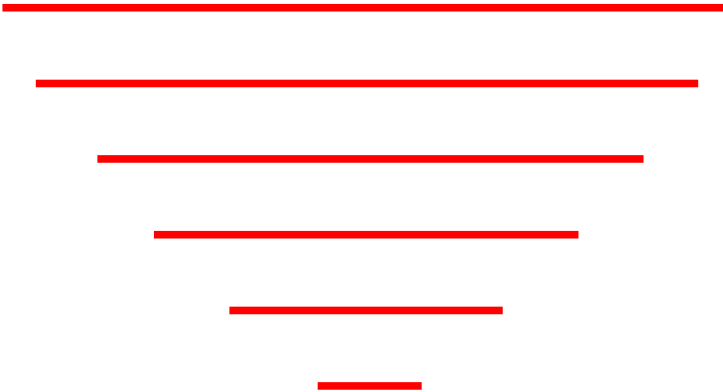




Apuntes de guerrilla



Index

| | |
|---------------------------------|--------------|
| Tablas | 3-4 |
| Variables | 4-5 |
| Clases | 5-6 |
| Objetos | 6 |
| Métodos | 7-8 |
| Sentencias y estructuras | 9 |
| Vectores | 10 |
| Matrices | 10 |
| Códigos útiles cortos | 11-13 |
| Códigos de ejemplo | 14-21 |

Tablas

Tabla de tipos de datos

| Tipo de datos | Información representada | Rango | Descripción |
|---------------|-----------------------------------|---|---|
| Byte | Datos enteros | -128 a +127 | Se utilizan 8 bits (1 byte) |
| Short | Dato enteros | -32768 a +32767 | Dato de 16 bits |
| Int | Datos enteros | -2147483648 a +2147483647 | Dato de 32 bits |
| Long | Datos enteros | -9223372036854775808 a +9223372036854775807 | Dato de 64 bits |
| Char | Datos enteros y cácteres | 0 a 65535 | Juego de caracteres ASCII |
| Float | Datos en coma flotante de 32 bits | Precisión aproximada de 7 dígitos | Coma flotante de 32 bits (1 bit signo, 8 el exponente y 24 mantisa) |
| Double | Datos en coma flotante de 64 bits | Precision aproximada de 16 dígitos | Coma flotante de 64 bits (1 bit signo, 11 exponente y 52 mantisa) |
| Boolean | Valores booleanos | True/false | |

Tabla de operadores aritméticos

| Operador | Uso | Operación |
|----------|-----|----------------|
| + | A+B | Suma |
| - | A-B | Resta |
| * | A*B | Multiplicación |
| / | A/B | División |
| % | A%B | Módulo o Resto |

Tabla de operadores racionales

| Operador | Uso | Operación |
|----------|--------|---------------------------|
| < | A<B | A menor que B |
| > | A>B | A mayor que B |
| <= | A<=B | A mayor o igual que B |
| >= | A>=B | A mayor que o igual que B |
| != | A!=B | A distinto que B |
| == | A == B | A igual que B |

Tabla de operadores lógico

| Operador | Uso | Operación |
|----------|----------------|---|
| && o & | A&& B o A&B | A AND B.El resultado será true si ambos son true sino false |
| o | A B o A B | A OR B.EL resultado será false si ambos son false sino true |
| ! | !A | Not A.Se invierte el resultado del operando. True es false. |
| ^ | A^B | A XOR B. True si un operando es true y el otro false. |

Tabla de operadores bits

| Operador | Uso | Operación |
|----------|-------|-------------|
| = | A = B | Asignación. |
| *= | A *=B | A=A*B |
| /= | A/=B | A=A/B |
| %= | A%=B | A=A%B |
| += | A+=B | A=A+B |
| -= | A-=B | A=A-B |

Tabla de operadores unitarios

| Operador | Uso | Operación |
|----------|-----|-------------------|
| ~ | ~A | Completo de 1 a A |
| - | -A | Cambio de signo. |
| -- | A-- | Decremento A. |
| ++ | A++ | Incremento de A |
| ! | !A | Not A. |

