

## Visão geral

wxWidgets foi lançada em 1992 por Julian Smart. Ela permite que um programa seja compilado e executado em diversas plataformas de computação, com poucas ou nenhuma modificação. Ela suporta sistemas como: Microsoft Windows, Apple Inc. Mac OS, Linux/Unix (para X11).

A biblioteca é implementada em C++, mas outras implementações estão disponíveis para várias das linguagens de programação mais comuns, entre elas: Python, Smalltalk, Perl e Java.

wxWidgets é melhor descrita como um utilitário nativo. Ao invés de emular a apresentação de widgets utilizando primitivas gráficas nas diferentes plataformas suportadas, ela fornece uma pequena abstração para código nativo. Isso a torna mais rápida, e com um visual melhor adaptado à plataforma, que utilitários como o Java Swing.

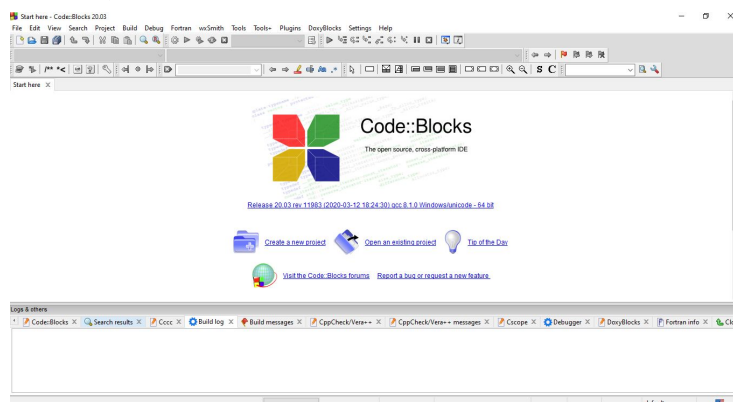
A biblioteca não é restrita ao desenvolvimento de interfaces gráficas, tendo suporte a conexão a bancos de dados ODBC, conectividade por sockets e mais.

## Instalação e configuração do ambiente

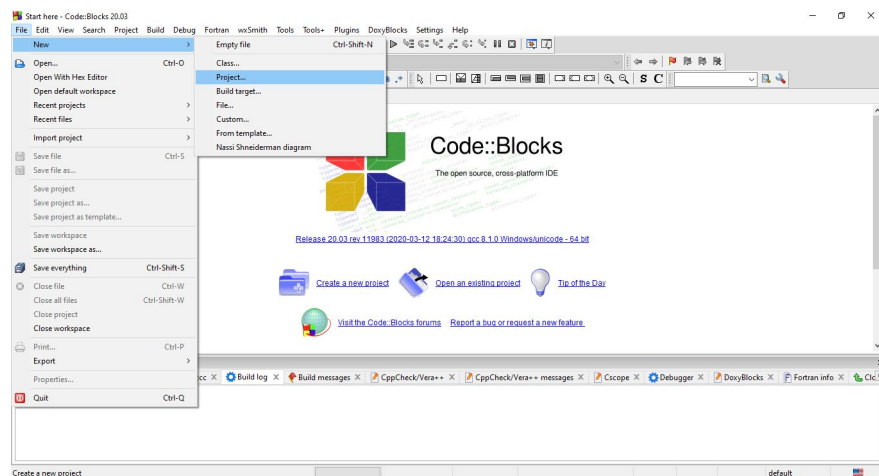
Primeiramente o wxWidgets deve ser baixado do site: <https://www.wxwidgets.org/downloads/>.

