede heredar atributos y métodos de otra (y sus superclases). Es decir, que no hay herencia múltiple en Java como hay en lenguajes como C++ (aunque se puede reproducir la técnica de herencia múltiple usando interfaces). De todas formas, la manera más habitual para tratar está tema es simplemente usar una clase dentro de otra. Por ejemplo, si hay una clase para el aparcamiento de una empresa que ya es una extensión de una clase base aparcamiento, dicha clase no puede heredar ninguna otra clase, por lo tanto, se incluirán las clases de coches, camiones, motos, etc., así:

```

public aparcamientoEmpresa extends aparcamiento {
    private String nombreEmpresa;
    private cocheDirector = new cocheDeportivo(...);
    public aparcamientoEmpresa(){}
    //etc.

    //para llamar a algún método en una clase hay que especificar
    //la variable de la instancia...
    //cocheDirector.setCapacidadMotor("4.5l");
}

```

### **3.- Escenario de la Práctica – Curso 2022 / 2023**

En el mundo rural, es bastante frecuente encontrar explotaciones agrarias familiares, de extensiones variables y de productos de los más diversos: desde aceitunas hasta uvas pasando por pistachos, naranjos o melones. En un mundo globalizado como en el que nos encontramos, encontrar una unión entre estos pequeños productores es factor clave a la hora de alcanzar el éxito en lo que a distribución y ventas se refiere. Aquí es donde entran en juego las cooperativas, que permite la colaboración entre agricultores favoreciendo los intereses de los mismos.

Se pretende, por tanto, facilitar la gestión de una cooperativa agrícola mediante el desarrollo de una aplicación que contemple la realidad del entorno en el que se mueve. Para ello, se va a mostrar a continuación las diferentes casuísticas a tener en cuenta, los requisitos que deben cumplirse, así como las restricciones que se plantean, junto con las funcionalidades que deben implementarse.

En lo que se refiere a los actores que participan del funcionamiento de la aplicación, tenemos los siguientes:

#### **Productores**

Desde un punto de vista estricto, los productores van a ser cualquier actor que proporciona a la empresa un determinado producto, y que cobrará por el mismo el precio que se estipule. Dentro de los productores, podemos encontrar dos tipos: pequeños productores, que son aquellos que tienen una extensión inferior o igual al límite en hectáreas que se determine en cada año fiscal (por ejemplo, 5 hectáreas, aunque cada año ese valor puede cambiar) y los grandes productores, que son aquellos que superan esa cantidad.

Los productores proporcionan a la cooperativa una diversidad de productos. Esta cantidad de productos está relacionada con el hecho de ser pequeño productor (pueden como máximo cinco productos diferentes) o gran productor (pueden proporcionar un número no limitado de productos).

Existen también lo que se consideran “productores federados”, que son aquellos productores que por la extensión total no superan el límite de los pequeños productores, pero que uno de sus productos lo “cede” a una federación de pequeños productores que se dedica a un producto determinado, de modo que esa agrupación de *n* pequeños productores son también (en su conjunto) un productor federado.

Para ejemplificar y aclarar la situación, veamos el siguiente caso concreto:

Juan P. tiene 1,5 hectáreas (en adelante, ha) de naranjos, 0,5 ha de algodón y 1,5 ha de melocotoneros. Como la suma total de extensiones (3,5 ha) no supera el límite establecido para el presente ejercicio (que se ha fijado en 5 ha), se considera que Juan P. es un pequeño productor. Sonia R. tiene 2,6 ha de ciruelos, 1,3 ha de trigo y 0,2 ha de algodón. Como en total la suma total de extensiones es de 4,1 ha, se considera igualmente que es una pequeña productora.

Juan P. ha decidido que sus naranjos y sus melocotoneros les interesa venderlos a la cooperativa como pequeño productor, pero forma junto con Sonia R. una federación de productores cediendo a esta unión sus 0,5 ha de algodón, que se unirán a las 0,2 ha de algodón de Sonia R., la cuál ha decidido igualmente que sus ciruelos y su trigo los venderá a la cooperativa como pequeña productora.

