



| | | | | | |
|-----------------------|---------------------------|----------|------|-------------------------------|----|
| Nombre de la práctica | PROGRAMAS C "Apuntadores" | | | No. | 15 |
| Asignatura: | MÉTODOS NUMÉRICOS | Carrera: | ISIC | Duración de la práctica (Hrs) | 4 |

Valdez Robles Víctor Hugo isic 321

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

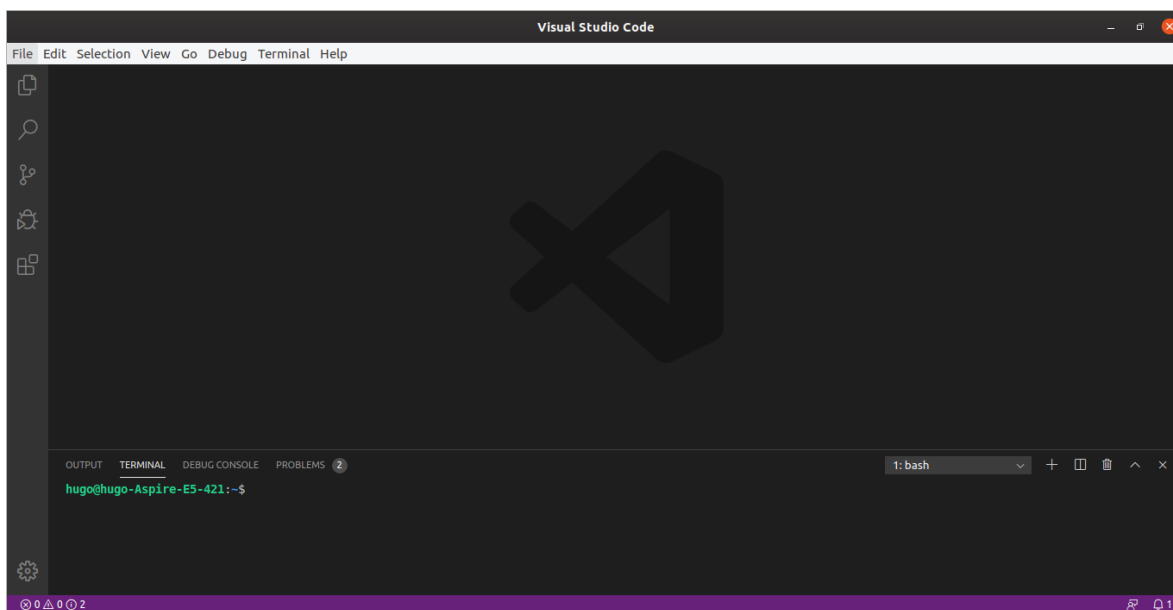
TALLER

III. Material empleado:

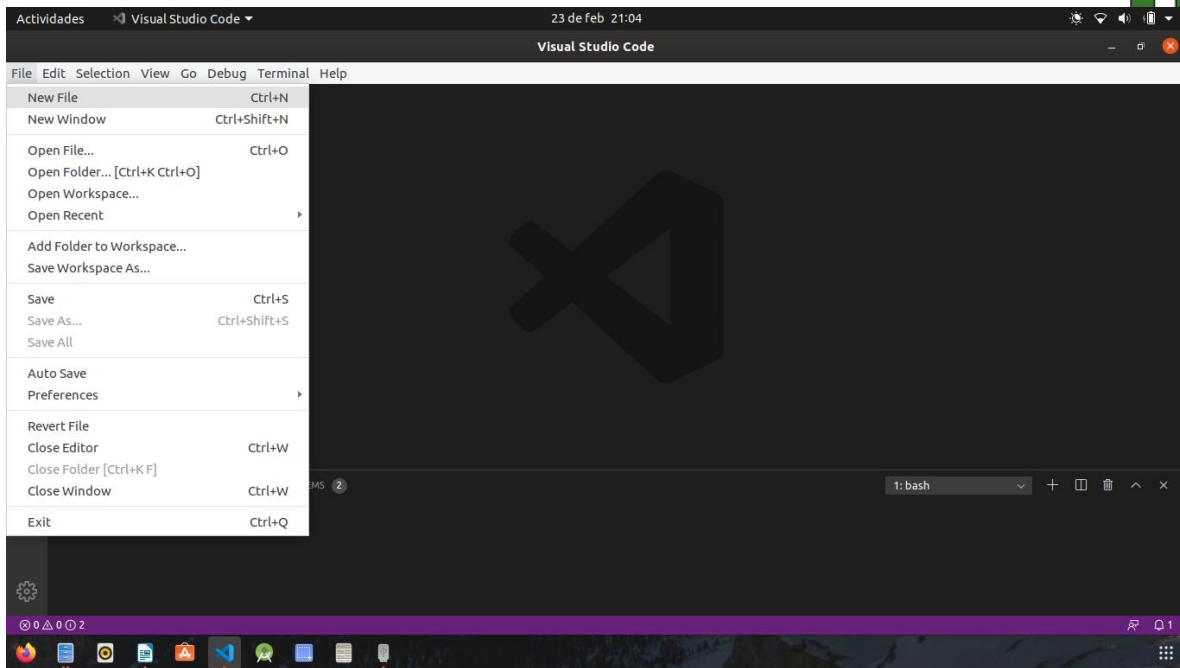
Dev c++

IV. Desarrollo de la práctica:

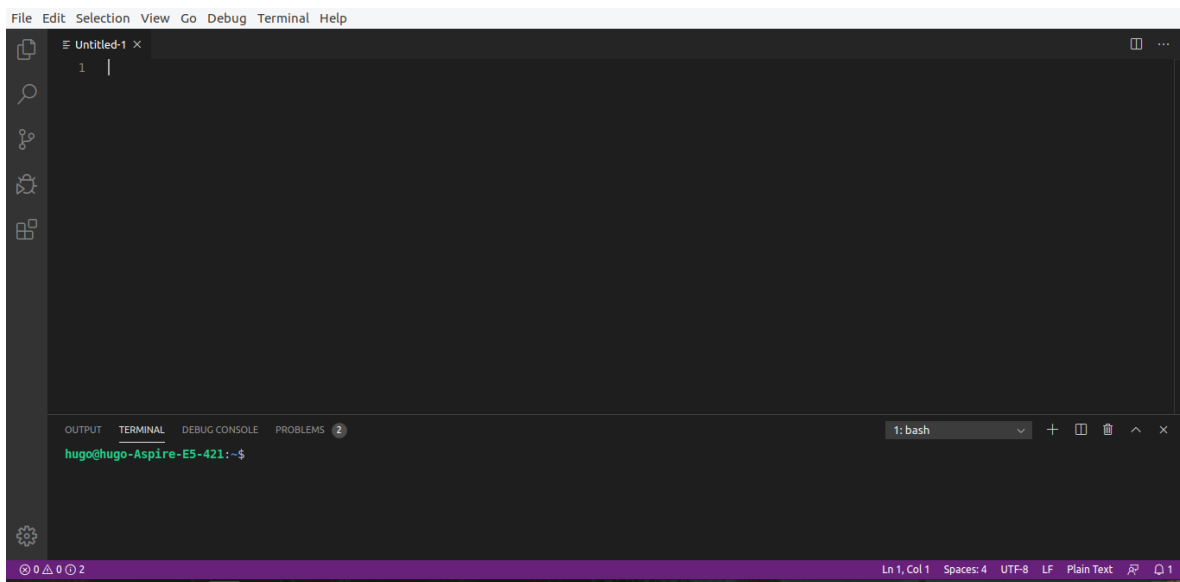
se abre la aplicación de visual studio code



Para crear un proyecto, nos dirigimos a la parte de file y señecionamos "newfile " Al dar clic nos dara una pantalla para desarrollar



En esta parte se muestra la platilla para el código, donde los proyectos se deben de guardar con punto c, para poder ejecutar y compilar en caso de errores poder corregir para que realice su función de manera correcta.



Para empezar con nuestros programas se van a incluir los estándares de entrada y salida, las bibliotecas, el método principal donde estará nuestro código de forma estructurada.

```
home > hugo > Escritorio > programasC > C Aa.c > main()  
1  #include <stdio.h>  
2  #include <stdlib.h>  
3
```

Para todos los codigos siguientes se va a crear con un metodo principal para ejecutar los codigos que se hace de la siguiente manera

```
home > hugo > Escritorio > programasC > C Aa.c > main()  
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  
4  int main(){  
5  
6  
7      system("pause");  
8  
9      return 0;  
10 }
```

en todos los codigos siguientes , en la parte superior se mostraran la estructura de los mismos y en la inferior la compilación y ejecucion

Apuntadores

para los siguientes ejercicios de apuntadores se requiere :generar o crear un apuntador, el cual se genera de la siguiente forma <tipo de dato apuntador> *<identificador del apuntador>

```
int * p;
```

“p” es el nombre que le queramos dar a nuestro apuntador.

