



# Universidade Estadual de Campinas

## Instituto de Computação

Prof. Dr. Bruno Barbieri de Pontes Cafeo  
cafeo@ic.unicamp.br  
<https://ic.unicamp.br/~cafeo/>



## INF1900 - Recursos Avançados de C++ (2023)

### Exercício 4 - Windows API

#### Contexto

Imagine que você está desenvolvendo uma aplicação simples de desktop usando a API Win32 em C++. Seu objetivo é criar uma aplicação básica que possa interagir com o usuário. Esta aplicação permitirá ao usuário inserir um texto e, ao clicar em um botão, o texto será exibido em uma `MessageBox`.

#### Parte 1 - Configuração da Janela e Controles

Implemente a função **WinMain** para iniciar sua aplicação e criar uma janela. Adicione um controle de “caixa de texto” onde os usuários podem inserir uma string e um botão que, quando pressionado, recupera o texto da caixa de texto.

Certifique-se de registrar a classe da janela e criar a janela com os estilos apropriados para acomodar os controles. Use a função **CreateWindow** para criar os controles.

#### Parte 2 - Manipulação de Eventos

Implemente a função de procedimento da janela **WndProc** para manipular os eventos do botão e da caixa de texto. Quando o botão é clicado, sua aplicação deve recuperar o texto da caixa de texto e exibi-lo em uma `MessageBox` usando a função **MessageBox**. Sua aplicação deve tratar as seguintes mensagens:

1. **WM\_CREATE**: Este evento é acionado após a janela ser criada. Utilize-o para definir configurações iniciais, criar controles de interface do usuário (como caixas de texto e botões) e realizar outras configurações necessárias.
2. **WM\_COMMAND**: Ocorre quando o usuário interage com um controle da interface, como pressionar um botão. Você precisará identificar qual controle iniciou o evento (por exemplo, verificando o ID do controle) e decidir as ações apropriadas a serem tomadas, como coletar dados da caixa de texto e iniciar processamentos.
3. **WM\_DESTROY**: É recebido quando a janela está prestes a ser destruída, marcando a oportunidade perfeita para realizar a limpeza, liberar recursos e encerrar a aplicação. Não se esqueça de chamar **PostQuitMessage(0)** para indicar que o aplicativo deve ser encerrado, o que, por sua vez, iniciará o evento **WM\_QUIT**.

Adicionalmente, enriqueça a interatividade do seu aplicativo implementando uma mensagem personalizada. Configure seu aplicativo para enviar uma mensagem personalizada quando o usuário pressionar o botão. Esta mensagem deve ser capturada e tratada dentro de **WndProc**, levando a função **MessageBox** a ser chamada para exibir o conteúdo da caixa de texto. Nesse contexto, defina **WM\_MINHA\_MENSAGEM** acima de **WinMain**, envie-a com **PostMessage** durante **WM\_COMMAND**, e, ao recebê-la em **WnProc**, exiba o texto da caixa em uma `MessageBox`.

#### Parte 3 - Execução e Teste

Após implementar as partes acima, compile e execute sua aplicação. Teste a funcionalidade inserindo diferentes strings na caixa de texto e clicando no botão para exibir a `MessageBox`. Verifique se o texto correto está sendo mostrado.