

RA6. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases (10%) - **(PARTE PRÁCTICA 100%)**

Una agencia de viajes dispone de un fichero con información de los viajes que oferta. Este fichero es un fichero secuencial de objetos **Viaje**. Cada objeto representa un viaje y se identifica por el campo **id**. El fichero se llama: **Viajes.dat**. La estructura es:

```
public class Viaje implements Serializable {
    private static final long serialVersionUID = 1L;

    private int id;           // identificación del viaje
    private String descripcion; // descripción del viaje
    private LocalDate fechasalida; // fecha de salida
    private double pvp;       // pvp del viaje por viajero
    private int dias;         // nº días que dura el viaje
    private int viajeros;     // nº de viajeros que hacen el viaje, inicialmente es 0

    Constructores
    Métodos get, set, y método toString().
    //no cambies nada de la clase
}
```

La agencia de viajes también dispone un fichero de los clientes que reservan los viajes. Este fichero es un fichero secuencial de objetos **Cliente**. Cada objeto representa un cliente y se identifica por el campo **id**. El fichero se llama: **Clientes.dat**. La estructura es:

```
public class Cliente implements Serializable {
    private static final long serialVersionUID = 1L;

    private int id;           //identificación del cliente
    private String nombre;    //nombre del cliente
    private int viajescontratados; //nº viajes contratados, inicialmente 0
    private double importetotal; //importe total de los viajes realizados,
                                //inicialmente es 0

    Constructores
    Métodos get, set, y método toString().
    //no cambies nada de la clase
}
```

Por último se tiene otro fichero secuencial con las reservas de los clientes sobre los viajes. Cada registro es una reserva de un cliente en un viaje y almacena el id del viaje, el id del cliente y el número de plazas reservadas por el cliente. El fichero se llama **Reservas.dat** y es de tipos de datos primitivos. Un cliente puede hacer varias reservas en los viajes que quiera. El formato es el siguiente:

Id viaje	Id cliente	Plazas
int	int	int

Este fichero puede contener id de viajes que no existen y también id de clientes que no existen.

Todos los ficheros tienen datos.

SE PIDE:

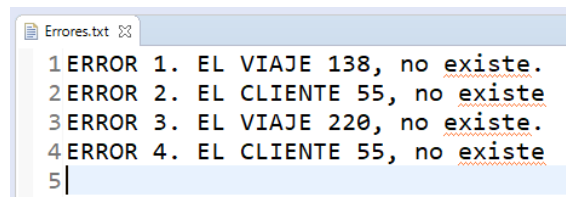
1) [1,5 puntos] Actualizar en el fichero **Viajes.dat**:

- El campo **viajeros** del fichero para que contenga el nº de viajeros que hacen el viaje. Este campo será igual a la suma de las plazas reservadas en el viaje correspondiente.

2) [2,5 puntos] Actualizar en el fichero **Clientes.dat**:

- El campo **viajescontratados** para que almacene el nº de viajes contratados por el cliente. Cada registro en el fichero de reservas es un viaje contratado.
- El campo **importetotal** debe almacenar la suma de lo que valen las reservas que ha realizado el cliente. El importe de cada reserva es igual a la multiplicación del pvp del viaje por el número de plazas.

3) [1 punto] Para los identificadores de viajes y clientes que están en el fichero de **Reservas.dat** y que no existen en los ficheros de viajes y clientes, se debe crear un mensaje que se insertará en un fichero de texto acompañado de un número de error a la izquierda indicando el número de error que se crea. El fichero se llama **Errores.txt** La información para este fichero se obtiene a partir de las reservas (es decir no generéis el fichero escribiendo los datos tal cual). El contenido del fichero debe ser similar a lo mostrado aquí:



```
1 ERROR 1. EL VIAJE 138, no existe.
2 ERROR 2. EL CLIENTE 55, no existe
3 ERROR 3. EL VIAJE 220, no existe.
4 ERROR 4. EL CLIENTE 55, no existe
5
```

4) [1 punto] Una vez actualizado el fichero se mostrará el siguiente listado con el campo número de viajeros actualizado:

```
=====
```

ID	DESCRIPCION	FEC SALIDA	VIAJEROS
===	=====	=====	=====
14	Alemania Oeste	01/08/2020	9
16	China Gran Muralla	25/09/2020	12
18	Croacia, Perla del Adriático	01/08/2020	0
20	Crucero por el mar Mediterráneo	01/08/2020	9
22	Cuba y Miami	15/09/2020	2
26	Moscu San Petersburgo	10/08/2020	0
28	Noruega Mágica	20/08/2020	0
30	Países Bajos	10/08/2020	0
32	Praga, Viena, Budapest	20/08/2020	0
34	Costa Brava	01/09/2020	0
35	Costa de Almería	15/09/2020	0
45	París romántico	01/08/2020	0
36	Extremadura al completo	25/09/2020	0
38	Huelva	20/08/2020	3
39	Oropesa del Mar con balneario	01/09/2020	0
40	Zaragoza y Teruel	10/08/2020	0
37	Galicia Costa da Morte	10/08/2020	7
24	La Toscana	01/09/2020	19

```
=====
```