Tabla de operadores de asignación

| Operador | Uso | Operación |
|----------|-------|---|
| & | A&B | AND lógico. |
| | A B | OR lógico. |
| ^ | A^B | XOR lógico. |
| << | A<<B | Desplazamiento a la izquierda de A B bits rellenos con ceros por la derecha. |
| >> | A>>B | Desplazamiento a la derecha de A B bits relleno con el BIT de signo por la izquierda. |
| >>> | A>>>B | Desplazamiento a la derecha de A B bits relleno con ceros por la izquierda. |

Variables

Constante No puede variar su valor en todo el programa. `Final static double PI=3.141592`

Variable local

Solo podrá ser usada por los metodos de esa clase. Pero podrá ser llamada desde otras clases.

```
public class ejemplo {
    public static void ejemplo2{
        int variable=0;
    }
    public static void main(String[] args) {
        ejemplo2.variable
    }
}
```

Variable global

Podrá ser usada por cualquier metodo de cualquier clase.

```
public class ejemplo {
    int variable=0;
    public static void main(String[] args){}
```

Variable private

Solo será posible ser llamada desde la clase en la que ha sido creada.

```
public class ejemplo {  
    private int variable=0;  
    public static void ejemplo2{  
    }  
    public static void main(String[] args) {  
        ejemplo2.variable // esto no funciona  
    }  
}
```

Clases

Clase no definida

Se podrá acceder a sus métodos desde todas las otras clases que estén en el mismo paquete.

```
class ejemplo {}
```

Clase publica

```
public class ejemplo {}
```

Si la clase es publica todos sus métodos podrán ser llamados desde cualquier clase de cualquier paquete. Una clase main no podrá contener otras clases públicas.

Clase privada Las clases no pueden ser declaradas private. Sus métodos si, solo serán accesibles por métodos internos a la misma clase.

Clase Protegida Se podrá acceder a los métodos de esta clase desde las otras que estén en el mismo paquete o sean subclases.

```
protected class ejemplo {}
```

Clase void La clase no devuelve ningún valor

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

Clase static

Estas clases están anidadas dentro de otras clases. Y podrán ser usadas sin crear una estancia de esa clase. Y serán modificadas por todas las subclases y/o objetos que se creen de esa clase.

```
public class ejemplo {  
    public static void ejemplo2{}  
    public static void main(String[] args) {}  
}
```

Extends

Las clases pueden heredar atributos y métodos de las clases padres que les asignemos. Cuando herede la clase tendrá sus propios atributos y los de la clase padre

```
public class padre {  
    String atributo1;}  
}
```

```
public class hija extends padre {  
    String atributo2;}  
}
```

Objetos

Construcción de objetos

Los objetos heredan los métodos y variables de la clase con la que es creado. En este caso la variable a del “objeto1” valdrá 6.

```
public class crear_objetos {  
    public static class ejemplo1 {  
        private static int variable=1;  
        public void setValor(int a){  
            a = a + variable;  
        }  
    }  
    public static void main(String[] args) {  
        ejemplo1 objeto1;  
        objeto1 = new ejemplo1();  
        objeto1.setValor(5);  
    }  
}
```

Métodos

Métodos de la clase

Los métodos pueden ser declarados igual que las clases, con sus mismas características.

```
public class ejemplo {  
    public static class void ejemplo2 {  
        public void setMetodo1()  
    }  
}  
  
public static void main(String[] args) {}
```

Llamar a un método de una clase desde main

Se llama al método getColor de la clase ejemplo1 y nos lo muestra por pantalla.

```
public class prueba {  
    static class ejemplo1 {  
        static String color;  
        public static String getColor() {  
            color = "verde";  
            return color;  
        }  
    }  
    public static void main(String[] args) {  
        System.out.println(ejemplo1.getColor());  
    }  
}
```

Introducir dato a un método de una clase desde main

Cambiamos el valor de color del método setColor de la clase ejemplo1 por la cadena de texto "Rojo".

```
public class introducir_dato_metodo {  
    static class ejemplo1 {  
        static String color;  
        public static void setColor(String a) {  
            color = a;  
        }  
    }  
    public static void main(String[] args) {  
        ejemplo1.setColor("Rojo");  
    }  
}
```

Método void

El método no devuelve ningún valor

```
void ejemplo() {}
```

Método con return

En caso de no declararla como void esta tiene que devolver algún valor. También se puede retornar clases, métodos,...

