

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ingeniería, Arquitectura Y Diseño



Ingenieria en Computacion

Programación Estructurada

\*\*\*

ALUMNO: Miguel Angel Portillo Attwell

MATRÍCULA: 370097

GRUPO: 432

PROFESOR: Pedro Nunez Yepiz



## PARTE 1:

```
MAPA_PE_ACT6_01.cpp • MAPA_PE_ACT6_02.cpp • VSD.cpp
ACTIVIDAD 7 > MAPA_PE_ACT6_01.cpp > cant_dig(void)
1 // MIGUEL ANGEL PORTILLO ATTWELL
2 // FECHA: 18 DE SEPTIEMBRE DEL 2023
3 // DESCRIPCION: PROGRAMA QUE MANEJE UN ESQUELETO DE SELECCION MULTIPLE
4 // QUE FUNCIONE A BASE DE ACTIVIDADES FORMADAS POR FUNCIONES
5 // MAPA_PE_ACT6_01.cpp
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 void menu(void);
10 void fibonacci(void);
11 void factorial(void);
12 void cant_dig(void);
13
14 int main()
15 {
16     menu();
17
18     return 0;
19 }
20
21 void menu(void)
22 {
23     int operacion;
24     do
25     {
26         printf("M E N U\n");
27         printf("ESCOGE LA SIGUIENTE OPERACION CON EL DIGITO CORRESPONDIENTE\n");
28         printf("[1-FIBONACCI]\n");
29         printf("[2-FACTORIAL]\n");
30         printf("[3-CANTIDAD DE DIGITOS]\n");
31         printf("[0-SALIR]\n");
32         printf("OPERACION: ");
33         scanf("%d", &operacion);
34         switch (operacion)
35         {
36             case 1:
37                 fibonacci();
38                 break;
39             case 2:
40                 factorial();
41                 break;
42             case 3:
43                 cant_dig();
44                 break;
45             default:
46                 break;
47         }
48     } while (operacion != 0);
49 }
50
51 > void fibonacci(void) ...
108
109 > void factorial(void) ...
164
165 > void cant_dig(void) ...
```



## Programación Estructurada



```
MAPA_PE_ACT6_01.cpp • MAPA_PE_ACT6_02.cpp • VSD.cpp
ACTIVIDAD 7 > MAPA_PE_ACT6_01.cpp > fibonacci(void)
1 // MIGUEL ANGEL PORTILLO ATTWELL
2 // FECHA: 18 DE SEPTIEMBRE DEL 2023
3 // DESCRIPCION: PROGRAMA QUE MANEJE UN ESQUELETO DE SELECCION MULTIPLE
4 // QUE FUNCIONE A BASE DE ACTIVIDADES FORMADAS POR FUNCIONES
5 // MAPA_PE_ACT6_01.cpp
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 void menu(void);
10 void fibonacci(void);
11 void factorial(void);
12 void cant_dig(void);
13
14 > int main() ...
15
16
17
18
19
20
21 > void menu(void) ...
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51 void fibonacci(void)
52 {
53     int metodo;
54     int n;
55     int a = -1;
56     int s = 1;
57     int r = 0;
58     int i, j;
59     printf("MENU DE FIBONACCI\n");
60     printf("ESCOGE CON QUE TIPO DE CICLO SERA REALIZARA EL PROGRAMA\n");
61     printf("[1-CICLO [FOR]]\n");
62     printf("[2-CICLO [WHILE]]\n");
63     printf("[3-CICLO [DO-WHILE]]\n");
64     printf("CICLO: ");
65     scanf("%d", &metodo);
66     switch (metodo)
67     {
68     case 1:
69         printf("CUANTAS VECES QUIERE QUE SE REPITA EL CICLO? ");
70         scanf("%d", &n);
71         for (i = 0; i < n; i++)
72         {
73             r = a + s;
74             printf("[%d]\n", r);
75             a = s;
76             s = r;
77         }
78         break;
79     case 2:
80         printf("CUANTAS VECES QUIERE QUE SE REPITA EL CICLO? ");
81         scanf("%d", &n);
82         while (n != 0)
83         {
84             r = a + s;
85             printf("[%d]\n", r);
86             a = s;
87             s = r;
88             n--;
89         }
90         break;
91     case 3:
92         printf("CUANTAS VECES QUIERE QUE SE REPITA EL CICLO? ");
93         scanf("%d", &n);
94         do
95         {
96             r = a + s;
97             printf("[%d]\n", r);
98             a = s;
99             s = r;
100             n--;
101         } while (n != 0);
102         break;
103     default:
104         break;
105     }
106 }
107
108
109 > void factorial(void) ...
110
111
112
113
114
115
116 > void cant_dig(void) ...
```



