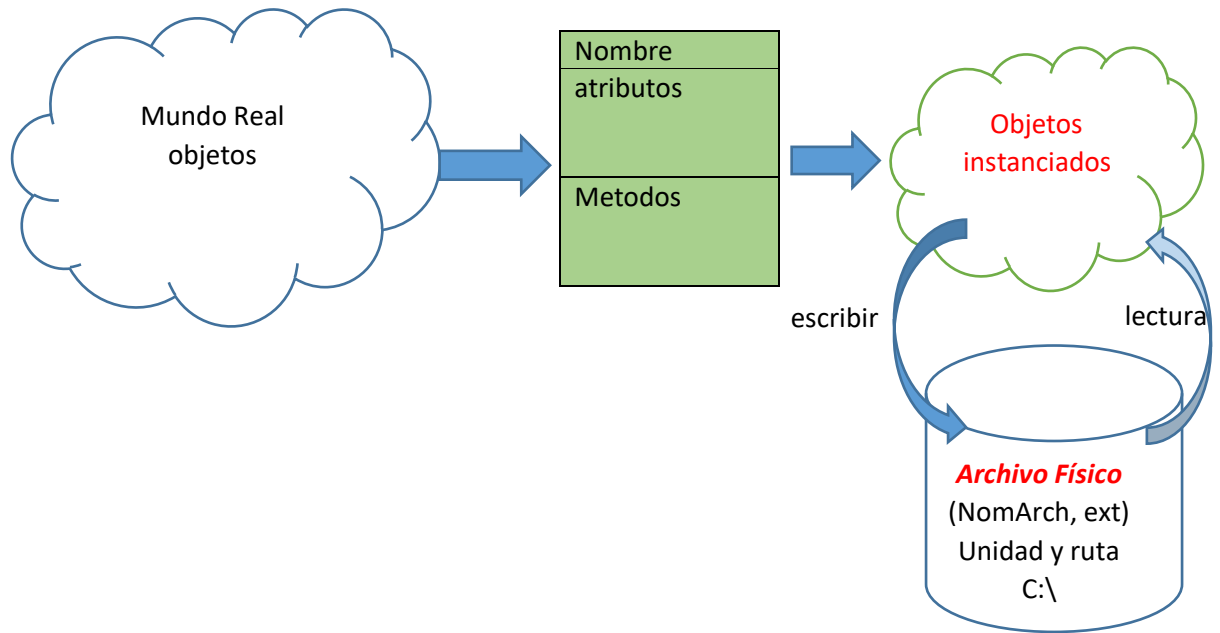


PERSISTENCIA DE OBJETOS



Archivo

Atributo1	Atributo2	AtributoN
.....			

