

# DESARROLLO DE APLICACIONES INFORMATICAS

## Programación en Lenguajes Estructurados

APELLIDOS: PEREZ VEGA\_\_\_\_\_

CALIFICACIÓN

NOMBRE : JAVIER\_\_\_\_\_

1.- Define brevemente cada una de las fases en el desarrollo de software o ciclo de vida de una aplicación informática. (1 Punto)

Análisis: Fase en la que se definen todos los requerimientos que debe de cumplir la aplicación, entradas, salidas, funciones etc. En resumen se define QUE debe hacer la aplicación.

Diseño: aquí se definirán las estructuras de datos que se van a utilizar, datos, ficheros, etc. Aquí se define el COMO se va desarrollar la aplicación.

Codificación: Pasar lo anterior a uno o varios lenguajes de programación.

Pruebas: fase en la que se probaran todas las funciones que realiza la aplicación y su correcto funcionamiento.

Implantación: instalación en un entorno real de funcionamiento.

Explotación: configuración y gestión de la aplicaciones en el entorno de la empresa.

Mantenimiento: una vez instalada en esta fase se arreglaran o actualizaran a lo largo del tiempo de vida de la aplicación.

2.- Indicar las ventajas y desventajas de los lenguajes de bajo nivel como el ensamblador. (1 Puntos)

La principal ventaja de un lenguaje de bajo nivel es la velocidad de proceso y el acceso a los dispositivos físicos del sistema.

Desventajas: Normalmente son específicos para cada procesador.  
Difíciles de comprender.  
Difíciles de programar (edición y corrección).

3.- Define brevemente: (1 Punto)

- Dato constante
- Variable estática
- Variable compuesta
- Tabla unidimensional o vector

- Dato Constante: Dato que mantiene su valor durante toda la ejecución del programa.

- Variable estática: dato cuyo valor puede variar pero no su tamaño y situación en la memoria.

- Variable Compuesta: conjunto de datos de distintos tipos, que pueden ser utilizados en su conjunto como si fuera uno solo o independientemente.

- Tabla unidimensional o Vector: Conjunto de datos del mismo tipo que se almacenan en memoria en posiciones conjuntas que usualmente son utilizados mediante un índice que indica la posición que ocupa dentro del vector.

4.- Dada la siguiente expresión lógica indicar si es verdadera o falsa según los siguientes valores de los datos:(1 Punto)

( NOT ( ( Num > 5) AND ( Sabor ? Dulce ) ) ) OR ( Nombre > "Luis" )

- a) Num = 5, Sabor = Amargo, Nombre = " Francisco"
- b) Num = 10, Sabor = Ácido, Nombre ="Ana"
- c) Num = 20, Sabor = Dulce, Nombre="Luis"

- A) VERDADERO
- B) FALSO
- C) VERDADERO

( 2 Puntos )

5.- Elaborar un algoritmo que lea números enteros hasta que el usuario introduzca un valor igual a cero y calcule la media de los números pares y de los impares. El programa me mostrará el número total de valores introducidos y cual de las dos medias tiene un valor mayor: la de los pares o la de los impares. Diseñar el algoritmo utilizando: **Pseudocódigo, Ordinogramas y Diagramas N-S.**

Ejemplo de ejecución:

Suponer los siguientes valores introducidos:

30, 7, 8, 2, 10, 15, 19, 23, 3, 7, 8, 157, 20, 31, 3, 0 <- Indica fin

El programa debería mostrar:

Nº de datos introducidos = 15.

El valor de la media de los impares es mayor.

ENTORNO;  
ENTEROS valor, cont\_pares, cont\_imp, total, mediapares, mediaimp, totalpares, totalimp, totalvalor, mediavalpar, mediavalimp

### PROCESO

Inicio

Cont\_pares=0

Cont\_imp=0

Total=0

Totalpares=0

Totalimp=0

Mediapares=0

Mediaimp=0

mediavalpar=0

mediavalimp=0

Mostrar "Introduce valor:"

Leer valor

Mientras(valor?0)

    Si((valor mod 2)=0)

        Entonces

            Cont\_pares=cont\_pares+1

            Totalpares=totalpares+valor

        Sino

            Cont\_imp=cont\_imp+1

            Totalimp=totalimp+valor

    Fin\_si

    Total=total+1

    Totalvalor=totalvalor+valor

    Mostrar "Introduce valor:"

    Leer valor

Fin\_mientras

Si(total=0)

