

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma</b>	
Centro Educativo <b>IES Campanillas</b> (sede PTA)		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>	Profesor <b>Luis José Sánchez González</b>
Curso lectivo: <b>2019 / 2020</b>	Grupo: <b>1º DAM</b>	Trimestre: <b>Segundo – Recuperación</b>	Fecha: <b>16 de abril de 2020</b>

## RECUPERACIÓN DEL SEGUNDO TRIMESTRE

En el segundo trimestre, las dos actividades evaluables han sido un **examen sobre colecciones y Programación Orientada a Objetos** y una **aplicación en JSP y MySQL de tipo CRUD**.

### RECUPERACIÓN DEL EXAMEN

La parte correspondiente al examen se recuperará con las siguientes actividades:

- Resolución de todos y cada uno de los ejercicios del examen (modelo A). Se deben subir a un repositorio de GitHub.
- Realización de una aplicación de **gestión de comerciales, clientes y ventas basada en el ejercicio 4 del examen** con los siguientes requisitos:
  - La jerarquía de clases estará basada en el ejercicio 4 del examen.
  - Se podrá hacer en consola o con JSP.
  - Debe permitir operaciones de listado, altas, bajas y modificaciones sobre comerciales y clientes.
  - Los comerciales podrán vender a los clientes y se podrán consultar esas ventas.
  - Debe haber una cantidad suficiente de datos precargados como para poder hacer listados.
  - La persistencia de los datos en disco (ya sea BBDD o ficheros) es opcional.

El lunes 11 de mayo, cada alumno dispondrá de 15 minutos para exponer su aplicación de gestión de comerciales, clientes y ventas de forma individual y el profesor podrá hacer preguntas antes, durante o al final de la exposición. Dicha exposición se realizará de forma presencial o telemática en función de las circunstancias. A continuación se muestran los horarios de exposición:

Cano Santana, Miguel	08:30 – 08:45
Cisneros Santana, Pedro	08:45 – 09:00
Cisneros Santana, Pedro	09:00 – 09:15
González Mora, Miguel	09:15 – 09:30
Gutiérrez Pardo, Miguel Valerio	09:30 – 09:45
López Ramos, Rodrigo	09:45 – 10:00
López Sánchez, Javier	10:00 – 10:15
Pavón López, Juan	10:15 – 10:30
Rosales Girón, José Antonio	10:30 – 10:45
Torres Barba, Pedro	10:45 – 11:00
Toscano Díaz, Sergio	11:00 – 11:15

Si un alumno termina la aplicación antes de la fecha indicada y la quiere presentar, puede acordar un día y una hora con el profesor.

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma</b>		
Centro Educativo <b>IES Campanillas</b> (sede PTA)		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>		Profesor <b>Luis José Sánchez González</b>
Curso lectivo: <b>2019 / 2020</b>	Grupo: <b>1º DAM</b>	Trimestre: <b>Segundo – Control 1</b>	Modelo: <b>A</b>	Fecha: <b>10 de febrero de 2020</b>

### INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por “Ex” seguido del número de lista, seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- ➔ En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el modelo.
- ➔ Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class.**

1. Realiza un test de inglés. La información se debe almacenar en un diccionario (HashMap) en el que las entradas (al menos 8) están formadas por pares (palabra en inglés, palabra en español). El programa hará 5 preguntas. No es necesario controlar que no se repitan las preguntas. En cada pregunta elegirá al azar una palabra en inglés y otra en español y preguntará si es correcto.
  - a Al final se mostrará el número de respuestas acertadas.

Ejemplo:

```
¿raspberry es frambuesa? Conteste (s/n): s
¿house es coche? Conteste (s/n): s
¿apple es manzana? Conteste (s/n): n
¿green es armario? Conteste (s/n): n
¿water es hola? Conteste (s/n): s
Respuestas acertadas: 2
```

- b Amplía el programa para que, además del número de respuestas correctas, se muestren las soluciones de las palabras cuya respuesta no ha sido correcta.