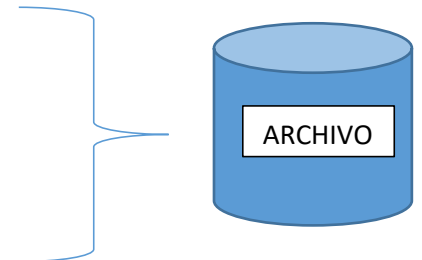
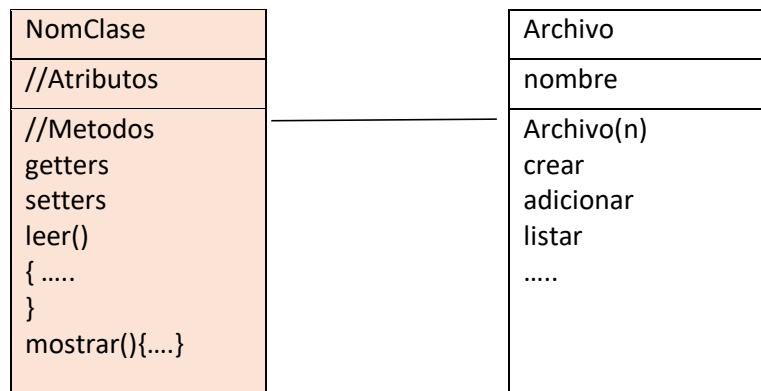
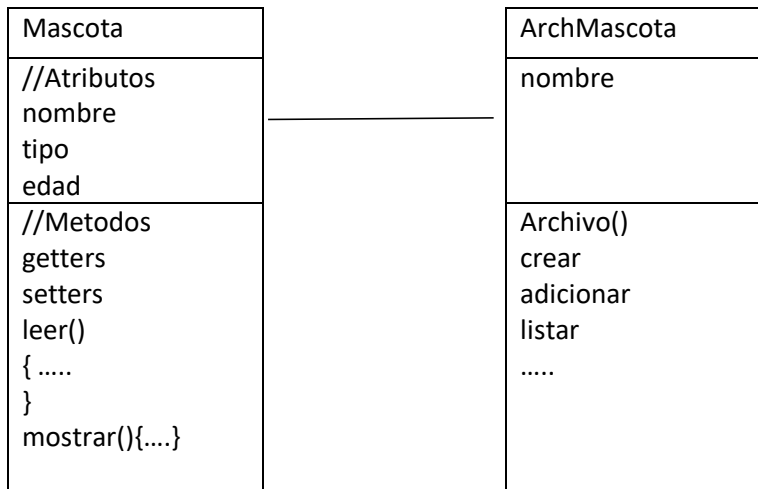


DIAGRAMA DE CLASES

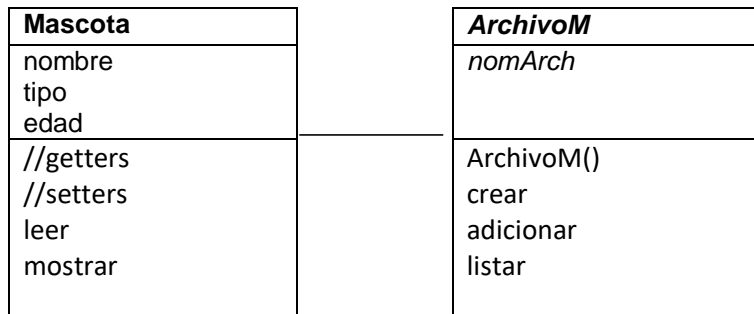


EJEMPLO: Archivo para almacenar datos de objetos Mascota



Ejemplo en JAVA

Considerando el almacenamiento de mensajes en un archivo, se tiene el siguiente diagrama de clases



PASO 1: Implementar la clase Mascota

- **Agregar en la cabecera:**
import java.util.*;
import java.io.Serializable;
- **Agregar en la clase el atributo serializable para que la clase se pueda serializar**
public class NomClase implements Serializable {

PASO 2: Implementar la clase ArchivoE

- **Agregar en la cabecera:**

<pre>import java.io.*; import java.util.Scanner;</pre>	<pre>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.Scanner;</pre>
--	---

- **El atributo nombre es tipo String, para almacenar la unidad y la ruta donde se almacenará el archivo.**
- **El constructor recibe un parámetro tipo cadena para inicializar nombre**
- **Implementar los métodos Crear, adicionar y listar**

<pre>class ArchivoM { private String nomArch; public ArchivoM(String nom) { nomArch= nom; } }</pre>	<pre>public void crear() throws ClassNotFoundException, IOException { Scanner sc=new Scanner(System.in); File archivoFisico = new File(nomArch); try{ if(!archivoFisico.exists()){ ObjectOutputStream archi = new ObjectOutputStream(new FileOutputStream(nomArch)); archi.close(); System.out.println("Archivo creado"); } else{ System.out.println("El archivo ya existe...!! "); System.out.println("Desea borrar sus datos??? s/n"); String resp=sc.next(); if(resp.equals("s")){ ObjectOutputStream archi = new ObjectOutputStream(new FileOutputStream(nomArch)); archi.close(); } } } catch(IOException e){ System.out.println("Error al crear.... "+e.getMessage()); } }</pre>
<pre>public void adicionar() throws ClassNotFoundException, IOException { Scanner sc=new Scanner(System.in); String resp; ObjectOutputStream archi=null; try{ if(new File(nomArch).exists()) { archi = new ObjectOutputStream(new FileOutputStream(nomArch,true)); } } }</pre>	<pre>public void listado()throws IOException, ClassNotFoundException { ObjectInputStream archi=null; try{ archi=new ObjectInputStream(new FileInputStream(nomArch)); while(true){ Mascota x=(Mascota)archi.readObject(); x.mostrar(); } } }</pre>