    Entonces

        mostrar "no se ha introducido ningún número"

    sino

        Mostrar "El Numero de datos introducidos =", Total

        mediapares=cont\_pares/total

        mediavalpar=totalpares/cont\_pares

        mediaimp=cont\_imp/total

        mediavalimp=totalimp/cont\_imp

        Si (mediapares=mediaimp)

            Entonces

                Mostrar "La media de números pares e impares introducidos es igual"

            Sino

                Si(mediapares>mediaimp)

                    Entonces

                        mostrar "La media de números pares introducidos es mayor"

                    Sino

                        mostrar "La media de números impares introducidos es mayor"

                Fin\_si

        Fin\_si

        Si (mediavalpar=mediavalimp)

            Entonces

                Mostrar "La media de la suma de los valores introducidos es igual"

            Sino

                Si (mediavalpar>mediavalimp)

                    Entonces

                        Mostrar "la media de la suma de los valores pares introducidos es mayor"

                    Sino

                        Mostrar "la media de la suma de los valores impares introducidos es

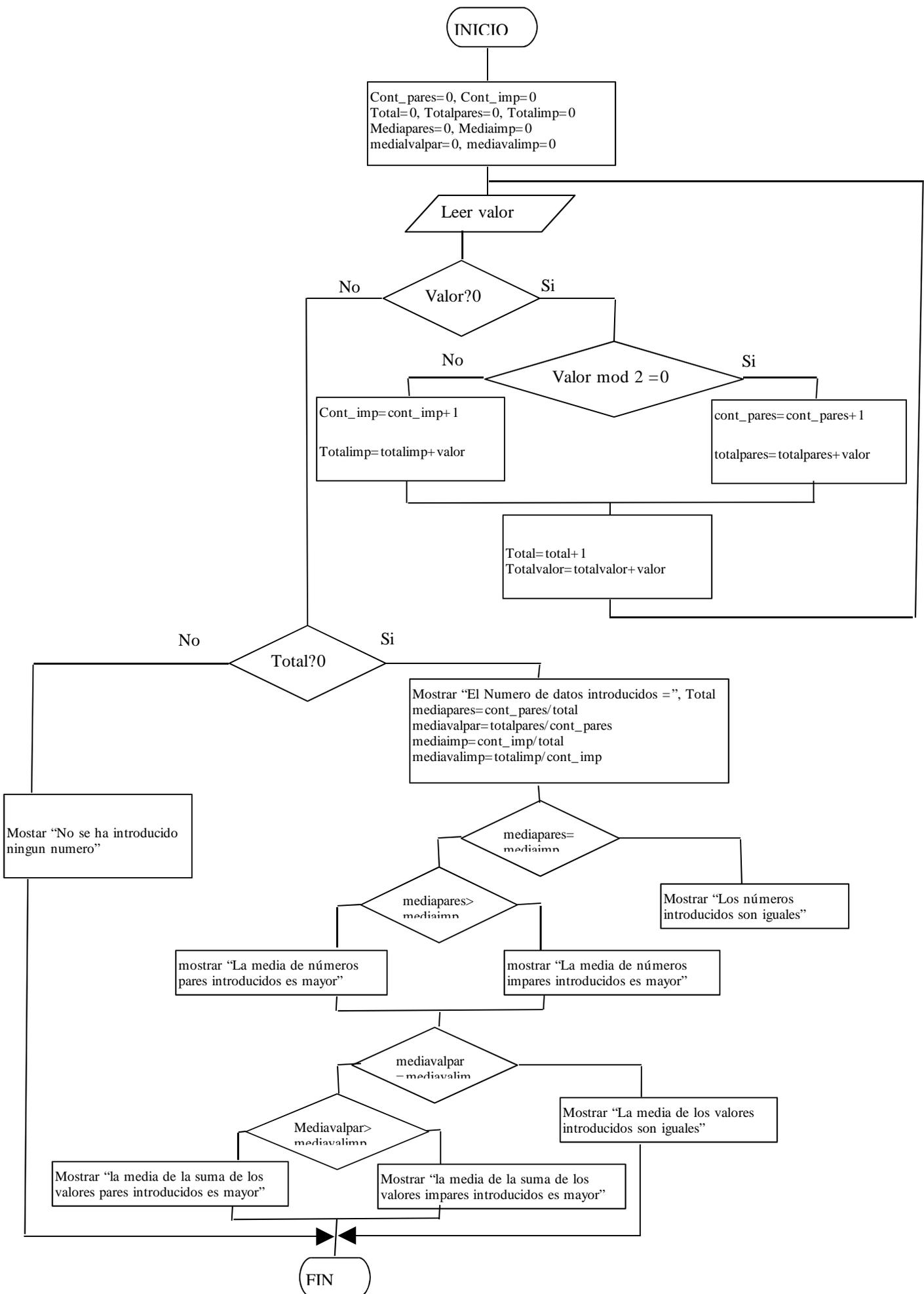
mayor"