5) [2 puntos] A continuación se debe realizar un proceso repetitivo en el que se introduzca el identificador del viaje y se muestre su nombre, el número de viajeros y los clientes que han reservado ese viaje. El proceso finalizará cuando el identificador sea 0. Si el viaje no existe se debe

mostrar un mensaje indicándolo. Si el viaje no tiene reservas se debe mostrar un mensaje indicándolo. Se debe mostrar el número de clientes que ha reservado ese viaje. Ejemplo de salida:

=====

Introduce id del viaje: 38

Huelva, Viajeros: 3

=====

ID	NOMBRE	PLAZAS
5	Julio Reyes	3

Número de clientes: 1

=====

Introduce id del viaje: 24

La Toscana, Viajeros: 19

=====

ID	NOMBRE	PLAZAS
2	Raquel Martinez	2
1	Alicia Sanz	2
11	Fernando Alia	5
4	Dora Suela	1
5	Julio Reyes	2
10	Maria Sanz	1
15	Pedro Martin	3
12	Daniel Sanchez	2
18	Maria Jose Perez	1

Número de clientes: 9

=====

Introduce id del viaje: 99

NO EXISTE EL ID DE VIAJE

=====

Introduce id del viaje: 40

NO HAY RESERVAS

=====

Introduce id del viaje: 0

- 6) **[2 puntos]** Por último se realizará otro proceso repetitivo en el que se introduce un identificador de cliente y se muestre su nombre, el número de viajes contratados y la lista de los viajes con el número de plazas que ha contratado. Al final se debe mostrar el importe total. El proceso finalizará cuando el identificador sea 0. Si el cliente no existe se debe mostrar un mensaje indicándolo. Si el cliente no ha contratado ningún viaje se debe mostrar un mensaje indicándolo. Ejemplo de salida:

=====

Introduce id del cliente: 11

Fernando Alia, Viajes contratados: 2

=====

ID	DESCRIPCION	FEC SALIDA	PVP	PLAZAS
20	Crucero por el mar Mediterráneo	01/08/2020	1.340,00	2
24	La Toscana	01/09/2020	800,00	5

Importe total : 6.680,00

=====

Introduce id del cliente: 2

Raquel Martinez, Viajes contratados: 4

=====

ID	DESCRIPCION	FEC SALIDA	PVP	PLAZAS
----	-------------	------------	-----	--------

16	China Gran Muralla	25/09/2020	2.100,00	3
20	Crucero por el mar Mediterráneo	01/08/2020	1.340,00	1
37	Galicia Costa da Morte	10/08/2020	1.100,00	1
24	La Toscana	01/09/2020	800,00	2

Importe total : 10.340,00

=====

Introduce id del cliente: 99

NO EXISTE EL ID DE CLIENTE

=====

Introduce id del cliente: 19

NO HA CONTRATADO NINGÚN VIAJE

=====

Introduce id del cliente: 0

Normas:

- DEBES CREAR UN PROYECTO ECLIPSE DE NOMBRE **RA5_FICTunombreyApe** SIN ESPACIOS EN BLANCO, TILDES, EÑES, ETC...
- DEBES COPIAR (O IMPORTAR) EL PAQUETE **datos** QUE CONTIENE LAS CLASES **Cliente y Viaje** (se incluye en los recursos).
- **OJO!!**. DEL PAQUETE **datos** NO CAMBIAS NADA.
- DEBES COPIAR Y EJECUTAR EL PROGRAMA JAVA **CrearFicheros.java** PARA CREAR E INSERTAR DATOS EN LOS FICHEROS DEL ENUNCIADO, ESTE TIENE QUE ESTAR EN EL PAQUETE POR DEFECTO **src** (se incluye en los recursos).
- EL PROGRAMA QUE REALIZA LO QUE PIDE EL ENUNCIADO SE DEBE LLAMAR **Main_tunombreyApe.java**.
- PUEDES CREAR LAS CLASES AUXILIARES QUE NECESITES.
- CREA EL FICHERO JAR LLAMADO **Main.jar** PARA EJECUTAR LA APLICACIÓN.

FORMATO DE ENTREGA:

- CREA UN **FICHERO ZIP** DE NOMBRE **RA5_NOMBREAPELLIDO.ZIP** QUE CONTENGA EL PROYECTO ECLIPSE CON LAS CLASES NECESARIAS.
- EL PROGRAMA SE EJECUTARÁ DESDE LA LINEA DE COMANDOS DEL DOS.
- SE EJECUTARÁ EL FICHERO **Main.jar**
- CUALQUIER EXCEPCIÓN NO CONTROLADA **RESTARÁ 3 PUNTOS**.
- CUALQUIER ERROR AL EJECUTAR EL FICHERO JAR **RESTARÁ 3 PUNTOS**.

Tiempo: 3 horas.