<pre> do{ Mascota x=new Mascota(); x.leer(); archi.writeObject(x); System.out.println("Desea continuar s/n"); resp=sc.next(); }while(resp.equals("s")); } else System.out.println("No existe el archivo.... debe crear!!!"); } catch(Exception e){ System.out.println("error de archivo"); } finally{ archi.close(); } } </pre>	<pre> } catch(Exception e){ System.out.println("Fin listado!!!"); } finally{ archi.close(); } } </pre>
---	--

Paso 3: Copiar o crear la clase

```

import java.io.*;

public class AddObjectOutputStream extends ObjectOutputStream {
    public AddObjectOutputStream(OutputStream out) throws IOException{
        super(out);
    }
    protected void writeStreamHeader() throws IOException{
        this.reset();
    }
}

```

Paso 4: PROGRAMA PRINCIPAL CON UN MENU PARA EL MANEJO DEL ARCHIVO

```

import java.io.*;
import java.util.*;
public class ArchivosJava {

    public static void main(String[] args) throws IOException, ClassNotFoundException {
        boolean sw=true;
        Scanner sc=new Scanner(System.in);
        ArchMascota miArch=new ArchMascota("misascotas.dat");
        while(sw)
        {
            System.out.println("1. Crear");
            System.out.println("2. Adicionar");
            System.out.println("3. Listar");
            System.out.println("0. Salir ");
            System.out.println("Elegir una opcion: ");
            String resp=sc.next();
            switch(resp){
                case "1": miArch.crear();break;
                case "2": miArch.adicionar();break;
                case "3": miArch.listado();break;
                case "4": {
                    System.out.print("Tipo : ");
                    String tip=sc.next();
                    miArch.tipoX(tip);}break;
            }
        }
    }
}

```

```

        case "5": {
            System.out.print("Tipo : ");
            String tip=sc.next();
            miArch.contar("gato");}break;
        default: sw=false;
    }
}
}
}
}

