

ENCUESTA INFEDILIDAD.....	1
PROGRAMADOR JUNIOR/SENIOR/JEFE PROYECTO.....	4
LISTAR TEMPORADAS/EPISODIOS JUEGO DE TRONOS.....	5
CUADRANTE CENTRO CIRCUNFERENCIA.....	6
PRODUCTO DE DOS NÚMEROS-POSITIVO/NEGATIVO.....	6
ACERTAR NÚMERO ENTRE 0 Y 100.....	7
PIEDRA/PAPEL/TIJERA.....	8
NÚMERO MÁGICO.....	9
CALCULADORA.....	10
CONVIERTE NÚMEROS EN PALABRAS.....	13
ARRAYLIST CENA DE NAVIDAD.....	14
DICCIONARIO.....	15
GRUPOS DE MÚSICA.....	17
PACO AIRWAYS.....	19
COLA CINE.....	24
BIBLIOTECA.....	28
AGENDA.....	35
LIBRERIA.....	41

INFIDELIDAD

```
import java.util.*;

public class Ejer1 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int puntuacion = 0;

        String respuesta;

        System.out.println("Tu pareja parece estar más inquieta de lo normal sin ningún
        motivo aparente. (verdadero/falso)");

        respuesta = scanner.nextLine();

        if (respuesta.equals("verdadero")) {

            puntuacion += 3;

        }

        System.out.println("Ha aumentado sus gastos de vestuario. (verdadero/falso)");

        respuesta = scanner.nextLine();

        if (respuesta.equals("verdadero")) {

            puntuacion += 3;

        }

        System.out.println("Ha perdido el interés que mostraba anteriormente por ti.
        (verdadero/falso)");

        respuesta = scanner.nextLine();

        if (respuesta.equals("verdadero")) {

            puntuacion += 3;

        }

        System.out.println("Ahora se asea con más frecuencia. (verdadero/falso)");

        respuesta = scanner.nextLine();

        if (respuesta.equals("verdadero")) {

            puntuacion += 3;

        }

        System.out.println("No te deja que mires la agenda de su teléfono móvil.
        (verdadero/falso)");

        respuesta = scanner.nextLine();

        if (respuesta.equals("verdadero")) {

            puntuacion += 3;

        }

    }

}
```

```

System.out.println("A veces tiene llamadas que dice no querer contestar cuando estás tú
delante. (verdadero/falso)");

respuesta = scanner.nextLine();

if (respuesta.equals("verdadero")) {
puntuacion += 3;
}

System.out.println("Últimamente se preocupa más en cuidar la línea y/o estar
bronceado/a. (verdadero/falso)");

respuesta = scanner.nextLine();

if (respuesta.equals("verdadero")) {
puntuacion += 3;
}

System.out.println("Muchos días viene tarde después de trabajar porque dice tener
mucho más trabajo. (verdadero/falso)");

respuesta = scanner.nextLine();

if (respuesta.equals("verdadero")) {
puntuacion += 3;
}

System.out.println("Has notado que últimamente se perfuma más. (verdadero/falso)");

respuesta = scanner.nextLine();

if (respuesta.equals("verdadero")) {
puntuacion += 3;
}

System.out.println("Se confunde y te dice que ha estado en sitios donde no ha ido
contigo. (verdadero/falso)");

respuesta = scanner.nextLine();

if (respuesta.equals("verdadero")) {
puntuacion += 3;
}

if (puntuacion >= 0 && puntuacion <= 10) {
System.out.println("¡Enhorabuena! tu pareja parece ser totalmente fiel.");
} else if (puntuacion >= 11 && puntuacion <= 22) {
System.out.println("Quizás exista el peligro de otra persona en su vida o en su mente,"
+ "\naunque seguramente será algo sin importancia. No bajes la
guardia.");
} else if (puntuacion >= 23 && puntuacion <= 30) {

```

```

System.out.println("Tu pareja tiene todos los ingredientes para estar viviendo
un romance con otra persona."
+ "\Te aconsejamos que indagues un poco más y averigües que es lo que
está pasando por su cabeza.");
} else {
System.out.println("Puntuación fuera de rango. Por favor, revisa tus
respuestas.");
}
}
}
}

```

PROGRAMADOR JUNIOR/SENIOR/JEFE PROYECTO

```

import java.util.Scanner;

public class Ejer2 {
    int diasViaje=0;
    double sueldoBase = 0;
    int cargo=0;
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double sueldoBase = 0;
        double dietas = 0;
        double sueldoBruto = 0;
        double irpf = 0;
        double sueldoNeto = 0;
        System.out.println("Introduce el cargo del empleado:\n1 - Programador Junior\n2 -
Programador Senior\n3 - Jefe de proyecto");
        int cargo = scanner.nextInt();
        System.out.println("Introduce los días que ha estado de viaje visitando clientes
durante el mes:");
        int diasViaje = scanner.nextInt();
        System.out.println("Introduce el estado civil:\n1 - Soltero\n2 - Casado");
        int estadoCivil = scanner.nextInt();
        double sueldoBase = 0;
        if (cargo == 1) {
            sueldoBase = 950;
        } else if (cargo == 2) {

```

```

sueldoBase = 1200;
else if (cargo == 3) {
sueldoBase = 1600;
}
dietas = diasViaje * 30;
sueldoBruto = sueldoBase + dietas;
if (estadoCivil == 1) {
irpf = sueldoBruto * 0.25;
} else if (estadoCivil == 2) {
irpf = sueldoBruto * 0.20;
} sueldoNeto = sueldoBruto - irpf;
System.out.println("Nómina del empleado:");
System.out.println("Sueldo base: " + String.format("%.2f",sueldoBase) + " euros");
System.out.println("Dietas (" + diasViaje + " días de viaje): " +
String.format("%.2f",dietas) + " euros");
System.out.println("Sueldo bruto: " + String.format("%.2f",sueldoBruto)
+ " euros");
System.out.println("IRPF: " + String.format("%.2f",irpf) + "
euros");
System.out.println("Sueldo neto: " + String.format("%.2f",sueldoNeto) + " euros");
}
}

```

JUEGO DE TRONOS

```

public class Ejer4JDT {
public static void main(String[] args) {
for (int i=1; i <= 8; i++) {
for (int j=1; j <=10; j++) {
if ((i == 3) && (j == 9)) {
System.out.println("Capitulo 3x9: La boda
roja");
}else if ((i == 6) && (j == 10)){
System.out.println("Capitulo 6x10: Vientos de invierno");
}
else {
System.out.println("Capitulo "+i+"x"+j);
}
}
}
}