Así, se puede considerar que en el sistema hay tres productores: Juan P. (pequeño productor), Sonia R. (pequeña productora) y el productor federado resultante de la unión de las extensiones de

algodón de Juan P. y de Sonia R. No puede haber más de un productor federados por cada producto (es decir, no puede haber además de Juan P. y de Sonia R., que están federados con el algodón, otro productor federado con algodón). El número total de pequeños productores que forma un productor federado no está limitado, si bien se exige que la suma total de extensiones no supere el límite de extensión existente para ser considerado pequeño productor (es decir, no hay un productor federado sobre un producto que, en el ejemplo que estamos tratando, tenga una extensión superior a 5 ha)

Por otro lado, cada producto (por ejemplo, algodón) tiene lo que se denomina un “rendimiento por hectárea”, en el que se indica cuántas toneladas se obtienen por hectárea de producto (por ejemplo, 2,5 toneladas de algodón por cada hectárea). Además, cada producto tiene un valor de referencia por kilogramo libre de impuestos (sin IVA). Por ejemplo, el precio del algodón podría fijarse en torno a los 80 céntimos de euro por kilogramo.

**Nota:** una tonelada son 1000 kg La forma de implementar las “tablas” de correspondencia entre productos y rendimiento y productos y precio de referencia libre de impuestos se deja a criterio del alumno.

### Logística

Este grupo de actores tienen como función principal la de proporcionar a los clientes finales y distribuidores los productos disponibles en la cooperativa. La logística se divide principalmente en dos grandes grupos: productos perecederos y productos no perecederos. Así, los productores de productos perecederos requieren de una logística más “costosa” (debe mantenerse unas condiciones de transporte determinadas) y el coste en sí depende tanto del producto como de la distancia. Esto se debe a que si la distancia de transporte del producto es inferior a 100km, entonces la propia empresa de logística realiza todo el proceso. Si la distancia es superior a los 100km, entonces hay que emplear una doble logística: una “gran logística” que implica el envío del producto a la capital de la provincia donde se va a surtir el producto, y después una “pequeña logística” realiza el reparto desde la capital de la provincia hasta el destino final (se asegura que esta pequeña logística siempre realiza trayectos inferiores a los 100 km).

Respecto a los productos no perecederos, la filosofía es la misma que en el caso anterior con la particularidad de que la “gran logística” requiere la contratación de diferentes tramos múltiples de 50 km. Es decir, enviar un producto a 140 kilómetros de distancia de la cooperativa implicará la contratación de dos tramos de 50 km de “gran logística” y un tramo final de distribución de 40 km de “pequeña logística”.

El coste de la gran logística está relacionado con el coste del producto, número de trayectos y kilómetros. Así, cada trayecto tiene un precio fijo de  $0,5 * \text{Precio de Referencia del Producto Libre de Impuestos} * \text{Kilogramos contratados}$ . Cada kilómetro se pagará a precio fijo. Como máximo cada trayecto contratado puede gestionar un máximo de 1 Tonelada de producto. El coste de la pequeña logística va relacionado simplemente con el kilometraje recorrido.

Es importante tener en cuenta (por las implicaciones en lo que al diseño se refiere) que desde el punto de vista de la logística, pueden existir (y de hecho, deben existir) diferentes ofertas para poder surtir los productos tanto a los distribuidores como a los consumidores finales (lo que se traducirá en costes diferentes en la logística).

**Nota:** La forma de implementar las “tablas” de kilometrajes desde la cooperativa hasta los diferentes puntos de referencia en cada provincia y los precios del kilometraje se deja a criterio del alumno.

### **Distribuidores y Consumidores Finales**

Los últimos actores son los distribuidores y consumidores finales. Los primeros se dedican a comprar grandes cantidades para posteriormente ponerlos a la venta a unos clientes finales. Los segundos adquieren productos a la cooperativa directamente.