Baixe a versão zip e depois extraia ela onde for de sua preferência. Depois você deve acessar o cmd. Entre na pasta wxWidgets\build\msw pelo cmd. Feito isso você usará o comando “mingw32-make.exe -f makefile.gcc BUILD=release SHARED=1 MONOLITHIC=1 UNICODE=1” no cmd. Isso pode levar alguns minutos, entre 10 e 15. Nesse momento ele linka e compila as bibliotecas do wxWidgets. A partir do momento que é feita a instalação correta do wxwidgets você será capaz de criar uma aplicação constituída por interfaces. Os passos para criação de um projeto que utiliza o wxwidgets serão demonstrados a seguir:

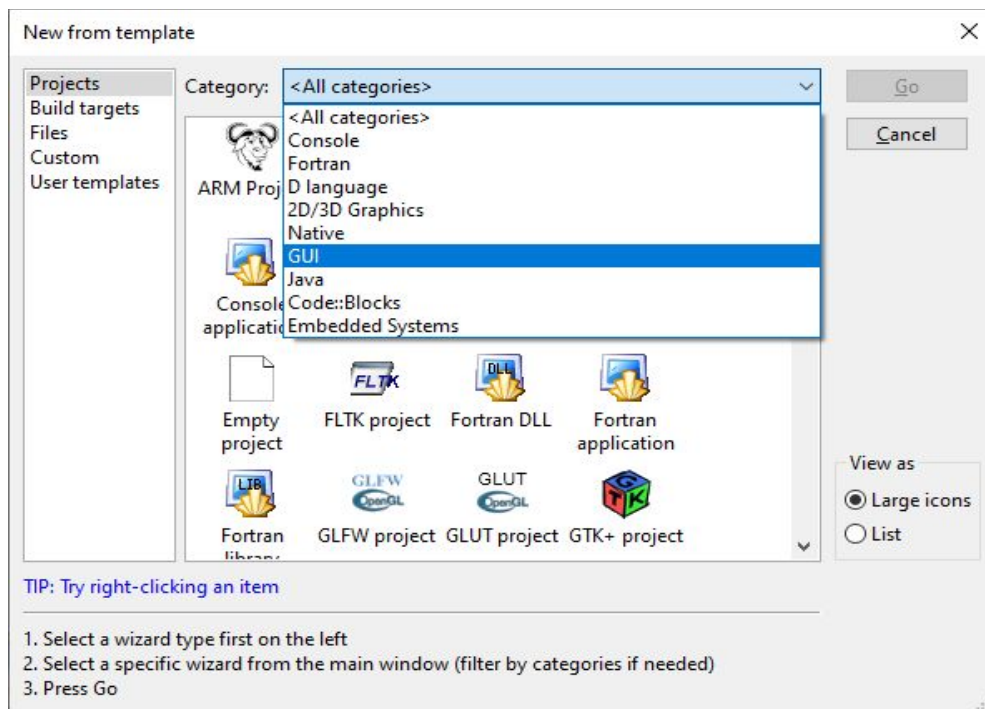
- Tela inicial, ao abrir a ferramenta codeBlocks.



