```
--------------------------------RANDOM--------------------------

public static int random(){
      Random r = new Random();
      int random = r.nextInt(6)+1;
      return random;
}

//otra forma de hacerlo

public int random(int multiplicador){

return (int)(Math.random()*multiplicador);
}


--------------------------------SCANNER--------------------------

public static scanner(){
      Scanner sc = new Scanner(System.in);
      int num= sc.nextInt();
      return num;
}


-------------------------recorrer Array-------------------------

public static void listar(int [] Array){

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

      System.out.println();

      }

}

-------------------------recorrer Array 2D----------------------

public static void listar(int[][]Array){

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

         System.out.println();
         }
      }}
```

```
-------------------------Bubble Sort------------------------------
int temporal=0;
for (int i =b.length; i >0 ; i--) {
            for (int j = 0; j < b.length-1; j++) {
                if (b[j]>b[j+1]) {
                    temporal=b[j];
                    b[j]=b[j+1];
                    b[j+1]=temporal;
                }
            }
        }



------------------------Menu
Switch---------------------------------
Scanner sc = new Scanner(System.in);
int choice;
     do{
            choice=sc.nextInt();
            switch(choice){
             case 0: System.out.println("Bye") ;
              break;
             case 1: ;
              break;
             case 2: ;
              break;
             case 3: ;
              break;
             case 4: ;
              break;
          default: System.out.println("puso un numero
incorrecto");
          }

          }while(choice!=0);




-----------------MENU MOSTRAR--------------------
    private static void menu() {
        System.out.println("----MENU-----\n"+
                           "1- \n"+
                           "2- \n"+
                           "3- \n"+
                           "4- \n"+
```

```
                            "5- \n");
 }

------------------MIN MAX-----------------------

public static void min_max(int[]Array) {
    int min=Array[0], max=Array[0];

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

           if(min<Array[i]){min=Array[i];};
           if(max>Array[i]){max=Array[i];};

            System.out.println("min -->"+min+"\n max -->"+max);

        }


    }

-----------------------------------------------sort--------
 Pokemon implements Serializable, Comparable<Pokemon>

@Override
    public int compareTo(Pokemon t) {
    int compare=this.nombre.compareTo(t.getNombre());
    if(compare!=0){
        return compare;
    }else{
        return Integer.compare(this.CP, t.getCP() );
    }

    }

private void sortBag(ArrayList<Pokemon> bag) {// clase pokemonDAO
     Collections.sort(bag);

    }
```

--
CHULETA

--
length(); devuelve la longitud de cadena

indexOF('char'); devuelve la posicion de la primera aparicion del char

lastIndexOf('char'); devuelve la posicion de la ultima aparicion del char

charAt(n); devuelve el caracter que esta en la posicion n

substring(n1, n2); devuelve la subcadena comprendida entre las pocisiones n1 y n2-1

toUpperCase(); devuelve en mayusculas

toLowerCase(); devuelve en minusculas

equals("string"); compara dos cadenas y devuelve si son iguales

equalsIgnoreCase("string"); compara dos cadenas ignorando si es lower o upper

comapreTo(OtroString); devuelve 0 si las dos cadenas son iguales. <0 si la primera es alfabeticamente menor que la seguda o >0 si la primera es alfabeticamente superior a segunda.

compareToIgnoreCase(OtroString); Igual que compare to pero ignore case

valueOf(N); metodo estatico. convierte el valor N a string. N puede ser de cualquier tipo

charAt();
sc.next().charAt(0);
public static Scanner sc = new Scanner(System.in);

TEMA FICHEROS

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package probaficheros;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.InputMismatchException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author mati
 */
public class ProbaFicheros {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {
            Scanner sc=new Scanner(System.in);

    int choice;
        do {
            System.out.println("1 leer 2 escrivir 3 leerStrings  4 escribir Strings 0 salir");
            choice=sc.nextInt();

            if (choice==1) {
                leer_numero();
            }
            else if(choice==2){
            escrivir_numero();
            }
```

```
         else if(choice==3){
        leer_string();
        }
        else if(choice==4){
        escrivir_Strings();
        }

    } while (choice!=0);



}

public static void leer_numero(){

  try{
    File f=new File("numero.txt");
    Scanner lector_f=new Scanner(f);
    while(lector_f.hasNext()){
        int valor=lector_f.nextInt();
        System.out.println("he leido "+valor);

    }
    lector_f.close();

    } catch (FileNotFoundException ex) {
        Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
    }
    catch(InputMismatchException ex){
        System.out.println("he leido algo que no es un numero");
    }
}

public static void escrivir_numero() throws IOException{
   Scanner sc=new Scanner(System.in);
   File f=new File("numero.txt");
   FileWriter fw=new FileWriter(f, true);
   int valor;
    do {
        System.out.println("pon cualquier numero y -1 para parar");
       valor=sc.nextInt();
        if (valor!=-1) {
            fw.write(valor+" ");
        }
    } while (valor!=-1);
     fw.close();
```