Desde el punto de vista de los distribuidores, excluyendo la logística, la cooperativa vende sus productos añadiendo un 5% al precio de referencia libre de impuestos del producto. Un distribuidor está obligado a comprar al menos 1 tonelada del producto que sea, y deberá pagar la logística de la forma en la que se ha indicado en el apartado anterior. Así, una distribuidora que compra 2 toneladas de aceite (no perecedero) a la cooperativa, que se encuentra a una distancia de 180 km, tendría que pagar lo siguiente:

- Para una tonelada, tendría que abonar  $0,5 \times 0,50 \text{ €/kg}$  (precio libre de impuestos del aceite)  $\times 1000 \text{ kg} = 250 \text{ €}$ . Esto sería el precio por cada trayecto y tonelada.
- Como hay que enviarlo a 180 km, implicaría tres trayectos (de 50 km cada uno) en “gran logística” y 30 km en “pequeña logística”. Luego  $250 \text{ €} \times 3 = 750 \text{ €}$  por cada tonelada en “gran logística”. Si el precio del kilómetro se fija en  $0,05 \text{ €}$  por kilómetro, el precio sería  $750 \text{ €} + 150 \text{ km} \times 0,05 \text{ €/km} = 750 \text{ €} + 7,5 \text{ €} = 757,5 \text{ €}$ . Esto sería el coste de la “gran logística” por cada tonelada. Como la compra son 2 toneladas, el coste total sería de  $1515 \text{ €}$ .
- A esto habría que añadir los 30 km de “pequeña logística”. Aquí el coste sería de  $0,01 \text{ €}$  por kilogramo y km. Luego como son 2 toneladas y 30 km, el coste a sumar sería  $2000 \text{ kg} \times 30 \text{ km} \times 0,01 = 600 \text{ €}$ .
- Por lo tanto, la distribuidora pagaría, por la logística,  $1515 + 600 = 2115 \text{ €}$  por las dos toneladas de aceite.
- El coste del producto en sí al ser adquirido a la cooperativa es de  $2000 \text{ kg} \times 0,5 \text{ €/kg}$  (precio libre de impuestos del aceite)  $\times 1,05$  (el beneficio de la cooperativa)  $= 1050 \text{ €}$ .
- Por lo tanto, sumando coste del producto y de la logística, se obtiene un total de  $1050 \text{ €} + 2115 \text{ €} = 3165 \text{ €}$ , lo que supondría un coste por kilo de  $1,5825 \text{ €}$ .

**Nota:** La forma de implementar las “tablas” de precios fijos por kilómetros en “gran logística” y en “pequeña logística” se deja a criterio del alumno.

Desde el punto de vista de los consumidores finales, excluyendo la logística, la cooperativa vende sus productos añadiendo un 15% al precio de referencia libre de impuestos del producto. Un consumidor final no puede adquirir más de 100 kg de producto, y deberá pagar la logística de la misma forma en la que lo hacen las distribuidoras, añadiendo el correspondiente IVA (que por simplicidad, supondremos siempre que es el 10%). El IVA se aplica tanto al producto en sí (que vende la cooperativa) como al precio de la logística. Así, si un cliente compra 50 kg de aceituna y su precio de referencia libre de impuestos es  $0,60 \text{ €/kg}$  (por poner un ejemplo), en verdad el cliente final paga a la cooperativa  $50 \times (1,15) \times 0,60 \text{ €} = 34,5 \text{ €}$ . A este importe habría que añadir la logística. Si (por poner un ejemplo) el coste de la logística fuese de  $80 \text{ €}$ , al monto total ( $34,5 \text{ €} + 80 \text{ €} = 114,5 \text{ €}$ ) le aplicamos el porcentaje de IVA:  $114,5 \text{ €} \times 1,10 = 125,95 \text{ €}$  sería la cantidad total a abonar por el cliente final.

**Nota:** La forma de implementar las “tablas” de precios fijos por kilómetros en “gran logística” y en “pequeña logística” se deja a criterio del alumno.

