

Arreglos multidimensionales

miércoles, 22 de septiembre de 2021 07:11 a. m.

Sintaxis de declaración:

<tipo de dato> identificadorArreglo[TamD1][...][Tam Dn];

Ejemplo de dos dimensiones:

Sintaxis de declaración:

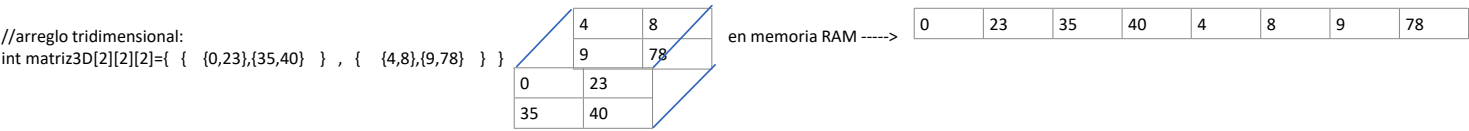
<tipo de dato> identificadorArreglo[TamD1][TamD2];

int matriz[5][5];
char matrizCar[10][2];

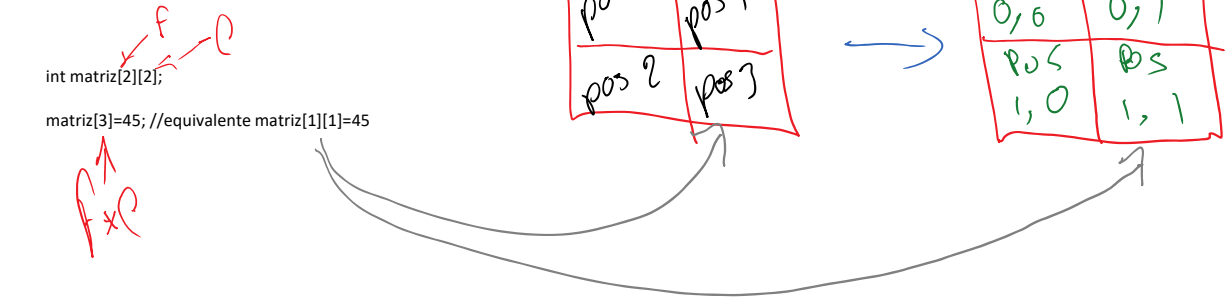
Sintaxis de inicialización:

<tipo de dato> identificadorArreglo[TamD1]
[TamD2]={ <valorPos0,0>,<valorPos0,1>,...,<valorPos0,TAMD1-1>>},{},{},...,{<valorPosTamD1-1,TamD2-1>};

Ejemplo:



Ejemplo de acceso a posiciones determinadas



Cadenas--> Arreglos de caracteres

martes, 28 de septiembre de 2021 07:15 a. m.

En C no existe un tipo predefinido para la manipulación de cadenas de caracteres (Strings). Sin embargo, el estándar de C provee de algunas funciones para el tratamiento de cadenas.

Una cadena en C es un **arreglo de caracteres de una sola dimensión** que en su última posición encontraremos un carácter especial, llamado fin de cadena: `'\0'`

Declaración de cadenas:

```
char cadena[n+1]; //donde n es la cantidad de letras que contiene la palabra
```

Debido a que en la representación interna de una cadena de caracteres es terminada por el símbolo `'\0'`, para un texto de "n" caracteres, debemos reservar "n+1" cantidad de elementos. Aunque el símbolo fin de cadena es parte del arreglo, este no aparece con las funciones `printf`, `puts`, `putchar`, etc.

Inicialización de cadenas:

```
char cadena[5]="Hola"; //solo es posible en la declaración e inicialización de una cadena
```

```
char cadena[5]={'H','o','l','a','\0'};
```

```
char cadena[5]={'H','o','l','a',0};
```

```
char cadena[5]={'\0'};
cadena[0]='H';
cadena[1]='o';
cadena[2]='l';
cadena[3]='a';
```

//Todas las cadenas son arreglos de caracteres, pero no todos los arreglos de caracteres son cadenas.