```
    }

    private static void leer_string() {


      try{
        File f=new File("hola.txt");
        Scanner lector_f=new Scanner(f);
        while(lector_f.hasNext()){
            String frase=lector_f.nextLine();
            System.out.println("he leido "+frase);

        }
        lector_f.close();

        } catch (FileNotFoundException ex) {
            Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        }
        catch(InputMismatchException ex){
            System.out.println("he leido algo que no es un numero");
        }
    }

    private static void escrivir_Strings() {
        try {
       Scanner sc=new Scanner(System.in);
       File f=new File("hola.txt");
       FileWriter fw;

            fw = new FileWriter(f, true);

       String frase;
       do {
            System.out.println("escribe fin para termianar");
           frase=sc.nextLine();
            if (!frase.equals("fin")) {
                fw.write(frase+"\n");
            }
        } while (!frase.equals("fin"));
       fw.close();

        } catch (IOException ex) {
            Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```
--------------------Lector bytes----------------------
  public static void leerBinario(String rutaFile){

      int cont=0, byteLeido;
      boolean finFile=false;
      try{
      FileInputStream FicheroLectura =new FileInputStream(rutaFile);
          do {
            byteLeido=FicheroLectura.read();
             if (byteLeido==-1) {
                finFile=true;

             }else{

                System.out.println("byte leido "+byteLeido);
                cont++;
             }

          } while (!finFile);
          FicheroLectura.close();
          System.out.println("bytes leidos "+cont);



      } catch (FileNotFoundException ex) {
           Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
       } catch (IOException ex) {
           Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
       }
  }
```

---------Escribit binario--------------------

```
  public static void escribirBinario(String rutaFile){

      int cont=0, byteLeido;
      boolean finFile=false;
      FileInputStream fileLectura=null;
      FileOutputStream fileCopia=null;

      try{
       fileLectura =new FileInputStream(rutaFile);
      fileCopia=new FileOutputStream("directorio/file.xd");
          do {
            byteLeido=fileLectura.read();
             if (byteLeido==-1) {
                finFile=true;

             }else{
                fileCopia.write(byteLeido);
                System.out.println("byte leido "+byteLeido);
                cont++;
             }

          } while (!finFile);
          fileLectura.close();
          System.out.println("bytes leidos "+cont);



      } catch (FileNotFoundException ex) {
          Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
       } catch (IOException ex) {
          Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
       }
  }
```

----------OBJETO SERIALIZBLE----------

```
public static void binarioObjetoSerializable(){ //class implements serializable deberia ser una
class
```

```
  ArrayList<Amigo>agenda=new ArrayList<>();
  agenda.add(new Amigo("Paco", 12));
  agenda.add(new Amigo("Lusia", 42));
  agenda.add(new Amigo("Pol pot", 62));
  agenda.add(new Amigo("Una vaca que rie", 22));

  escribirFicheroBinario(agenda);

   ArrayList<Amigo>copiaAgenda=new ArrayList<>();
   copiaAgenda=lecturaFicheroBinario(copiaAgenda);

      for (int i = 0; i < copiaAgenda.size(); i++) {
          System.out.println("nombre "+copiaAgenda.get(i).getNombre());
      }
  }

private static void escribirFicheroBinario(ArrayList<Amigo> agenda) {
  FileOutputStream escritura= null;
  ObjectOutputStream StreamDatosEscritura=null;


    try{
     escritura=new FileOutputStream("file.xd");
     StreamDatosEscritura=new ObjectOutputStream(escritura);

   /**     for (int i = 0; i < 10; i++) {
           StreamDatosEscritura.writeObject(agenda.get(i));
        }**/

   StreamDatosEscritura.writeObject(agenda);
   StreamDatosEscritura.close();

    } catch (FileNotFoundException ex) {
        Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        System.out.println("error escritura");
    } catch (IOException ex) {
        Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        System.out.println("fichero no encontrado");
    }


}

private static ArrayList<Amigo> lecturaFicheroBinario(ArrayList<Amigo> copiaAgenda) {

    FileInputStream fileLectura=null;
    ObjectInputStream fichero=null;
    try{
```

```
        fileLectura=new FileInputStream("elmismoFIleDeAntes.xd");
        fichero=new ObjectInputStream(fileLectura);
        copiaAgenda=(ArrayList<Amigo>) fichero.readObject();


        } catch (FileNotFoundException ex) {
            Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
        }

        try {
            fichero.close();
        } catch (IOException ex) {
            Logger.getLogger(ProbaFicheros.class.getName()).log(Level.SEVERE, null, ex);
            System.out.println("error cerrando fichero");
        }
return copiaAgenda;
    }


}
```

----------Objeto Amigos---------

```
public class Amigo {
private String nombre;
private int edad;

    public String getNombre() {
        return nombre;
    }




    public Amigo(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }



}
```