



UNIVERSIDAD ABIERTA Y A DISTANCIA DE MÉXICO

División de Ciencias Exactas, Ingeniería y Tecnología

Desarrollo de software

Semestre: Primer semestre.

Unidad didáctica: 01 – Fundamentos de programación.

Unidad de aprendizaje: **Unidad 2.** Introducción al lenguaje C.

Actividad: Evidencia de aprendizaje. Programación en DEV C.

Nombre del estudiante: Orlando Antonio Maturano Pizaña

Matrícula: ES251107915

Grupo: DS-DFPR-2501-B1-013

Figura académica: JOSÉ MANUEL NORIEGA BARRERA

Fecha de entrega: 3 de marzo de 2025

Estado de México, a 26 de febrero del 2025.

Diseño: DL-CPL

ÍNDICE

INTRODUCCIÓN	1
DESARROLLO DE LA ACTIVIDAD	2
Análisis del problema	2
Pseudocódigo	4
Diagrama de flujo	7
Prueba de escritorio	8
Desarrollo del código en lenguaje C	11
CONCLUSIÓN	17
CITAS DE AUTOR	18
REFERENCIAS BIBLIOGRÁFICAS	18

INTRODUCCIÓN

El lenguaje C, desarrollado por Dennis Ritchie en 1972 y estandarizado posteriormente como ANSI C, es un pilar fundamental en la programación moderna debido a su eficiencia y portabilidad. Su diseño estructurado permite comprender conceptos esenciales como tipos de datos, variables, operadores y funciones, sentando las bases para lenguajes posteriores. En el contexto educativo, dominar C facilita el pensamiento lógico-algorítmico y la resolución de problemas mediante estructuras de control selectivas, repetitivas y anidadas.

Esta actividad tiene como objetivo principal implementar soluciones programáticas en Dev-C++, aplicando metodologías basadas en problemas y casos. Se abordarán aspectos clave como la sintaxis del lenguaje, la gestión de bibliotecas, la codificación de algoritmos y la depuración del código.

Mediante ejemplos prácticos se busca fortalecer el diseño de algoritmos, traducción de soluciones de programación a código y uso de herramientas de compilación. La integración de estructuras de control y funciones estándar permitirá resolver problemas complejos, fomentando un enfoque sistemático y estructurado en la programación.

DESARROLLO DE LA ACTIVIDAD

Análisis del problema

En primer lugar se deben imprimir mis datos de estudiante que son: Nombre, matrícula y grupo. Después se debe mostrar un menú de opciones las cuales son:

1. Ingresar el puntaje del estudiante.
2. Salir del programa.

Al seleccionar la opción 1., e ingresar el puntaje del estudiante, el programa debe imprimir en pantalla una recomendación de la carrera a la que el estudiante podría aspirar, mostrando en pantalla la carrera.

Si el puntaje no alcanza el mínimo requerido se debe imprimir en pantalla: "El puntaje no es adecuado para cursar una carrera en la UnADM.

Para esto el programa debe interactuar con los rangos de puntaje de ingreso para las carreras de la UnADM. Los cuales son:

Carreras	Puntaje
Ingeniería en Desarrollo de Software	100
Ingeniería en Logística y Transporte	90-99
Ingeniería en Telemática	80-89
Licenciatura en Enseñanza de las Matemáticas	70-79

Variables:

1. int opcion: Almacena la elección del usuario en el menú (1 o 2).
2. int puntaje: Guarda el puntaje ingresado por el usuario.

Constantes:

1. Puntaje mínimo para Ingeniería en Desarrollo de Software: 100

```
#define MIN_SOFTWARE 100
```
2. Límite superior para Ingeniería en Logística y Transporte: 99

```
#define MIN_LOGISTICA 99
```
3. Límite inferior para Ingeniería en Logística y Transporte: 90

```
#define MIN_LOGISTICA 90
```
4. Límite superior para Ingeniería en Telemática: 89

```

#define MIN_TELEMATICA 89

5. Límite inferior para Ingeniería en Telemática: 80
#define MIN_TELEMATICA 80

6. Límite superior para Licenciatura en Enseñanza de las Matemáticas: 79

#define MIN_MATEMATICAS 79

7. Límite inferior para Licenciatura en Enseñanza de las Matemáticas: 70

#define MIN_MATEMATICAS 70

8. Puntaje mínimo para ser admitido: 70

#define PUNTAJE_MINIMO 70

```

Expresiones matemáticas (comparaciones):

- Evaluación de rangos:

```

if (puntaje >= MIN_SOFTWARE) { ... }
else if (puntaje >= MIN_LOGISTICA && puntaje <= MAX_LOGISTICA) { ... }
else if (puntaje >= MIN_TELEMATICA && puntaje <= MAX_TELEMATICA) { ... }
else if (puntaje >= MIN_MATEMATICAS && puntaje <= MAX_MATEMATICAS) { ... }
}
else { ... } // Puntaje < PUNTAJE_MINIMO

```

Librerías:

- #include <stdio.h>: Para funciones printf() y scanf().
- #include <stdlib.h>: Para la función de system("pause");
- #include <locale.h>: Para la configuración regional en sistemas UNIX
- #ifdef _WIN32: Si el sistema en el que se abre el programa es windows.
- #include <windows.h>: Para las funciones de la API de windows como UTF-8.
- #endif

Funciones:

- main(): Contiene el bucle principal (do-while), menú y lógica de evaluación.

Pseudocódigo:

Se utilizaron las siguientes abreviaturas: L: Logística, T: Telemática, M: Matemáticas.

Orlando Antonio Maturano Pizaña
ES251107975

Inicio

// Librerías y configuración

Incluir <stdio.h>

Incluir <stdlib.h>

Incluir <locale.h>

Incluir <windows.h>

// Definir constantes

Definir MIN_SOFTWARE = 100

Definir MIN_L = 90, MAX_L = 99

Definir MIN_T = 80, MAX_T = 89

Definir MIN_M = 70, MAX_M = 79

Definir PUNTAJE_MINIMO = 70

// Configuración UTF-8

Comprobar _WIN32_UTF-8

Condición regional actual en.US.UTF-8

// Mostrar datos de estudiante.

Imprimir "Nombre: Orlando Antonio
Maturano Pizaña"

Orlando Antonio Maturano Pizanq
ES251107915

Imprimir "Matrícula: ES251107915"

Imprimir "Grupo: DS-DFPR-2501-B1-
013"

// Bucle principal del menú

Repetir:

Imprimir "(1) Ingresar puntaje del
aspirante"

Imprimir "(2) Salir del programa"

Imprimir "Introduce la opción deseada"

Leer opción

Según opción:

Caso 1:

Imprimir "Introduce el puntaje:"

Leer puntaje

Si puntaje >= MIN_SOFTWARE:

Imprimir "Ingeniería en Desarrollo de
Software"

Sino Si puntaje entre MIN_L y MAX_L:

Imprimir "Ingeniería en Logística y
Transporte"

Orlando Antonio Maturano Pizaña
ES251107915

Sino Si puntaje entre MIN_T y MAX_T:

Imprimir "Ingeniería en Telemática"

Sino Si puntaje entre MIN_M y MAX_M:

Imprimir "Licenciatura en Enseñanza
de Matemáticas"