Se acceden a la posición de una cadena como si de un arreglo simple tratara. Las cadenas se pueden manipular de forma conjunta o separada: `puts(cadena)`; ciclo: `putchar(cadena[i])`

La librería `<string.h>` provee de las siguientes funciones para la manipulación de cadenas:

- `strlen`
- `strcpy`
- `strcat`
- `strcmp`
- `strchr`

0x206					0x207					0x208					0x209				
	E	s	c	u	e	l	a												

0x306					0x307					0x308					0x309.....				
						E	s	c	u	e	l	a							

```
#include<stdio.h>
#include<string.h>

void main(){
    //cadenas locales
    char cadena[4]="IPN"; //Una sola dimensión es para manejar 1 sola palabra
    char cadena2[8]={'E','s','c','u','e','l','a',0};
    int i,j;
    //lista de cadenas:
    //Hola
    //adios
    //saludos
    char nombres[3][8]={"Hola","adios","saludos"};
    //Lista de grupos, cada una con 3 nombres:
    //Grupo 0: Alan, Jose, Maria, Grupo 1: Jaime, Rosa, Laura, Grupo 2: Alex, Robert, Mau
    char grupos[3][3][8]={
        {"Alan","Jose","Maria"},
        {"Jaime","Rosa","Laura"},
        {"Alex","Robert","Mau"}
    };

    char copia[15]={0};
    puts(cadena);
    puts(cadena2);
    printf("Esta es la cadena 1: %s y su longitud es: %d\n", cadena, strlen(cadena));
    printf("Esta es la cadena 2: %s y su longitud es: %d\n", cadena2, strlen(cadena2));
    //copia de un lugar a otro una cadena:
    strcpy(copia,cadena2);
```

```

printf("Esta es la cadena copia: %s y su longitud es: %d\n", copia, strlen(copia));
//Concatenación de cadenas
strcat(copia, cadena);
printf("Esta es la cadena copia concatenada: %s y su longitud es: %d\n", copia, strlen(copia));
printf(" El resultado de la comparación es: %d \n", strcmp(cadena,cadena2));
puts("Lista de palabras:");
//ciclo para recorrer una lista de palabras
for(i=0; i<3; i++){
    puts(nombres[i]);
}
puts("Lista de grupo 1:");
//ciclo para recorrer una lista de grupos con cada grupo de 3 nombres
for(i=0; i<3; i++){
    printf("El grupo %d es:\n",i);
    for(j=0; j<3; j++){
        puts(grupos[i][j]);
    }
}
}
}

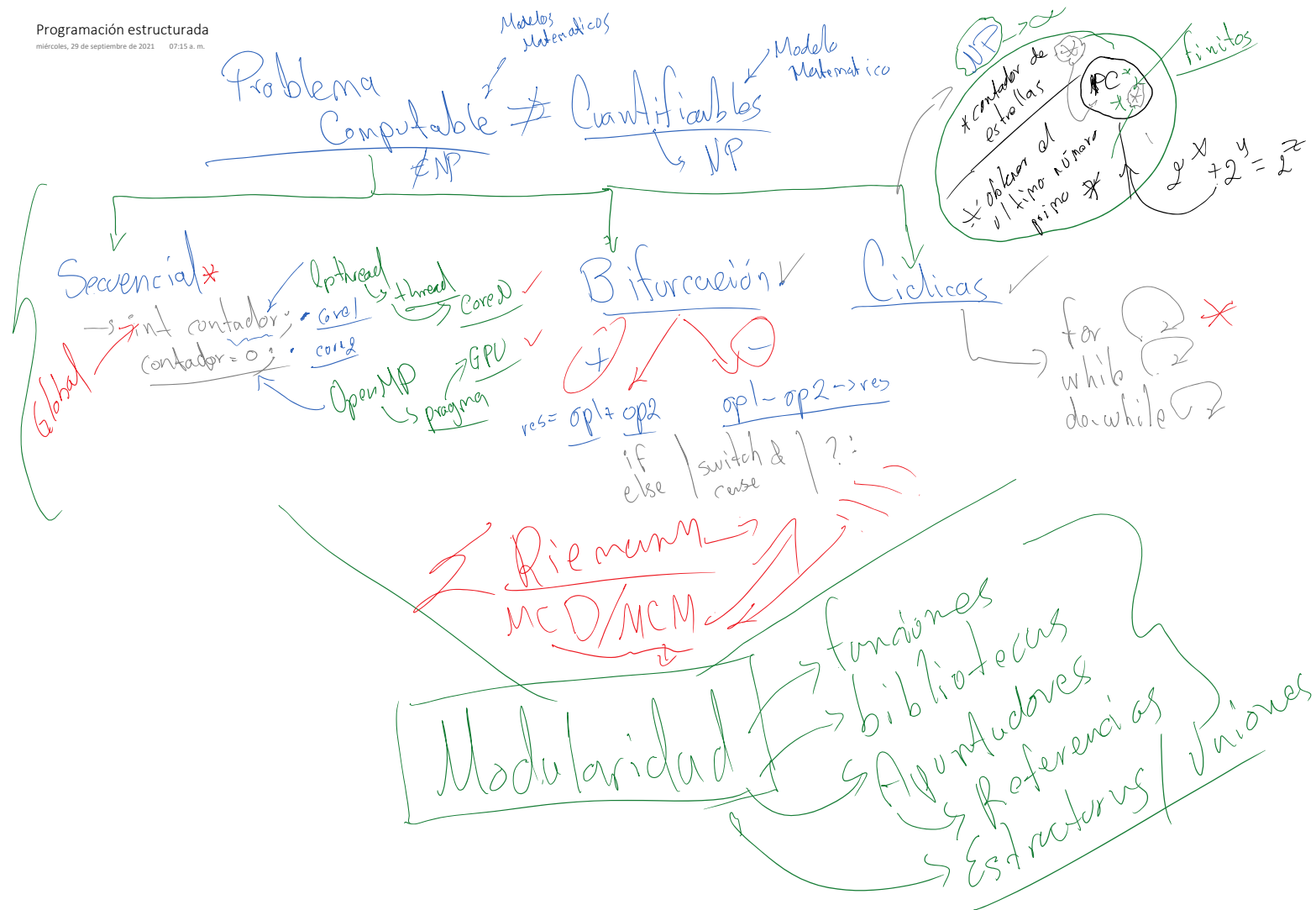
```