Llegados a este punto, se quiere desarrollar una aplicación que permita gestionar de manera informática toda la gestión planteada de la cooperativa, aplicando un paradigma de Orientación a Objetos. No se va a querer desarrollar la aplicación en su conjunto, sino que se va a solicitar la realización de unas determinadas funcionalidades del total de las mismas posibles. Será necesario que cada una de ellas se realice de manera independiente en tantos proyectos de BlueJ como sea necesario, pudiendo utilizarse código ya desarrollado en unas funcionalidades en la implementación de otras. Por lo tanto, se va a solicitar realizar las siguientes tareas:

1) **(3,0 puntos)**. Plantee, aplicando el paradigma de Programación Orientada a Objetos una jerarquía de clases que permita modelar, de la manera más genérica posible y permitiendo la mayor variabilidad admisible dentro de las restricciones impuestas, la gestión de la cooperativa, incluyendo los aspectos tratados en dicho paradigma como son herencia, abstracción, polimorfismo y encapsulamiento entre otros. Desde el punto de vista de la logística, pueden existir (y de hecho, deben existir) diferentes ofertas para poder surtir los productos tanto a los distribuidores como a los consumidores finales (esto se traducirá en costes diferentes en la logística). Toda la casuística que se ha enunciado en el documento de descripción ha de implementarse necesariamente, quedando a criterio del alumno cuantas decisiones puedan ser necesarias y que en modo alguno contradigan lo ya especificado ni impongan algún tipo de restricción a los requisitos expuestos.

2) **(3,0 puntos)**. Toda vez que se ha modelado la casuística de la cooperativa, se quiere generar el conjunto de estructuras (colecciones) y objetos que sean necesarios de modo que se cumplan las siguientes condiciones específica (se proporcionará también el código que crea estas estructuras):

- Las compras a la cooperativa se realizan en una determinada fecha y se sirven dentro de un plazo máximo de diez días. Si al realizar la petición, se solicita que se entregue en un plazo superior a diez días, entonces habrá que revisar el valor del producto en el momento de proporcionarlo (siempre diez días antes de la fecha de entrega solicitada).
- Los precios de los productos se revisan semanalmente, y pueden experimentar subidas o bajadas.
- Cada pedido lleva asociado un número de pedido.
- Se lleva un registro de pedidos por cada distribuidor y/o consumidor final.
- El importe que recibe cada productor (sea del tipo que sea) será proporcional a lo que se genera en cada pedido. Así, si se realiza un pedido de algodón de 300 kg, y hay tres productores de 1 ha, 3 ha y 0,5 ha, cada uno de los productores recibe la parte proporcional a lo que aportan (es decir, sería como si le hubiese comprado 66,67 kg al primero, 200 kg al segundo y 33,33 kg al tercero).
- No se puede vender más cantidad de producto del existente en la cooperativa.
- Importante tener en cuenta que los productores reciben el importe establecido como precio de referencia libre de impuestos (que es diferente al precio que el distribuidor o consumidor final compra, puesto que hay un porcentaje adicional que se añade y que revierte en la cooperativa).

3) **(4,0 puntos)**. Se quiere generar, a final de año, un resumen anual de totales de gestión de la cooperativa, proporcionando además datos estadísticos de los mismos. Se quiere que se diseñe una clase, con sus correspondientes objetos, que permita recoger esta información y que permita generar las estadísticas que se consideren oportunas, si bien se espera que al menos proporcione:

- Ventas totales en un periodo determinado de cada uno de los productos de la cooperativa.
- Importes obtenidos por cada uno de los productores (desglosados por productos)
- Importes obtenidos por cada una de las empresas de logística.
- Beneficios de la cooperativa por cada uno de los productos.
- Evolución de los precios de referencias de cada productos.

## **4.- Plan de Trabajo**

Para realizar la práctica se seguirá el siguiente método de trabajo:

- En primer lugar se leerá detenidamente el enunciado de esta práctica.
- A continuación hay que diseñar, utilizando un paradigma orientado a objetos, los elementos necesarios para cada nivel de la aplicación explicada en el apartado anterior. Debe hacerse uso de los mecanismos de herencia siempre que sea posible. Se valorará un buen diseño que favorezca la reutilización de código y facilite su mantenimiento.
- El código estará debidamente comentado.
- La clase principal que lleve método `main()` de ejecución deberá llamarse "cooperativa.class".