```

```
}  
}  
}  
}  
}
```

CUADRANTE CENTRO CIRCUNFERENCIA

```
package ejerciciosClase;  
  
import java.util.Scanner;  
  
public class Ejer5 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Introduce la coordenada x del centro: ");  
        int x = scanner.nextInt();  
        System.out.print("Introduce la coordenada y del centro: ");  
        int y = scanner.nextInt();  
        if (x > 0 && y > 0) {  
            System.out.print("Primer cuadrante");  
        } else if (x < 0 && y > 0) {  
            System.out.print("Segundo cuadrante");  
        } else if (x < 0 && y < 0) {  
            System.out.print("Tercer cuadrante");  
        } else if (x > 0 && y < 0) {  
            System.out.print("Cuarto cuadrante");  
        } else if (x == 0 && y != 0) {  
            System.out.print("Sobre el eje Y");  
        } else if (y == 0 && x != 0) {  
            System.out.print("Sobre el eje X");  
        } else {  
            System.out.print("En el origen");  
        }  
    }  
}
```

PRODUCTO DE DOS NÚMEROS-POSITIVO/NEGATIVO

```
import java.util.*;  
  
public class Ejer6 {
```

```

public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Introduce el primer número (num1): ");
int num1 = scanner.nextInt();
System.out.print("Introduce el segundo número (num2): ");
int num2 = scanner.nextInt();
if ((num1 > 0 && num2 > 0) || (num1 < 0 && num2 < 0)) {
System.out.println("El producto de los 2 números es positivo");
} else if ((num1 > 0 && num2 < 0) || (num1 < 0 && num2 >
0)) {
System.out.println("El producto de los 2 números es negativo");
} else {
System.out.println("Uno de los números es cero, por lo que el producto es cero");
}
}
}
}

```

ACERTAR NÚMERO ENTRE 0 Y 100

```

package ejerciciosClase;
import java.util.*;
public class Ejer12 {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
int numeroSecreto = 42;
int intento;
boolean adivinado = false;
System.out.println("¡Bienvenido al juego de adivinar el número!");
System.out.println("He pensado en un número entre 1 y 100. ¿Puedes adivinar cuál es?");
while (!adivinado) {
System.out.print("Introduce tu adivinanza: ");
intento = scanner.nextInt();
if (intento == numeroSecreto) {
System.out.println("¡Felicidades! ¡Has adivinado el número!");
adivinado = true;
} else if (intento < numeroSecreto) {
System.out.println("El número es mayor que " + intento + ".");
}
}
}
}

```

```

    } else {
        System.out.println("El número es menor que " + intento + ".");
    }
    if (Math.abs(numeroSecreto - intento) <= 5 && intento
        != numeroSecreto) {
        System.out.println("¡Estás muy cerca!");
    } else if (Math.abs(numeroSecreto - intento) <= 10 && intento != numeroSecreto) {
        System.out.println("Estás cerca.");
    } else if (intento != numeroSecreto) {
        System.out.println("Estás lejos.");
    }
}
}
}
}

```

PIEDRA/PAPEL/TIJERA

```

public static void main(String[] args) {
    Scanner teclado = new Scanner(System.in);

    String jugador1;
    String jugador2;
    String resultado;

    while (true) {
        System.out.println("Jugador 1, introduce tu jugada (piedra, papel o tijera): ");
        jugador1 = teclado.nextLine().toLowerCase();
        if (jugador1.equals("piedra") || jugador1.equals("papel") || jugador1.equals("tijera")) {
            break;
        } else {
            System.out.println("Entrada no válida. Por favor, introduce 'piedra', 'papel' o 'tijera'.");
        }
    }

    while (true) {
        System.out.println("Jugador 2, introduce tu jugada (piedra, papel o tijera): ");
        jugador2 = teclado.nextLine().toLowerCase();
        if (jugador2.equals("piedra") || jugador2.equals("papel") || jugador2.equals("tijera")) {
            break;
        } else {

```



```

System.out.println("Entrada no válida. Por favor, introduce 'piedra', 'papel' o 'tijera.'");
}
}
if (jugador1.equals(jugador2)) {
    resultado = "¡Es un empate!";
} else if ((jugador1.equals("piedra") && jugador2.equals("tijera")) ||
(jugador1.equals("papel") && jugador2.equals("piedra")) ||
(jugador1.equals("tijera") && jugador2.equals("papel"))) {
    resultado = "¡Jugador 1 gana!";
} else {
    resultado = "¡Jugador 2 gana!";
}
System.out.println(resultado);
teclado.close();
}
}

```

NÚMERO MÁGICO

```

import java.util.*;

public class Ejer04 {

    public static void main(String[] args) {

        Scanner teclado = new Scanner(System.in);

        Random numero = new Random();

        int objetivo = numero.nextInt(100)+1;

        int tirada = 0;

        int intentos = 1;

        boolean jugada = false;

        System.out.println("El juego del número mágico");

        while (jugada == false) {

            System.out.println("Introduzca numero:");

            tirada = teclado.nextInt();

            if (tirada == objetivo) {

                System.out.println("Enhorabuena! Has acertado el numero que era el "+tirada+"!");

                System.out.println("Lo has conseguido en " + intentos + " tiradas!");

                jugada = true;

            }else {

```

```

intentos++;

if (objetivo > tirada) {

System.out.println("El numero mágico es mayor!");

}

if (objetivo < tirada) {

System.out.println("El numero mágico es menor!");

}

}

}

}

teclado.close();

}

}

```

CALCULADORA

```

import java.util.*;

public class Ejer01 {

public static void main(String[] args) {

int opcion = 0;

Scanner teclado = new Scanner(System.in);

int num1 = 0;

int num2 = 0;

while (opcion != 9) {

System.out.println("1.- Sumar \r\n"

+ "2.- Restar \r\n"

+ "3.- Multiplicar \r\n"

+ "4.- Dividir \r\n"

+ "5.- Potencia \r\n"

+ "6.- Raíz cuadrada \r\n"

+ "9.- Salir ");

System.out.println("Introduce una opción");

opcion = teclado.nextInt();

switch (opcion) {

case 1:

System.out.println("Introduce el primer número");

num1 = teclado.nextInt();

System.out.println("Introduce el segundo número");

