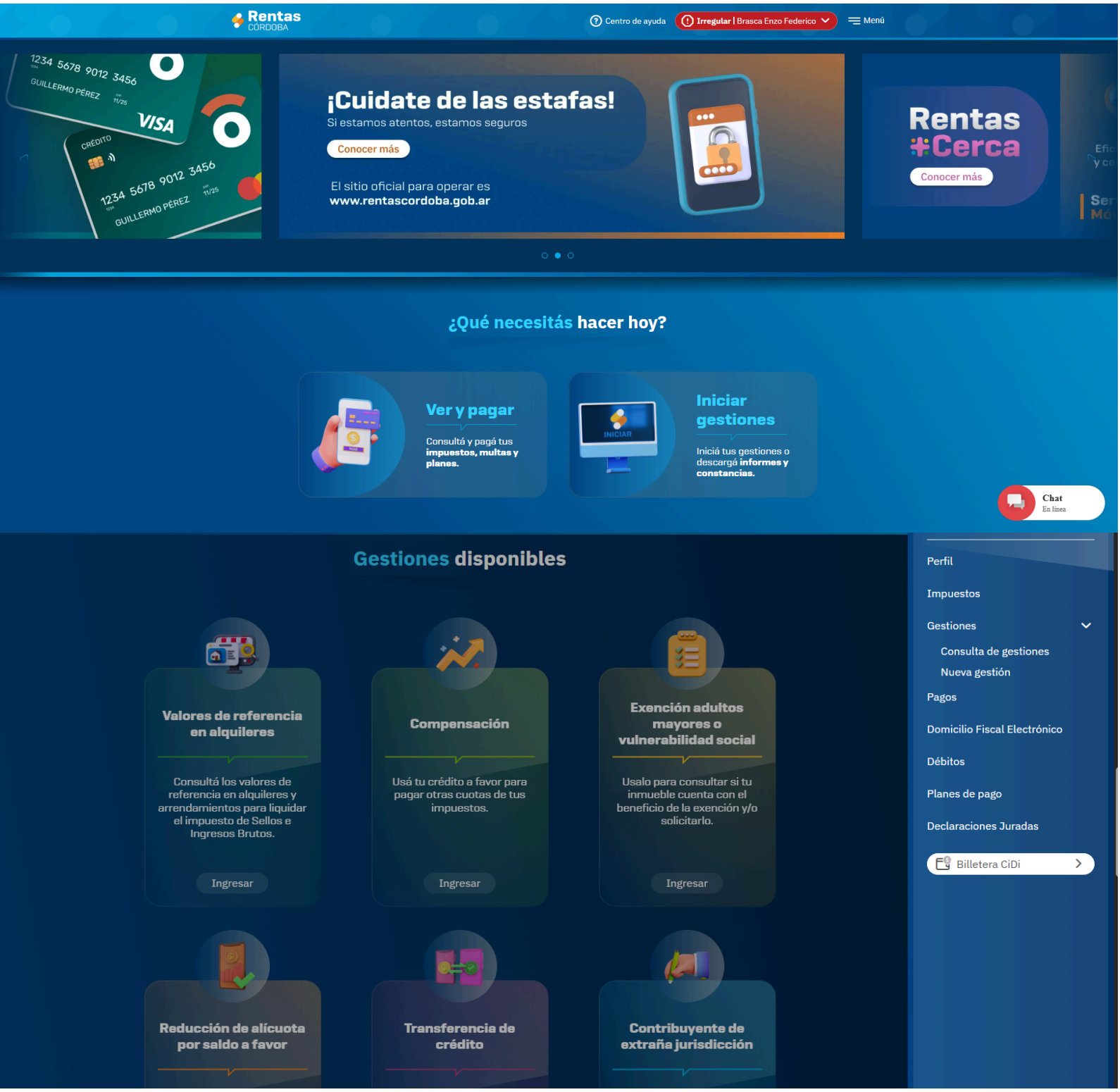

En la página de Rentas Córdoba hay un problema en cómo se muestran las funcionalidades de la página, principalmente en el header. La página actualmente tiene un menú en el que se muestran todas las funciones posibles de forma comprimida, cuando seria mas practico que se presenten de forma individual en el espacio no utilizado del header.
Además existen algunas funcionalidades que se presentan de forma redundante, como pueden ser los impuestos que aparecen en el menú o en el gestor de usuarios. Para esto se eliminaría el botón dentro del menú de usuario y se dejaría solo un botón en el navbar

Otro problema que presenta la página tiene que ver con las gestiones que ofrece, mientras que en el menú solo se muestran 2 opciones (consultar y crear una nueva), en la página principal se muestran todas las gestiones disponibles de forma más ordenada, para esto se podría adaptar el botón dentro del menú para que te muestre todas las opciones disponibles.
Finalmente, el problema principal de la pagina es que cada vez que se quiere acceder a una nueva ventana, se genera una ventana totalmente desde cero. Esto hace que se ralentice el uso de la pagina al tener que cargar nuevos elementos todo el tiempo.