Expresiones regulares

```

#include<stdio.h>
void main(){
    char cadena[40];
    //scanf("%[A-Za-z ,.]",cadena); //Formato de símbolos aceptados
    //scanf("%[^.]",cadena); //Formato de símbolo rechazado
    scanf("%[0-9]",cadena);
    printf("Esta es la cadena ingresada: %s\n", cadena);
    fflush(stdin);
    char caractereEnBuffer=getchar();
    printf("Este es el caracter leído:%c",caractereEnBuffer);
}

```



Algoritmos

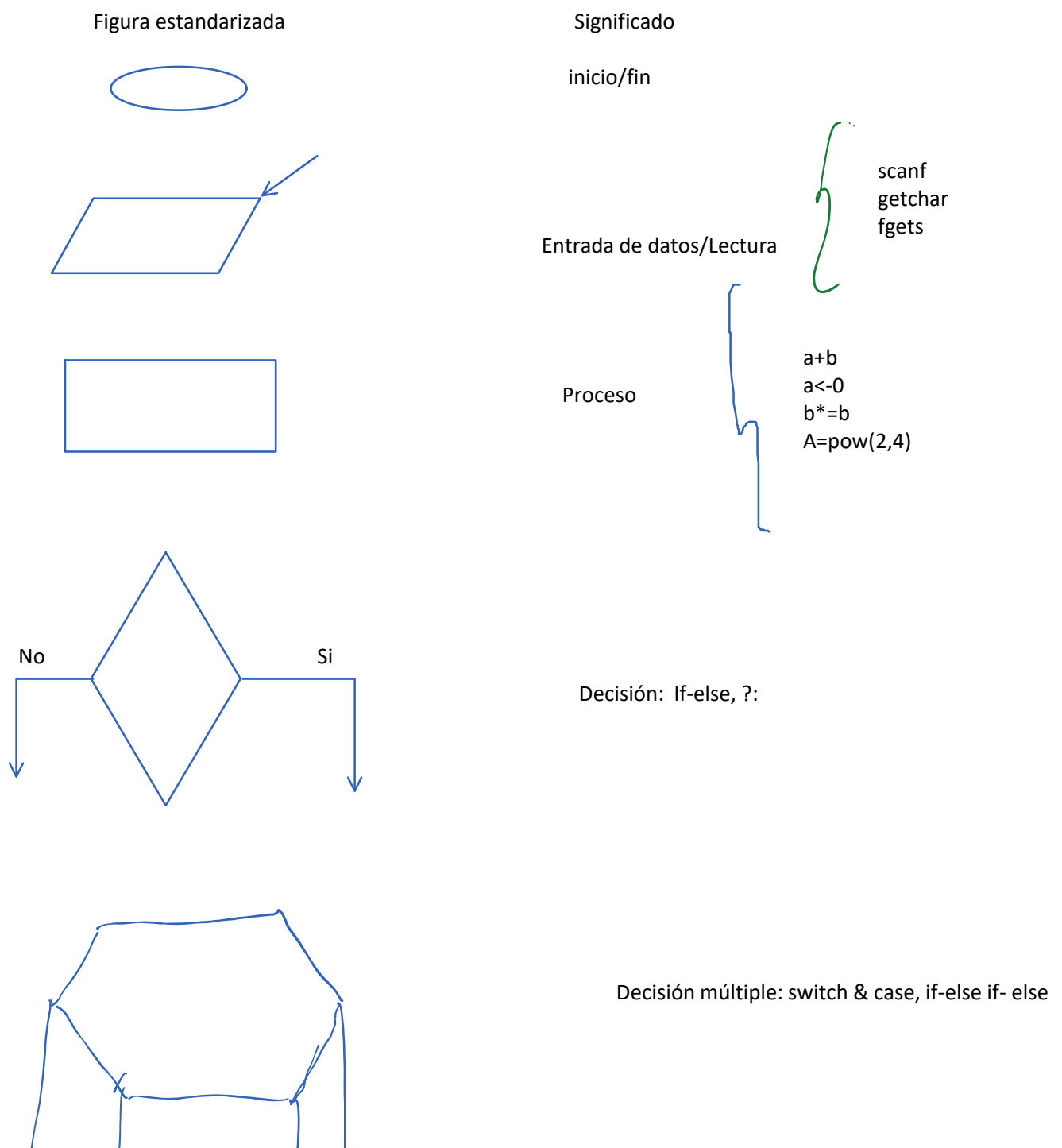
martes, 5 de octubre de 2021

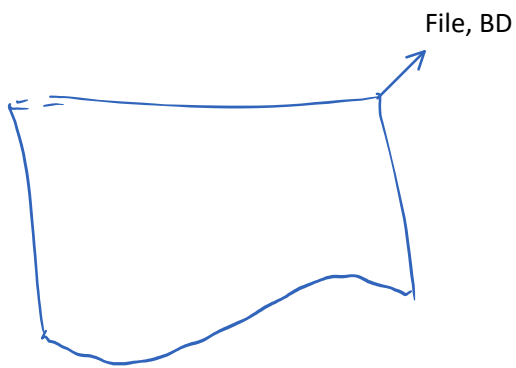
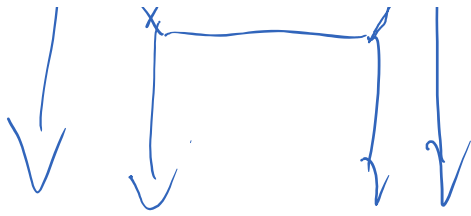
07:33 a. m.

Es un conjunto de instrucciones o reglas bien definidas, ordenadas y finitas, que permite realizar una actividad mediante pasos no ambiguos y efectivos que no generen dudas a quien lo ejecute. (Receta)

Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

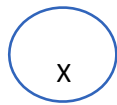
Diagrama de flujo



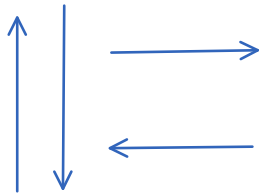


Salida de datos

printf
putchar
puts
fputs



Conector.



Flujo de datos

Ejemplo: Contador de 0 hasta un valor que determine el usuario.