```

```
num2 = teclado.nextInt();
System.out.println(sumar(num1, num2));
break;
case 2:
System.out.println("Introduce el primer número");
num1 = teclado.nextInt();
System.out.println("Introduce el segundo número");
num2 = teclado.nextInt();
System.out.println(restar(num1, num2));
break;
case 3:
System.out.println("Introduce el primer número");
num1 = teclado.nextInt();
System.out.println("Introduce el segundo número");
num2 = teclado.nextInt();
System.out.println(multiplicar(num1, num2));
break;
case 4:
System.out.println("Introduce el primer número");
num1 = teclado.nextInt();
System.out.println("Introduce el segundo número");
num2 = teclado.nextInt();
System.out.println(dividir(num1, num2));
break;
case 5:
System.out.println("Introduce el primer número");
num1 = teclado.nextInt();
System.out.println("Introduce el segundo número");
num2 = teclado.nextInt();
System.out.println(potencia(num1, num2));
break;
case 6:
System.out.println("Introduce un número");
num1 = teclado.nextInt();
System.out.println(square(num1));
```

```
break;

case 9:

System.out.println("Saliendo... ¡Adiós!");

break;

default:

System.out.println("Opción no válida");

}

}

teclado.close();

}

public static int sumar(int num1, int num2) {

return num1 + num2;

}

public static int restar(int num1, int num2) {

return num1 - num2;

}

public static int multiplicar(int num1, int num2) {

return num1 * num2;

}

public static int dividir(int num1, int num2) {

if (num2 != 0) {

return num1 / num2;

} else {

System.out.println("Error: División por cero.");

return 0;

}

}

public static double potencia(int num1, int num2) {

return Math.pow(num1, num2);

}

public static double square(int num1) {

return Math.sqrt(num1);

}

}
```

CONVIERTE NÚMEROS EN PALABRAS

```
import java.util.*;

public class Ejer03 {

    public static void main(String[] args) {

        Scanner teclado = new Scanner(System.in);

        System.out.println("Introduzca un numero:");

        int n = teclado.nextInt();

        String cadena = Integer.toString(n);

        String resultado = "";

        for (int i = 0; i < cadena.length(); i++) {

            String digito = transforma_digito(cadena.charAt(i));

            resultado = resultado + digito + " ";

            teclado.close();

        }

        System.out.println("El resultado es: " + resultado);

    }

    public static String transforma_digito(char digito) {

        String valor="";

        switch(digito) {

            case '0': valor = "cero";

            break;

            case '1': valor = "uno";

            break;

            case '2': valor = "dos";

            break;

            case '3': valor = "tres";

            break;

            case '4': valor = "cuatro";

            break;

            case '5': valor = "cinco";

            break;

            case '6': valor = "seis";

            break;

            case '7': valor = "siete";

            break;

        }

    }

}
```

```

case '8': valor = "ocho";
break;
case '9': valor = "nueve";
break;
default: valor = "error";
break;
}
return valor;
}
}

```

ARRAYLIST CENA DE NAVIDAD

```

import java.util.*;

public class Ejer02 {

    public static void main(String[] args) {

        Scanner teclado = new Scanner(System.in);

        ArrayList<String> asistentes = new ArrayList<>();

        System.out.println("Introduce los nombres de los asistentes (escribe 'fin' para terminar):");

        String nombre="";

        while (!(nombre.equalsIgnoreCase("fin"))) {

            nombre = teclado.nextLine();

            asistentes.add(nombre.toLowerCase());

        }

        System.out.println("Introduce los nombres de los alumnos que no asistiran (escribe 'fin' para terminar):");

        String nombreBaja="";

        while (!(nombreBaja.equalsIgnoreCase("fin"))) {

            nombreBaja = teclado.nextLine();

            Iterator<String> iterador = asistentes.iterator();

            while (iterador.hasNext()) {

                String asistente = iterador.next();

                if (asistente.equalsIgnoreCase(nombreBaja)) {

                    iterador.remove();

                }

            }

        }

        System.out.println("Introduce el nombre del alumno que solo asistira si

```

```

otro alumno especifico tambien asiste:");

String alumnoCondicional = teclado.nextLine().toLowerCase();

System.out.println("Introduce el nombre del alumno del que depende la
asistencia del anterior:");

String alumnoDependiente = teclado.nextLine().toLowerCase();

if (asistentes.contains(alumnoDependiente)) {
if (!asistentes.contains(alumnoCondicional)) {
asistentes.add(alumnoCondicional);
}
} else {
Iterator<String> iterador = asistentes.iterator();
while (iterador.hasNext()) {
String asistente = iterador.next();
if (asistente.equalsIgnoreCase(alumnoCondicional)) {
iterador.remove();
break;
}
}
}

System.out.println("Lista final de asistentes a la cena de Navidad:");

System.out.println(asistentes);

System.out.println("Introduce el nombre del alumno que quieres buscar en la lista de
asistentes:");

String alumnoBuscar = teclado.nextLine().toLowerCase();

if (asistentes.contains(alumnoBuscar)) {

System.out.println(alumnoBuscar + " está en la lista de asistentes.");
} else {

System.out.println(alumnoBuscar + " no esta en la lista de
asistentes.");
}

teclado.close();
}
}

```

DICCIONARIO

```

import java.util.*;

public class Ejer05 {

```

```
public static void main(String[] args) {  
    HashMap<String, String> diccionario = new HashMap<>();  
    diccionario.put("hola", "hello");  
    diccionario.put("adiós", "goodbye");  
    diccionario.put("por favor", "please");  
    diccionario.put("gracias", "thank you");  
    diccionario.put("sí", "yes");  
    diccionario.put("no", "no");  
    diccionario.put("amigo", "friend");  
    diccionario.put("familia", "family");  
    diccionario.put("comida", "food");  
    diccionario.put("agua", "water");  
    diccionario.put("casa", "house");  
    diccionario.put("escuela", "school");  
    diccionario.put("libro", "book");  
    diccionario.put("perro", "dog");  
    diccionario.put("gato", "cat");  
    diccionario.put("coche", "car");  
    diccionario.put("ciudad", "city");  
    diccionario.put("país", "country");  
    diccionario.put("música", "music");  
    diccionario.put("película", "movie");  
    Scanner teclado = new Scanner(System.in);  
    String palabra;  
    do {  
        System.out.print("Introduce una palabra en español (o 'salir' para terminar): ");  
        palabra = teclado.nextLine().toLowerCase();  
        if (!palabra.equals("salir")) {  
            String traduccion = diccionario.get(palabra);  
            if (traduccion != null) {  
                System.out.println("La traducción en inglés es: " + traduccion);  
            } else {  
                System.out.println("La palabra no está en el diccionario.");  
                System.out.print("¿Quieres añadirla al diccionario? (si/no): ");  
                String respuesta = teclado.nextLine().toLowerCase();  

