

```

package Examen.vacunas;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;

public class CentroSalud {

    static String rutaFichero = "centrosalud.dat";
    static ArrayList<Paciente> pacientes = null;

    public static void main(String[] args) {
        leerArchivoBinario();

        int opcionMenu = menu();

        while (opcionMenu != 4) {
            switch (opcionMenu) {
                // Funciones segun la opcion del menu
                case 1 :
                    registrarPaciente();
                    break;
                case 2 :
                    mostrarPacientesCompleta();
                    break;
                case 3 :
                    mostrarPacientePendiente();
                    break;
            }
            opcionMenu = menu();
        }
        escribirArchivoBinario();
    }

    private static void leerArchivoBinario() {
        // Leemos el archivo binario
        File fichero = new File(rutaFichero);
        if (fichero.exists()) {
            ObjectInputStream objInputStream = null;
            try {
                objInputStream = new ObjectInputStream(
                    new FileInputStream(fichero));
                pacientes = (ArrayList<Paciente>) objInputStream.readObject();
            } catch (ClassNotFoundException e) {
                System.err.println("Error abriendo el fichero");
            }
        }
    }
}

```

```

        } catch (IOException e) {
            System.err.println("Formato del fichero incorrecto");
        } finally {
            try {
                if (objInputStream != null)
                    objInputStream.close();
            } catch (IOException e) {
                System.err.println("Error cerrando el fichero");
            }
        }
    } else {
        pacientes = new ArrayList<Paciente>();
    }
}

private static void escribirArchivoBinario() {
    // Escribimos el archivo binario
    File fichero = new File(rutaFichero);
    ObjectOutputStream objOutputStream = null;
    try {
        objOutputStream = new ObjectOutputStream(
            new FileOutputStream(fichero));
        objOutputStream.writeObject(pacientes);
    } catch (IOException e) {
        System.err.println("Formato del fichero incorrecto");
    } finally {
        try {
            if (objOutputStream != null)
                objOutputStream.close();
        } catch (IOException e) {
            System.err.println("Error cerrando el fichero");
        }
    }
}

private static int menu() {
    // Texto y petición de opción del menú
    String textoMenu = "*****MENU*****\n" + "1. Registrar pacientes \n"
        + "2. Mostrar pacientes con dosis completa \n"
        + "3. Mostrar pacientes pendiente de segunda dosis \n"
        + "4. Salir \n";
    int opcionMenu = Teclado.leerEntero(textoMenu, 1, 4);
    return opcionMenu;
}

private static void registrarPaciente() {

```

```

// Registramos el paciente, pedimos todos los datos, hay 2 funciones al
// final para validar telefono y fecha
String nombreApellidos = "";
nombreApellidos = Teclado.leerTexto("Indica el nombre del paciente");
String numeroTelefono = "";
boolean formatoTelefonoCorrecto = false;
while (!formatoTelefonoCorrecto) {
    numeroTelefono = Teclado
        .leerTexto("Indica el telefono del paciente");
    if (comprobarFormatoTelefono(numeroTelefono)) {
        formatoTelefonoCorrecto = true;
    }
}
String fechaPrimeraDosis = "";
boolean formatoFechaCorrecto = false;
while (!formatoFechaCorrecto) {
    fechaPrimeraDosis = Teclado.leerTexto(
        "Indica la fecha de la primera dosis (Formato dd/mm/aaaa)");
    if (comprobarFormatoFecha(fechaPrimeraDosis)) {
        formatoFechaCorrecto = true;
    }
}
String nombreVacuna = "";
nombreVacuna = Teclado.leerTexto("Indica el nombre de la vacuna");
int opcionVacuna = Teclado.leerEntero("Indica tipo de vacuna \n"
    + "1 = Vacuna Monodosis\n" + "2 = Vacuna Multidosis", 1, 2);
Paciente nuevoPaciente = null;
Vacuna nuevaVacuna = null;
switch (opcionVacuna) {
    case 1 :

        // Vacuna monodosis
        // Si es monodosis instanciamos Vacuna como monodosis
        nuevaVacuna = new VacunaMonodosis(nombreVacuna,
            fechaPrimeraDosis);
        nuevoPaciente = new Paciente(nombreApellidos,
numeroTelefono,
            nuevaVacuna);
        pacientes.add(nuevoPaciente);
        break;
    case 2 :
        // Vacuna multidosis
        // Si es multidosis instanciamos Vacuna como multidosis
        int semanasParaSegundaVacuna = Teclado.leerEntero(
            "Indica el numero de semanas para la siguiente dosis")
;

        nuevaVacuna = new VacunaMultidosis(nombreVacuna,
            fechaPrimeraDosis, semanasParaSegundaVacuna);
        nuevoPaciente = new Paciente(nombreApellidos,