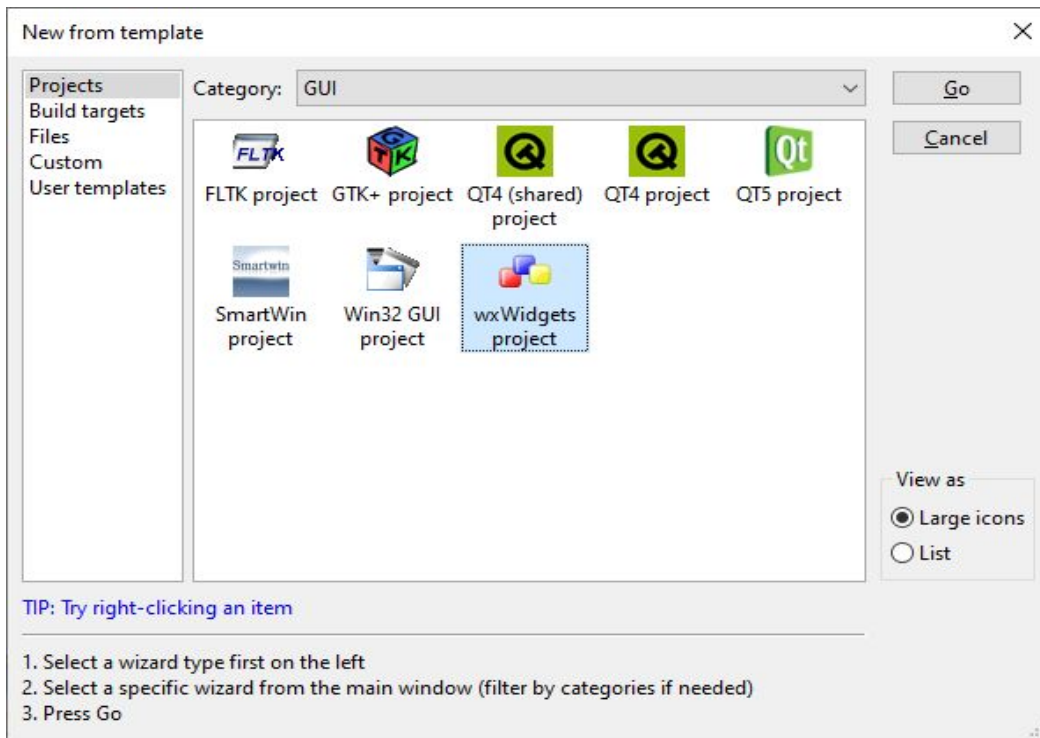
- Para criar um aplicação que utiliza o wxwidgets você cria um novo projeto normalmente.



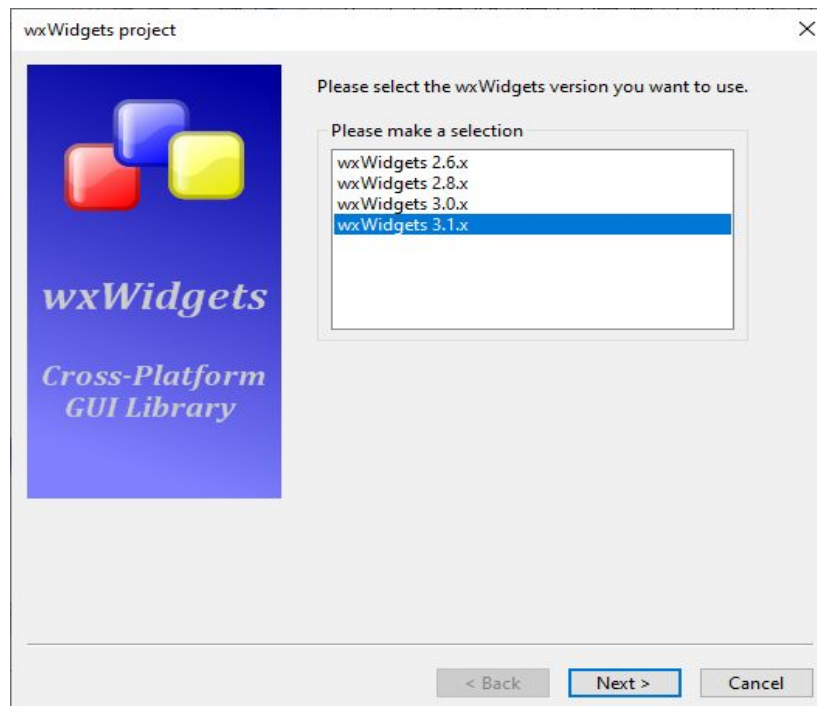
- Ao selecionar a opção de criação do projeto você pode usar a comboBox presente na janela que aparecerá para selecionar uma categoria. No caso será a GUI que demonstrará os que você pode usar para criação de interfaces.