```



```
grupos.add(grupo);
System.out.println("Grupo añadido al final de la lista.");
break;
case 2:
System.out.print("Introduce la posición: ");
int posicion = teclado.nextInt();
teclado.nextLine();
if (posicion >= 0 && posicion <= grupos.size()) {
System.out.print("Introduce el nombre del grupo: ");
grupo = teclado.nextLine();
grupos.add(posicion, grupo);
System.out.println("Grupo añadido en la posición " + posicion + ".");
} else {
System.out.println("Posición no válida.");
}
break;
case 3:
System.out.print("Introduce la posición: ");
posicion = teclado.nextInt();
if (posicion >= 0 && posicion < grupos.size()) {
System.out.println("Grupo en la posición " + posicion + ": " +
grupos.get(posicion));
} else {
System.out.println("Posición no válida.");
}
break;
case 4:
System.out.print("Introduce la posición: ");
posicion = teclado.nextInt();
teclado.nextLine(); // Consumir la nueva línea
if (posicion >= 0 && posicion < grupos.size()) {
System.out.println("Grupo antiguo en la posición " + posicion + ": " +
grupos.get(posicion));
System.out.print("Introduce el nuevo nombre del grupo: ");
grupo = teclado.nextLine();
```

```

grupos.set(posicion, grupo);
System.out.println("Grupo en la posición " + posicion + " ha sido modificado.");
} else {
System.out.println("Posición no válida.");
}
break;
case 5:
System.out.print("Introduce la posición: ");
posicion = teclado.nextInt();
if (posicion >= 0 && posicion < grupos.size()) {
System.out.println("Grupo en la posición " + posicion + ": " + grupos.get(posicion));
} else {
System.out.println("Posición no válida.");
}
break;
case 6:
System.out.println("Lista de grupos:");
for (int i = 0; i < grupos.size(); i++) {
System.out.println(i + ": " + grupos.get(i));
}
break;
case 7:
System.out.println("Saliendo del programa. ¡Adiós!");
break;
default:
System.out.println("Opción no válida.");
}
} while (opcion != 7);
teclado.close();
}
}

```

PACO AIRWAYS

--CLASE VUELO--

```

public class Vuelo {
private String origen;

```

```
private String destino;

private int dia_ida;

private int dia_vuelta;

private String estado;

private int id;

static int cantidad_vuelos=0;

public Vuelo(String o, String d, int di, int dv) {

    origen = o;

    destino = d;

    dia_ida = di;

    dia_vuelta = dv;

    estado = "Vigente";

    cantidad_vuelos++;

    id = cantidad_vuelos;

}

public void estadovuelo() {

    String mensaje = id + ":" + origen + "-" + destino + " IDA: dia "

+dia_ida+ " VUELTA: dia " +dia_vuelta;

    if (estado.contentEquals("Vigente")) {

        System.out.println(mensaje);

    }else {

        System.out.println(mensaje + " CANCELADO ");

    }

}

public void cancela billete() {

    if (estado.equals("Vigente")) {

        estado = "Cancelado";

        System.out.println("Billete cancelado!!");

    }

    else {

        System.out.println("Este billete ya estaba cancelado");

    }

}

public int get_id() {

    return id;

}
```

```

    }

    public String get_estado() {
        return estado;
    }
}

--CLASE MAIN--

import java.util.*;

public class Main {

    private static Scanner teclado = new Scanner(System.in);

    public static void main(String[] args) {
        char opcion=' ';
        int opcion_num=0;
        ArrayList<Vuelo> lista_vuelos = new ArrayList<Vuelo>();

        System.out.println("Bienvenido a las aerolíneas Paco Airways, ¿qué desea hacer?");

        while(true) {
            System.out.println("");
            System.out.println("1-Comprar billete de avion");
            System.out.println("2-Cancelar billete de avion");
            System.out.println("3-Estado de mi reserva");
            System.out.println("S-Salir");
            System.out.print("OPCION:");
            opcion = teclado.nextLine().charAt(0);
            if ((opcion == 'S') || (opcion == 's')) {
                break;
            } else {
                opcion_num = Character.getNumericValue(opcion);
                switch (opcion_num) {
                    case 1: comprar_billetes(lista_vuelos);
                        break;
                    case 2: cancelar_billetes(lista_vuelos);
                        break;
                    case 3: estado_reserva(lista_vuelos);
                        break;
                }
            }
        }
    }
}

```

```

teclado.close(); // Cerrar el Scanner aquí al final del programa

System.out.println("Paco Airways no le agradece su confianza. Gracias
por nada.");
}

public static void comprar_billetes(ArrayList<Vuelo> lista_vuelos) {
    System.out.println("Introduzca origen, destino, día de ida y día de vuelta por @:");
    String respuesta = teclado.nextLine();
    String[] datos = respuesta.split("@");
    Vuelo v = new
    Vuelo(datos[0],datos[1],Integer.parseInt(datos[2]),Integer.parseInt(datos[3]));
    lista_vuelos.add(v);
    System.out.println("Billete comprado!!");
}

public static void cancelar_billetes(ArrayList<Vuelo> lista_vuelos) {
    int id_billete=0;
    int id_recorrer=0;
    char confirmar=' ';
    boolean encontrado=false;
    System.out.println("Introduzca id del billete a cancelar:");
    id_billete = teclado.nextInt();
    teclado.nextLine();
    Vuelo v;
    Iterator iter = lista_vuelos.iterator();
    while (iter.hasNext()) {
        v = (Vuelo)iter.next();
        id_recorrer = v.get_id();
        if (id_recorrer == id_billete) {
            encontrado = true;
            System.out.println("Billete encontrado, ¿Confirma su
cancelación(S/N)?");
            confirmar = teclado.nextLine().charAt(0);
            if ((confirmar == 'S') || (confirmar == 's')) {
                v.cancela_billete();
            } else {
                System.out.println("Operación de cancelación cancelada a su vez");
            }
        }
    }
}

```

```

}
}
}
if (!encontrado) {
    System.out.println("Billete NO encontrado.");
}
}

public static void estado_reserva(ArrayList<Vuelo> lista_vuelos) {
    Vuelo v;
    char tipo=' ';
    String estado_vuelo = " ";
    Iterator iter = lista_vuelos.iterator();
    while (true) {
        System.out.println("¿Que vuelos desea ver? (V-Vigentes,C-Cancelados-T-Todos)");
        tipo = teclado.nextLine().charAt(0);
        if ((tipo == 'V') || (tipo == 'v') || (tipo == 'C') || (tipo == 'c') ||
            (tipo == 'T') || (tipo == 't')) {
            break;
        }
    }

    while (iter.hasNext()) {
        v = (Vuelo)iter.next();
        estado_vuelo = v.get_estado();
        if ((tipo == 'T') || (tipo == 't')) {
            v.estadovuelo();
        }

        if (((tipo == 'V') || (tipo == 'v')) &&
            (estado_vuelo.equals("Vigente"))){
            v.estadovuelo();
        }

        if (((tipo == 'C') || (tipo == 'c')) &&
            (estado_vuelo.equals("Cancelado"))){
            v.estadovuelo();
        }
    }
}