## Programación Estructurada



```
MAPA_PE_ACT6_01.cpp • MAPA_PE_ACT6_02.cpp • VSD.cpp
ACTIVIDAD 7 > MAPA_PE_ACT6_01.cpp > fibonacci(void)
1 // MIGUEL ANGEL PORTILLO ATTWELL
2 // FECHA: 18 DE SEPTIEMBRE DEL 2023
3 // DESCRIPCION: PROGRAMA QUE MANEJE UN ESQUELETO DE SELECCION MULTIPLE
4 // QUE FUNCIONE A BASE DE ACTIVIDADES FORMADAS POR FUNCIONES
5 // MAPA_PE_ACT6_01.cpp
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 void menu(void);
10 void fibonacci(void);
11 void factorial(void);
12 void cant_dig(void);
13
14 > int main() ...
20
21 > void menu(void) ...
50
51 > void fibonacci(void) ...
108
109 void factorial(void)
110 {
111     int metodo;
112     int n;
113     int r;
114     int i;
115     printf("MENU DE FACTORIAL\n");
116     printf("ESCOGE CON QUE TIPO DE CICLO SERA REALIZARA EL PROGRAMA\n");
117     printf("[1-CICLO [FOR]]\n");
118     printf("[2-CICLO [WHILE]]\n");
119     printf("[3-CICLO [DO-WHILE]]\n");
120     printf("CICLO: ");
121     scanf("%d", &metodo);
122     switch (metodo)
123     {
124     case 1:
125         printf("CUANTAS VECES QUIERE QUE SE REPITA EL CICLO? ");
126         scanf("%d", &n);
127         r = n;
128         for (i = n - 1; i > 0; i--)
129         {
130             printf("[%d] X [%d] ", r, i);
131             r = r * i;
132             printf("= [%d]\n", r);
133         }
134         break;
135     case 2:
136         printf("CUANTAS VECES QUIERE QUE SE REPITA EL CICLO? ");
137         scanf("%d", &n);
138         r = n;
139         while (n != 1)
140         {
141             n--;
142             printf("[%d] X [%d] ", r, n);
143             r = r * n;
144             printf("= [%d]\n", r);
145         }
146         break;
147     case 3:
148         printf("CUANTAS VECES QUIERE QUE SE REPITA EL CICLO? ");
149         scanf("%d", &n);
150         r = n;
151         do
152         {
153             n--;
154             printf("[%d] X [%d] ", r, n);
155             r = r * n;
156             printf("= [%d]\n", r);
157         } while (n != 1);
158         break;
159     default:
160         break;
161     }
162 }
163 }
164
165 > void cant_dig(void) ...
```