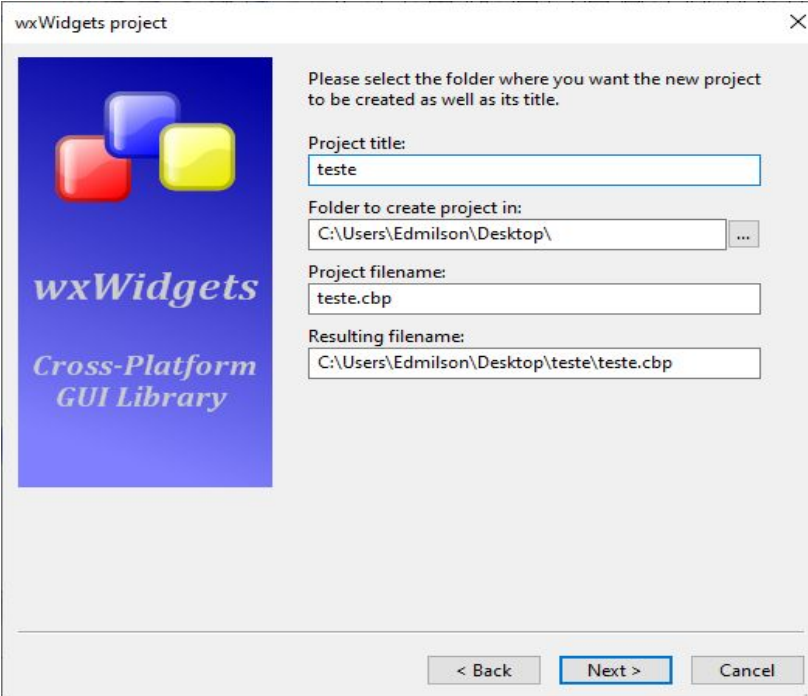
- Com isso o wxWidgets estará disponível para que você possa selecioná-lo.



- Será solicitado a versão do wxWidgets que está presente no meu sistema. Após escolhê-la clique em Next>.



- Será então solicitado o nome do projeto que será criado junto com o nome da pasta onde o mesmo será salvo. Após fornecer clique em Next>.



wxWidgets project

Please select the folder where you want the new project to be created as well as its title.

Project title:

Folder to create project in:  ...

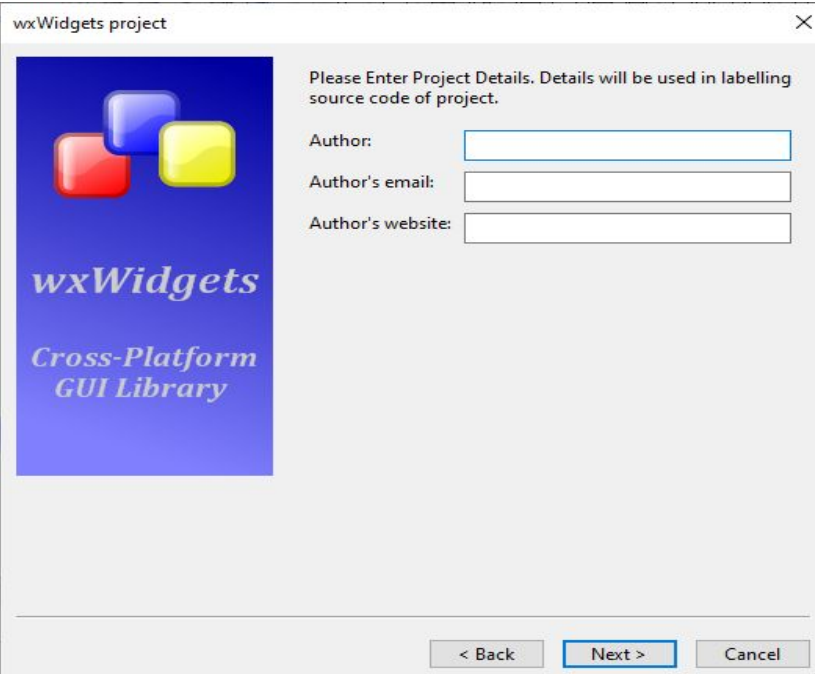
Project filename:

Resulting filename:

< Back   Next >   Cancel

*wxWidgets*  
*Cross-Platform*  
*GUI Library*

- Depois será solicitado alguns dados opcionais que são usados para se identificar o criador da aplicação. Esses dados podem ser deixados em branco caso o usuário não queira preenchê-los. Após decidir clique em Next>.



wxWidgets project

Please Enter Project Details. Details will be used in labelling source code of project.