```

```
}
```

```
}
```

COLA CINE

--CLASE PERSONAS--

```
public class Personas {  
    private String nombre;  
    private int edad;  
    public Personas(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
    public void imprimir_persona() {  
        System.out.println(this.nombre + "," + edad + ".");  
    }  
}
```

--CLASE COLACINE--

```
import java.util.*;  
public class ColaCine {  
    private Queue<Personas> lista_clientes = new LinkedList<Personas>();  
    static int num_clientes=0;  
    public ColaCine() {  
        num_clientes++;  
    }  
    public void recorrerCola() {  
        Iterator iter = lista_clientes.iterator();  
        Personas pers;  
        while (iter.hasNext()) {  
            pers = (Personas)(iter.next());  
            pers.imprimir_persona();  
        }  
    }  
    public void anyadirPersona(Personas p) {  
        lista_clientes.add(p);  
    }  
    public void quitarPersona() {
```



```
lista_clientes.remove();
```

```
}
```

```
}
```

```
--CLASE MAIN--
```

```
import java.util.*;
```

```
public class Main {
```

```
private static Scanner teclado = new Scanner(System.in); // Scanner
```

```
compartido
```

```
public static void main(String[] args) {
```

```
int opcion_menu=0;
```

```
int idcola=0;
```

```
int numcolas=0;
```

```
HashMap<Integer,ColaCine> colas = new HashMap<Integer,ColaCine>();
```

```
System.out.println("Bienvenido a Kinopolis, ¿de cuantas salas dispone tu cine?");
```

```
numcolas = teclado.nextInt();
```

```
teclado.nextLine();
```

```
for (int i=1; i <= numcolas; i++) {
```

```
ColaCine cola1 = new ColaCine();
```

```
colas.put(i,cola1);
```

```
}
```

```
opcion_menu = menu();
```

```
while (opcion_menu != 5) {
```

```
switch(opcion_menu) {
```

```
case 1: idcola = solicitud_cola(numcolas);
```

```
encolar_cliente(idcola, colas);
```

```
break;
```

```
case 2: idcola = solicitud_cola(numcolas);
```

```
desencolar_cliente(idcola, colas);
```

```
break;
```

```
case 3: idcola = solicitud_cola(numcolas);
```

```
informacion_sala(idcola, colas);
```

```
break;
```

```
case 4: estado_cine(colas);
```

```
break;
```

```
}
```

```

opcion_menu = menu();
}

teclado.close(); // Cerrar el Scanner aquí al final del programa

System.out.println("Gracias por venir a Kinopolis.");
}

public static int menu() {
    int opcion=0;

    System.out.println("----- Menu Kinopolis -----");

    System.out.println("1-Encolar cliente en una sala");
    System.out.println("2-Desencolar cliente de la sala");
    System.out.println("3-Información de una sala");
    System.out.println("4-Estado del cine:");
    System.out.println("5-Salir");

    opcion = teclado.nextInt();

    teclado.nextLine(); // Consumir el salto de línea

    return opcion;
}

public static void encolar_cliente(int idcola, HashMap<Integer,ColaCine>
colas) {
    ColaCine c = colas.get(idcola);
    String nombre="";
    int edad=0;

    System.out.println("Introduzca nombre del nuevo cliente para la cola " +
    idcola + ":");

    nombre = teclado.nextLine();

    System.out.println("Introduzca su edad:");

    edad = teclado.nextInt();

    teclado.nextLine(); // Consumir el salto de línea

    Personas pers = new Personas(nombre,edad);

    c.anyadirPersona(pers);

    System.out.println("Añadimos a la cola " + idcola + " el cliente " +
    nombre + " de " + edad + " años.");
}

public static void desencolar_cliente(int idcola,
HashMap<Integer,ColaCine> colas) {

```

```

System.out.println("Desencolamos al primero de la cola de la sala " +
idcola);
ColaCine c = colas.get(idcola);
c.quitarPersona();
}
public static void informacion_sala(int idcola,
HashMap<Integer,ColaCine> colas) {
System.out.println("La sala " + idcola + " tiene esperando a los
siguientes clientes: ");
ColaCine c = colas.get(idcola);
c.recorrerCola();
}
public static int solicitudCola(int numcolas) {
int opcola=0;
System.out.println("¿Sobre qué sala desea realizar la operación?");
opcola = teclado.nextInt();
teclado.nextLine(); // Consumir el salto de línea
while (true) {
if ((opcola >= 1) && (opcola <= numcolas)) {
break;
} else {
System.out.println("Valor de sala introducido erróneo. Introduzca
valores desde el 1 hasta el " + numcolas + ":");
opcola = teclado.nextInt();
teclado.nextLine();
}
}
return opcola;
}
public static void estado_cine(HashMap<Integer,ColaCine> colas) {
for (int i=1; i <= colas.size(); i++) {
System.out.println("Clientes de la sala " + i + ":");
ColaCine c = colas.get(i);
c.recorrerCola();
}
}

```

```
}
```

```
}
```

BIBLIOTECA

--CLASE ALUMNO--

```
public class Alumno {  
    private String NIA;  
    private String nombre;  
    private int edad;  
    private static int numAlumno;  
    public Alumno (String NIA,String nombre, int edad) {  
        this.NIA=NIA;  
        this.nombre=nombre;  
        this.edad=edad;  
        numAlumno++;  
    }  
    public int getnumAlumno() {  
        return numAlumno;  
    }  
    public String getNIAlumno() {  
        return NIA;  
    }  
    public String getnombreAlumno() {  
        return nombre;  
    }  
    public int getedadAlumno() {  
        return edad;  
    }  
    public void datosAlumno() {  
        System.out.println("\t-Alumno de NIA: "+NIA+" y nombre: "+nombre+" tiene  
        "+ edad + " años.");  
    }  
}
```

--CLASE LIBRO--

```
public class Libro {  
    private String titulo;
```

```
private String autor;
private int numPaginas;
private String estado;
private static int numlibro=1;
public Libro (String titulo,String autor, int numPaginas) {
this.titulo=titulo;
this.autor=autor;
this.numPaginas=numPaginas;
this.estado="D";
numlibro++;
}
public int getnumLibro() {
return numlibro;
}
public String gettituloLibro() {
return titulo;
}
public String getautorLibro() {
return autor;
}
public int getnumPaginasLibro() {
return numPaginas;
}
public String getestadoLibro() {
return estado;
}
public void cambiarEstadoLibro () {
if (estado.equalsIgnoreCase("D")) {
estado="P";
}else {
estado="D";
}
}
public void mostrarLibro() {
System.out.println(("t-" + titulo + " escrito por: " + autor + " tiene
```

```
" + numPaginas + " pags. | " + estado));
```

```
}
```

```
}
```