Solución Propuesta e Implementación en el Prototipo

La solución que se propone en el prototipo para estos problemas es una barra de navegación donde se disponen los botones, que se encuentran dentro del menú de la página, de forma individual. Esto facilita la visualización de las funcionalidades y facilita su acceso. Además al problema del apartado de gestiones se lo adaptó para que al apretar el botón se despliegan hacia abajo una serie de botones para poder acceder a los diferentes tipos de gestiones. Finalmente se eliminaron los botones redundantes, haciendo que la barra de navegación se muestre de forma generalizada en todas las ventanas, lo que permitió además la reutilización de código para aumentar la eficiencia.
Con respecto al problema de las ventanas, el prototipo propone el uso de dos ventanas, una ventana principal donde no se muestra información y solo se muestran los botones. Y una ventana que se modifica dinámicamente dependiendo del botón que se presione, haciendo que los elementos de la ventana cambien pero sin generar una nueva ventana.

Capturas del código principal

Rentas (Global Scope)

```cpp
#include <windows.h>
#include <vector>

// ----- IDs para los botones de la VENTANA PRINCIPAL -----
// Cada botón representa una "vista" que queremos mostrar.
#define ID_BUTTON_SHOW_VIEW1 101
#define ID_BUTTON_SHOW_VIEW2 102
#define ID_BUTTON_SHOW_VIEW3 103
#define ID_BUTTON_SHOW_VIEW4 104
#define ID_BUTTON_SHOW_VIEW5 105
#define ID_BUTTON_SHOW_VIEW6 106
#define ID_BUTTON_SHOW_VIEW7 107


// Id para el slide down de la ventana principal
#define ID_BUTTON_SHOW_VIEW8 108
#define ID_MENU_OPCION1 201
#define ID_MENU_OPCION2 202
#define ID_MENU_OPCION3 203
#define ID_MENU_OPCION4 204
#define ID_MENU_SEPARADOR 205

// ID para el botón estático de regreso en la ventana secundaria
#define ID_BUTTON_RETURN_TO_MAIN 206

// WM_APP es el inicio del rango para mensajes definidos por la aplicación.
#define WM_APP_CHANGE_VIEW (WM_APP + 1)

// ----- Prototipos de los WndProc -----
LRESULT CALLBACK MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam);
LRESULT CALLBACK SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam);

// ----- Variables Globales -----
HWND g_hwndMain = NULL;
HWND g_hwndSecondary = NULL;
```