Ejemplo:

```
¿raspberry es frambuesa? Conteste (s/n): s
¿house es coche? Conteste (s/n): s
¿apple es manzana? Conteste (s/n): n
¿green es armario? Conteste (s/n): n
¿water es hola? Conteste (s/n): s
Respuestas acertadas: 2
Debe repasar las siguientes palabras:
house: casa
apple: manzana
water: agua
```

2. Crea una función que mezcla dos ArrayList de números enteros a partir de otros dos que se pasan como parámetros. Para mezclarlos, se va cogiendo un elemento del primer ArrayList y otro del segundo. Cuando se acaben los elementos de una de las listas lista, se continúan pasando los de la otra. La cabecera es la siguiente:

```
public static ArrayList<Integer> mezclaListas(ArrayList<Integer> a1,
ArrayList<Integer> a2)
```

Contenido del main:

```
ArrayList<Integer> lista1 = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 5, 6));
ArrayList<Integer> lista2 = new ArrayList<>(Arrays.asList(77, 88, 99));
ArrayList<Integer> lista3 = new ArrayList<>();

System.out.println(mezclaListas(lista1, lista2));
System.out.println(mezclaListas(lista2, lista1));
System.out.println(mezclaListas(lista1, lista3));
System.out.println(mezclaListas(lista2, lista3));
```

Salida:

```
[1, 77, 2, 88, 3, 99, 4, 5, 6]
[77, 1, 88, 2, 99, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6]
[77, 88, 99]
```

3. Modifica el programa Gestisimal realizado en clase con ArrayList (ejercicio 13 del capítulo 10) añadiendo las siguientes mejoras:
- El listado de artículos debe aparecer ordenado alfabéticamente por el código de artículo.
  - Al final del listado debe aparecer un resumen con los siguientes datos: número total de artículos, precio de compra medio, precio de venta medio, margen de beneficio (precio de venta menos precio de compra) medio.
  - Añade al menú la opción “Listado de artículos bajos de stock” mediante la cual se obtiene un listado con todos los datos de los artículos que tienen un stock inferior o igual a un número (tope) que se introduce por teclado. Por ejemplo, si el usuario introduce el 20, aparecerán todos los artículos con un stock inferior o igual a 20.

4. Una empresa nos pide implementar la gestión de los comerciales y los clientes. El primer paso es implementar las clases e interfaces con los métodos básicos. Crea la clase **Comercial** de tal forma que de un comercial se conozca su nombre, correo electrónico, cargo que ocupa y los artículos que ha vendido, es decir, una lista con los nombres de los artículos que ha vendido y la cantidad de cada uno de ellos (crea una clase adicional **Articulo** con **nombre** y **cantidad**). La empresa tiene clientes que pueden ser usuarios u otras empresas. Crea la clase **Usuario**. De cada usuario se quiere saber su nombre, su correo electrónico y su teléfono. Crea la clase **Empresa** de tal forma que de cada una de ellas se pueda saber su CIF, su nombre y su teléfono. Como los Comerciales y los usuarios son personas, para no repetir innecesariamente código, crea la clase **Persona** como superclase abstracta de **Comercial** y de **Usuario** con los atributos y métodos adecuados. Las clases **Usuario** y **Empresa** deben implementar la interfaz **Cliente** que contiene el método **compra()**.

Contenido del main:

```
Usuario pepe = new Usuario("Pepe", "pepe@yahoo.com", "555 123456");
Usuario pablo = new Usuario("Pablo", "pablo@gmail.com", "555 112233");
Empresa ofiMalaga = new Empresa("B123456", "Ofi Málaga", "555 654321");
Comercial juan = new Comercial("Juan", "juan@hotmail.com", "jefe de
ventas");
Comercial luis = new Comercial("Luis", "luis@gmail.com", "coordinador de
ventas");
pepe.compra(juan, new Articulo("Impresora Láser", 1));
ofiMalaga.compra(juan, new Articulo("Impresora Láser", 2));
pepe.compra(juan, new Articulo("Paquete de folios", 10));
pablo.compra(luis, new Articulo("Clasificador anillas", 12));
pepe.compra(luis, new Articulo("Paquete de folios", 50));
pablo.compra(luis, new Articulo("Mesa escritorio", 6));
pepe.compra(juan, new Articulo("Paquete de folios", 5));
System.out.println(juan);
System.out.println(luis);
```

Salida:

```
Nombre:Juan
Correo electrónico:juan@hotmail.com
Cargo: jefe de ventas
Ventas realizadas
=====
Impresora Láser, 3 unid.
Paquete de folios, 15 unid.

Nombre:Luis
Correo electrónico:luis@gmail.com
Cargo: coordinador de ventas
Ventas realizadas
=====
Clasificador anillas, 12 unid.
Paquete de folios, 50 unid.
Mesa escritorio, 6 unid.
```