Sino Si puntaje < PUNTAJE_MINIMO:

Imprimir "El puntaje no es adecuado
para cursar una carrera en UnADM"

Caso 2:

Terminar bucle

Mientras la opción no sea 2

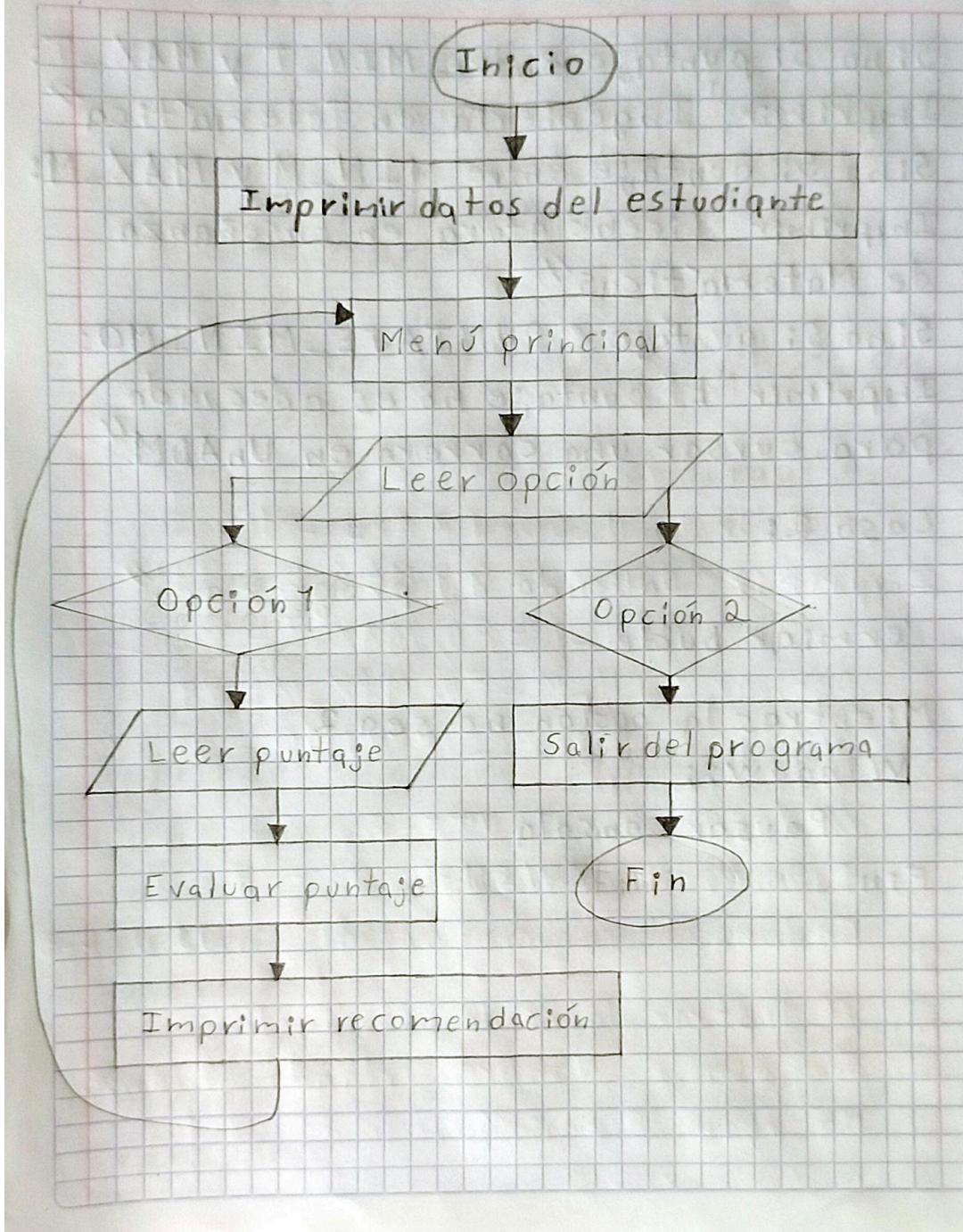
Windows:

Pausar consola

Fin

Diagrama de flujo

Orlando Antonio Maturano Pizaña
ES 25 7107915



Prueba de escritorio

Orlando Antonio Maturano Pizana
ES257107975

Caso 1: Puntaje válido (100 → Software)

Paso	Acción	Entrada	Salida esperada	Resultado
1	Iniciar programa	-	Datos del estudiante	✓
2	Mostrar menú	-	Opciones 1 y 2	✓
3	Seleccionar opción 1	1	Solicitar puntaje	✓
4	Ingresar puntaje	100	Ingeniería en Desarrollo de Software	✓
5	Volver al menú	-	Reaparece el menú	✓

Caso 2: Puntaje invalido (65 → No apto)

Paso	Acción	Entrada	Salida esperada	Resultado
1	Iniciar programa	-	Datos del estudiante	✓

Orlando Antonio Maturano Pizana
ES251107975

2	Mostrar menú	-	Opciones 1 y 2	✓
3	Seleccionar opción 1	1	Solicitar puntaje	✓
4	Ingresar puntaje	65	"El puntaje no es adecuado para cursar una carrera en UnADM"	✓
5	Volver al menú	-	Reaparece el menú	✓

Caso 3: Opción invalida en el menú (Opción 3)

Paso	Acción	Entrada	Salida esperada	Resultado
1	Iniciar programa	-	Datos del estudiante.	✓
2	Mostrar menú	-	Opciones 1 y 2	✓
3	Ingresar opción invalida	3	"Opción invalida, intente de nuevo."	✓
4	Volver al menú	-	Reaparece el menú.	✓

Orlando Antonio Maturano Pizaña
ES251107915

Caso 4: Salir del programa (Opción 2)

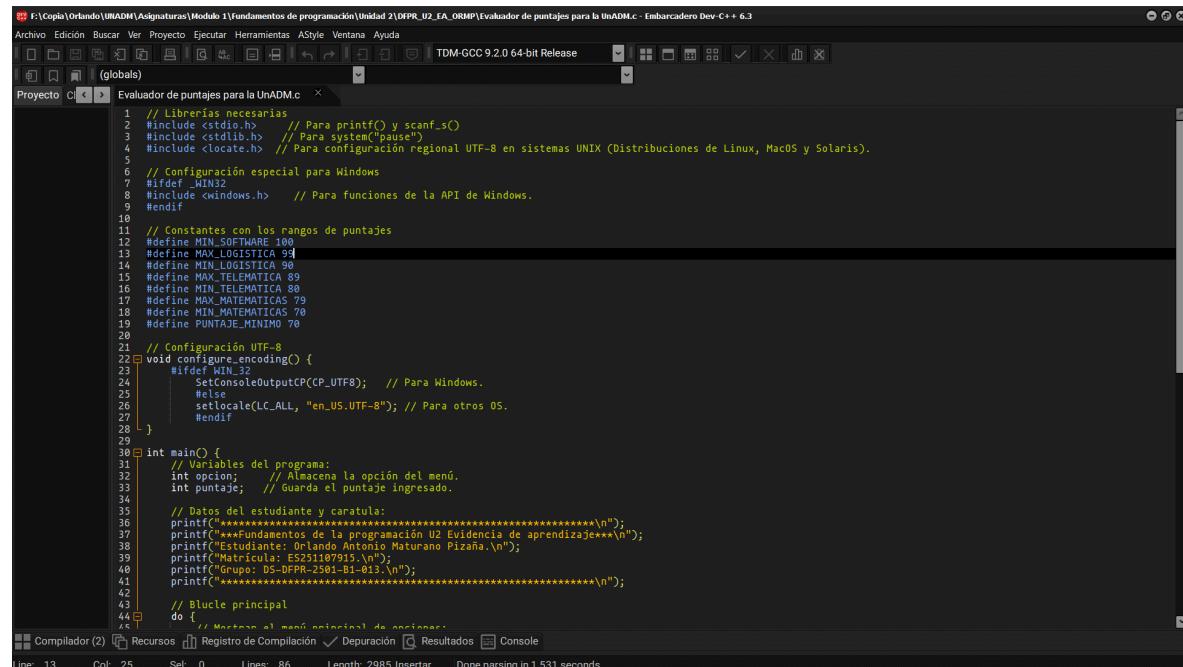
Paso	Acción	Entrada	Salida esperada	Resultado
1	Iniciar programa	-	Datos del estudiante	✓
2	Mostrar menú	-	Opciones 1 y 2	✓
3	Seleccionar opción 2	2	-	✓
4	Finalizar programa	-	El programa termina	✓