## **5.- Control de Plagio en las Prácticas**

Tal y como está indicado en el Apartado 9 de este enunciado, las prácticas son esenciales en las titulaciones de Informática porque permiten a los alumnos adquirir conocimientos importantes sobre los aspectos más aplicados de una asignatura. Por lo tanto, dado el hecho de que la práctica de esta asignatura es un trabajo individual y obligatorio que cuenta para la nota final de la asignatura y que implica un esfuerzo por parte de los alumnos, es necesario garantizar la originalidad de dicho trabajo.

Para evitar este problema, una vez terminado el plazo de entrega de la práctica (indicado en el curso virtual), el equipo docente usará un software de control de plagio para revisar las prácticas. En los casos donde haya plagio se informará al Servicio de Inspección de la UNED para que tome las medidas disciplinarias apropiadas.

## **6.- Normas de Realización de la Práctica**

1. La realización de la práctica es obligatoria. **Sólo se evaluará el examen si la práctica ha sido previamente aprobada.**
2. Aunque, si bien el desarrollo de aplicaciones orientadas a objetos usando el lenguaje de programación Java no requiere el uso concreto de ningún entorno de desarrollo, **esta práctica ha de desarrollarse íntegramente empleando el entorno de desarrollo BlueJ**, que es el que se muestra en el libro de texto básico de la asignatura.
3. **La práctica es un trabajo individual.** Las prácticas cuyo código coincida total o parcialmente con el de otro alumno serán motivo de suspenso para todos los implicados (copiadores y copiados) además de cualquiera medida disciplinaria que pudiera aplicar el Servicio de Inspección.
4. Cada tutor organizará, al menos, una sesión de control de la realización de la práctica:
  - a. **Es el tutor el que marca la fecha de dicha sesión o sesiones y no el equipo docente.** Los tutores enviarán posteriormente listados de los alumnos que han asistido a dicha sesión.
  - b. **La asistencia a dicha sesión (o una de ellas, si son varias) es obligatoria y se debería realizar antes de la entrega de la práctica en la plataforma.**
  - c. El tutor puede organizar la sesión hacia el final del cuatrimestre para poder comprobar que los alumnos han hecho bien el trabajo y para ayudar al tutor a calificar el trabajo.
  - d. El tutor entrará en el espacio virtual de la asignatura antes del 1 de junio, para meter las notas para sus alumnos. **En ningún caso esto significa que el plazo de entrega de la práctica será el 1 de Junio.** Esta fecha es fijada exclusivamente por cada tutor en cada centro asociado y será



anterior al desarrollo de las pruebas presenciales (primera y segunda semana). Por lo tanto, aún cuando el tutor es el único responsable de fijar esta fecha, **no podrá establecerse como fecha de entrega de la práctica obligatoria cualquier fecha posterior al 21 de Mayo de 2023.**

- e. En el caso de que un alumno no pueda asistir a la sesión de control debido a una causa mayor (p.ej., por motivos médicos o de trabajo), deberá informar al equipo docente (adjuntando un certificado médico o carta de su empresa) antes del 21 de Mayo de 2023. Una vez empezado el periodo de exámenes no se aceptarán estos avisos.
- f. Con vistas a la convocatoria de Septiembre, cada tutor habilitará un plazo extraordinario de entrega de las mismas. Como en el caso anterior, es responsabilidad exclusiva de cada tutor el establecimiento de dicha fecha y la comunicación a sus alumnos. Es importante además tener en cuenta que, salvo que el tutor específicamente así lo indique y exclusivamente a criterio suyo, **no se establecerán sesiones de control extraordinarias con vista a la convocatoria de Septiembre.** Por lo tanto, inexcusablemente, los alumnos, **con independencia de la convocatoria a la que vayan a presentar la práctica, deberán acudir a la sesión de control establecida en periodo ordinario** (es decir, entre Febrero y Mayo).
- g. Puede ocurrir, que por motivos organizativos del centro asociado, **algunos tutores puede que no lleven a cabo correcciones de prácticas con vistas a la convocatoria de Septiembre.** En este caso (y solo para los alumnos cuyo tutor no corregirá prácticas en Septiembre), se indicará pertinentemente la forma de proceder y cómo enviar la práctica para esa convocatoria en el curso virtual.