```

```

numeroTelefono,
        nuevaVacuna);
    pacientes.add(nuevoPaciente);
    break;
}

}

private static void mostrarPacientesCompleta() {
    // Recorremos el ArrayList y mostramos los que tienen pauta completa
    for (Paciente valor : pacientes) {
        if (valor.getVacuna() instanceof VacunaMonodosis) {
            System.out.println(valor.toString());
        } else if (valor.getVacuna() instanceof VacunaMultidosis
            && ((VacunaMultidosis) valor.getVacuna())
                .comprobarFechaSegundaDos()) {
            String vacunaMultidosis = valor.toString();
            vacunaMultidosis += "\n Tipo de vacuna: Vacuna Multidosis\n"
                + "Fecha Segunda Dosis: "
                + ((VacunaMultidosis) valor.getVacuna())
                    .getFechaSegundaVacunaString()
                + "\n-----";
            System.out.println(vacunaMultidosis);
        }
    }
}

private static void mostrarPacientePendiente() {
    // Recorremos el ArrayList y mostramos los que les falta la segunda
    // dosis
    for (Paciente valor : pacientes) {
        if (valor.getVacuna() instanceof VacunaMultidosis
            && !((VacunaMultidosis) valor.getVacuna())
                .comprobarFechaSegundaDos()) {
            String vacunaMultidosis = valor.toString();
            vacunaMultidosis += "\n Tipo de vacuna: Vacuna Multidosis\n"
                + "Fecha Segunda Dosis: "
                + ((VacunaMultidosis) valor.getVacuna())
                    .getFechaSegundaVacunaString()
                + "\n-----";
            System.out.println(vacunaMultidosis);
        }
    }
}

private static boolean comprobarFormatoTelefono(String telefono) {

```

```

        if (telefono.matches("[6-7]{1}[0-9]{8}")) {
            System.out.println("Formato telefono correcto");
            return true;
        } else {
            System.err.println("Formato de telefono incorrecto");
        }
        return false;
    }

    private static boolean comprobarFormatoFecha(String fecha) {
        if (fecha.matches("[0-9]{2}/[0-9]{2}/[0-9]{4}")) {
            System.out.println("Formato fecha correcto");
            return true;
        } else {
            System.err.println("Formato de fecha incorrecto");
        }
        return false;
    }
}

```

```
package Examen.vacunas;
import java.util.Calendar;

public class Fecha {
    static Calendar cal = Calendar.getInstance();
    static int dia = cal.get(Calendar.DAY_OF_MONTH);

    static int mes = cal.get(Calendar.MONTH);
    static int year = cal.get(Calendar.YEAR);
    static int diasSegundaFecha;
    static int mesSegundaFecha;
    static int yearSegundaFecha;
    static Calendar fecha2;

    public static Calendar sumarDias(int dia, int mes, int anio, int dias) {
        Calendar fechaCalendar = Calendar.getInstance();
        fechaCalendar.set(anio, mes, dia);
        fechaCalendar.add(Calendar.DAY_OF_YEAR, dias);
        return fechaCalendar;
    }
}
```

```

package Examen.vacunas;
import java.io.Serializable;

public class Paciente implements Serializable {
    private String nombreApellidos;
    private String telefono;
    private Vacuna vacuna;

    public Paciente(String nombreApellidos, String telefono, Vacuna vacuna) {
        this.nombreApellidos = nombreApellidos;
        this.telefono = telefono;
        this.vacuna = vacuna;
    }

    public String getNombreApellidos() {
        return nombreApellidos;
    }

    public void setNombreApellidos(String nombreApellidos) {
        this.nombreApellidos = nombreApellidos;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    public Vacuna getVacuna() {
        return vacuna;
    }

    public void setVacuna(Vacuna vacuna) {
        this.vacuna = vacuna;
    }

    @Override
    public String toString() {
        String toString = "\nPaciente" + "\n-----"
            + "\n Nombre Apellidos: " + getNombreApellidos()
            + "\n Telefono: " + getTelefono() + "\n Nombre Vacuna: "
            + getVacuna().getNombre() + "\n Fecha primera dosis "
            + getVacuna().getFecha() + "\n-----";
        return toString;
    }
}

```

```

package Examen.vacunas;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Teclado {

    static Scanner sc = new Scanner(System.in);
    static Scanner scString = new Scanner(System.in);

    public static int leerEntero(String mensaje, int min, int max) {
        int numero = 0;
        boolean respuestValida = false;
        System.out.println(mensaje);
        while (!respuestValida) {
            try {
                numero = sc.nextInt();
                if (numero >= min && numero <= max) {
                    respuestValida = true;
                } else {
                    System.err.println(
                        "Introduce un numero entre " + min + " y " + max);
                }
            } catch (InputMismatchException e) {
                sc.nextLine();
                System.err.println("Introduce un numero entero");
            }
        }
        return numero;
    }

    public static int leerEntero(String mensaje) {
        int numero = 0;
        boolean respuestValida = false;
        System.out.println(mensaje);
        while (!respuestValida) {
            try {
                numero = sc.nextInt();
                respuestValida = true;
            } catch (InputMismatchException e) {
                sc.nextLine();
                System.err.println("Introduce un numero entero");
            }
        }
        return numero;
    }

    public static String leerTexto(String mensaje) {
        String texto;
        System.out.println(mensaje);
    }
}