Se puede asignar la devolución del "return" a una variable.

```
public class ejemplo {
    static int ejemplo2(){
        int variable=0;
        return variable;
    }
    public static void main(String[] args) {}
}
```

```
public static void main(String[] args) {
    int variable2;
    variable2=ejemplo.emeplo2();
}
```

Referencia a atributos generales de la clase

Para referirse dentro de un método a un atributo de de la clase contenedora.

```
public class ejemplo {
    int variable;
    void ejemplo2(){
        this.variable=0;
    }
}
```

Métodos de la clase object

Hace una comparación entre un objeto y otro.

```
public class ejemplo {
    public static void main(String[] args) {
        ejemplo e1 = new Ejemplo();
        ejemplo e2 = new Ejemplo();
        resultado=e1.equals(r2); }
}
```

Devuelve el nombre de la clase desde donde se creo el objeto y su código hash.

```
public class ejemplo {
    public static void main(String[] args) {
        ejemplo e1 = new Ejemplo();
        ejemplo e2 = new Ejemplo();
        resultado=e1.toString(); }
}
```

Libera la memoria ocupada por el objeto y sus métodos.

```
public class ejemplo {
    public static void main(String[] args) {
        ejemplo e1 = new Ejemplo();
        ejemplo e2 = new Ejemplo();
        resultado=e1.finalize(); }
}
```


Sentencias y estructuras

Estructuras IF

1- Estructura IF

```
if (a==0){ b = c;}
```

2-Estructura else

```
if (a==0){ b = c;}else{b=e}
```

3-Estructura if-elseif-else

```
if (a==0){ b =c}
    else if (a==1){b=e}
    else if (a==2){b=f}
else{b=g}
```

Estructura switch

Para hacer que el switch elija únicamente un paso de los 3 se pondrá ;break; al final de la sentencia case.

```
switch (a){
    case 1: b = 3;break;
    case 2: b = 0;break;
    case 3: b = 5;break;
    default: b =1;break;
}
```

La opción “default” se ejecuta cuando ninguna de las otras opciones son activadas.

Estructura while

```
while (a<=1){
    b = c;
}
```

Estructura do while

```
do{
    b = c;
}while(a<=1);
```

Estructura For

```
for (int i=0;i<=10;i++){
    b = c;
}
```

Vectores

Crear vector

Se puede declarar global

```
public class vectores {  
    public static int vector[];  
    public static void main(String[] args) {  
    }  
}
```

O local

```
public class vectores {  
    public static void main(String[] args) {  
        int vector[];  
    }  
}
```

Declarar vector

Se puede crear y declarar de forma separada.

```
public class vectores {  
    public static int vector[];  
    public static void main(String[] args) {  
        vector = new int [10];  
    }  
}
```

O todo junto, crear y declarar.

```
public class vectores {  
    public static void main(String[] args) {  
        int [] vector = new int [10];  
    }  
}
```

Matrices

Crear matriz

Es igual que los vectores pero con dos valores.

```
public class matriz {  
    public static int matriz[][];  
    public static void main(String[] args) {  
    }  
}
```

Declarar matriz

```
public class matriz {  
    public static void main(String[] args) {  
        int [][] matriz = new int [10][10];  
    }  
}
```

Códigos útiles cortos

Introducción de datos

Introducción de datos por scanner

```
import java.util.Scanner;
public class introduccion_datos {
    public static void main(String[] args) {
        int v1;
        Scanner teclado = new Scanner(System.in);
        System.out.println("Introduce un numero");
        v1 = teclado.nextInt();
        System.out.println("El número es: "+ v1);
    }
}
```

Introducción de datos por bufferprint

```
import java.io.*;
public class introduccion {
    public static void main(String[] args) {
        String res="";
        InputStreamReader isr = new
        InputStreamReader(System.in);
        BufferedReader buff = new BufferedReader(isr);
        System.out.println("Escribe una letra: ");
        try {
            String ln = buff.readLine();
            res = ln;
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println("La letra es :"+ res);
    }
}
```

