

Nombre: Ian Axel Hernández Ortega **No. de Matrícula.:** Zap288

Materia: Fundamentos de Programación **Grupo:** 21-1 **Turno:** Matutino

Carrera: Licenciatura en Desarrollo de Software Interactivo y Videojuegos

Tema: Practica 11, Crear una aplicación de escritorio tradicional de Windows **No:** P.11

Fecha propuesta: 17 – 12 – 2020 **Fecha de Entrega:** 05 – 01 – 2021

Escuela: Instituto Universitario Amerike

Plantel: Guadalajara

Calle: Calle Montemorelos **No:** 3503 **Colonia:** Rinconada de la Calma **C.P.:** 45080

Teléfono: 33 3632 6100

Ciudad: Zapopan

Logotipo personal



Logotipo (de la escuela)



IanAxelHernándezOrtega

Firma del alumno (a)

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.)	

Nombre: Ian Axel Hernández Ortega

Práctica: Practical11, Crear una aplicación de escritorio tradicional de Windows

No. P-11

Página 1

Índice	pag 2
Teoría	pag 2
Cálculos	pag 2
Diagramas	pag 2 y 3
Tabla Comparativa	pag 4
Observaciones	pag 4
Conclusiones	pag 4
Bibliografía	pag 4

Teoría

Crear una aplicación de escritorio para Windows, la aplicación de ejemplo que nos brinda el programa de visual estudio es mostrar el texto “Hola Escritorio de Windows”, la cual nos permite el ingresar cualquier tipo de código para así poder crear después una aplicación la cual sea para diferentes usos.

Cálculos

```
int CALLBACK WinMain(
    _In_ HINSTANCE hInstance,
    _In_opt_ HINSTANCE hPrevInstance,
    _In_ LPSTR lpCmdLine,
    _In_ int nCmdShow
);
```

Diagramas

```
3  #include <windows.h>
4  #include <stdlib.h>
5  #include <string.h>
6  #include <tchar.h>
7
8
9  static TCHAR szWindowClass[] = _T("DesktopApp");
10
11
12 static TCHAR szTitle[] = _T("Windows Desktop Guided Tour Application");
13
14 HINSTANCE hInst;
15
16
17 LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
18
19 int CALLBACK WinMain(
20     _In_ HINSTANCE hInstance,
21     _In_opt_ HINSTANCE hPrevInstance,
22     _In_ LPSTR lpCmdLine,
23     _In_ int nCmdShow
24 )
25 {
26     WNDCLASSEX wcex;
27
28     wcex.cbSize = sizeof(WNDCLASSEX);
29     wcex.style = CS_HREDRAW | CS_VREDRAW;
30     wcex.lpfnWndProc = WndProc;
31     wcex.cbClsExtra = 0;
32     wcex.cbWndExtra = 0;
33     wcex.hInstance = hInstance;
34     wcex.hIcon = LoadIcon(hInstance, IDI_APPLICATION);
35     wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
36     wcex.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
```

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

    return 1;
}

hInst = hInstance;

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;
}

//
78     ShowWindow(hWnd,
79         nCmdShow);
80     UpdateWindow(hWnd);
81
82
83     MSG msg;
84     while (GetMessage(&msg, NULL, 0, 0))
85     {
86         TranslateMessage(&msg);
87         DispatchMessage(&msg);
88     }
89
90     return (int)msg.wParam;
91 }
92
93
94 LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
95 {
96     PAINTSTRUCT ps;
97     HDC hdc;
98     TCHAR greeting[] = _T("Hello, Windows desktop!");
99
100     switch (message)
101     {
102     case WM_PAINT:
103         hdc = BeginPaint(hWnd, &ps);
104
105         TextOut(hdc,
106             5, 5,
107             greeting, _tcslen(greeting));
108
109
110
111         EndPaint(hWnd, &ps);
    }
```

```

110
111     EndPaint(hWnd, &ps);
112     break;
113 case WM_DESTROY:
114     PostQuitMessage(0);
115     break;
116 default:
117     return DefWindowProc(hWnd, message, wParam, lParam);
118     break;
119 }
120
121 return 0;
122 }

```

Tabla comparativa

Datos ingresados	Datos Esperados	Datos Obtenidos
Hello Windows	Hello Windows	Hello Windows
Hola Windows	Hola Windows	Hola Windows

Observaciones

Al ingresar los datos de “Hola Windows” se dan como resultado los datos que fueron ingresados anteriormente, las diferentes formas que se logran con esta aplicación es la recompensa de obtener los datos ingresados.

Conclusiones

Como ya lo mencione antes esta app nos permite ingresar los datos que sean necesarios para así poder completar el programa, y para que al momento de ejecutar el código nos dé como resultado como nosotros completamos la app por las diferentes secciones que complementa el código.

Bibliografía

Crear una aplicación de escritorio tradicional de windows. (2020, 28 mayo). Microsoft Docs.

<https://docs.microsoft.com/es-es/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

