

Nombre: Miguel Alejandro Santiago Pérez

No. de Matrícula.: zap362

Materia: Fundamentos de Programación _____ Grupo: Dev 1

Turno: Matutino

Carrera: Desarrollo de Software Interactivo y Videojuegos

Tema: Practica 12

No: R.12

Fecha propuesta: 02/06/2021

Fecha de Entrega: 08/06/2021

Escuela: Instituto de estudios universitarios Amerike

Plantel Zapopan

Calle: Montemorelos

No: 3503 Colonia: Rinconada de la Calma

C.P.: 45080

Teléfono: 3336326100

Ciudad: Zapopan



AMERIKE

INSTITUTO UNIVERSITARIO

Firma del alumno (a)

Firma de revisión fecha

Qué se evalúa:	10 pts.	7 pts.	4pts.	Pts.
Entrega electrónica	Es en tiempo y forma al iniciar la clase. (1 pts.)	Después de 30 minutos de iniciada la clase. (.7 pts.)	Al minuto 40. (Posteriormente ya no se reciben) (.4pts.)	
Del formato.	Cumple con todos los elementos solicitados. (1 pts.)	No cumple con dos elementos solicitados. (.7 pts.)	No cumple con tres o más elementos solicitados. (.4pts.)	
La ortografía.	Tiene dos errores ortográficos. (1 pts.)	Tiene de tres a cuatro errores ortográficos. (.7 pts.)	Tiene cinco o más errores ortográficos. (.4pts.)	
Del tema y objetivo.	La teoría y ejemplos corresponden al tema tratado. (1 pts.)	La teoría o ejemplos no corresponden al tema tratado. (.7 pts.)	La teoría y ejemplos no corresponden al tema tratado. (.4pts.)	
El programa y los cálculos.	Los parámetros y componentes corresponden al 100% de lo planeado. (1 pts.)	El programa arroja un error o componente no corresponden al 100% de lo planeado. (7 pts.)	El programa arroja dos errores o componentes no corresponden al 100% de lo calculado. (.4pts.)	
Diagramas.	Los diagramas a bloques, de flujo y esquemáticos son acorde al de la práctica y siguen una secuencia lógica. (1 pts.)	Los diagramas a bloques, o de flujo o esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.7 pts.)	Los diagramas a bloques, de flujo y esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.4pts.)	
La tabla de valores.	Los valores calculados y medidos presentan una desviación máxima del 10%. (1 pts.)	Los valores calculados y medidos presentan una desviación máxima del 15%. (.7 pts.)	Los valores calculados y medidos presentan una desviación máxima del 20%. (.4pts.)	
Las observaciones y conclusiones.	Son específicas y congruentes con la práctica. (1 pts.)	Las observaciones o conclusiones son específicas y congruentes con la práctica. (.7 pts.)	Las observaciones y las conclusiones no son específicas y congruentes con la práctica. (.4pts.)	
Bibliografía.	Es acorde al (los) tema (s) tratado (s) y está completa (1 pts.)	Es acorde a algún (os) tema (s) tratado (s), le falta algún elemento que la conforman (.7 pts.)	No es acorde al (los) tema (s) tratado (s), le faltan 2 elementos que la conforma (.4pts.)	
Fuentes de consulta.	Es acorde al (los) tema (s) tratado (s) (1 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.7 pts.)	Es acorde a algún (los) tema (s) tratado (s) (.4pts.)	

Índice

2... Teoría

2, 3, 4... Diagramas e imágenes del programa

5...Observaciones, Conclusiones y Bibliografía

Teoría El código ejecutara una aplicación de escritorio de Windows

Diagramas

```
1  #include <windows.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <tchar.h>
5
6  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
7  {
8      PAINTSTRUCT ps;
9      HDC hdc;
10     TCHAR greeting[] = _T("Hello, Windows desktop!");
11
12     switch (message)
13     {
14     case WM_PAINT:
15         hdc = BeginPaint(hwnd, &ps);
16
17         // Here your application is laid out.
18         // For this introduction, we just print out "Hello, Windows desktop!"
19         // in the top left corner.
20         TextOut(hdc,
21             5, 5,
22             greeting, _tcslen(greeting));
23         // End application specific layout section.
24
25         EndPaint(hwnd, &ps);
26         break;
27     case WM_DESTROY:
28         PostQuitMessage(0);
29         break;
30     default:
31         return DefWindowProc(hwnd, message, wParam, lParam);
32         break;
33     }
34
35     return 0;
36 }
37
38 int CALLBACK WinMain(
39     _In_ HINSTANCE hInstance,
40     _In_opt_ HINSTANCE hPrevInstance,
41     _In_ LPSTR lpCmdLine,
42     _In_ int nCmdShow)
43 {
44     static TCHAR szWindowClass[] = _T("DesktopApp");
45     static TCHAR szTitle[] = _T("Windows Desktop Guided Tour Application");
46
47     WNDCLASSEX wcex;
```

```
WNDCLASSEX wcex;

wcex.cbSize = sizeof(WNDCLASSEX);
wcex.style = CS_HREDRAW | CS_VREDRAW;
wcex.lpfnWndProc = WndProc;
wcex.cbClsExtra = 0;
wcex.cbWndExtra = 0;
wcex.hInstance = hInstance;
wcex.hIcon = LoadIcon(hInstance, IDI_APPLICATION);
wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
wcex.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
wcex.lpszMenuName = NULL;
wcex.lpszClassName = szWindowClass;
wcex.hIconSm = LoadIcon(wcex.hInstance, IDI_APPLICATION);

if (!RegisterClassEx(&wcex))
{
    MessageBox(NULL,
        _T("Call to RegisterClassEx failed!"),
        _T("Windows Desktop Guided Tour"),
        NULL);

    return 1;
}

// The parameters to CreateWindow explained:
// szWindowClass: the name of the application
// szTitle: the text that appears in the title bar
// WS_OVERLAPPEDWINDOW: the type of window to create
// CW_USEDEFAULT, CW_USEDEFAULT: initial position (x, y)
// 500, 100: initial size (width, length)
// NULL: the parent of this window
// NULL: this application does not have a menu bar
// hInstance: the first parameter from WinMain
// NULL: not used in this application
HWND hwnd = CreateWindow(
    szWindowClass,
    szTitle,
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, CW_USEDEFAULT,
    500, 100,
    NULL,
    NULL,
    hInstance,
    NULL
```

```
if (!hWnd)
{
    MessageBox(NULL,
        _T("Call to CreateWindow failed!"),
        _T("Windows Desktop Guided Tour"),
        NULL);

    return 1;
}

// The parameters to ShowWindow explained:
// hWnd: the value returned from CreateWindow
// nCmdShow: the fourth parameter from WinMain
ShowWindow(hWnd,
    nCmdShow);
UpdateWindow(hWnd);

MSG msg;
while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}

return (int)msg.wParam;
};
```

Observaciones Visual ejecutaba el programa con lo escrito dentro de el en una ventana nueva de windows

Conclusiones Es útil para probar nuevas apps en ventana

Bibliografía [Tutorial: crear una aplicación de escritorio tradicional de Windows \(C++\) | Microsoft Docs](#)