Generar números random

```
public class numeros {
    public static int random(){
        int x=0;
        x=(int)(Math.random()*10);
        return x;
    }
    public static void main(String[] args) {
        System.out.println("Primer numero: " + random());
    }
}
```

Cuenta los dígitos de un número

```
public static int cuentaDigitos(int x){
    int con=0;
    do{
        x = x/10;
        con = con + 1;
    } while (x!=0);
    return con;}
}
```

**Suma los
dígitos de
un número**

```
public static int sumaDigitos(int x){
    int s=0,d=0;
    do{
        d = x%10;
        s = s + d;
        x = x/10;
    } while(x!=0);
    return s;}
```

**Eleva la variable
"potencia"
tantas veces
como dígitos
tiene el número
de la variable x**

```
public static void eleva(int x){
    int potencia=1;
    x = x/10;
    while (x!=0){
        x = x/10;
        potencia = potencia * 10;
    }
    System.out.println(potencia);}
```

**Cálculo
fecha
bisiesta**

```
public int calculo (int dia, int mes , int a){
    int sw = 0, max = 0;
    if (mes == 4 | mes == 6 | mes == 9 | mes == 11) {
        max = 30;
    } else {
        max = 31;
        if (mes == 2) {
            if (a % 4 == 0 & !(a % 100 == 0 & !(a % 400 == 0))) {
                max = 29;
            } else {
                max = 28;
            }
        }
    }
    if (dia > max) {
        sw = 1;
    }
    else {
        sw = 0;
    }
    return sw;}}
```

**Filtro para
variables**

```
do {
    System.out.println("Introduce el dia actual");
    intro2 = intro.nextInt();
} while (intro2 < 1 | intro2 > 31);
```

Filtro para variables

```
do {
    System.out.println("Introduce el dia actual");
    intro2 = intro.nextInt();
} while (intro2 < 1 | intro2 > 31);
```

Ordenar los dígitos de un vector de mayor a menor con método burbuja

```
public static void setOrdenar(int vector[],int posiciones) {
    int aux;
    for (int i=posiciones;i>=0;i--){
        for (int j=0;j<i-1;j++){
            if (vector[j]>vector[j+1]){
                aux = vector[j+1];
                vector[j+1] = vector[j];
                vector[j]=aux;
            }
        }
    }
}
```

Desglosar Un Número De 4 cifras

```
for (a=1;a<=4;a++){
    numero2 = numero%10;
    if (a==1){
        h = numero2;
    }
    if (a==2){
        k = numero2;
    }
    if (a==3){
        j = numero2;
    }
    if (a==4){
        i = numero2;
    }
    numero = numero/10;
}
```

Formatear Un Systemout

```
public class pruebasString {
    public static void main(String[] args) {
        float floatVar=3;
        int intVar=7;
        String stringVar="hola";
        System.out.printf("%6.2f %10d %10s " , floatVar ,
/*el porcentaje son los dígitos que se le darán al valor, f es float, d
es int y s es String*/
        intVar , stringVar);}}
```

Códigos ejemplo

Acciones sobre cifra

```
import java.util.Scanner;
public class calcul_numero {
    public static int numeroDigits(int x){
        return x;
    }
    public static int comptaDigits(int x){
        int con=0;
        do{
            x = x/10;
            con = con + 1;
        }while (x!=0);
        return con;
    }
    public static int sumaDigits(int x){
        int s=0,d=0;
        do{
            d = x%10;
            s = s + d;
            x = x/10;
        }while(x!=0);
        return s;
    }
    public static int inversDigits(int x){
        int i,h=1,num,num2=0,num3=0,numero=0,j,g=0;
        for (i=1;i<comptaDigits(x);i++){
            h = h * 10;
        }
        g = h;
        num = numeroDigits(x);
        for (j=1;j<=h;j=j*10){
            num3 = num%10;
            num2 = num3 * g;
            numero = numero + num2;
            num = num/10;
            g = g/10;
        }
        return numero;
    }
    public static void main(String[] args) {
        int x;
        Scanner intro = new Scanner(System.in);
        System.out.println("Escriu el número: ");
        x = intro.nextInt();
        calcul_numero.comptaDigits(x);
        System.out.println("El número te: " + comptaDigits(x) + " digits.");
        calcul_numero.sumaDigits(x);
        System.out.println("El número suma: " + sumaDigits(x) );
        System.out.println("");
        System.out.println("El número invertit es: " + inversDigits(x) );}
}
```