Rentas (Global Scope)

```cpp
// Un vector para guardar los handles de los controles que creamos dinámicamente.
// Esto nos permitirá destruirlos fácilmente después.
std::vector<HWND> g_vecChildControls;


// ----- Punto de Entrada (WinMain)-----
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {
    // El registro de clases y la creación de las ventanas principales es igual que antes.
    WNDCLASSEX wcMain = { 0 };
    wcMain.cbSize = sizeof(WNDCLASSEX);
    wcMain.lpfnWndProc = MainWndProc;
    wcMain.hInstance = hInstance;
    wcMain.hCursor = LoadCursor(NULL, IDC_ARROW);
    wcMain.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
    wcMain.lpszClassName = L"MainWindowClass";
    RegisterClassEx(&wcMain);

    WNDCLASSEX wcSecondary = { 0 };
    wcSecondary.cbSize = sizeof(WNDCLASSEX);
    wcSecondary.lpfnWndProc = SecondaryWndProc;
    wcSecondary.hInstance = hInstance;
    wcSecondary.hCursor = LoadCursor(NULL, IDC_ARROW);
    wcSecondary.hbrBackground = (HBRUSH)(COLOR_WINDOW + 2);
    wcSecondary.lpszClassName = L"SecondaryWindowClass";
    RegisterClassEx(&wcSecondary);

    g_hwndMain = CreateWindowEx(WS_EX_CLIENTEDGE, L"MainWindowClass", L"Menú Principal", WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT, 1200, 500, NULL, NULL, hI
    g_hwndSecondary = CreateWindowEx(WS_EX_CLIENTEDGE, L"SecondaryWindowClass", L"Ventana de Contenido", WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT, 1200, 450

    ShowWindow(g_hwndMain, nCmdShow);
    UpdateWindow(g_hwndMain);

    MSG msg;
    while (GetMessage(&msg, NULL, 0, 0) > 0) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return (int)msg.wParam;
}

static HWND g_hDropdownButton = NULL;

//Funcion para generar los botones en ambas ventanas evitando la redundancia de codigo

void generarBotones(HWND hwnd) {
    HDC hdc = GetDC(hwnd);
    if (!hdc) {
        return;
    }

    // Definimos las constantes para la posicion y tamaño de los botones
    const int Button_Y_Pos = 20;
    const int Button_Height = 30;
    const int Button_Width = 180;
    const int Initial_X_Pos = 20;
    const int Horizontal_Spacing = 20;
    const int Separador = 40;

    // Calculamos el ancho de cada boton segun el texto que contiene
    const wchar_t* btn1Txt = L"Impuestos";
    SIZE sizeBtn1;

    GetTextExtentPoint32W(hdc, btn1Txt, wcslen(btn1Txt), &sizeBtn1);

    int btn1Width = sizeBtn1.cx + Horizontal_Spacing;

    const wchar_t* btn2Txt = L"Pagos";
    SIZE sizeBtn2;
```

```
Rentas                                                    (Global Scope)
106         GetTextExtentPoint32W(hdc, btn2Txt, wcslen(btn2Txt), &sizeBtn2);
107
108         int btn2Width = sizeBtn2.cx + Horizontal_Spacing;
109
110         const wchar_t* btn3Txt = L"Domicilio Fiscal Electronico";
111         SIZE sizeBtn3;
112
113         GetTextExtentPoint32W(hdc, btn3Txt, wcslen(btn3Txt), &sizeBtn3);
114
115         int btn3Width = sizeBtn3.cx + Separador;
116
117         const wchar_t* btn4Txt = L"Debito";
118         SIZE sizeBtn4;
119
120         GetTextExtentPoint32W(hdc, btn4Txt, wcslen(btn4Txt), &sizeBtn4);
121
122         int btn4Width = sizeBtn4.cx + Separador;
123
124         const wchar_t* btn5Txt = L"Planes de Pagos";
125         SIZE sizeBtn5;
126
127         GetTextExtentPoint32W(hdc, btn5Txt, wcslen(btn5Txt), &sizeBtn5);
128
129         int btn5Width = sizeBtn5.cx + Separador;
130
131         const wchar_t* btn6Txt = L"Declaraciones Juradas";
132         SIZE sizeBtn6;
133
134         GetTextExtentPoint32W(hdc, btn6Txt, wcslen(btn6Txt), &sizeBtn6);
135
136         int btn6Width = sizeBtn6.cx + Separador;
137
138         const wchar_t* btn7Txt = L"Usuario";
139         SIZE sizeBtn7;
140
```

```
141         GetTextExtentPoint32W(hdc, btn7Txt, wcslen(btn7Txt), &sizeBtn7);
142
143         int btn7Width = sizeBtn7.cx + Separador;
144
145         const wchar_t* btn8Txt = L"Gestiones \u25BC";
146         SIZE sizeBtn8;
147
148         GetTextExtentPoint32W(hdc, btn8Txt, wcslen(btn8Txt), &sizeBtn8);
149
150         int btn8Width = sizeBtn8.cx + Separador;
151
152         // Creamos el boton desplegable y guardamos su handle en una variable global para usarlo luego
153         g_hDropdownButton = CreateWindow(
154             L"BUTTON",
155             btn8Txt,
156             WS_TABSTOP | WS_VISIBLE | WS_CHILD,
157             (Initial_X_Pos + btn1Width + btn2Width + btn3Width + btn4Width + btn5Width + btn6Width + btn7Width + 140), Button_Y_Pos, btn8Width, Button_Height,
158             hwnd, (HMENU)ID_BUTTON_SHOW_VIEW8, NULL, NULL);
159
160         // Creamos los 7 Botones principales
161         CreateWindow(L"BUTTON", btn1Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, Initial_X_Pos, Button_Y_Pos, btn1Width, Button_Height, hwnd, (HMENU)ID_BUTTON_SHOW_VIEW1, NU
162         CreateWindow(L"BUTTON", btn2Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, (Initial_X_Pos + btn1Width + Horizontal_Spacing), Button_Y_Pos, btn2Width, Button_Height, hw
163         CreateWindow(L"BUTTON", btn3Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, (Initial_X_Pos + btn1Width + btn2Width + Separador), Button_Y_Pos, btn3Width, Button_Height,
164         CreateWindow(L"BUTTON", btn4Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, (Initial_X_Pos + btn1Width + btn2Width + btn3Width + 60), Button_Y_Pos, btn4Width, Button_He
165         CreateWindow(L"BUTTON", btn5Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, (Initial_X_Pos + btn1Width + btn2Width + btn3Width + btn4Width + 80), Button_Y_Pos, btn5Widt
166         CreateWindow(L"BUTTON", btn6Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, (Initial_X_Pos + btn1Width + btn2Width + btn3Width + btn4Width + btn5Width + 100), Button_Y_
167         CreateWindow(L"BUTTON", btn7Txt, WS_TABSTOP | WS_VISIBLE | WS_CHILD, (Initial_X_Pos + btn1Width + btn2Width + btn3Width + btn4Width + btn5Width + btn6Width + 120
168
169     }
170
171     // ----- WndProc de la Ventana Principal -----
172     LRESULT CALLBACK MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) {
173         switch (msg) {
```

```
Rentas                                                    (Global Scope)                    MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)
171     // ----- WndProc de la Ventana Principal -----
172     LRESULT CALLBACK MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) {
173         switch (msg) {
174         case WM_CREATE:
175         {
176             // Creamos los botones y el label en la ventana principal
177             HDC hdc = GetDC(hwnd);
178             if (!hdc) {
179                 return -1;
180             }
181             const wchar_t* labelTxt = L"Bienvenido a la pagina de rentas Cordoba \n Porfavor seleccione alguna de las opciones disponibles para acceder a las funcionalid
182             SIZE sizeLabel;
183
184             GetTextExtentPoint32W(hdc, labelTxt, wcslen(labelTxt), &sizeLabel);
185
186             int labelWidth = sizeLabel.cx;
187             CreateWindow(L"STATIC", labelTxt, WS_VISIBLE | WS_CHILD, 50, 100, labelWidth, 50, hwnd, NULL, NULL, NULL);
188             generarBotones(hwnd);
189             break;
190         }
191
192         case WM_COMMAND:
193         {
194             // Manejo de los comandos de los botones
195             int wmId = LOWORD(wParam);
196             HMENU hMenu = CreatePopupMenu();
197             RECT rect;
198             //Verificacion de que boton se presiono
199             switch (wmId) {
200             case ID_BUTTON_SHOW_VIEW1:
201                 ShowWindow(g_hwndMain, SW_HIDE);
202
203                 SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);
204
205                 ShowWindow(g_hwndSecondary, SW_SHOW);
```