```

OTROS METODOS EN LA CLASE ARCHIVO

<pre> //MOSTRAR LAS MASCOTAS TIPO X public void tipoX(String tx)throws IOException, ClassNotFoundException { ObjectInputStream archi=null; try{ archi=new ObjectInputStream(new FileInputStream(nomArch)); while(true){ Mascota x=(Mascota)archi.readObject(); if(x.getTipo().equals(tx)) x.mostrar(); } } catch(Exception e){ System.out.println("Fin listado!!!"); } finally{ archi.close(); } } </pre>	<pre> //CONTAR LAS MASCOTAS TIPO X public void contar(String tx)throws IOException, ClassNotFoundException { ObjectInputStream archi=null; int c=0; try{ archi=new ObjectInputStream(new FileInputStream(nomArch)); while(true){ Mascota x=(Mascota)archi.readObject(); if(x.getTipo().equals(tx)) c++; } } catch(Exception e){ System.out.println("Fin listado!!!"); } finally{ archi.close(); System.out.println("Cantidad: "+c); } } </pre>
<pre> //ELIMINAR A LA MASCOTA CON NOMBRE X public void eliminar(String m)throws IOException, ClassNotFoundException { ObjectInputStream archi=null; ObjectOutputStream temporal=null; try{ archi=new ObjectInputStream(new FileInputStream(nomArch)); temporal=new ObjectOutputStream(new FileOutputStream("temporal.dat",true)); while(true){ Mascota zx=(Mascota)archi.readObject(); if(!zx.getNombre().equals(m)) temporal.writeObject(zx); } } catch(Exception e){ System.out.println("Fin listado!!!"); } finally{ temporal.close(); archi.close(); File f1=new File(nomArch); File f2=new File("temporal.dat"); f1.delete(); f2.renameTo(f1); } } </pre>	<pre> //MODIFICAR A LA MASCOTA CON NOMBRE X POR UNA NUEVA MASCOTA public void modificar(String m, Mascota nuevaM)throws IOException, ClassNotFoundException { ObjectInputStream archi=null; ObjectOutputStream temporal=null; try{ archi=new ObjectInputStream(new FileInputStream(nomArch)); temporal=new ObjectOutputStream(new FileOutputStream("temporal.dat",true)); while(true){ Mascota zx=(Mascota)archi.readObject(); if(zx.getNombre().equals(m)) temporal.writeObject(nuevaM); else temporal.writeObject(zx); } } catch(Exception e){ System.out.println("Fin listado!!!"); } finally{ temporal.close(); archi.close(); File f1=new File(nomArch); File f2=new File("temporal.dat"); f1.delete(); f2.renameTo(f1); } } </pre>

PROBLEMAS:

1. Mostrar

EJEMPLO: Sea un archivo para almacenar la evaluación de estudiantes

Evaluacion	ArchEval
nom pat mat parcial1 parcial2 exFinal auxi prac	<i>nom</i>
//getters //setters leer mostrar	ArchEval() crear adicionar listar

PROBLEMAS

1. Crear, Adicionar y Listar
2. Mostrar la nota total de cada Estudiante
3. Mostrar a los mejores estudiantes
4. ¿Cuántos aprobados y reprobados existe?
5. Verificar si el estudiante Pepito Perez Perez aprobó o reprobó
6. Mostrar a los mejores estudiantes

EJEMPLO: Sea un Archivo que almacena la información de las provincias del país

Provincia	ArchProvincia
departamento nomProvincia superficie poblacion	<i>nomprov</i>
//getters //setters leer mostrar	ArchProvincia() crear adicionar listar //prov del dpto. X dptoX //menor poblacio menor

