



Una entidad comercial quiere implementar un programa para gestionar a sus clientes así como poder ofertarles descuentos y productos en función de sus características. Para ello necesita almacenar información sobre los clientes, un identificador de cliente, nombre y apellidos, año de alta como cliente y un vector con sus últimas 20 transacciones con el comercio. Una transacción no es más que una breve información de una operación realizada en el comercio que tiene como datos la fecha de la transacción (de tipo String con formato dd/mm/aa) y el importe (será siempre un valor positivo). Las transacciones, a su vez, se dividen en compras y devoluciones, y serán almacenadas en un único array para cada cliente.

La empresa lleva una clasificación de sus clientes en función de la antigüedad que tiene como cliente y el importe total gastado en la empresa (compras menos devoluciones). La antigüedad de un cliente se calculará consultando al sistema el año actual. Esta clasificación de los clientes se traduce en un descuento que se aplica a cada cliente en todas sus compras según la siguiente tabla:

Antigüedad Importe gastado	<5	>=5 y <10	>=10
<100€	0%	0%	5%
>=100€ y <500€	5%	10%	15%
>=500€	10%	15%	20%

Los clientes deben implementar una interfaz llamada IDescuentos que defina un método descuento. Dicho método devolverá el descuento a aplicar a un determinado cliente según sus características.

El sistema debe almacenar el número de identificador del último cliente añadido ya que este será asignado automáticamente por el sistema y cada vez que se añada un cliente se le asignará el número siguiente al último añadido. Para ello usaremos variables estáticas.

Se pide:

- Diseñar el diagrama de clases para este sistema, agrupando elementos (atributos y métodos) comunes evitando duplicar elementos. Para ello se podrá añadir clases intermedias o abstractas si se considera oportuno. Se deberá entregar la representación de dicho diagrama por escrito.
- Implementar las clases descritas ajustándose al esquema diseñado en el apartado anterior. Se implementarán los métodos que se consideren necesarios y se valorará el correcto uso de los modificadores.
- Implementar un programa que contenga un método *main*. Dicho programa creará un vector de clientes (un único vector para todos ellos) y mostrará un menú con las siguientes opciones, cuya función se describe a continuación de cada una de ellas:
  1. **Alta de cliente:** si se elige esta opción se solicitará al usuario el nombre y apellidos del cliente y el año de alta. El identificador será asignado por el sistema de forma

correlativa tal y como se especifica en el enunciado. Al primer cliente se le asignará el número 1.

2. **Baja de cliente:** se le pedirá al usuario el identificador del cliente y si éste se corresponde con el de algún cliente, se elimina dicho cliente del vector. Se deberá informar al usuario si se ha podido eliminar el cliente o no.
3. **Listado de clientes:** mostrará un listado de todos los clientes. En el listado se mostrará una línea por cliente que mostrará el identificador, nombre y apellidos, año de alta, gasto total realizado, descuento que se le aplica, separado por guiones. Ejemplo:

*1 – Elena García López – 2010 – 625€ – 15%*  
*3 – José Díaz Méndez – 2016 – 56€ – 0%*

4. **Información de cliente:** mostrará la información del cliente así como un listado de sus transacciones (como mucho 20 pero pueden ser menos). Las transacciones mostrarán la fecha, importe. Aunque el importe almacenado en la transacción siempre es positivo, en el listado se mostrará positivo para las compras y negativo para las devoluciones. La salida tendrá el formato del siguiente ejemplo:

*Identificador: 1*  
*Nombre y Apellidos: Elena García López*  
*Año de alta: 2010*  
*Descuento: 15%*  
*Transacciones:*  
*10/04/2018 +1800€*  
*11/04/2018 -1175€*  
*Importe: 625€*

5. **Nueva transacción:** en esta opción se le preguntará al usuario el tipo de transacción (compra o devolución), el identificador de cliente y el importe. La fecha será la fecha actual que se calculará automáticamente. Se añadirá dicha transacción al vector de transacciones del cliente. Si el cliente indicado no existe, no se podrá añadir la transacción y se informará de ello al usuario.
6. **Salir:** el programa finalizará al seleccionar esta opción.

A no ser que se elija la opción salir después de cada acción se volverá a mostrar el menú. Cualquier decisión que se tome por no estar especificada en el enunciado y que no modifique ninguna de las especificaciones deberá ser explicada razonadamente.

Para aprobar el examen será imprescindible que éste no tenga errores de compilación.

#### **NOTAS:**

- No debe limitarse el número de clientes a almacenar en el vector.
- Para calcular la antigüedad de un cliente o la fecha podemos hacer uso de la clase Calendar (java.util.Calendar) que nos proporciona el año actual, el día o el mes. El código a utilizar para obtener el día, el mes o el año sería el siguiente:

```
Calendar cal= Calendar.getInstance();  
int dia = cal.get(Calendar.DAY_OF_MONTH);  
int mes = cal.get(Calendar.MONTH);
```



NOMBRE: \_\_\_\_\_

*int year= cal.get(Calendar.YEAR);*

## DIAGRAMA DE CLASES

Criterios de calificación			
Apartado	Criterio	Max	Puntos
Compilación	<b>Si el programa no compila la nota máxima del examen será de 2 puntos</b>		
Diseño del diagrama de clases	Elección adecuada de las clases añadiendo aquellas necesarias para evitar duplicidad	1,5	
Implementación del diagrama de clases	Correspondencia del diagrama con su implementación	0,25	
	Correcto uso de los modificadores	0,25	
	Uso correcto de clases abstractas e interfaces	0,5	
	Implementación de los atributos y métodos necesarios para cumplir con las especificaciones descritas	0,5	
	Implementación y uso de atributos y métodos estáticos	0,5	
	Correcto funcionamiento de los métodos de las clases siguiendo las especificaciones del enunciado.	1,75	
Programa principal	Organización del programa principal usando los métodos necesarios para tener un código claro y legible	0,5	
	Se añade un cliente correctamente al vector	1	
	Se elimina un cliente correctamente del vector	1	
	Se muestra correctamente el listado de clientes	0,5	
	Se muestra correctamente la información de un cliente	0,5	
	Se añade correctamente una transacción	1	
	Funcionamiento del menú	0,25	
Total		10	