Rentas (Global Scope) MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)

```
            break;
        case ID_BUTTON_SHOW_VIEW2:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW3:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW4:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW5:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW6:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
```

Rentas (Global Scope) MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)

```
            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW7:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;

        case ID_BUTTON_SHOW_VIEW8:
            // Crear el menú desplegable
            AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION1, L"Valores de Ref...");
            AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION2, L"Compensacion");
            AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION3, L"Exencion...");
            AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION4, L"Transferencia");

            AppendMenuW(hMenu, MF_SEPARATOR, ID_MENU_SEPARADOR, NULL);

            // Obtener la posición del botón desplegable para mostrar el menú justo debajo de él
            GetWindowRect(g_hDropdownButton, &rect);

            TrackPopupMenuEx(hMenu, TPM_LEFTALIGN | TPM_TOPALIGN, rect.left, rect.bottom, hwnd, NULL);

            DestroyMenu(hMenu);
            break;
        case ID_MENU_OPCION1:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_MENU_OPCION2:
            ShowWindow(g_hwndMain, SW_HIDE);
```

Rentas (Global Scope) MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)

```
            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_MENU_OPCION3:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_MENU_OPCION4:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        } break;
    }
    case WM_CLOSE:
        DestroyWindow(hwnd);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    default:
        return DefWindowProc(hwnd, msg, wParam, lParam);
    }
    return 0;
}
```

