

La elaboración de este ejercicio está dirigida a través de una secuencia de **10 preguntas**, agrupadas en **siete secciones**, en las que se solicita realizar determinadas operaciones y tareas de diseño, vistas en la asignatura. El objetivo de cada sección es evaluar una porción del aprendizaje, la madurez y las capacidades fundamentales exigidas en el programa. **La puntuación** es sobre un **total de 10**, (más 1 punto de la sección BP), pero **no todas** las cuestiones **puntúan igual** porque las preguntas correspondientes a la elaboración de los Casos de Uso no requieren capacidades tan próximas a los objetivos de la asignatura como la realización de los Diagramas de Colaboración. Puede utilizar la cantidad de papel que necesite, pero conteste a las preguntas de cada sección en hojas diferentes. No se permite ningún material de apoyo.

Enunciado y planteamiento del caso de estudio.

El dominio del problema es un **sistema de gestión de pedidos online en un supermercado** (SuperList).

El sistema se encarga de mostrar el ' escaparate ' de existencias, recoger los *Pedidos*, gestionar la preparación de los *Productos*, su *Empaquetado* y *Reparto* y actualizar el *Catalogo* de existencias en el *Almacen*.

Los *usuarios* acceden al supermercado mediante un navegador Web, app en dispositivo móvil u otra tecnología. Podrán ver el *Catalogo* de *Productos*, pero sólo los *usuarios* registrados (*Cientes*), tendrán acceso a realizar y enviar un *Pedido* (*RealizarPedido*). Mediante un 'carrito' virtual, el *Cliente* puede seleccionar *Productos* y cantidades e ir añadiéndolos al carrito. Al final, el sistema mostrará el coste de cada elemento y el precio final. Si el *Cliente* está de acuerdo, tendrá la opción de *Pagar*, en cuyo caso, el sistema recogerá la modalidad y los datos de *Pago*, así como el lugar y la *forma* de *entrega*. Cuando el sistema reciba la confirmación del *Pago*, mostrará un informe al *Cliente* y remitirá dicho *Pedido* a la etapa de gestión que ocupa este estudio. Por tanto, las operaciones anteriores se encapsulan en *GeneraPedido*, iniciada por el *Cliente* y que incluye la selección de *Productos*, *forma* de *entrega* y *Pago*. La consecuencia de *GeneraPedido* es el envío del *Pedido* al sistema, con la siguiente información: *ClienteID*, *ModoEntrega*, *ListaProd* (*RefProd*, *Cantidad*) y *Factura*.

Cuando el sistema recibe un *Pedido*, comprueba las existencias en el *Almacen* y envía órdenes de recogida (*PickupOrder*) a los terminales inalámbricos de los operarios del *Almacen*, según su zona de operación. Depositán la mercancía en un único contenedor y, cuando terminan, envían una notificación al sistema a través de sus terminales. Al concluir la recogida, el contenedor se deposita en la zona de *Empaquetado*. Mientras tanto, el sistema, actualiza las existencias en el *Catalogo* del *Almacen* y, tras consultar a otro sistema externo de optimización en el empaquetado, genera las instrucciones de empaquetado (*PackProced*) y las envía al muelle de la zona de *Empaquetado* en el que está el contenedor, para que el operario encargado realice el procedimiento.

Una vez preparado el *Pedido*, el sistema consulta el *ModoEntrega* y genera un *Albaran* de entrega que adjunta a la *Factura*. Si el *Cliente* ha elegido recoger el *Pedido* 'in situ', se deposita (con la documentación) en la zona de entrega (*Reparto*). En caso contrario, un planificador externo genera una orden de reparto (*DeliverOrder*) que, el sistema, adjunta al resto de documentación y, junto al *Pedido*, se entrega al transportista (*Reparto*) para su reparto.

Las operaciones concluyen cuando el sistema recibe la conformidad del *Cliente* tras la recepción del *Pedido*, *Albaran* y *Factura*.

Los detalles y simplificaciones admitidas son:

- El sistema recibe *Pedidos* directamente del *Cliente*. Es decir, se ignoran la identificación, selección de productos, pago, etc.
- Los *Productos* tienen un identificador único *RefProd* (no se componen de otros *Productos*). La *Cantidad* se refiere al número de unidades, si es un *Producto* empaquetado, o peso o volumen si es a granel.
- No se contempla la posibilidad de que falten existencias de algún producto del *Pedido*.
- El *Empaquetado* recibe los procedimientos desde un optimizador externo al sistema. Cuando concluye, devuelve un listado con los paquetes, sus dimensiones y pesos.
- De igual forma el *Reparto*, en el caso de que requiera transporte, recibe el listado de paquetes anterior, así como una orden de reparto calculada por un planificador externo al sistema, que se encarga de optimizar los viajes en función del tamaño del *Pedido*, urgencia de entrega, los vehículos disponibles y las zonas de reparto.