--CLASE PRESTAMO--

```
import java.util.*;
```

```
public class Prestamo {
```

```
private Libro prestamoLibro;
```

```
private Alumno prestamoAlumno;
```

```
public Prestamo(Libro prestamoLibro, Alumno prestamoAlumno) {
```

```
this.prestamoLibro=prestamoLibro;
```

```
this.prestamoAlumno=prestamoAlumno;
```

```
}
```

```
public void infoPrestamo() {
```

```
System.out.println("\t-" + prestamoLibro.getTituloLibro() + " | " +
```

```
prestamoLibro.getAutorLibro() + " | " +
```

```
prestamoLibro.getNumPaginasLibro() + " pags. prestado a: "
```

```
+ prestamoAlumno.getNIAlumno()+ " | " + prestamoAlumno.getNombreAlumno()
```

```
+ " | " +
```

```
prestamoAlumno.getEdadAlumno() + " años.");
```

```
}
```

```
public Libro getLibro() {
```

```
return prestamoLibro;
```

```
}
```

```
public Alumno getAlumno() {
```

```
return prestamoAlumno;
```

```
}
```

```
}
```

--CLASE MAIN--

```
import java.util.*;
```

```
public class Main {
```

```
public static Scanner teclado = new Scanner(System.in);
```

```
public static void main(String[] args) {
```

```
ArrayList<Libro> listaLibros = new ArrayList<>();
```

```
ArrayList<Alumno> listaAlumnos = new ArrayList<>();
```

```
ArrayList<Prestamo> listaPrestamos = new ArrayList<>();
```

```
while (true) {
```

```

int opcion = menu();
if (opcion == 0) {
    break;
} else {
    switch (opcion) {
        case 1:
            altaLibro(listaLibros);
            break;
        case 2:
            altaAlumno(listaAlumnos);
            break;
        case 3:
            hacerPrestamo(listaAlumnos, listaLibros, listaPrestamos);
            break;
        case 4:
            devolverLibro(listaAlumnos, listaLibros, listaPrestamos);
            break;
        case 5:
            resumenBiblioteca(listaAlumnos, listaLibros, listaPrestamos);
            break;
    }
}

public static int menu() {
    int opcion = 0;

    System.out.println("Bienvenido a la biblioteca del IES Henri Matisse");
    System.out.println("opcion 1: Dar de alta un libro");
    System.out.println("opcion 2: Dar de alta un alumno");
    System.out.println("opcion 3: Prestar un libro");
    System.out.println("opcion 4: Devolver un libro");
    System.out.println("opcion 5: Resumen");
    System.out.println("opcion 0: Salir");
    System.out.println("-----");
}

```

```

System.out.println("opcion: ");
opcion = teclado.nextInt();
teclado.nextLine(); // Consumir el salto de línea
return opcion;
}

public static void altaLibro(ArrayList<Libro> listaLibros) {
String titulo;
String autor;
int numPaginas;
System.out.println("introduzca el titulo del libro");
titulo = teclado.nextLine();
System.out.println("introduzca el autor del libro");
autor = teclado.nextLine();
System.out.println("introduzca el numero de paginas del libro");
numPaginas = teclado.nextInt();
teclado.nextLine();
Libro libro = new Libro(titulo, autor, numPaginas);
listaLibros.add(libro);
System.out.println("\nSe ha dado de alta este ejemplar: ");
libro.mostrarLibro();
}

public static void altaAlumno(ArrayList<Alumno> listaAlumnos) {
String NIA;
String nombre;
int edad;
System.out.println("introduzca el NIA del alumno");
NIA = teclado.nextLine();
System.out.println("introduzca el nombre del alumno");
nombre = teclado.nextLine();
System.out.println("introduzca la edad del alumno");
edad = teclado.nextInt();
teclado.nextLine();
Alumno alumno = new Alumno(NIA, nombre, edad);
listaAlumnos.add(alumno);
System.out.println("\nSe ha dado de alta este alumno: ");

```



```

alumno.datosAlumno();
}

public static void hacerPrestamo(ArrayList<Alumno> listaAlumnos,
ArrayList<Libro> listaLibros, ArrayList<Prestamo> listaPrestamos) {
    Libro libro = null;
    Alumno alumno = null;
    String NIA;
    String titulo;
    do {
        System.out.println("Indica el NIA del alumno: ");
        NIA = teclado.nextLine();
        alumno = buscarAlumno(listaAlumnos, NIA);
    } while (alumno == null || !alumno.getNIAAlumno().equalsIgnoreCase(NIA));
    do {
        System.out.println("Indica el titulo del libro: ");
        titulo = teclado.nextLine();
        libro = buscarLibro(listaLibros, titulo);
        if (libro != null) {
            libro.cambiarEstadoLibro();
        }
    } while (libro == null || !libro.getTituloLibro().equals(titulo));
    Prestamo = new Prestamo(libro, alumno);
    listaPrestamos.add(prestamo);
    prestamo.infoPrestamo();
}

public static void devolverLibro(ArrayList<Alumno> listaAlumnos,
ArrayList<Libro> listaLibros, ArrayList<Prestamo> listaPrestamos) {
    Libro libroDevolver = null;
    Libro = null;
    String titulo;
    System.out.println("Indica el titulo del libro que vas a devolver: ");
    titulo = teclado.nextLine();
    libro = buscarLibro(listaLibros, titulo);
    if (libro == null) {
        System.out.println("El libro con el título " + titulo + " no se

```

```

    encontró.");
    return;
}
if (libro.getestadoLibro().equalsIgnoreCase("D")) {
    System.out.println("El libro no está prestado o ya se ha devuelto");
} else {
    libro.cambiarEstadoLibro();
    System.out.println("Libro devuelto correctamente!");
}
libroDevolver = libro;
Iterator<Prestamo> iter = listaPrestamos.iterator();
while (iter.hasNext()) {
    Prestamo = iter.next();
    if (prestamo.getlibro().equals(libroDevolver)) {
        iter.remove();
        System.out.println("Prestamo dado de baja correctamente!");
    }
}
}

public static void resumenBiblioteca(ArrayList<Alumno> listaAlumnos,
    ArrayList<Libro> listaLibros, ArrayList<Prestamo> listaPrestamos) {
    for (Alumno alumno : listaAlumnos) {
        alumno.datosAlumno();
    }
    for (Libro libro : listaLibros) {
        libro.mostrarLibro();
    }
    for (Prestamo prestamo : listaPrestamos) {
        prestamo.infoPrestamo();
    }
}

public static Alumno buscarAlumno(ArrayList<Alumno> listaAlumnos, String
    NIA) {
    for (Alumno alumnoBuscado : listaAlumnos) {
        if (alumnoBuscado.getNIAlumno().equals(NIA)) {