Rentas | (Global Scope) | MainWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)

```cpp
// ----- Funciones para Crear y Destruir Vistas Dinámicamente -----
void DestroyChildControls() {
    for (HWND hControl : g_vecChildControls) {
        DestroyWindow(hControl);
    }
    g_vecChildControls.clear();
}
// Cada una de estas funciones crea una "vista" diferente en la ventana secundaria.
void CreateView1(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Aqui usted puede ver y pagar sus impuestos", WS_VISIBLE | WS_CHILD, 50, 100, 400, 20, hWndParent, NULL, NULL, NULL);

    g_vecChildControls.push_back(hLabel);
}

void CreateView2(HWND hWndParent) {
    HWND hLabel = CreateWindow(L"STATIC", L"Aqui usted puede ver su historial de pagos", WS_VISIBLE | WS_CHILD, 50, 100, 400, 20, hWndParent, NULL, NULL, NULL);

    g_vecChildControls.push_back(hLabel);
}
void CreateView3(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Aqui usted puede administrar y ver las notificaciones de su domicilio fiscal", WS_VISIBLE | WS_CHILD, 50, 100, 400, 30, h

    g_vecChildControls.push_back(hLabel);
}

void CreateView4(HWND hWndParent) {
    HWND hLabel = CreateWindow(L"STATIC", L"Aqui usted puede gestionar sus planes de debito automatico", WS_VISIBLE | WS_CHILD, 50, 100, 400, 20, hWndParent, NULL, N
        ;
    g_vecChildControls.push_back(hLabel);
}

void CreateView5(HWND hWndParent) {
    HWND hLabel = CreateWindow(L"STATIC", L"Aqui usted puede ver los planes de pago que tiene a su nombre", WS_VISIBLE | WS_CHILD, 50, 100, 400, 30, hWndParent, NULL

    g_vecChildControls.push_back(hLabel);
}

void CreateView6(HWND hWndParent) {
    HWND hLabel = CreateWindow(L"STATIC", L"Aqui usted puede consultar sus declaraciones juradas", WS_VISIBLE | WS_CHILD, 50, 100, 400, 20, hWndParent, NULL, NULL, N

    g_vecChildControls.push_back(hLabel);
}
void CreateView7(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Perfil de Usuario: Juan Pérez", WS_VISIBLE | WS_CHILD, 50, 100, 400, 20, hWndParent, NULL, NULL, NULL);
    HWND hButton = CreateWindow(L"BUTTON", L"Editar Perfil", WS_VISIBLE | WS_CHILD, 50, 180, 150, 30, hWndParent, NULL, NULL, NULL);

    g_vecChildControls.push_back(hLabel);
    g_vecChildControls.push_back(hButton);
}
void CreateViewOpcion1(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Aqui puede consultar los valores de referencia de los alquileres", WS_VISIBLE | WS_CHILD, 50, 50, 400, 40, hWndParent, NU

    g_vecChildControls.push_back(hLabel);
}
void CreateViewOpcion2(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Aqui puede solicitar su compensacion", WS_VISIBLE | WS_CHILD, 50, 50, 400, 20, hWndParent, NULL, NULL, NULL);

    g_vecChildControls.push_back(hLabel);
}
void CreateViewOpcion3(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Aqui puede solicitar una exncion por ser adulto mayor o vulnerble socialmente", WS_VISIBLE | WS_CHILD, 50, 50, 400, 50, h
```

Rentas | (Global Scope) | CreateViewOpcion3(HWND hWndParent)

```cpp
    g_vecChildControls.push_back(hLabel);
}
void CreateViewOpcion4(HWND hWndParent) {

    HWND hLabel = CreateWindow(L"STATIC", L"Aqui puede trasnferir su credito", WS_VISIBLE | WS_CHILD, 50, 50, 400, 20, hWndParent, NULL, NULL, NULL);

    g_vecChildControls.push_back(hLabel);
}

// ----- WndProc de la Ventana Secundaria -----
LRESULT CALLBACK SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) {
    switch (msg) {
    case WM_CREATE:
    {
        // Al crear la ventana secundaria, generamos los botones y el botón de regreso
        generarBotones(hwnd);
        CreateWindow(L"BUTTON", L"Regresar al Menú", WS_VISIBLE | WS_CHILD, 150, 300, 180, 40, hwnd, (HMENU)ID_BUTTON_RETURN_TO_MAIN, NULL, NULL);
        break;
    }

    case WM_APP_CHANGE_VIEW:
    {
        DestroyChildControls();

        // Leemos qué vista nos pidieron cargar (el ID del botón que pasamos en SendMessage)
        int viewId = (int)wParam;

        // Creamos la nueva interfaz según el ID.
        switch (viewId) {
        case ID_BUTTON_SHOW_VIEW1:
            CreateView1(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW2:
```