Caso 5: Entrada no numérica en el puntaje.

1	Iniciar programa	-	Datos del estudiante	✓
2	Mostrar menú	-	Opciones 1 y 2	✓
3	Seleccionar opción 1	1	Solicitar puntaje	✓
4	Ingresar texto	"Abc"	"Solo se permite caracteres numéricos de puntaje."	✓

Desarrollo del código en lenguaje C

Primero empecé escribiendo el código en el IDE Dev-C++:

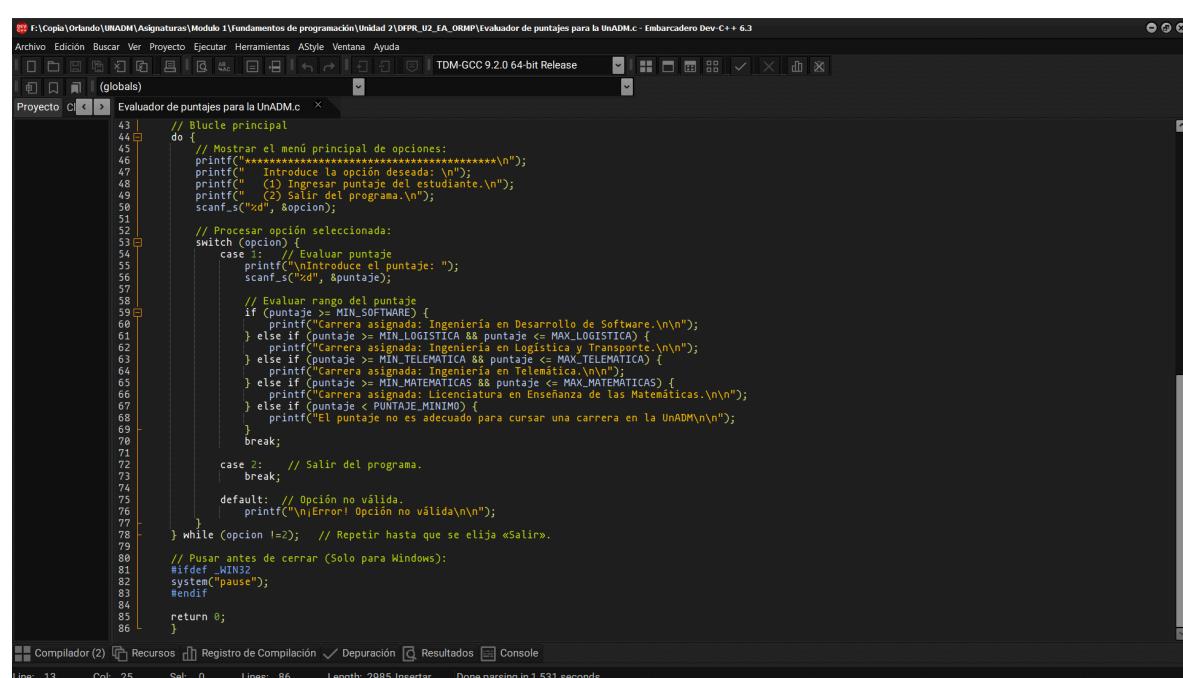


```

F:\Copia\Orlando\UNADM\Asignaturas\Modulo 1\Fundamentos de programación\Unidad 2\DFPR_U2_EA_ORMP\Evaluador de puntajes para la UnADM.c - Embarcadero Dev-C++ 6.3

Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
TDM-GCC 9.2.0 64-bit Release
(globals)
Proyecto Clásico Evaluador de puntajes para la UnADM.c
1 // Librerías necesarias
2 #include <stdio.h> // Para printf() y scanf_s()
3 #include <stdlib.h> // Para system("pause")
4 #include <locale.h> // Para configuración regional UTF-8 en sistemas UNIX (Distribuciones de Linux, MacOS y Solaris).
5
6 // Configuración especial para Windows
7 #ifndef _WIN32
8 #include <windows.h> // Para funciones de la API de Windows.
9 #endif
10
11 // Constantes con los rangos de puntajes
12 #define MIN_SOFTWARE 100
13 #define MAX_LOGISTICA 95
14 #define MIN_LOGISTICA 90
15 #define MAX_TELEMATICA 89
16 #define MIN_TELEMATICA 88
17 #define MAX_MATEMATICAS 79
18 #define MIN_MATEMATICAS 70
19 #define PUNTAJE_MINIMO 70
20
21 // Configuración UTF-8
22 void configure_encoding() {
23     #ifdef _WIN32
24         SetConsoleOutputCP(CP_UTF8); // Para Windows.
25     #else
26         setlocale(LC_ALL, "en_US.UTF-8"); // Para otros OS.
27     #endif
28 }
29
30 int main() {
31     // Variables del programa:
32     int opcion; // Almacena la opción del menú.
33     int puntaje; // Guarda el puntaje ingresado.
34
35     // Datos del estudiante y cursatua:
36     printf("*****\n***Fundamentos de la programación U2 Evidencia de aprendizaje***\n");
37     printf("*****\n");
38     printf("Estudiante: Orlando Antonio Matrano Pizáñ.\n");
39     printf("Matrícula: ES2510107915.\n");
40     printf("Grupo: DS-DFPR-2501-81-013.\n");
41     printf("*****\n");
42
43     // Bucle principal
44     do {
45         // Mostrar el menú principal de opciones:

```



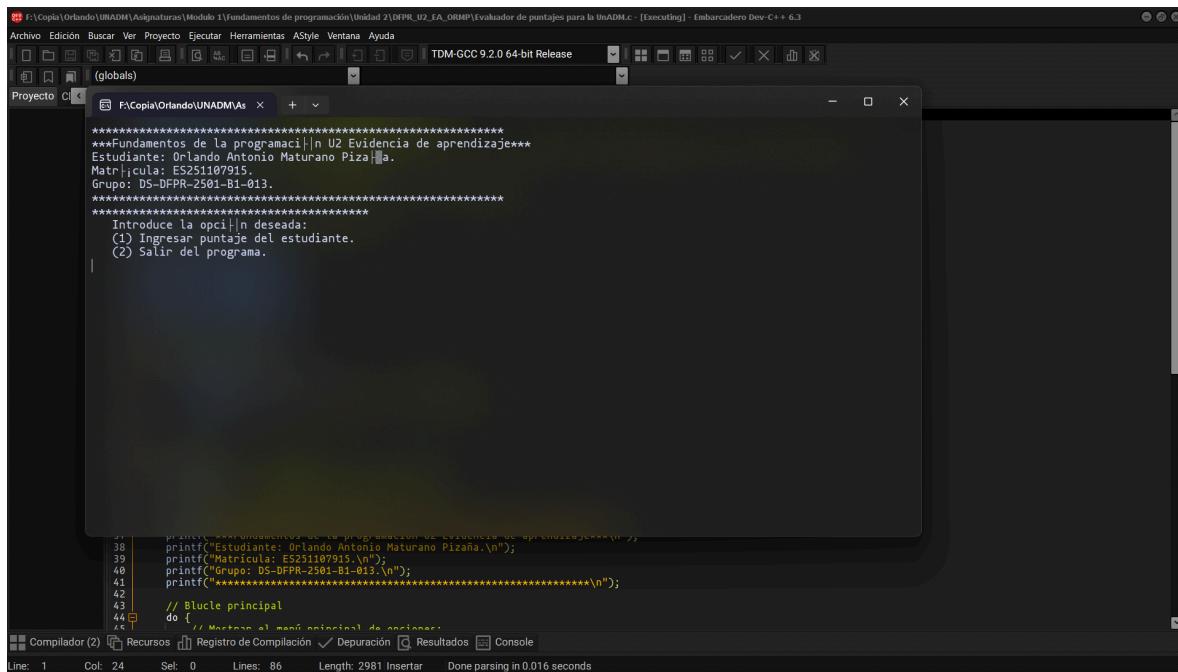
```

46             // Mostrar el menú principal de opciones:
47             printf("*****\n");
48             printf(" Introduce la opción deseada: \n");
49             printf(" (1) Ingresar puntaje del estudiante.\n");
50             printf(" (2) Salir del programa.\n");
51             scanf_s("%d", &opcion);
52
53             // Procesar opción seleccionada:
54             switch (opcion) {
55                 case 1: // Evaluar puntaje
56                     printf("Introduce el puntaje: ");
57                     scanf_s("%d", &puntaje);
58
59                     // Evaluar rango del puntaje
60                     if (puntaje >= MIN_SOFTWARE) {
61                         printf("Carrera asignada: Ingeniería en Desarrollo de Software.\n\n");
62                     } else if (puntaje >= MIN_LOGISTICA && puntaje <= MAX_LOGISTICA) {
63                         printf("Carrera asignada: Ingeniería en Logística y Transporte.\n\n");
64                     } else if (puntaje >= MIN_TELEMATICA && puntaje <= MAX_TELEMATICA) {
65                         printf("Carrera asignada: Ingeniería en Telemática.\n\n");
66                     } else if (puntaje >= MIN_MATEMATICAS && puntaje <= MAX_MATEMATICAS) {
67                         printf("Carrera asignada: Licenciatura en Enseñanza de las Matemáticas.\n\n");
68                     } else if (puntaje < PUNTAJE_MINIMO) {
69                         printf("El puntaje no es adecuado para cursar una carrera en la UnADM.\n\n");
70                     }
71                     break;
72
73                 case 2: // Salir del programa.
74                     break;
75
76                 default: // Opción no válida.
77                     printf("¡Error! Opción no válida.\n\n");
78             }
79         } while (opcion != 2); // Repetir hasta que se elija «Salir».
80
81         // Pesar antes de cerrar (Solo para Windows):
82         #ifndef _WIN32
83             system("pause");
84         #endif
85
86     return 0;
87 }

Compilador (2) Recursos Registro de Compilación Depuración Resultados Console
Line: 13 Col: 25 Sel: 0 Lines: 86 Length: 2985 Insertar Done parsing in 1.531 seconds

```

Al compilar el código hubo errores en los caracteres a pesar de agregar la configuración UTF-8:



```

F:\Copia\Orlando\UNADM\Asignaturas\Modulo 1\Fundamentos de programación\Unidad 2\DFPR_U2_FA_ORMP\Evaluador de puntajes para la UnADM.c [Executing] - Embarcadero Dev-C++ 6.3

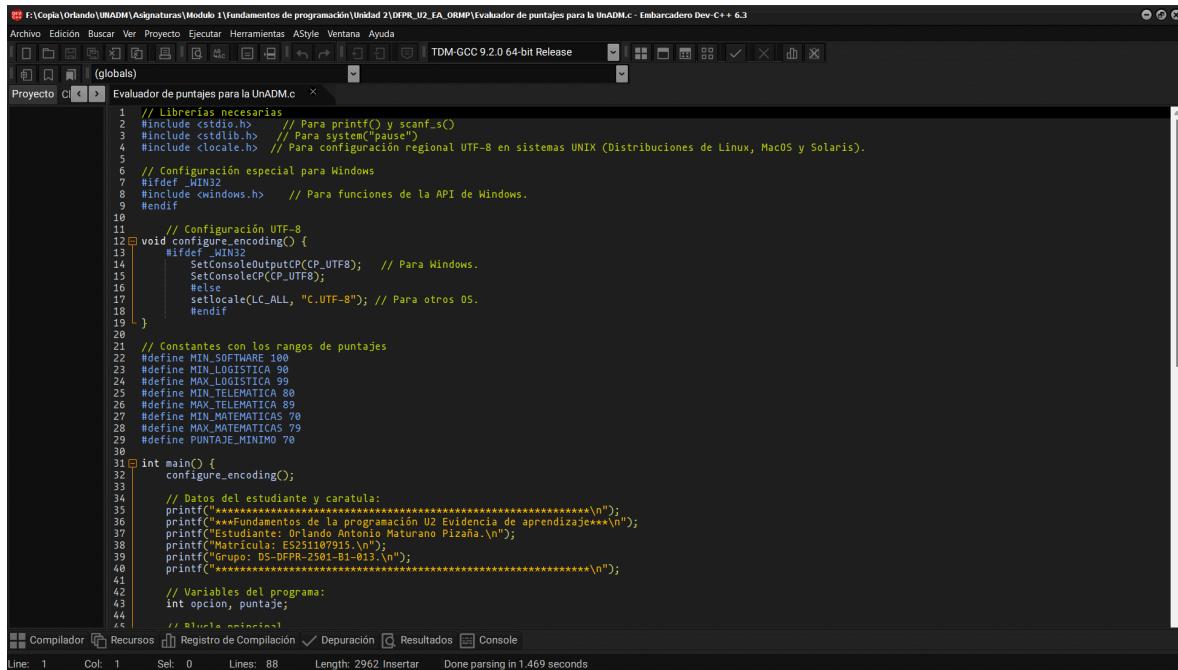
*****Fundamentos de la programaci|n U2 Evidencia de aprendizaje*****
Estudiante: Orlando Antonio Maturano Piza|a.
Matr|cula: ES251107915.
Grupo: DS-DFPR-2501-B1-013.
**********
Introduce la opci|n deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

d:   printf("*****Fundamentos de la programaci|n U2 Evidencia de aprendizaje*****\n");
38  printf("Estudiante: Orlando Antonio Maturano Piza|a.\n");
39  printf("Matr|cula: ES251107915.\n");
40  printf("Grupo: DS-DFPR-2501-B1-013.\n");
41  printf("**********\n");
42
43 // Blucle principal
44 do {
45     // Muestra el men| principal de opciones:
46
47     // Imprime el men| principal de opciones:
48
49 }

Compilador (2) Recursos | Registro de Compilaci|n Depuraci|n Resultados Console
Line: 1 Col: 24 Sel: 0 Lines: 86 Length: 2981 Insertar Done parsing in 0.016 seconds