```

```

return alumnoBuscado;
}
}
return null;
}
public static Libro buscarLibro(ArrayList<Libro> listaLibros, String
titulo) {
for (Libro libroBuscado : listaLibros) {
if (libroBuscado.getTituloLibro().equals(titulo)) {
return libroBuscado;
}
}
return null;
}
}

```

AGENDA

-- CLASE CONTACTOSERIALIZABLE--

```

import java.io.Serializable;

public class ContactoSerializable implements Serializable {

private static final long serialVersionUID = 1L;

String nombre, numero;

public ContactoSerializable(String nombre, String numero) {
this.nombre = nombre;
this.numero = numero;
}

public String getNombre() {
return nombre;
}

public String getNumero() {
return numero;
}

@Override
public String toString() {
return nombre + ": " + numero;
}
}

```

```

}
-- CLASE MAINSERIALIZABLE--

import java.io.*;
import java.util.*;

public class MainSerializable {

    private static ArrayList<ContactoSerializable> agenda = new ArrayList<>();

    public static Scanner teclado = new Scanner(System.in);

    private static final String nombreArchivo = "agenda.txt";

    public static void main(String[] args) {

        cargarAgenda();

        while (true) {

            int opcion = menu();

            if (opcion == 4) {

                guardarAgenda();

                break;

            } else {

                switch (opcion) {

                    case 1:

                        try {

                            altaContacto();

                        } catch (Exception e) {

                            System.out.println("Se ha producido un error: " + e.getMessage());

                        }

                        break;

                    case 2:

                        try {

                            bajaContacto();

                        } catch (Exception e) {

                            System.out.println("Se ha producido un error: " + e.getMessage());

                        }

                        break;

                    case 3:

                        try {

                            listarAgenda();

                        } catch (Exception e) {

                            System.out.println("Se ha producido un error: " + e.getMessage());

                        }

```

```

    }
    break;
}
}
}
System.out.println("Salimos");
}

public static int menu() {
    int opcion = 0;
    while (true) {
        System.out.println("AGENDA");
        System.out.println("-----");
        System.out.println("1: Dar de alta un contacto");
        System.out.println("2: Dar de baja un contacto");
        System.out.println("3: Listar contactos");
        System.out.println("4: Salir");
        System.out.println("-----");
        try {
            System.out.print("Opcion: ");
            if (!teclado.hasNextInt()) {
                teclado.nextLine();
                throw new InputMismatchException("\nERROR: INDICA UN NUMERO ENTERO\n");
            }
            opcion = teclado.nextInt();
            teclado.nextLine();
            if (opcion < 1 || opcion > 4) {
                throw new Exception("OPCION INCORRECTA: INDICA UN NUMERO ENTRE 1 Y 4.\n");
            } else {
                return opcion;
            }
        } catch (InputMismatchException e) {
            System.out.println(e.getMessage());
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

```

}
}

public static void altaContacto() throws Exception {
    System.out.println("Indica el número de teléfono");
    String numero = teclado.next();
    ContactoSerializable contacto = buscarContacto(numero);
    if (contacto != null) {
        throw new Exception("ERROR: EL NUMERO YA ESTA EN LA AGENDA");
    }
    System.out.println("Indica el nombre para el contacto.");
    String nombre = teclado.next();
    contacto = new ContactoSerializable(nombre, numero);
    agenda.add(contacto);
    System.out.println("Contacto agregado a la agenda.");
}

public static void bajaContacto() throws Exception {
    if (agenda.isEmpty()) {
        throw new Exception("ERROR: LA AGENDA ESTA VACIA");
    }
    System.out.println("Indica el número de teléfono");
    String numero = teclado.next();
    ContactoSerializable contacto = buscarContacto(numero);
    if (contacto == null) {
        throw new Exception("ERROR: EL NUMERO NO ESTA EN LA AGENDA");
    }
    agenda.remove(contacto);
    System.out.println("Contacto borrado de la agenda.");
}

public static void listarAgenda() throws Exception {
    if (agenda.isEmpty()) {
        throw new Exception("ERROR: LA AGENDA ESTA VACIA");
    } else {
        for (ContactoSerializable c : agenda) {
            System.out.println(c);
        }
    }
}

```

```

}
}

public static ContactoSerializable buscarContacto(String numero) {
    for (ContactoSerializable c : agenda) {
        if (c.getNumero().equals(numero)) {
            return c;
        }
    }
    return null;
}

public static void guardarAgenda() {
    ObjectOutputStream oos = null;
    try {
        oos = new ObjectOutputStream(new FileOutputStream(nombreArchivo));
        oos.writeObject(agenda);
        System.out.println("Agenda guardada correctamente.");
    } catch (IOException e) {
        System.out.println("Error al guardar la agenda.");
        e.printStackTrace();
    } finally {
        if (oos != null) {
            try {
                oos.close();
            } catch (IOException e) {
                System.out.println("Error al cerrar el archivo.");
                e.printStackTrace();
            }
        }
    }
}

public static void cargarAgenda() {
    ObjectInputStream ois = null;
    try {
        File archivo = new File(nombreArchivo);
        if (!archivo.exists()) {

```

```

System.out.println("El archivo de agenda no existe. Se creará uno nuevo al guardar.");
return;
}

ois = new ObjectInputStream(new FileInputStream(nombreArchivo));
Object obj = ois.readObject();
if (obj instanceof ArrayList<?>) {
    ArrayList<?> lista = (ArrayList<?>) obj;
    if (!lista.isEmpty() && lista.get(0) instanceof ContactoSerializable) {
        agenda = (ArrayList<ContactoSerializable>) lista;
        System.out.println("Agenda cargada correctamente.");
    } else {
        throw new ClassNotFoundException("El archivo no contiene una lista de
        ContactoSerializable.");
    }
} else {
    throw new ClassNotFoundException("El archivo no contiene una lista.");
}
} catch (FileNotFoundException e) {
    System.out.println("El archivo de agenda no existe. Se creará uno nuevo al guardar.");
} catch (EOFException e) {
    System.out.println("El archivo de agenda está vacío. Se creará uno nuevo al guardar.");
} catch (IOException | ClassNotFoundException e) {
    System.out.println("Error al cargar la agenda.");
    e.printStackTrace();
} finally {
    if (ois != null) {
        try {
            ois.close();
        } catch (IOException e) {
            System.out.println("Error al cerrar el archivo.");
            e.printStackTrace();
        }
    }
}
}
}
}
}

```