                Fin\_si

Fin\_si

Fin\_si

FIN PROCESO



Cont\_pares=0 , Cont\_imp=0 , Total=0  
 Totalpares=0 , Totalimp=0 , Mediapares=0  
 Mediaimp=0 , medialvalpar=0 , mediavalimp=0

Leer valor

(valor ? 0)

Valor mod 2 = 0

SI

NO

Cont\_pares=cont\_pares+1

Cont\_imp=cont\_imp+1

Totalpares=totalpares+valor

Totalimp=totalimp+valor

Total=total+1

Totalvalor=totalvalor+valor

Leer Valor

Total = 0

SI

NO

Mostrar “ No hay valores introducidos”

Mostrar “El Numero de datos introducidos =”, Total

mediapares=cont\_pares/total

mediavalpar=totalpares/cont\_pares

mediaimp=cont\_imp/total

mediavalimp=totalimp/cont\_imp

SI

mediapares=mediaimp

NO

Mostrar “la media de numeros introducidos es igual”

mediapares>mediaimp

Mostrar “los impares son mayores”

Mostrar “los pares son mayores”

mediavalpar=mediavalimp

mediavalpar>mediavalimp

SI

O

N

Mostrar la suma de los impares es mayor”

Mostrar “la suma de los pares es mayor”

(2 Puntos)

6.- Diseñar un procedimiento que tenga la siguiente cabecera:

A) void PonEco ( char Palabra [ ], entero nveces )

El procedimiento debe repetir cada carácter de la palabra, tanta veces como indique el parámetro nveces.

Ej.- Palabra = “Hola”, nveces = 3            Palabra = “HHHHoooooolllaaaa “

```
void poneco( char Palabra[], int nveces)
{
    int i, j, h, longcadena, longcadenafinal ;
    for (i=0; Palabra[i]!='\0';i++);
    longcadena=(i-1); // longitud de la cadena ( de 0 a N)
    longcadenafinal=(longcadena+1)*(nveces+1); // longitud de la cadena final (cadena + (nveces+1))

    Palabra[longcadenafinal+1]='\0'; // ultimo carácter de la cadena
    i=longcadenafinal;
    for(h=longcadena;h>=0;h--)
    {
        for(j=nveces; j>=0;j--)
        {
            Palabra[i]=Palabra[h];
            i--;
        }
    }
}
```

B) void QuitarLetrar ( char Palabra[ ], char ListaLetras[ ] )

El procedimiento debe suprimir de la Palabra aquellos caracteres que aparezcan en ListaLetras.

Ej.- Palabra = “Buena suerte” ListaLetras = “atepoi”    Palabra = “Bun sur”

```
void quitarletra (char palabra[], char listalettras[])
{
    int i, j, h;
    i=0;
    while (listalettras[i]!='\0')
    {
        j=0;
        h=0;
        while (palabra[j]!='\0')
        {
            if(palabra[h]!=listalettras[i])
            {
                palabra[j]=palabra[h];
                j++;
            }
            h++;
        }
        palabra[j]='\0';
        i++;
    }
}
```

```
}  
}
```

(3 Puntos)

7.- Diseñar un programa que realiza notas de compra. Primero se leerán los precios de 20 artículos y se almacenarán en una tabla. Posteriormente se solicitan números de artículos y cantidad para ir elaborando la nota preguntando al usuario después de leer cada número y cantidad si desea introducir más artículos, mostrando a continuación el importe de cada unidad y el importe total. Al final mostrará el importe total de la nota indicando la cantidad de descuento a aplicar. Se realiza un descuento del 10 % si el importe total supera las 10.000 Pts ( 4 Puntos )

Ejemplo de Ejecución:

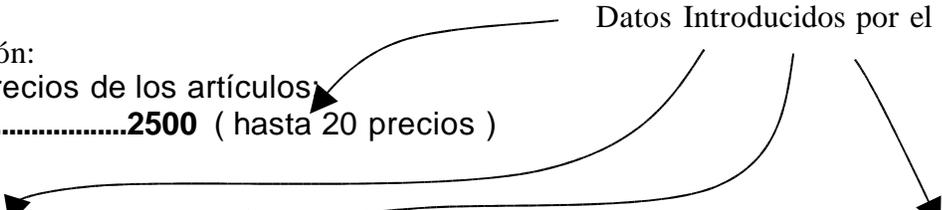
Introduzca los precios de los artículos  
**50,120,100,.....2500** ( hasta 20 precios )