```

Hice las debidas correcciones en el código, un error señalable fue que al colocar la librería `#include <locale.h>`, escribí la librería con una t en vez de una l (`#include <locate.h>`). Después de eso compilé el código otra vez y aún así seguían apareciendo mal los caracteres, así que coloqué el bloque de configuración UTF-8 arriba de los rangos de puntajes, coloqué `configure_encoding()`; antes de `printf`, cambié el orden de los rasgos de puntajes empezando por el puntaje menor, y agregué el comando UTF-8 genérico para la configuración para sistemas operativos que no sean Windows (“C.UTF-8”).



```

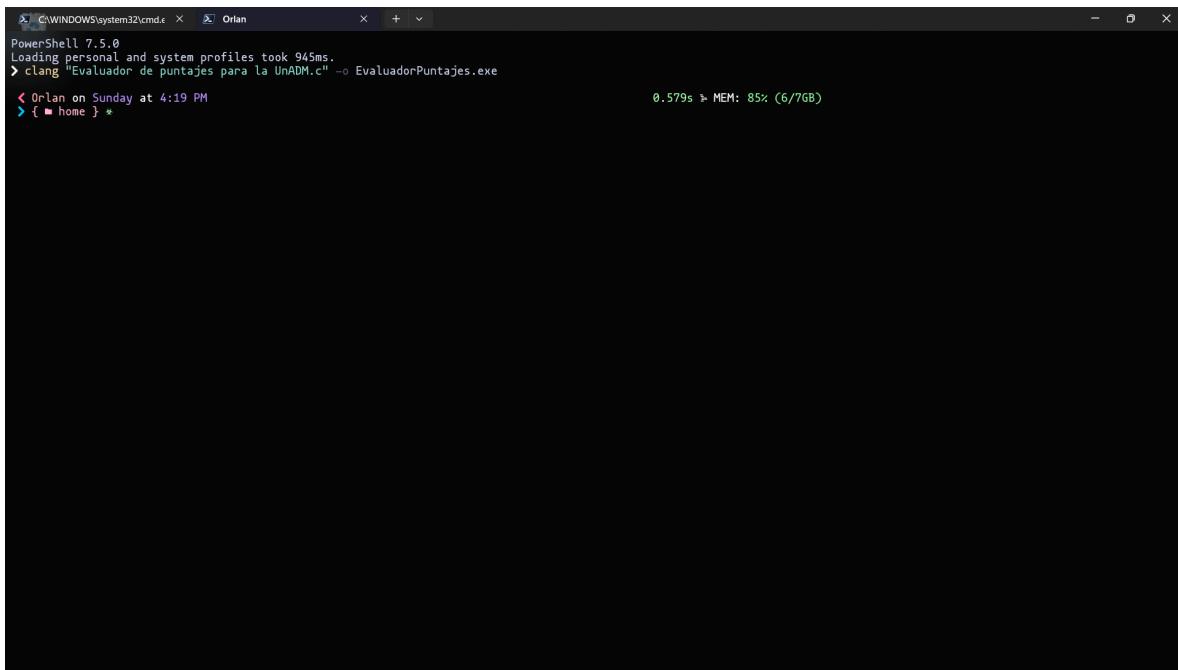
F:\Copia\Orlando\UNADM\Asignaturas\Modulo 1\Fundamentos de programación\Unidad 2\DFPR_U2_FA_ORMP\Evaluador de puntajes para la UnADM.c - Embarcadero Dev-C++ 6.3

// Librerias necesarias
1 #include <stdio.h> // Para printf() y scanf_s()
2 #include <stdlib.h> // Para system("pause")
3 #include <locale.h> // Para configuraci|n regional UTF-8 en sistemas UNIX (Distribuciones de Linux, MacOS y Solaris).
4
5 // Configuraci|n especial para Windows
6 #ifndef _WIN32
7 #include <windows.h> // Para funciones de la API de Windows.
8 #endif
9
10 // Configuraci|n UTF-8
11 void configure_encoding() {
12     #ifdef _WIN32
13         SetConsoleOutputCP(CP_UTF8); // Para Windows.
14         SetConsoleCP(CP_UTF8);
15     #else
16         setlocale(LC_ALL, "C.UTF-8"); // Para otros OS.
17     #endif
18 }
19
20 // Constantes con los rangos de puntajes
21 #define MIN_SOFTWARE 100
22 #define MIN_LOGISTICA 90
23 #define MAX_LOGISTICA 99
24 #define MIN_TELEMATICA 80
25 #define MAX_TELEMATICA 89
26 #define MIN_MATEMATICAS 70
27 #define MAX_MATEMATICAS 79
28 #define PUNTAJE_MINIMO 70
29
30 int main() {
31     configure_encoding();
32
33     // Datos del estudiante y matrula:
34     printf("*****Fundamentos de la programaci|n U2 Evidencia de aprendizaje*****\n");
35     printf("*****\n");
36     printf("Estudiante: Orlando Antonio Maturano Piza|a.\n");
37     printf("Matr|cula: ES251107915.\n");
38     printf("Grupo: DS-DFPR-2501-B1-013.\n");
39     printf("**********\n");
40
41     // Variables del programa:
42     int opcion, puntaje;
43
44     // El bucle principal
45
46 }

Compilador Recursos | Registro de Compilaci|n Depuraci|n Resultados Console
Line: 1 Col: 1 Sel: 0 Lines: 88 Length: 2962 Insertar Done parsing in 1.469 seconds

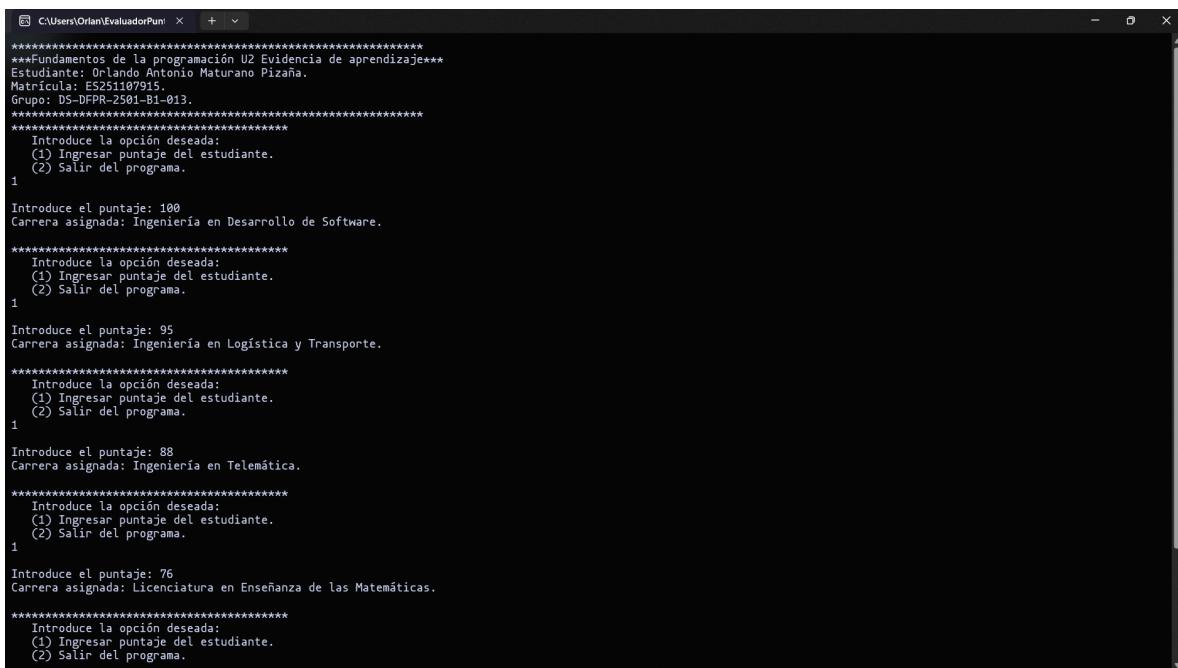
```