```
}
```

LIBRERIA

--CLASE LIBRO--

```
import java.io.Serializable;

public class Libro implements Serializable{

    private String titulo;

    private String autor;

    private double precio;

    public Libro (String titulo, String autor, double precio) {

        this.titulo=titulo;

        this.autor=autor;

        this.precio=precio;

    }

    public String getTitulo() {

        return titulo;

    }

    public void infoLibro() {

        String formatPrecio= String.format("%.2f", precio

        System.out.println("Datos del libro: "+"\\nTitulo: "+titulo+"\\nAutor: "+autor+"\\nPrecio:

        "+formatPrecio+" euros");

    }

}
```

--CLASE MAIN--

```
public class Main {

    public static Scanner teclado = new Scanner(System.in);

    public static ArrayList<Libro> inventario = new ArrayList<>();

    private static final String nombreArchivo = "inventario.txt";

    public static void main(String[] args) {

        cargarInventario();

        while (true) {

            int opcion = menu();

            switch (opcion) {

                case 1:

                    try {

                        altaLibro();

                    } catch (Exception e) {

                        e.printStackTrace();

                    }

                case 2:

                    bajaLibro();

                case 3:

                    modificarLibro();

                case 4:

                    mostrarInventario();

                case 5:

                    salir();

            }

        }

    }

}
```

```
} catch (Exception e) {  
    System.out.println("Se ha producido un error: " + e.getMessage());  
}  
  
break;  
  
case 2:  
    try {  
        bajaLibro();  
    } catch (Exception e) {  
        System.out.println("Se ha producido un error: " + e.getMessage());  
    }  
  
    break;  
  
case 3:  
    try {  
        listarInventario();  
    } catch (Exception e) {  
        System.out.println("Se ha producido un error: " + e.getMessage());  
    }  
  
    break;  
  
case 4:  
    try {  
        guardarInventario();  
    } catch (Exception e) {  
        System.out.println("Se ha producido un error: " + e.getMessage());  
    }  
  
    System.out.println("Salimos");  
  
    break;  
}  
  
if (opcion == 4) {  
    break;  
}  
  
}  
  
public static int menu() {  
    int opcion = 0;  
    while (true) {
```

```

System.out.println("-----");
System.out.println("LIBRERIA");
System.out.println("-----");
System.out.println("1: Dar de alta un libro");
System.out.println("2: Dar de baja un libro");
System.out.println("3: Listar inventario");
System.out.println("4: Salir");
System.out.println("-----");
try {
    System.out.print("Opcion: ");
    if (!teclado.hasNextInt()) {
        teclado.nextLine();
        throw new InputMismatchException("\nERROR: INDICA UN NUMERO ENTERO\n");
    }
    opcion = teclado.nextInt();
    teclado.nextLine();
    if (opcion < 1 || opcion > 4) {
        throw new Exception("OPCION INCORRECTA: INDICA UN NUMERO ENTRE 1 Y 4.\n");
    } else {
        return opcion;
    }
} catch (InputMismatchException e) {
    System.out.println(e.getMessage());
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}

public static Libro buscarLibro(String titulo) {
    for (Libro libro : inventario) {
        if (libro.getTitulo().equals(titulo)) {
            return libro;
        }
    }
    return null;
}

```

```

}

public static void altaLibro() throws Exception {
    System.out.println("Titulo del libro: ");
    String titulo = teclado.nextLine();
    Libro libro = buscarLibro(titulo);
    if (libro != null) {
        throw new Exception("ERROR: EL LIBRO YA ESTA EN EL INVENTARIO");
    }
    System.out.println("Indica el autor: ");
    String autor = teclado.nextLine();
    System.out.println("Indica el precio: ");
    double precio = teclado.nextDouble();
    libro = new Libro(titulo, autor, precio);
    inventario.add(libro);
    System.out.println("Has agregado el libro al inventario.");
}

public static void bajaLibro() throws Exception {
    if (inventario.isEmpty()) {
        throw new Exception("ERROR: EL INVENTARIO ESTA VACIO");
    }
    System.out.println("Indica el titulo del libro: ");
    String titulo = teclado.nextLine();
    Libro libro = buscarLibro(titulo);
    if (libro == null) {
        throw new Exception("ERROR: EL LIBRO NO ESTA EN EL INVENTARIO");
    }
    inventario.remove(libro);
    System.out.println("Has eliminado el libro del inventario.");
}

public static void listarInventario() throws Exception {
    if (inventario.isEmpty()) {
        throw new Exception("ERROR: EL INVENTARIO ESTA VACIO");
    } else {
        for (Libro libro : inventario) {
            libro.infoLibro();
        }
    }
}

```

```

}
}
}
public static void guardarInventario() {
    ObjectOutputStream oos = null;
    try {
        oos = new ObjectOutputStream(new FileOutputStream(nombreArchivo));
        for (Libro libro : inventario) {
            oos.writeObject(libro);
        }
        System.out.println("Inventario guardado correctamente.");
    } catch (IOException e) {
        System.out.println("ERROR AL GUARDAR EL INVENTARIO");
        e.printStackTrace();
    } finally {
        if (oos != null) {
            try {
                oos.close();
            } catch (IOException e) {
                System.out.println("ERROR AL CERRAR EL ARCHIVO.");
                e.printStackTrace();
            }
        }
    }
}

public static void cargarInventario() {
    ObjectInputStream ois = null;
    try {
        File archivo = new File(nombreArchivo);
        if (!archivo.exists()) {
            System.out.println("El archivo del inventario no existe. Se creara uno nuevo al guardar.");
            return;
        }
        ois = new ObjectInputStream(new FileInputStream(nombreArchivo));
    }
}

```

```

inventario.clear();
while (true) {
    try {
        Libro libro = (Libro) ois.readObject();
        inventario.add(libro);
    } catch (EOFException e) {
        break;
    }
}

System.out.println("Inventario cargado correctamente.");
} catch (FileNotFoundException e) {
    System.out.println("El archivo de inventario no existe. Se creará uno nuevo al
guardar.");
} catch (EOFException e) {
    System.out.println("El archivo de inventario esta vacío. Se creará uno nuevo al
guardar.");
} catch (ClassCastException e) {
    System.out.println("Error: El archivo no contiene una lista de Libros.");
    e.printStackTrace();
} catch (IOException | ClassNotFoundException e) {
    System.out.println("Error al cargar el inventario.");
    e.printStackTrace();
} finally {
    if (ois != null) {
        try {
            ois.close();
        } catch (IOException e) {
            System.out.println("Error al cerrar el archivo.");
            e.printStackTrace();
        }
    }
}
}
}
}
}
}

```