Sección 1. Evaluación de **Casos de Uso**

1. (0'5 puntos) Identifique al menos 4 casos de uso primarios y sus actores correspondientes. Represente los resultados en un diagrama de casos de uso de UML.
2. (1 punto) Escriba el caso de uso <<*RealizarPedido*>> en un estilo y formato completo, esencial y breve. Incluya tanto el escenario principal de éxito (flujo básico correspondiente a que el sistema reciba un *Pedido* –iniciado por el *Cliente*, generado y registrado con éxito por *GeneraPedido*—, se prepare, empaquete, se entregue y se reciba la conformidad del *Cliente*) como 2 extensiones o flujos alternativos que pudieran ser frecuentes. No escriba un encabezamiento demasiado elaborado del caso de uso (es decir, omita *propósito*, *resumen*...); en su lugar, afronte directamente el transcurso típico de los acontecimientos. ATENCIÓN: límitese a describir la parte indicada de *RealizarPedido*, omitiendo los detalles de la identificación del *Cliente*, la selección de *Productos* y el *Pago*, acciones incluidas en *GeneraPedido*.

Sección 2. Evaluación del **Modelado Conceptual**

3. (2 puntos) En relación al caso de uso anterior <<*RealizarPedido*>>, construya un Modelo de Dominio y represéntelo en notación UML. Represente los objetos conceptuales, las asociaciones y los atributos.

Sección 3. Evaluación de los **Eventos del Sistema**

4. (0'5 puntos) Circunscrito al caso de uso anterior <<RealizarPedido>>, construya un Diagrama de Secuencia del Sistema (DSS) en UML. Represente los actores y los eventos del sistema. A partir de este DSS, especifique los contratos de dos operaciones principales: <<se omite la operación A>> y <<se omite la operación B>>. ATENCIÓN: lo que hay entre corchetes <<se omite>> es un ejemplo, usted lo debe sustituir por su propia operación.

Sección 4. Evaluación de la **Asignación de Responsabilidades y Diseño de Colaboraciones**

5. (2'5 puntos) A partir del contrato de la operación <<se omite la operación A>> que haya indicado en el punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente. ATENCIÓN: lo que hay entre corchetes <<se omite>> es un ejemplo, usted lo debe sustituir por su propia operación.
6. (2'5 puntos) A partir del contrato de la operación <<se omite la operación B>> que haya indicado en el punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente. ATENCIÓN: lo que hay entre corchetes <<se omite>> es un ejemplo, usted lo debe sustituir por su propia operación.

Sección 5. Evaluación de los **Diagramas de Clases** de diseño

7. (0'5 puntos) Elabore un diagrama de clases parcial: centrado en la clase <<Pedido>> y en sus relaciones con <<Cliente>>, <<Almacen>>, <<Empaquetado>>, <<Reparto>> u otras que considere, analizadas en el Modelo de Dominio. Represente los nombres de todos sus atributos, asociaciones (con la navegabilidad) y métodos.

Sección 6. Evaluación de la **Transformación del Diseño en Código**

8. (0'5 puntos) A partir de los anteriores diagramas de clases y colaboraciones, elabore y defina la clase <<Pedido>>. Incluya las definiciones de todas las variables que la componen (miembros), pero escriba solamente la definición completa del cuerpo para el método (función miembro): <<se omite el método>>. Ignore los pequeños detalles de sintaxis -el objetivo es evaluar la capacidad fundamental para transformar el diseño en código-. Utilice la sintaxis de Java. ATENCIÓN: lo que hay entre corchetes <<se omite>> es un ejemplo, usted lo debe sustituir por su propio método.

Sección 7. Preguntas opcionales **BP**. Motivación.

9. (0'5 puntos) Contraste brevemente los términos 'Análisis' y 'Diseño' que ha estudiado en este curso. ¿Qué objetivos quiere cumplir cada actividad?
10. (0'5 puntos) Indique qué patrones ha utilizado en el ejercicio y para qué le han servido.