Aún así, seguían apareciendo de forma incorrecta los caracteres al compilar el código en Dev-C++, así que decidí compilar el código con clang, desde mi terminal, con mi comando programado, y se compiló sin errores:



```
C:\WINDOWS\system32\cmd.exe x Orlan + 0.579s 85% (6/7GB)
PowerShell 7.5.0
Loading personal and system profiles took 945ms.
> clang "Evaluador de puntajes para la UnADM.c" -o EvaluadorPuntajes.exe
< Orlan on Sunday at 4:19 PM
> { home } *
```

El programa funciona como debe y los caracteres se ven bien:



```
C:\Users\Orlan\EvaluadorPuntajes x + 0.579s 85% (6/7GB)
*****
***Fundamentos de la programación U2 Evidencia de aprendizaje***
Estudiante: Orlando Antonio Maturano Pizaña.
Matrícula: ES251107915.
Grupo: DS-DPR-2501-B1-013.
*****
Introduce la opción deseada:
(1) Ingresar_puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 100
Carrera asignada: Ingeniería en Desarrollo de Software.

*****
Introduce la opción deseada:
(1) Ingresar_puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 95
Carrera asignada: Ingeniería en Logística y Transporte.

*****
Introduce la opción deseada:
(1) Ingresar_puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 88
Carrera asignada: Ingeniería en Telemática.

*****
Introduce la opción deseada:
(1) Ingresar_puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 76
Carrera asignada: Licenciatura en Enseñanza de las Matemáticas.

*****
```

```

C:\Users\Orlan\EvaluadorPuntaje> + -
1

Introduce el puntaje: 100
Carrera asignada: Ingeniería en Desarrollo de Software.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 95
Carrera asignada: Ingeniería en Logística y Transporte.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 88
Carrera asignada: Ingeniería en Telemática.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 76
Carrera asignada: Licenciatura en Enseñanza de las Matemáticas.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

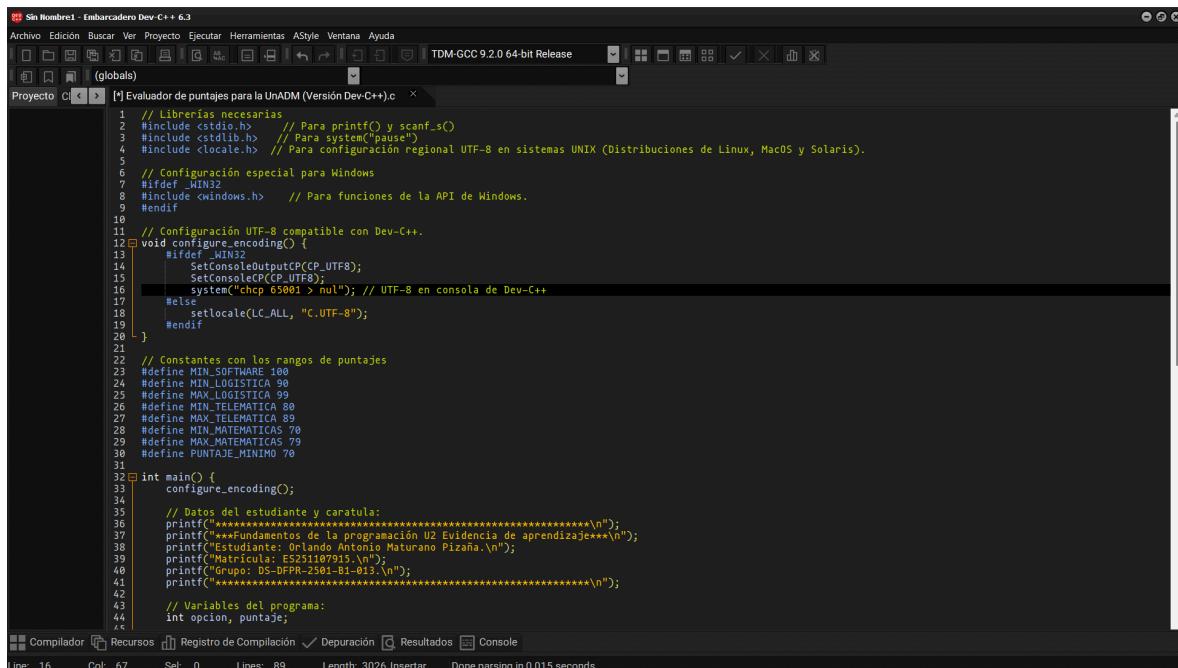
1

Introduce el puntaje: 66
El puntaje no es adecuado para cursar una carrera en la UnADM

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.
|
```

Una cosa a destacar es que al compilar el código en Dev-C++, el archivo .exe pesa 327 kilobytes, y al compilar el código con clang desde mi terminal, el archivo .exe pesa 170 kilobytes.