- 5. La única vía de entrega de la práctica es a través de la plataforma de los cursos virtuales siguiendo las indicaciones del Apartado 7.
- 6. El equipo docente podrá tener en cuenta prácticas con notas altas para aquellos alumnos cuyo examen esté cercano al aprobado.
- 7. El alumno debería dirigirse a su tutor para cualquier duda que tenga sobre su práctica y solamente al equipo docente (por correo electrónico) en el caso de que su tutor no pueda resolver su problema. En este caso, pediremos al alumno que, además de sus datos personales, nos envíe el nombre del centro asociado en el que está matriculado y el de su tutor.
- 8. Evidentemente se pueden usar los foros para realizar consultas a los compañeros pero nunca para intercambiar código.

## **7.- La entrega de la Práctica**

La práctica se entrega a través de la plataforma de cursos virtuales en el apartado “Entrega de trabajos”. El archivo que hay que subir debe ser un archivo comprimido (rar o zip), que se puede preparar con el software de compresión que traen la mayoría de los sistemas operativos hoy en día o usando un software libre como 7-Zip (<https://www.7-zip.org/>). No se deben usar acentos en los nombres de los archivos ni las carpetas. El archivo comprimido debe estar compuesto por una carpeta con el nombre del alumno que contiene dos cosas:

- 1. Memoria: La memoria constará de los siguientes apartados:
  - a. Portada con título “Práctica de Programación Orientada a Objetos – Curso 2022 / 2023” y los datos del alumno: Nombre, Apellidos, dirección de correo electrónico y teléfono de contacto.



- b. Análisis de la aplicación realizada, estrategias implementadas, decisiones de diseño establecidas y, en general, toda aquella información que haga referencia a las diferentes decisiones tomadas a lo largo del desarrollo de la práctica, junto a una justificación de dichas decisiones.
  - c. Diagrama de clases, detallando claramente el tipo de relación entre ellas (uso, agregación, herencia, ...).
  - d. Un texto en el que se describa cada clase/objeto, justificación de su existencia, métodos públicos que contiene y funcionalidad que realizan.
  - e. Anexo con el código fuente de las clases implementadas.
2. Una carpeta con el código: incluyendo todos los ficheros \*.java y \*.class, así como la memoria en formato electrónico (preferiblemente pdf).

#### **NOTAS:**

- Al hacer la entrega del trabajo se acepta que tanto el código fuente Java como la memoria de la práctica es original. Aquellos aportes intelectuales de otros autores (como por ejemplo, el tutor) deben estar referenciados debidamente en el texto de dicho trabajo.
- Si el archivo subido por parte del alumno no sigue estas indicaciones, está infectado con algún virus, o que no se puede descomprimir, el tutor (y por ende, el equipo docente) no aceptará la práctica y se calificará con una nota de 0.
- La no inclusión de memoria alguna de la práctica igualmente acarreará una penalización en la calificación final de la misma, la cuál podría suponer el suspenso de la misma.
- **Se vuelve a insistir en que la fecha de entrega de la práctica depende exclusivamente del tutor asignado para su corrección en el centro asociado en el que el alumno esté matriculado. Es fundamental que el alumno contacte con sus tutor para conocer la fecha de entrega de la misma. Este Equipo Docente no atenderá ningún tipo de reclamación al respecto.**

## **8.- Normas para los Tutores**

Como se puede apreciar, el papel del tutor es fundamental en todos los aspectos de la práctica, tanto el planteamiento del problema, el diseño orientado a objetos del programa, su desarrollo y su depuración.