Y para guardar la dirección del “apuntado” se usa lo siguiente `p= &x;`
la “x” es una variable de la cual se va a extraer su dirección en memoria

1. En primer ejercicio se busca el valor de “x” y “y” con el uso de apuntadores para obtener su dirección en memoria.

```
C Apuntadores.c X
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores.c > main(void)
1  #include <stdio.h>
2
3  int main(void){
4      int x=17,y;
5
6      int * p;
7
8      p= &x;
9
10
11     printf("El valor de x es: %d\n",*p);
12
13     y=*p+3;
14
15     printf("El valor de y es: %d\n",y);
16
17     return 0;
18 }
19

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ gcc Apuntadores.c -o Apuntadores
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores
El valor de x es: 17
El valor de y es: 20
```

2. en el siguiente ejercicio crean dos variables globales donde una de ellas es un apuntador, el siguiente ejercicio consta en la siguiente estructura: se crea una función donde se obtendrá la dirección de "x" y asignarle un valor al apuntador. En el método principal se muestra el mensaje los valores que se les asignaron a las variables.

```
C Apuntadores1.c X
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores1.c > main(void)
1  #include <stdio.h>
2  int * p,y;
3  void func(){
4      int x=40;
5      p=&x;
6      y=*p;
7      *p=23;
8  }
9  int main(void){
10     func();
11
12     y=*p;
13     *p=25;
14
15     printf("El valor de y es %d \nEl valor de *p es %d \n El valor de p es %p\n",y,*p,p);
16
17
18
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores1
El valor de y es 23
El valor de *p es 25
El valor de p es 0x7ffd4373bf04
```

3. el siguiente ejercicio consta de paso de parámetros por referencia,

```
C Apuntadores2.c X
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores2.c > ...
1  #include <stdio.h>
2
3  void intercambio(int *a, int *b);
4  int main(void){
5  int x = 2;
6  int y = 5;
7
8  printf ("Antes x = %d, y = %d \n", x, y);
9
10 intercambio (&x, &y);
11
12 printf ("Despues x = %d, y = %d \n", x, y);
13
14 }
15
16 void intercambio(int *a, int *b){
17 int temp;
18 temp = *b;
19 *b = *a;
20 *a = temp;
21 }
22

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores2
Antes x = 2, y = 5
Despues x = 5, y = 2
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$
```

4. Devuelve el tamaño en bytes que ocupa un tipo de variable en memoria.

```
C Apuntadores3_Sizeof.c ×
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores3_Sizeof.c > ...
1  #include <stdio.h>
2
3
4  int main(void){
5  char cadena [10];
6  printf ("un int ocupa %ld bytes\n", sizeof(int));
7  printf ("un char ocupa %ld bytes\n", sizeof(char));
8  printf ("un float ocupa %ld bytes\n", sizeof(float));
9  printf ("un double ocupa %ld bytes\n", sizeof(double));
10 printf ("cadena ocupa %ld bytes\n", sizeof(cadena));
11 }
12
13

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS

hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores3
un int ocupa 4 bytes
un char ocupa 1 bytes
un float ocupa 4 bytes
un double ocupa 8 bytes
cadena ocupa 10 bytes
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$
```

5. el siguiente ejercicio nos mostrara el tamaño de bytes que tiene el array y con ayofa del metodo sizeof nos da el tamaño en byts de cualquier variable



```
C Apuntadores4_Sizeof.c X
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores4_Sizeof.c > main(void)
1  #include <stdio.h>
2
3
4  int main(void){
5      int array[10]={1,2,3,4,5,6,7,8,9,10};
6      int len=sizeof(array)/sizeof(int);
7
8      printf("Los bytes del arreglo son: %ld\n",sizeof(array));
9      printf("Cada entero tiene : %ld\n",sizeof(int));
10     printf("El arreglo tiene %d elementos",len);
11 }
12
13
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores4
Los bytes del arreglo son: 40
Cada entero tiene : 4
El arreglo tiene 10 elementoshugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$
```

6.en el siguiente ejercicio se hace uso del método malloc() que reserva memoria sin importar el tipo de datos que almacenará en ella. En el cual se pide el tamaño que se le quiera dar a una cadena de caracteres



```
C Apuntadores6.c C Apuntadores5.c X
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores5.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void){
5      int i,n;
6      char * buffer;
7
8      puts("Teclea la longitud dela cadena");
9      scanf("%d",&i);
10
11     buffer=(char*)malloc((i+1)*sizeof(char));
12
13     if(buffer==NULL)exit(1);
14
15     for ( n = 0; n <i; n++)
16         buffer[n]=rand()%26+'a';
17     buffer[i]='\0';
18
19     printf("Random String: %s\n",buffer);
20     free(buffer);
21
22 }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores5
Teclea la longitud dela cadena
5
Random String: nwlrb
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$
```