Nota de compras:

|                          |                    |             |       |      |                                 |
|--------------------------|--------------------|-------------|-------|------|---------------------------------|
| Nºde Artículo: <b>1</b>  | Cantidad: <b>8</b> | Precio 50   | ..... | 400  | Desea continuar (s/n): <b>s</b> |
| Nºde Artículo: <b>3</b>  | Cantidad: <b>5</b> | Precio 100  | ..... | 600  | Desea continuar (s/n): <b>s</b> |
| Nºde Artículo: <b>8</b>  | Cantidad: <b>2</b> | Precio 2000 | ..... | 4000 | Desea continuar (s/n): <b>s</b> |
| Nºde Artículo: <b>10</b> | Cantidad: <b>6</b> | Precio 1000 | ..... | 6000 | Desea continuar(s/n): <b>n</b>  |

Importe .....11000  
10% Descuento.....1100  
TOTAL .....9900

Datos Introducidos por el usuario



**Nota:** Se debe validar todos los datos que introduce el usuario:  
Precios > 0  
Nºde artículo entre [1-20]  
Cantidad > 0  
Se valorará la calidad y claridad del programa propuesto.

```

/*-----
* PROGRAMA : examen.c
* GESTION DE PRECIOS Y LISTA DE LA COMPRA:
* DESCRIPCIÓN: Diseñar un programa que realiza notas de compra
*
* AUTOR: Javier Pérez Vega
*
* VERSION 1.0
*
* Fecha: 7/12/04
*
* -----*/
#include <stdio.h>
#include <conio.h>

#define N 20 // Número de Precios

/*-----
PROGRAMA PRINCIPAL
-----*/
void main()
{
    int precios[N]; /* Tabla precios */
    int articulo, i, cantidad, linea, columna, parcial, total, continua, desc;
    desc=0;
    articulo=0;
    total=0;

    // rellena la tabla con los precios de los articulos
    for (i=0; i<N; i++)
    {
        do
        {
            printf ("Introduce valor del articulo N°%d : ", i+1);
            scanf("%d", &precios[i]);
        }
        while(precios[i]==0); //comprueba que el valor del articulo no sea 0
    }

    clrscr();
    printf(" ");
    linea=2, columna=1;
    printf("NOTA DE COMPRAS\n");

    do
    {
        gotoxy(columna, linea);
        printf ("N° de Artículo: ");
        scanf("%d", &articulo);
    }
    while(!((articulo>0) && (articulo<=20)));

    columna=20;
    gotoxy(columna, linea);
    printf ("Cantidad : ");
    scanf("%d", &cantidad);

    columna=34;
    parcial=cantidad*precios[articulo-1];
    total=total+parcial;

    do
    {
        gotoxy(columna, linea);
        printf("Precio %d .... %d Desea continuar(S/N): ", precios[(articulo-1)], parcial);
        continua= getche();
    }
    while((continua!='N') && (continua !='n') && (continua !='S') && (continua!='s'));

    while((continua=='s') || (continua=='S'))
    {
        linea=linea++;
        columna=1;
    }
}

```

```

do
{
gotoxy(columna, linea);
printf ("N° de Artículo: ");
scanf("%d", &articulo);
}
while(!((articulo>0) && (articulo<=20)));

columna=20;
gotoxy(columna, linea);
printf ("Cantidad : ");
scanf("%d", &cantidad);

columna=34;
parcial=cantidad*precios[articulo-1];
total=total+parcial;
do
{
gotoxy(columna, linea);
printf("Precio %d .... %d Desea continuar(S/N): ", precios[(articulo-1)], parcial);
continua= getche();
}
while((continua!='N') && (continua !='n') && (continua !='S') && (continua !='s'));
}
linea=linea+2;
columna=1;
gotoxy(columna, linea);
printf("\t\t\t\t\t Importe .... %d", total);
if(total>10000)
{
linea++ ;
columna=1;
desc=total/10;
gotoxy(columna, linea);
printf("\t\t\t\t\t 10% Descuento ...");
printf(" %d", desc);
total=total-desc;
}
linea++;
columna=1;
gotoxy(columna, linea);
printf("\t\t\t\t\t TOTAL ..... %d", total);
continua=getche();
}

```