Investigando, encontré que el UTF-8 en Dev-C++ se maneja en comando de CMD e hice la corrección del bloque correspondiente:



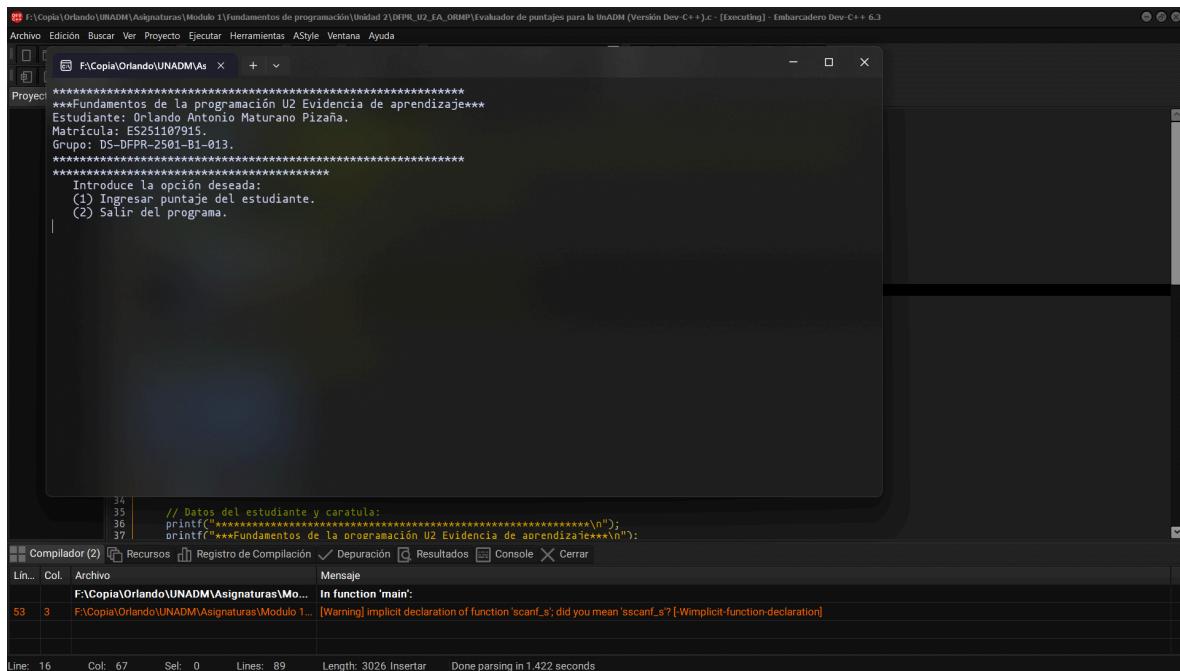
```

Sin Nombre - Embarcadero Dev-C++ 6.3
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
TDM-GCC 9.2.0 64-bit Release
(globals)
Proyecto C:\...\Evaluador de puntajes para la UnADM (Versión Dev-C++.c) □
1 // Librerías necesarias
2 #include <stdio.h> // Para printf() y scanf_s()
3 #include <stdlib.h> // Para system("pause")
4 #include <locale.h> // Para configuración regional UTF-8 en sistemas UNIX (Distribuciones de Linux, MacOS y Solaris).
5
6 // Configuración especial para Windows
7 #ifndef _WIN32
8 #include <windows.h> // Para funciones de la API de Windows.
9 #endif
10
11 // Configuración UTF-8 compatible con Dev-C++.
12 void configure_encoding() {
13     #ifdef _WIN32
14         setConsoleOutputCP(CP_UTF8);
15         SetConsoleCP(CP_UTF8);
16         system("chcp 65001 > null"); // UTF-8 en consola de Dev-C++
17     #else
18         setlocale(LC_ALL, ".UTF-8");
19     #endif
20 }
21
22 // Constantes con los rangos de puntajes
23 #define MIN_SOFTWARE 100
24 #define MIN_LOGISTICA 90
25 #define MAX_LOGISTICA 99
26 #define MIN_TELEMATICA 80
27 #define MAX_TELEMATICA 89
28 #define MIN_MATEMATICAS 70
29 #define MAX_MATEMATICAS 79
30 #define PUNTAJE_MINIMO 70
31
32 int main() {
33     configure_encoding();
34
35     // Datos del estudiante y caratula:
36     printf("*****\n");
37     printf("Estudiante: Orlando Maturano Pizáña.\n");
38     printf("Matrícula: ES2510187915\n");
39     printf("Grupo: DS-DFPR-2501-81-013.\n");
40     printf("*****\n");
41
42     // Variables del programa:
43     int opcion, puntaje;
44 }
```

Compilador Recursos Registro de Compilación ✓ Depuración Resultados Console

Line: 16 Col: 67 Sel: 0 Lines: 89 Length: 3026 Insertar Done parsing in 0.015 seconds

Al compilar el código con esta nueva corrección, el programa imprime bien todos los caracteres:



```

F:\Copia\Orlando\UNADM\Asignaturas\Modulo 1\Fundamentos de programación\Unidad 2\DFPR_U2_EA_ORIMP\evaluador de puntajes para la UnADM (Versión Dev-C++).c - [Executing] - Embarcadero Dev-C++ 6.3

Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda

Proyecto ****Fundamentos de la programación U2 Evidencia de aprendizaje****
Estudiante: Orlando Antonio Maturano Pizaña.
Matrícula: ES251107915.
Grupo: DS-DFPR-2501-B1-013.
***** Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

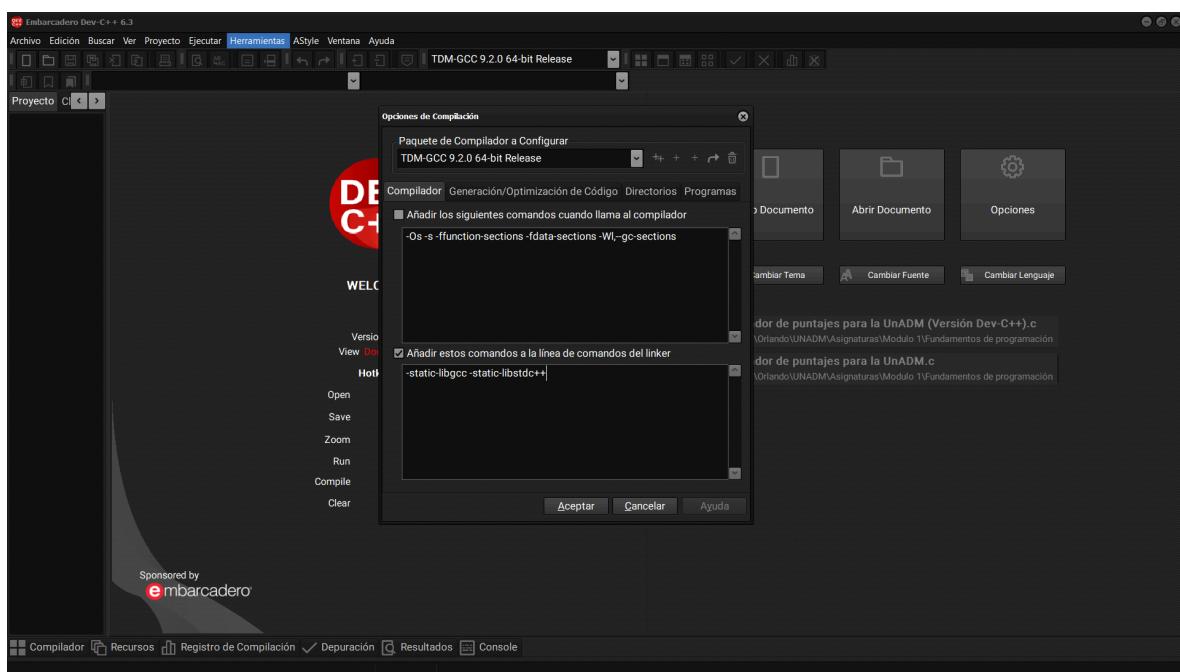
34 // Datos del estudiante y matrícula:
35 printf("*****\n");
36 orintf("*****Fundamentos de la programación U2 Evidencia de aprendizaje*****\n");
37

Compilador (2) Recursos Registro de Compilación Depuración Resultados Console Cerrar
Lín... Col... Archivo Mensaje
F:\Copia\Orlando\UNADM\Asignaturas\Modulo 1\evaluador de puntajes para la UnADM (Versión Dev-C++).c In function 'main'.
53 3 F:\Copia\Orlando\UNADM\Asignaturas\Modulo 1\evaluador de puntajes para la UnADM (Versión Dev-C++).c [Warning] implicit declaration of function 'scanf_s'; did you mean 'sscanf_s'? [-Wimplicit-function-declaration]

Line: 16 Col: 67 Sel: 0 Lines: 89 Length: 3026 Insertar Done parsing in 1.422 seconds