## Programación Estructurada



```
MAPA_PE_ACT6_01.cpp • MAPA_PE_ACT6_02.cpp • VSD.cpp
ACTIVIDAD 7 > MAPA_PE_ACT6_01.cpp > fibonacci(void)
1 // MIGUEL ANGEL PORTILLO ATTMELL
2 // FECHA: 18 DE SEPTIEMBRE DEL 2023
3 // DESCRIPCION: PROGRAMA QUE MANEJE UN ESQUELETO DE SELECCION MULTIPLE
4 // QUE FUNCIONE A BASE DE ACTIVIDADES FORMADAS POR FUNCIONES
5 // MAPA_PE_ACT6_01.cpp
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 void menu(void);
10 void fibonacci(void);
11 void factorial(void);
12 void cant_dig(void);
13
14 > int main() ...
15
16 > void menu(void) ...
17
18 > void fibonacci(void) ...
19
20 > void factorial(void) ...
21
22 > void cant_dig(void)
23 {
24     int metodo;
25     int dig = 0;
26     int n;
27     int r;
28     int i;
29     int x;
30     printf("MENU DE CANTIDAD DE DIGITOS\n");
31     printf("ESCOGE CON QUE TIPO DE CICLO SERA REALIZARA EL PROGRAMA\n");
32     printf("[1-CICLO [FOR]]\n");
33     printf("[2-CICLO [WHILE]]\n");
34     printf("[3-CICLO [DO-WHILE]]\n");
35     printf("CICLO: ");
36     scanf("%d", &metodo);
37     switch (metodo)
38     {
39     case 1:
40         dig = 0;
41         printf("INGRESE EL NUMERO DEL QUE DESEA SABER CUANTOS DIGITOS TIENE: ");
42         scanf("%d", &n);
43         x = n;
44         for (i = 0; x > 0; i++)
45         {
46             r = n % 10;
47             dig++;
48             x = x / 10;
49         }
50
51         printf("EL NUMERO [%d]\n", n);
52         printf("TIENES [%d] DIGITO/S\n", dig);
53
54         break;
55     case 2:
56         printf("INGRESE EL NUMERO DEL QUE DESEA SABER CUANTOS DIGITOS TIENE: ");
57         scanf("%d", &n);
58         while (n != 0)
59         {
60             r = n % 10;
61             dig++;
62             n = n / 10;
63         }
64
65         printf("EL NUMERO [%d]\n", n);
66         printf("TIENES [%d] DIGITO/S\n", dig);
67
68         break;
69     case 3:
70         printf("INGRESE EL NUMERO DEL QUE DESEA SABER CUANTOS DIGITOS TIENE: ");
71         scanf("%d", &n);
72         do
73         {
74             r = n % 10;
75             dig++;
76             n = n / 10;
77         } while (n != 0);
78
79         printf("EL NUMERO [%d]\n", n);
80         printf("TIENES [%d] DIGITO/S\n", dig);
81
82         break;
83     default:
84         break;
85     }
86 }
87 }
```



## PARTE2:

```
ACTIVIDAD 7 > MAPA_PE_ACT6_02.cpp > calif_intent(void)
1 // MIGUEL ANGEL PORTILLO ATTWELL
2 // FECHA: 18 DE SEPTIEMBRE DEL 2023
3 // DESCRIPCION: PROGRAMA QUE MANEJE UN ESQUELETO DE SELECCION MULTIPLE
4 // QUE FUNCIONE A BASE DE ACTIVIDADES FORMADAS POR FUNCIONES
5 // MAPA_PE_ACT6_02.cpp
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <time.h>
9 #define NUM_ALUMNOS 40
10 #define NUM_UNIDADES 5
11
12 void menu(void);
13 void exa_dere(void);
14 void tab_mult(void);
15 void rango_cant(void);
16 void kg_turista(void);
17 void calif_intent(void);
18
19 int main()
20 {
21     menu();
22
23     return 0;
24 }
25
26 void menu(void)
27 {
28     int operacion;
29     do
30     {
31         printf("M E N U\n");
32         printf("ESCOGE LA SIGUIENTE OPERACION CON EL DIGITO CORRESPONDIENTE\n");
33         printf("[1-DERECHO A EXAMEN]\n");
34         printf("[2-TABLAS DE MULTIPLICAR]\n");
35         printf("[3-CANTIDAD DE NUMEROS DENTRO DEL RANGO]\n");