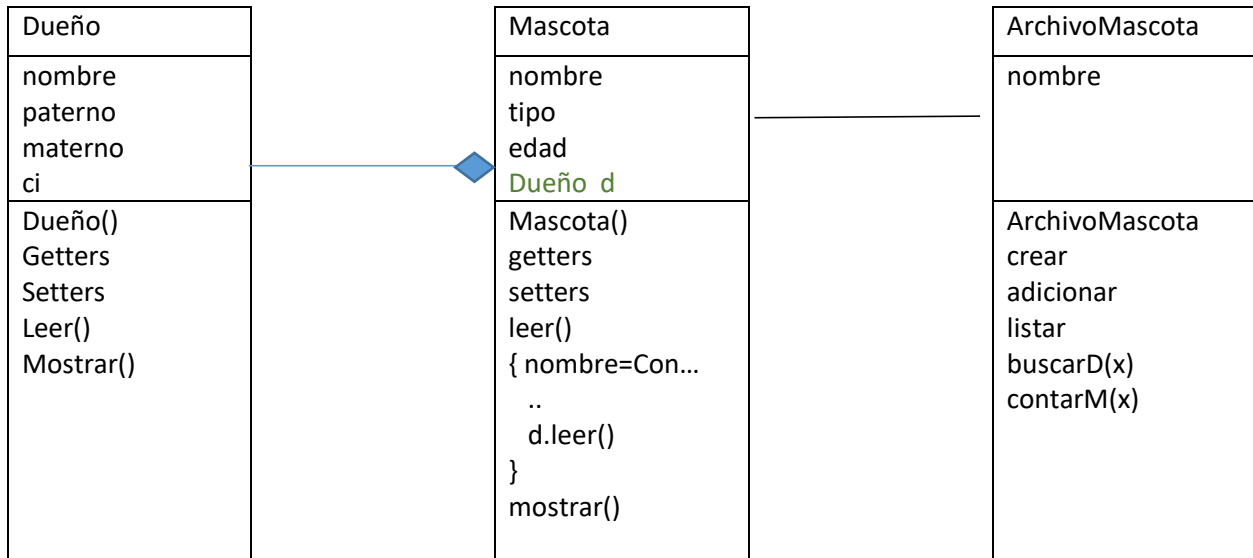
PROBLEMAS

1. Crear, adicionar y listar
2. Mostrar las provincias del departamento X
3. Mostrar las provincias con menor población

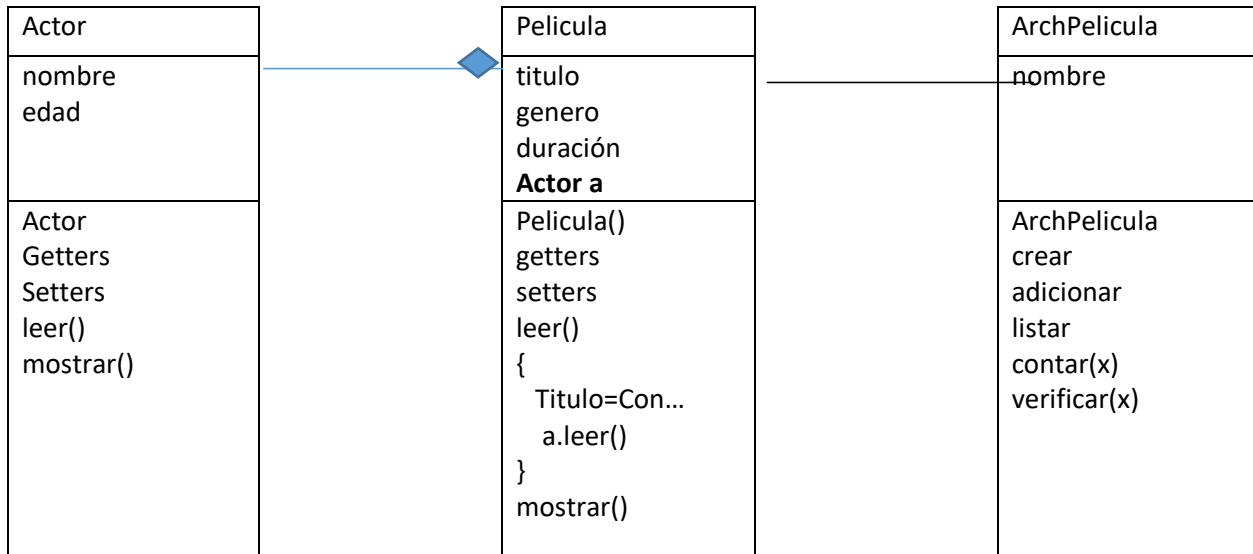
4. Verificar si la provincia X pertenece al departamento de La Paz

EJEMPLO

DIAGRAMA DE CLASES



EJEMPLO



Ejercicios

1. Mostrar las películas del genero x
2. Mostrar las películas con duración mayor 120

Contar las películas con duración menor a x
Mostrar las películas con mayor duración
Mostrar las películas con menor duración

Ejemplo: Sea un archivo que almacena la información de los proyectos de curso

Proyecto	ArchProy
titulo nombre p1 p2 pf	nombre
Proyecto() getters setters leer() { } mostrar()	ArchProy crear adicionar listar contar(x) verificar(x)

titulo	nombre	P1	P2	Pf
cultura	juan	5	5	5

Defensa	ArchDef
nro tema nombre ndef //s 8pts ninf // s 2pts	nombre
getters setters leer() mostrar() ...	ArchDef(n) crear() adicionar() listar()