```



```
        texto = scString.nextLine();  
        return texto;  
    }  
}
```

```
package Examen.vacunas;
import java.io.Serializable;

public class Vacuna implements Serializable {
    protected String nombre;
    protected String fecha;

    public Vacuna(String nombre, String fecha) {
        this.nombre = nombre;
        this.fecha = fecha;
    }
    // Getters y setters
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getFecha() {
        return fecha;
    }
    public void setFecha(String fecha) {
        this.fecha = fecha;
    }
}
```

```
package Examen.vacunas;

public class VacunaMonodosis extends Vacuna {

    public VacunaMonodosis(String nombre, String fecha) {
        super(nombre, fecha);
    }

}
```

```

package Examen.vacunas;
import java.text.SimpleDateFormat;
import java.util.Calendar;

public class VacunaMultidosis extends Vacuna {
    private int semanasSegundaVacuna;
    private String fechaSegundaVacunaString = "";

    // Constructor
    public VacunaMultidosis(String nombre, String fecha,
        int semanasSegundaVacuna) {
        super(nombre, fecha);
        this.semanasSegundaVacuna = semanasSegundaVacuna;
        // TODO Auto-generated constructor stub
    }

    public boolean comprobarFechaSegundaDosis() {
        // Comprobamos si se ha pasado la fecha de la segunda dosis
        int dia = extraerDiaStringFecha(fecha);
        int mes = extraerMesStringFecha(fecha);
        int year = extraerYearStringFecha(fecha);
        int diasParaSegundaVacuna = semanasSegundaVacuna * 7;

        SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd");
        Fecha.sumarDias(dia, mes, year, diasParaSegundaVacuna);
        fechaSegundaVacunaString = format1
            .format(Fecha.sumarDias(dia, mes, year,
                diasParaSegundaVacuna));
        if (Fecha.cal.after(Fecha.fecha2)) {
            System.out.println(Fecha.cal.after(Fecha.fecha2));
            return true;
        } else {
            System.out.println(Fecha.cal.after(Fecha.fecha2));
            return false;
        }
    }

    public boolean comprobarFechaSegundaDos() {
        // Comprobamos si se ha pasado la fecha de la segunda dosis
        int dia = extraerDiaStringFecha(fecha);
        int mes = extraerMesStringFecha(fecha);
        int year = extraerYearStringFecha(fecha);
        int diasParaSegundaVacuna = semanasSegundaVacuna * 7;

        Calendar fechaSegundaVacuna = Fecha.sumarDias(dia, mes, year,
            diasParaSegundaVacuna);
        int diaSegundaVacuna = fechaSegundaVacuna.get(Calendar.
            DAY_OF_MONTH);
    }
}

```

```

        int mesSegundaVacuna = fechaSegundaVacuna.get(Calendar.MONTH);
        int yearSegundaVacuna = fechaSegundaVacuna.get(Calendar.YEAR);
        fechaSegundaVacunaString = diaSegundaVacuna + "/" +
mesSegundaVacuna
        + "/" + yearSegundaVacuna;
        if (Fecha.cal.after(fechaSegundaVacuna)) {
            return true;
        } else {
            return false;
        }
    }

    private int extraerDiaStringFecha(String fecha) {
        String dia = fecha.substring(0, 2);
        int diaNumero = Integer.parseInt(dia);
        return diaNumero;
    }

    private int extraerMesStringFecha(String fecha) {
        String mes = fecha.substring(3, 5);
        int mesNumero = Integer.parseInt(mes);
        return mesNumero;
    }

    private int extraerYearStringFecha(String fecha) {
        String year = fecha.substring(6, 10);
        int yearNumero = Integer.parseInt(year);
        return yearNumero;
    }

    public int getSemanasSegundaVacuna() {
        return semanasSegundaVacuna;
    }

    public void setSemanasSegundaVacuna(int semanasSegundaVacuna) {
        this.semanasSegundaVacuna = semanasSegundaVacuna;
    }

    public String getFechaSegundaVacunaString() {
        return fechaSegundaVacunaString;
    }

    public void setFechaSegundaVacunaString(String fechaSegundaVacunaString) {
        this.fechaSegundaVacunaString = fechaSegundaVacunaString;
    }
}

```