Calcular areas

```
import java.util.Scanner;
public class calcular_area_scanner {
    public static void main(String[] args) {
        int opcion;
        double pi=3.1416,resultado=0,num1,num2;
        Scanner intro = new Scanner(System.in);
        do{
            System.out.println("Calculo de area");
            System.out.println("Introduce:");
            System.out.println("1-Para calculo de rectangulo");
            System.out.println("2-Para calculo de Triangulo");
            System.out.println("3-Para calculo de Circulo");
            System.out.println("4-Para calculo de Cuadrado");
            System.out.println("5-Para salir");
            opcion = intro.nextInt();
            switch (opcion){
                case 1:
                    System.out.println("Rectangulo, introduce la altura: ");
                    num1 = intro.nextInt();
                    System.out.println("Y la base: ");
                    num2 = intro.nextInt();
                    resultado = num1 * num2;
                    System.out.println("El area del rectangulo es : " + resultado);break;
                case 2:
                    System.out.println("Triangulo, introduce la altura: ");
                    num1 = intro.nextInt();
                    System.out.println("Y la base: ");
                    num2 = intro.nextInt();
                    resultado = (num1 * num2)/2;
                    System.out.println("El area del triangulo es : " + resultado);break;
                case 3:
                    System.out.println("Circulo, introduce el radio: ");
                    num1 = intro.nextInt();
                    resultado = (num1 * num1)*pi;
                    System.out.println("El area del circulo es : " + resultado);break;
                case 4:
                    System.out.println("Cuadrado, introduce la altura: ");
                    num1 = intro.nextInt();
                    resultado = num1 * num1;
                    System.out.println("El area del cuadrado es : " + resultado);break;
                case 5:
                    System.out.println("Saliendo del programa... ");break;
            }
        } while (opcion!=5);
    }
}
```

Suma de matriz

```
import java.util.Scanner;
public class sumar_matrus {
    public static int matriu[][];
    public static int sumafil[];
    public static int sumacol[];
    public static void setPlenar(int col,int fil){
        int i,j;
        for (j=0;j<col;j++){
            for (i=0;i<fil;i++){
                matriu [j][i] = (int)(Math.random()*10);
            }
        }
    }
    public static void getMostrar(int col, int fil){
        int i, j;
        for (j=0;j<col;j++){
            for (i=0;i<fil;i++){
                System.out.println("Columna: " + j + " Fila: " + i + " Numero: " + matriu [j][i]);
            }
        }
    }
    public static void getSumacolumnas(int col,int fil){
        int i,j;
        for (j=0;j<col;j++){
            for (i=0;i<fil;i++){
                sumacol[j] = sumacol[j] + matriu[j][i];
            }
        }
        for (j=0;j<col;j++){
            System.out.println("La suma de la columna " + j + " es : " + sumacol[j]);
        }
    }
    public static void getSumafilas(int col,int fil){
        int i,j;
        for (j=0;j<fil;j++){
            for (i=0;i<col;i++){
                sumafil[j] = sumafil[j] + matriu[i][j];
            }
        }
        for (j=0;j<fil;j++){
            System.out.println("La suma de la fila " + j + " es : " + sumafil[j]);
        }
    }
    public static void main(String[] args) {
        int fil,col;
        Scanner teclat = new Scanner(System.in);
        System.out.println("Cuantes files vols fer?");
        fil = teclat.nextInt();
        System.out.println("Cuantes columnas vols fer?");
        col = teclat.nextInt();
        matriu = new int [col][fil];
        sumafil = new int [fil];
        sumacol = new int [col];
        setPlenar(col,fil);
        getMostrar(col,fil); //mostre tota la matriu
        getSumacolumnas(col,fil);
        getSumafilas(col,fil);}
    }
```