Tratándose de una asignatura obligatoria, cada alumno debería tener acceso a un tutor. Los tutores deben seguir los siguientes pasos:

1. Ayudar a los alumnos al principio del curso con el planteamiento de la práctica y las normas que tienen que seguir.
2. Para explicar ciertos conceptos relacionados con la solución de la práctica, el tutor puede dar fragmentos de código fuente a los alumnos. Los pequeños fragmentos no tendrán importancia a la hora de llevar a cabo el control de plagio por parte del equipo docente. No obstante, si un alumno va a incluir un fragmento de código en su práctica, debe incluir un comentario al respecto directamente anterior al código y también una nota al respecto en su memoria.
3. Indicar a los alumnos que habrá una sesión obligatoria de seguimiento y evaluación de la práctica.
4. Una vez terminada y entregada la práctica, el tutor debe entrar en el espacio virtual de la asignatura, antes del 1 de junio, para introducir las notas de sus estudiantes.
5. Comunicar la calificación a sus alumnos.

## **9.- Centros Asociados vs. Prácticas en Asignaturas Obligatorias**

Las prácticas son esenciales en las titulaciones de Informática porque, entre otras cosas, permiten a los alumnos adquirir conocimientos importantes sobre los aspectos más aplicados de ciertas asignaturas, lo cual resulta de gran relevancia e interés a la hora de acceder a un puesto laboral relacionado con la Informática. Para orientar y ayudar a los alumnos, así como para comprobar que realmente un alumno ha realizado su práctica de forma satisfactoria, ésta se debe realizar en un Centro Asociado bajo la supervisión de un tutor, quien decide, en última instancia, la forma en la cual se organiza el desarrollo de la misma en su Centro Asociado (existencia o no de sesiones presenciales obligatorias, forma de entrega, etc.)

De vez en cuando sucede que un alumno se pone en contacto con un Equipo Docente del Departamento de Lenguajes y Sistemas Informáticos (L.S.I.) porque se ha matriculado en una asignatura obligatoria en un Centro Asociado que no le proporciona un tutor para supervisar la práctica, aún cuando se le ha permitido matricularse. El alumno busca en el Equipo Docente que se le proporcione una solución a este problema, como por ejemplo, la posibilidad de asistir a unas sesiones extraordinarias de prácticas en la Sede Central de la U.N.E.D. en Madrid o la posibilidad de realizar la práctica por su cuenta en casa, enviándola a continuación al Equipo Docente para su corrección. Sin embargo, los Equipos Docentes de L.S.I. no disponen de recursos para poder llevar a cabo ninguna de estas dos alternativas.

Un Centro Asociado que ha permitido a un alumno matricularse en una asignatura obligatoria de una carrera de Informática debería ayudarle a encontrar una solución al problema de la realización de las prácticas. Si se trata de una asignatura donde no se han matriculado muchos alumnos, quizás el centro no cuente con recursos para proporcionar un tutor específicamente para la asignatura. Si hay otro Centro Asociado cerca que dispone de tutor, quizás el alumno pueda realizar la práctica allí. Pero si no es así, el Centro Asociado debería proporcionar un tutor para supervisar y corregir las prácticas de sus alumnos. Lo más razonable sería que fuera un tutor de otra asignatura de Informática en el mismo Centro el que hiciera la sesión de prácticas para los alumnos de la asignatura en cuestión, y al final de la sesión evaluara los trabajos de los alumnos, según las pautas marcadas por el Equipo Docente, haciendo llegar a éste las calificaciones otorgadas.

Por lo tanto, cualquier alumno o alumna que tras haberse matriculado en una asignatura obligatoria en un Centro Asociado, se encuentre con que el centro no tiene tutor para dicha asignatura, debería dirigirse a la Dirección del Centro Asociado, para solicitar de la misma una solución, tal como se ha presentado aquí, es decir, alguien que pueda supervisar y corregir su práctica con plenas garantías. En el caso de que no se le proporcione una solución, la persona afectada debería comunicárselo, por escrito, lo antes posible, a la Directora del Departamento de L.S.I., Dra. Raquel Martínez Unanue.