Rentas (Global Scope) SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lPa

```
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW2:
            CreateView2(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW3:
            CreateView3(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW4:
            CreateView4(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW5:
            CreateView5(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW6:
            CreateView6(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_BUTTON_SHOW_VIEW7:
            CreateView7(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_MENU_OPCION1:
            CreateViewOpcion1(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_MENU_OPCION2:
            CreateViewOpcion2(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_MENU_OPCION3:
            CreateViewOpcion3(hwnd);
```

Rentas (Global Scope) SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lPa

```
            break;
        case ID_MENU_OPCION3:
            CreateViewOpcion3(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        case ID_MENU_OPCION4:
            CreateViewOpcion4(hwnd);
            SetWindowText(hwnd, L"Rentas Cordoba");
            break;
        }
        break;
    }

    case WM_COMMAND:
    {
        // Manejamos el comando del botón de regreso y de los demas botones
        int wmId = LOWORD(wParam);
        HMENU hMenu = CreatePopupMenu();
        RECT rect;
        switch (wmId) {
        case ID_BUTTON_SHOW_VIEW1:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW2:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW3:
```

Rentas (Global Scope) SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lPa

```
            break;
        case ID_BUTTON_SHOW_VIEW3:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW4:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW5:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW6:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
            break;
        case ID_BUTTON_SHOW_VIEW7:
            ShowWindow(g_hwndMain, SW_HIDE);

            SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);

            ShowWindow(g_hwndSecondary, SW_SHOW);
```

```
Rentas                    (Global Scope)                    SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lPa
511
512         case ID_BUTTON_SHOW_VIEW8:
513
514
515
516             AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION1, L"Valores de Ref...");
517             AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION2, L"Compensacion");
518             AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION3, L"Exencion...");
519             AppendMenuW(hMenu, MF_STRING, ID_MENU_OPCION4, L"Transferencia");
520
521             AppendMenuW(hMenu, MF_SEPARATOR, ID_MENU_SEPARADOR, NULL);
522
523
524             GetWindowRect(g_hDropdownButton, &rect);
525
526
527             TrackPopupMenuEx(hMenu, TPM_LEFTALIGN | TPM_TOPALIGN, rect.left, rect.bottom, hwnd, NULL);
528
529             DestroyMenu(hMenu);
530             break;
531         case ID_MENU_OPCION1:
532             ShowWindow(g_hwndMain, SW_HIDE);
533
534             SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);
535
536             ShowWindow(g_hwndSecondary, SW_SHOW);
537             break;
538         case ID_MENU_OPCION2:
539             ShowWindow(g_hwndMain, SW_HIDE);
540
541             SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);
542
543             ShowWindow(g_hwndSecondary, SW_SHOW);
544             break;
545         case ID_MENU_OPCION3:
```

```
Rentas                    (Global Scope)                    SecondaryWndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lPa
544             break;
545         case ID_MENU_OPCION3:
546             ShowWindow(g_hwndMain, SW_HIDE);
547
548             SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);
549
550             ShowWindow(g_hwndSecondary, SW_SHOW);
551             break;
552         case ID_MENU_OPCION4:
553             ShowWindow(g_hwndMain, SW_HIDE);
554
555             SendMessage(g_hwndSecondary, WM_APP_CHANGE_VIEW, (WPARAM)wmId, 0);
556
557             ShowWindow(g_hwndSecondary, SW_SHOW);
558             break;
559         case ID_BUTTON_RETURN_TO_MAIN:
560             ShowWindow(g_hwndSecondary, SW_HIDE);
561             ShowWindow(g_hwndMain, SW_SHOW);
562             break;
563         }
564     }break;
565     case WM_CLOSE:
566         DestroyWindow(g_hwndMain);
567         break;
568     case WM_DESTROY:
569         PostQuitMessage(0);
570         break;
571     default:
572         return DefWindowProc(hwnd, msg, wParam, lParam);
573     }
574     return 0;
575 }
```

Video del prototipo en funcionamiento

Grabación 2025-08-18 144931.mp4