Introducir nombres en matriz

```
import java.util.*;

public class noms {
    static String []nombres;
    static int POS;
    public static void muestra(){
        int i;
        for (i=0;i<=POS-1;i++){
nombres[i]);
            System.out.println("Nombre numero " + i + " : " +
                }
        }
    public static void introducir(){
        Scanner teclado = new Scanner(System.in);
        System.out.println("Cuantos alumnos vas a introducir? ");
        POS = teclado.nextInt();
        nombres = new String [POS];
    }
    public static void llenar(){
        int i;
        String intro;
        Scanner teclado = new Scanner(System.in);
        for (i=0;i<=POS-1;i++){
            System.out.println("Introduce el nombre en la posicion " + i );
            intro = teclado.next();
            nombres[i] = intro;
        }
    }
    public static void main(String[] args) {
        introducir();
        llenar();
        muestra();
    }
}
```

Ordenar vectores de string

```
import java.util.Scanner;
public class matrialumnes {
    static int POSalu; static int POSasig; static String []nombre; static String
    []apellido1; static String []apellido2; static String []asignaturas; static float []
    []notas;
    static Scanner teclado = new Scanner(System.in);
    public static void introducir(){
        System.out.println("Cuantos alumnos vas a introducir? ");
        POSalu = teclado.nextInt();
        nombres = new String [POSalu];
        apellido1 = new String [POSalu];
        apellido2 = new String [POSalu];
        System.out.println("Cuantas asignaturas vas a introducir?");
        POSasig = teclado.nextInt();
        asignaturas = new String [POSasig];
        notas = new float [POSasig][POSalu];
    }
    public static void llenar(int vector){
        int i,j;
        if (vector==0){
            for (i=0;i<POSalu;i++){
                System.out.println("Introduce el nombre de alumno " +
                (i+1) );
                nombres[i] = teclado.next();
                System.out.println("Introduce el primer apellido del alumno
                " + (i+1));
                apellido1[i] = teclado.next();
                System.out.println("Introduce el segundo apellido del
                alumno " + (i+1));
                apellido2[i] = teclado.next();
            }
        }
        if(vector==1){
            for (i=0;i<POSasig;i++){
                System.out.println("Introduce el nombre de asignatura en la
                posicion " + (i+1) );
                asignaturas[i] = teclado.next();
            }
        }
        if(vector==2){
            for (j=0;j<POSalu;j++){
                for (i=0;i<POSasig;i++){
                    System.out.println("Introduce la nota de la asignatura " +
                    asignaturas[i] + " , para el alumno " +
                    apellido1[j]+ " " + apellido2[j] + " , " + nombres[j] +
                    " numero " + (j+1));
                    notas[i][j] = teclado.nextFloat();}}}}
        .
        .
        .
    }
```

```

.
.
.
public static void comparar(){
    int i,j,com=0;
    String aux;
    for (i=POSalu-1;i>=0;i--){
        for (j=0;j<i;j++){
            com = apellido1[j].compareTo(apellido1[j+1]);
            if (com > 0){
                aux = apellido1[j+1];
                apellido1[j+1] = apellido1[j];
                apellido1[j] = aux;
                aux = apellido2[j+1];
                apellido2[j+1] = apellido2[j];
                apellido2[j] = aux;
                aux = nombres[j+1];
                nombres[j+1] = nombres[j];
                nombres[j] = aux;
            }
            if (com == 0){
                com = apellido2[j].compareTo(apellido2[j+1]);
                if (com > 0){
                    aux = apellido1[j+1];
                    apellido1[j+1] = apellido1[j];
                    apellido1[j] = aux;
                    aux = apellido2[j+1];
                    apellido2[j+1] = apellido2[j];
                    apellido2[j] = aux;
                    aux = nombres[j+1];
                    nombres[j+1] = nombres[j];
                    nombres[j] = aux;
                }
            }
            if (com == 0){
                com = nombres[j].compareTo(nombres[j+1]);
                if (com > 0){
                    aux = apellido1[j+1];
                    apellido1[j+1] = apellido1[j];
                    apellido1[j] = aux;
                    aux = apellido2[j+1];
                    apellido2[j+1] = apellido2[j];
                    apellido2[j] = aux;
                    aux = nombres[j+1];
                    nombres[j+1] = nombres[j];
                    nombres[j] = aux;
                }
            }
        }
    }
}
}
}
}
.
.
.