7.Crea un arreglo entero de tamaño x, en donde x es ingresado por teclado. Llena todos los elementos del arreglo con datos ingresados por el usuario.Muestra los valores



```
C Apuntadores6.c ×
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores6.c > main(void)
4  int main(void){
5      int i,n;
6
7      puts("Ingrese el tamaño del arreglo");
8      scanf("%d",&n);
9
10     int a[n];
11
12     for ( i = 0; i <n; i++){
13         printf("ingresa el %d:\n",(i+1));
14         scanf("%d",&a[i]);
15     }
16
17     for ( i = 0; i <n; i++){
18         printf("el %d numero es: %d\n",(i+1),a[i]);
19     }
20 }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores6
Ingrese el tamaño del arreglo
3
ingresa el 1:
7
ingresa el 2:
2
ingresa el 3:
9
el 1 numero es: 7
el 2 numero es: 2
el 3 numero es: 9
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$
```



8. en el siguiente programa nos mostrara el contenido que tiene una array con el uso de apuntadores, los cuales deberán ser ingresados por el usuario

```
C Apuntadores9.c  C Apuntadores8.c X  C Apuntadores2.c
home > hugo > Escritorio > programasC > APUNTADES > C Apuntadores8.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void){
5      int i,n;
6      int * buffer,* p_buffer;
7      puts("Ingrese el tamaño del arreglo");
8      scanf("%d",&n);
9      buffer=(int*)malloc((n)*sizeof(int));
10     if(buffer==NULL)exit (1);
11
12     p_buffer=buffer;
13     for ( i = 0; i <n; i++){
14
15         printf("ingresa el %d:\n",i);
16         scanf("%d",p_buffer++);
17     }
18
19     printf("Los valores son\n");
20     for ( i = 0; i <n; i++){
21         printf("el arreglo[%d] es: %d\n",i,*p_buffer++);
22     }
23     return 0;
24 }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
3
ingresa el 0:
2
ingresa el 1:
6
ingresa el 2:
8

Los valores son
el arreglo[0] es: 2
el arreglo[1] es: 6
el arreglo[2] es: 8
```

9. Crea un arreglo de tipo char de tamaño x, en donde x es ingresado por teclado. Llena elemento por elemento del arreglo con letras ingresadas por el usuario. Muestra el arreglo impreso en forma inversa. Todo debe ser manejado con apuntadores.

```
C Apuntadores9.c × C Examen5.c C Apuntadores2.c
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores9.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void){
5      int i,n;
6      int * buffer,* p_buffer;
7      puts("Ingrese el tamaño de la cadena");
8      scanf("%d",&n);
9
10     char cadena[n];
11     buffer=(int*)malloc((n)*sizeof(int));
12     if(buffer==NULL)exit (1);
13
14     p_buffer=buffer;
15     for ( i = 0; i <n; i++){
16
17         printf("ingresa el %d:",i);
18
19         if(i%4==0)printf("\n");
20     }
21     printf("\n");
22     for ( i = n-1; i >= 0; i--){
23         printf("%c",cadena[i]);
24     }
25     printf("\n");
26     return 0;
27 }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
ingresa el 2:l
ingresa el 3:a
aloh
hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$ ./Apuntadores9
Ingrese el tamaño de la cadena
4
ingresa el 0:h
ingresa el 1:o
ingresa el 2:l
ingresa el 3:a
aloh
holahugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES$
```

10.

```
C Apuntadores7.c ×
home > hugo > Escritorio > programasC > APUNTADORES > C Apuntadores7.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(void){
4      int i[10],x;
5      float fl[10];
6      puts("\t\tEntero\t\tFlotante");
7      float *c;
8      for ( x = 0; x<10; x++){
9          c=&fl[x];
10         printf("Elemento %d\t%p\t%p\n",x, &i[x],c);
11     }
12 }
13
14
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

compilation terminated.

hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES\$ gcc Apuntadores7.c -o A

hugo@hugo-Aspire-E5-421:~/Escritorio/programasC/APUNTADORES\$./Apuntadores7

| | Entero | Flotante |
|------------|----------------|----------------|
| Elemento 0 | 0x7ffdc2a6a470 | 0x7ffdc2a6a4a0 |
| Elemento 1 | 0x7ffdc2a6a474 | 0x7ffdc2a6a4a4 |
| Elemento 2 | 0x7ffdc2a6a478 | 0x7ffdc2a6a4a8 |
| Elemento 3 | 0x7ffdc2a6a47c | 0x7ffdc2a6a4ac |
| Elemento 4 | 0x7ffdc2a6a480 | 0x7ffdc2a6a4b0 |
| Elemento 5 | 0x7ffdc2a6a484 | 0x7ffdc2a6a4b4 |
| Elemento 6 | 0x7ffdc2a6a488 | 0x7ffdc2a6a4b8 |
| Elemento 7 | 0x7ffdc2a6a48c | 0x7ffdc2a6a4bc |
| Elemento 8 | 0x7ffdc2a6a490 | 0x7ffdc2a6a4c0 |
| Elemento 9 | 0x7ffdc2a6a494 | 0x7ffdc2a6a4c4 |