36         printf("[4-PESO TURISTAS]\n");
37         printf("[5-CALIFICACIONES SEMESTRE]\n");
38         printf("[0-SALIR]\n");
39         printf("OPERACION: ");
40         scanf("%d", &operacion);
41         switch (operacion)
42         {
43             case 1:
44                 exa_dere();
45                 break;
46             case 2:
47                 tab_mult();
48                 break;
49             case 3:
50                 rango_cant();
51                 break;
52             case 4:
53                 kg_turista();
54                 break;
55             case 5:
56                 calif_intent();
57                 break;
58             default:
59                 break;
60         }
61     } while (operacion != 0);
62 }
```



```
ACTIVIDAD 7 > MAPA_PE_ACT6_02.cpp > calif_intent(void)
1 > // MIGUEL ANGEL PORTILLO ATTWELL ...
6 > #include <stdio.h>...
9 #define NUM_ALUMNOS 40
10 #define NUM_UNIDADES 5
11
12 void menu(void);
13 void exa_dere(void);
14 void tab_mult(void);
15 void rango_cant(void);
16 void kg_turista(void);
17 void calif_intent(void);
18
19 > int main() ...
25
26 > void menu(void) ...
63
64 void exa_dere(void)
65 {
66     int calificaciones[NUM_ALUMNOS][NUM_UNIDADES];
67     int promedios[NUM_ALUMNOS];
68     int countNoExamen = 0;
69
70     for (int i = 0; i < NUM_ALUMNOS; i++)
71     {
72
73         printf("Ingrese las calificaciones para el alumno %d (en el formato 'calif1 calif2 calif3 calif4 calif5')\n", i);
74
75         for (int j = 0; j < NUM_UNIDADES; j++)
76         {
77
78             if (scanf("%d", &calificaciones[i][j]) != 1)
79             {
80
81                 printf("Error al leer calificación. Asegúrese de ingresar números.\n");
82             }
83         }
84     }
85
86     for (int i = 0; i < NUM_ALUMNOS; i++)
87     {
88
89         int suma = 0;
90
91         for (int j = 0; j < NUM_UNIDADES; j++)
92         {
93
94             suma += calificaciones[i][j];
95         }
96
97         promedios[i] = suma / NUM_UNIDADES;
98
99         if (promedios[i] < 50)
100         {
101             countNoExamen++;
102         }
103     }
104
105     printf("\nCantidad de alumnos que no tienen derecho al examen de nivelación: %d\n", countNoExamen);
106 }
107
108 > void tab_mult(void) ...
128
129 > void rango_cant(void) ...
168
169 > void kg_turista(void) ...
213
214 > void calif_intent(void) ...
```



```
ACTIVIDAD 7 > MAPA_PE_ACT6_02.cpp > calif_intent(void)
1 > // MIGUEL ANGEL PORTILLO ATTWELL ...
6 > #include <stdio.h> ...
9 #define NUM_ALUMNOS 40
10 #define NUM_UNIDADES 5
11
12 void menu(void);
13 void exa_dere(void);
14 void tab_mult(void);
15 void rango_cant(void);
16 void kg_turista(void);
17 void calif_intent(void);
18
19 > int main() ...
25
26 > void menu(void) ...
63
64 > void exa_dere(void) ...
107
108 v void tab_mult(void)
109 {
110     int i;
111     int j;
112     int resultado;
113     int tabla;
114
115 v for (i = 1; i < 11; i++)
116 {
117
118     printf("TABLA DEL [%d]\n", i);
119 v for (j = 1; j < 11; j++)
120 {
121     resultado = i * j;
122
123     printf("[%d] X [%d] = [%d]\n", i, j, resultado);
124 }
125     printf("\n");
126 }
127 }
128
129 > void rango_cant(void) ...
168
169 > void kg_turista(void) ...
213
214 > void calif_intent(void) ...
```