```

```

.
.
.
public static void media(int vector){
int i,intro;
float acum=0;
if (vector==0){
System.out.println("Lista de alumnos");
for (i=0;i<POSalu;i++){
System.out.println("Alumno " + (i+1) + " : " +
apelido1[i] + " " + apellido2[i] + " , " +
nombres[i]);
}
System.out.println("De que alumno quieres sacar la
media?");
intro = teclado.nextInt();
intro = intro-1;
for (i=0;i<POSasig;i++){
System.out.println("Alumno: " +
apellido1[intro] + " " + apellido2[intro] + " , " +
nombres[intro] + " con nota: " + notas[i][intro]);
acum = acum + notas[i][intro];
}
acum = acum/POSasig;
System.out.println("El alumno " + nombres[intro] + " tiene
una media de " + acum);
}
if (vector==1){
System.out.println("Lista de asignaturas");
for (i=0;i<POSasig;i++){
System.out.println("Asignatura " + (i+1) + " : " +
asignaturas[i]);
}
System.out.println("De que asignatura quieres sacar la media?");
intro = teclado.nextInt();
intro = intro-1;
for (i=0;i<POSalu;i++){
System.out.println("Asignatura: " + asignaturas[intro] + "
con nota: " + notas[intro][i]);
acum = acum + notas[intro][i];
}
acum = acum/POSalu;
System.out.println("La asignatura " + asignaturas[intro] + " tiene una
media de " + acum);
}
}
.
.
.

```

```

        .
        .
public static void muestra(int vector){
int i,j;
    if (vector==0){
        for (i=0;i<POSalu;i++){
            System.out.println("Nombre numero " + (i+1) +
                ":" + apellido1[i] + " " + " " + apellido2[i]
                ,"+ nombres[i]);
        }
    }
    if(vector==1){
        for (i=0;i<POSasig;i++){
            System.out.println("Asignatura numero " + (i+1)
                + " : " + asignaturas[i]);
        }
    }
    if(vector==2){
        System.out.println("+-----+");
        System.out.println("");
        System.out.print("|Asignaturas ");
        for (i=0;i<POSasig;i++){
            System.out.print(" | " + asignaturas[i]);
        }
        System.out.print("|");
        System.out.println();
        System.out.println("+-----+");
        System.out.println(" | Alumnos ");
        for (j=0;j<POSalu;j++){
            System.out.print(" | " + apellido1[j] + " " +
                apellido2[j] + " , " + nombres[j]);
            for (i=0;i<POSasig;i++){
                System.out.print(" | " + notas[i][j]);
            }
            System.out.println(" |");
        }
        System.out.println("+-----+");
    }
}
public static void main(String[] args) {
    introducir();
    llenar(0);
    llenar(1);
    comparar();
    muestra(0);
    muestra(1);
    llenar(2);
    muestra(2);
    media(0);
    media(1);
}
}

```


Al igual que el mapa en el bolsillo, la cantimplora en el cinturón y el machete cruzado en la espalda son esenciales para la vida de un guerrillero estos manuales facilitan la lucha constante contra los trabajos de programación.

Los informáticos tenemos la suerte de tener internet cerca de nosotros, pero cuando esa suerte no existe entonces tenemos que recurrir al papel o pdf. Estos pequeños manuales no atienden a explicaciones para principiantes ni avanzados, simplemente sacian las consultas de las dudas que pueden surgir programando en cualquier sitio, en el día a día.

Cuando estas lejos de tu puesto de trabajo, internet no está ahí o simplemente la red no funciona el guerrillero informático tiene el manual en el bolsillo, la botella de agua en la mochila y el portátil cruzado en la espalda.

Autor: Jesús Benages Sales

Contacto: jobinary@hotmail.com