```

Configuré la compilación con Dev-C++, para que al compilar con GCC, el archivo .exe pese lo mínimo posible, utilicé los siguientes comandos:



Esta configuración es sumamente importante, ya que los usuarios preferimos instalar software más ligero para optimizar nuestro disco local, lo cual añade competitividad en la industria además de no utilizar espacio innecesario en el disco local.

Añadiendo esta configuración, el archivo .exe final pesa 18 kilobytes en comparación con los 170 KB de la compilación con clang. Después de este cambió tan significativo del peso del programa, es crucial hacer pruebas para comprobar que parte del código no se haya eliminado con esa comprensión tan grande. Al realizar las pruebas el programa sigue funcionando como debe:

```

P:\Copia\Orlando\UNADMAs <--> + -
***Fundamentos de la programación U2 Evidencia de aprendizaje***
Estudiante: Orlando Antonio Maturano Pizaña.
Matrícula: ES251107915.
Grupo: DS-DPR-2501-B1-013.
*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 100
Carrera asignada: Ingeniería en Desarrollo de Software.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 95
Carrera asignada: Ingeniería en Logística y Transporte.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 85
Carrera asignada: Ingeniería en Telemática.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.
1
Introduce el puntaje: 75
Carrera asignada: Licenciatura en Enseñanza de las Matemáticas.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

```

```

F:\Copia\Orlando\UNADMAs > + -
1

Introduce el puntaje: 100
Carrera asignada: Ingeniería en Desarrollo de Software.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 95
Carrera asignada: Ingeniería en Logística y Transporte.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 85
Carrera asignada: Ingeniería en Telemática.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 75
Carrera asignada: Licenciatura en Enseñanza de las Matemáticas.

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.

1

Introduce el puntaje: 65
El puntaje no es adecuado para cursar una carrera en la UnADM

*****
Introduce la opción deseada:
(1) Ingresar puntaje del estudiante.
(2) Salir del programa.
|
```

Nota:

Sería ideal configurar el código del programa para cada sistema operativo, para garantizar la experiencia del usuario en pequeños detalles, como por ejemplo: Al salir del programa, para que en Windows salga solo eligiendo la opción 2 sin pulsar ninguna tecla adicional.

Pero por cuestiones de tiempo se dejará un código portable.

CONCLUSIÓN

El desarrollo de este programa en C, orientado a evaluar puntajes para el ingreso a las carreras de la UnADM, permitió consolidar competencias esenciales en programación estructurada y gestión de entornos multilingües. A través de la implementación de estructuras de control selectivas y repetitivas, se logró un sistema interactivo que no solo cumple con los requisitos funcionales del caso de estudio, sino que también garantiza usabilidad mediante un menú intuitivo. Uno de los mayores desafíos fué asegurar la correcta visualización de caracteres especiales (como ñ y acentos) en distintas plataformas, lo que demandó el dominio de configuraciones regionales (setlocale) y ajustes específicos para Windows (SetConsoleOutputCP). Además, la optimización del tamaño del ejecutable reveló la importancia de ajustar opciones de compilación, equilibrando eficiencia y portabilidad.

Este proyecto reforzó la relevancia de adaptar soluciones técnicas a contextos educativos, donde la claridad y accesibilidad son prioritarias. Los aprendizajes abarcan desde el manejo de codificaciones hasta la depuración del código en entornos diversos, destacando la necesidad de enfoques multiplataforma en un mundo tecnológicamente heterogéneo. En un ámbito académico, herramientas como esta no solo agilizan procesos administrativos, sino que también sirven como modelo pedagógico para entender la

programación aplicada. Así, el trabajo evidencia cómo la integración de fundamentos de programación con buenas prácticas técnicas puede resolver problemas reales, mientras se fomentan habilidades críticas para el desarrollo de software robusto y escalable.

CITAS DE AUTOR

"UTF-8 permite la representación de todos los caracteres Unicode, siendo compatible con sistemas heredados" (Unicode Consortium, 2023).

"El lenguaje C proporciona construcciones de control de flujo que permiten decisiones y repeticiones lógicas" (Kernighan & Ritchie, 1988, p. 55).

"SetConsoleOutputCP establece la codificación de la consola para soportar caracteres Unicode" (Microsoft, 2023).

"La opción -Os optimiza para reducir el tamaño del código, eliminando redundancias" (GCC Team, 2023).

REFERENCIAS BIBLIOGRÁFICAS

Embarcadero. (2024). Product documentation. C++ Builder. De:
https://docwiki.embarcadero.com/RADStudio/Athens/en/Main_Page

GCC Documentation. (2025). 3.12 Options That Control Optimization. De:
<https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html>

Y: <https://gcc.gnu.org/onlinedocs/gcc.pdf>

ISO. (2024). Information technology — Programming languages — C, De:
<https://www.open-std.org/jtc1/sc22/wg14/www/docs/n3220.pdf>

Javier Ceballos Sierra. (2019). C(C++) Curso de programación 5 ed. De:
<https://es.scribd.com/doc/9838927/Ceballos-C-C-Curso-de-programacion-5Ed>

Joyanes, L. (2005). Programación en C. Algoritmos, estructuras de datos y objetos. McGraw-Hill, De:
<https://intprog.wordpress.com/wp-content/uploads/2013/08/programacion-en-c-metodologia-algoritmos-y-estructura-de-datos-editorial-mcgraw-hill1.pdf>

Learn Microsoft. (2023). Función SetConsoleOutputCP. De:
<https://learn.microsoft.com/es-es/windows/console/setconsoleoutputcp>

Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). De:

Inglés:

https://www.cimat.mx/ciencia_para_jovenes/bachillerato/libros/%5BKernighan-Ritchie%5DThe_C_Programming_Language.pdf

Español:

https://frrq.cvg.utn.edu.ar/pluginfile.php/13741/mod_resource/content/0/El-lenguaje-de-programacion-C-2-ed-kernighan-amp-ritchie.pdf

UnADM. (s.f.) Unidad 2. Introducción al lenguaje C. De:

https://dmd.unadmexico.mx/contenidos/DCEIT/Compartidas/FPR/U2/descargables/FPR_U2_Contenido.pdf

Unicode Consortium. (2022). Unicode 15.0.0. 2 General structure. 2.5 Encoding Forms.

De: <https://www.unicode.org/versions/Unicode15.0.0/>