## Programación Estructurada



```
ACTIVIDAD 7 > MAPA_PE_ACT6_02.cpp > calif_intent(void)
1 > // MIGUEL ANGEL PORTILLO ATTWELL ...
2 > #include <stdio.h> ...
3 > #define NUM_ALUMNOS 40
4 > #define NUM_UNIDADES 5
5
6 > void menu(void);
7 > void exa_dere(void);
8 > void tab_mult(void);
9 > void rango_cant(void);
10 > void kg_turista(void);
11 > void calif_intent(void);
12
13 > int main() ...
14
15 > void menu(void) ...
16
17 > void exa_dere(void) ...
18
19 > void tab_mult(void) ...
20
21 > void rango_cant(void)
22 {
23     int numeros[100];
24     int cantidadNumeros = 0;
25     int suma = 0;
26     int rangoInicio, rangoFin;
27
28     printf("Ingrese el rango (inicio fin): ");
29     char input[100];
30     gets(input);
31     sscanf(input, "%d %d", &rangoInicio, &rangoFin);
32
33     while (1)
34     {
35         printf("Ingrese un número (o -1 para terminar): ");
36         gets(input);
37
38         int numero = atoi(input);
39
40         if (numero == -1)
41         {
42             break;
43         }
44
45         if (numero >= rangoInicio && numero <= rangoFin)
46         {
47             numeros[cantidadNumeros] = numero;
48             suma += numero;
49             cantidadNumeros++;
50         }
51     }
52
53     float media = (float)suma / cantidadNumeros;
54
55     printf("Suma de los números dentro del rango: %d\n", suma);
56     printf("Media aritmética de los números dentro del rango: %.2f\n", media);
57 }
58
59 > void kg_turista(void) ...
60
61 > void calif_intent(void) ...
```



```
ACTIVIDAD 7 > MAPA_PE_ACT6_02.cpp > calif_intent(void)
1 > // MIGUEL ANGEL PORTILLO ATTWELL ...
6 > #include <stdio.h>...
9 #define NUM_ALUMNOS 40
10 #define NUM_UNIDADES 5
11
12 void menu(void);
13 void exa_dere(void);
14 void tab_mult(void);
15 void rango_cant(void);
16 void kg_turista(void);
17 void calif_intent(void);
18
19 > int main() ...
25
26 > void menu(void) ...
63
64 > void exa_dere(void) ...
107
108 > void tab_mult(void) ...
128
129 > void rango_cant(void) ...
168
169 > void kg_turista(void)
170 {
171     int numTuristas;
172     double pesoTurista, pesoTotal = 0;
173     int capacidadTuristas = 10;
174     double capacidadPesoKilos = 700;
175     double sobrepesoPermitido = 0.15; // 15% de sobrepeso
176
177     printf("Por favor, ingrese el número de turistas: ");
178     scanf("%d", &numTuristas);
179
180     if (numTuristas <= 0)
181     {
182         printf("Debe ingresar al menos un turista.\n");
183         return;
184     }
185
186     for (int i = 1; i <= numTuristas; i++)
187     {
188         printf("Peso del turista %d (en kilos): ", i);
189         scanf("%lf", &pesoTurista);
190         pesoTotal += pesoTurista;
191     }
192
193     double promedioPeso = pesoTotal / numTuristas;
194
195     if (numTuristas <= capacidadTuristas && pesoTotal <= capacidadPesoKilos &&
196         pesoTotal <= (1 + sobrepesoPermitido) * capacidadPesoKilos)
197     {
198         printf("La embarcación puede salir a navegar.\n");
199         printf("Promedio de peso de los turistas: %.2lf kilos.\n", promedioPeso);
200         printf("Se cumplió la condición de cantidad de turistas.\n");
201     }
202     if (pesoTotal <= capacidadPesoKilos && pesoTotal <= (1 + sobrepesoPermitido) * capacidadPesoKilos)
203     {
204         printf("La embarcación puede salir a navegar.\n");
205         printf("Promedio de peso de los turistas: %.2lf kilos.\n", promedioPeso);
206         printf("Se cumplió la condición de peso total.\n");
207     }
208     else
209     {
210         printf("La embarcación NO puede salir a navegar.\n");
211     }
212 }
213
214 > void calif_intent(void) ...
```



```
ACTIVIDAD 7 > MAPA_PE_ACT6_02.cpp > calif_intent(void)
1 > // MIGUEL ANGEL PORTILLO ATTWELL ...
6 > #include <stdio.h> ...
9 #define NUM_ALUMNOS 40
10 #define NUM_UNIDADES 5
11
12 void menu(void);
13 void exa_dere(void);
14 void tab_mult(void);
15 void rango_cant(void);
16 void kg_turista(void);
17 void calif_intent(void);
18
19 > int main() ...
25
26 > void menu(void) ...
63
64 > void exa_dere(void) ...
107
108 > void tab_mult(void) ...
128
129 > void rango_cant(void) ...
168
169 > void kg_turista(void) ...
213
214 > void calif_intent(void)
215 {
216     int intentos = 3;
217     int calif[3];
218     float sumaCalif = 0;
219
220     for (int i = 0; i < intentos; i++)
221     {
222         printf("Ingrese la calificación del examen parcial %d: ", i + 1);
223         char input[100];
224         gets(input);
225
226         calif[i] = atoi(input);
227
228         sumaCalif += calif[i];
229     }
230
231     float prome = sumaCalif / intentos;
232
233     if (prome >= 60)
234     {
235         printf("Aprobado. Promedio final: %.2f\n", prome);
236     }
237
238     if (prome >= 40)
239     {
240         printf("Repetir materia. Promedio final: %.2f\n", prome);
241     }
242
243     else
244     {
245         printf("Baja temporal. Promedio final: %.2f\n", prome);
246     }
247 }
```

GIT HUB: [https://github.com/MAIKYPORTILLO67/PROGRAMACION\\_ESTRUCTURADA](https://github.com/MAIKYPORTILLO67/PROGRAMACION_ESTRUCTURADA)