Author:

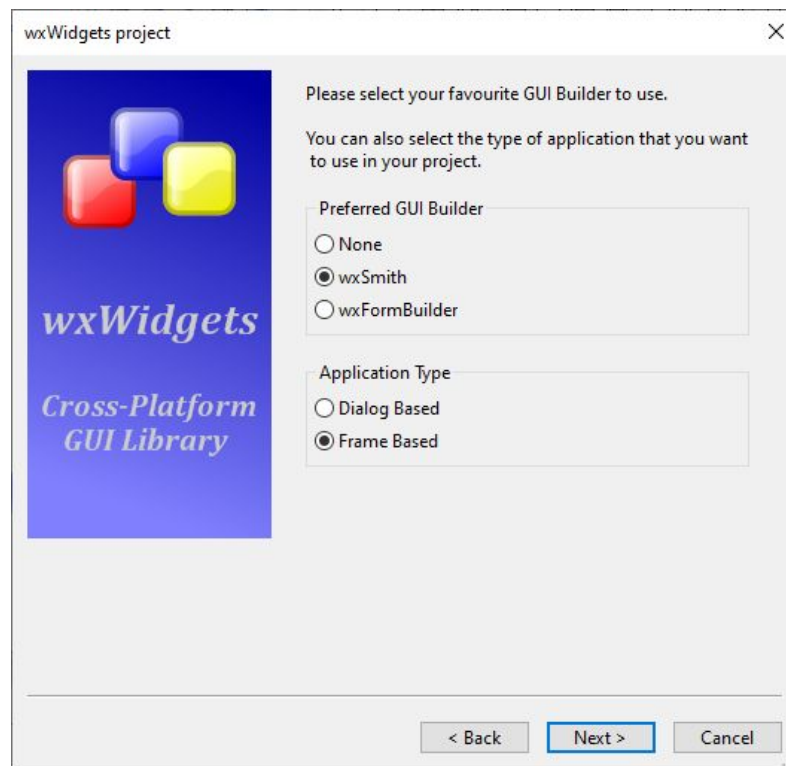
Author's email:

Author's website:

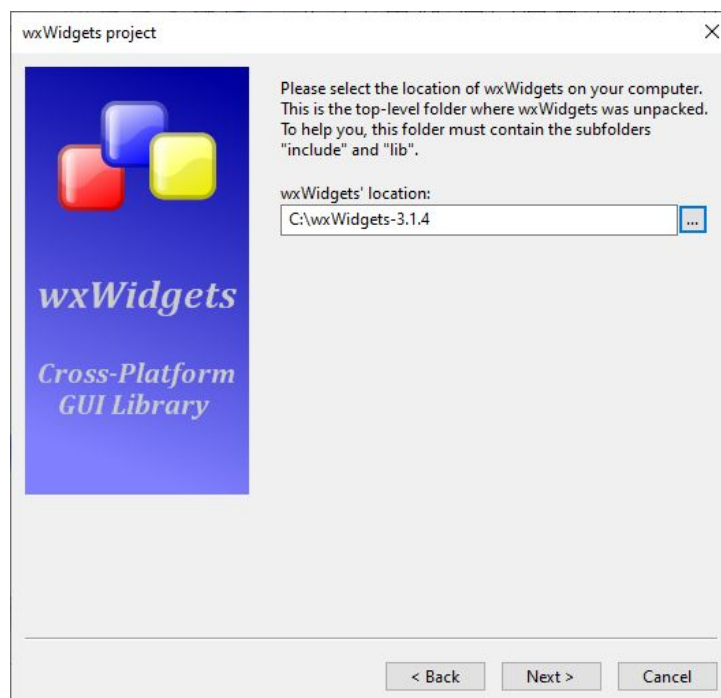
< Back   Next >   Cancel

*wxWidgets*  
*Cross-Platform*  
*GUI Library*

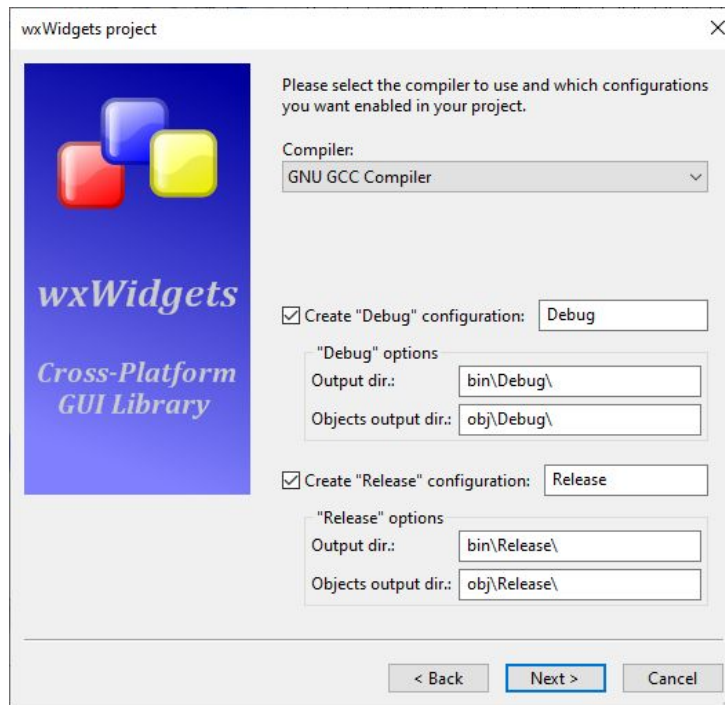
- Aqui será escolhida o construtor de GUI que se deseja usar e que no caso foi escolhido o wxSmith que já vem no codeblocks. Após escolher clique em Next>.



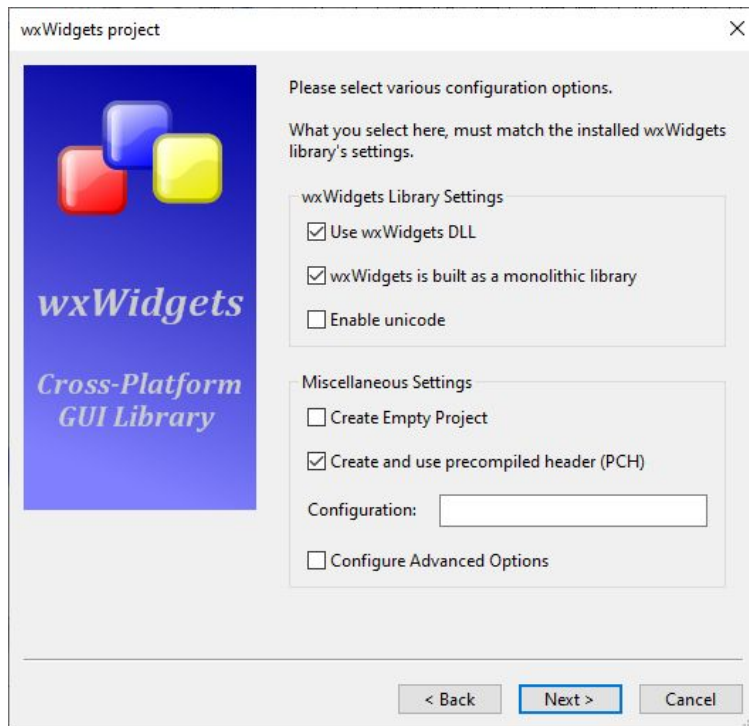
- Será então solicitada a pasta que contem o WxWidgets no seu computador, ele ainda fornece a dica que essa pasta deve ser a mais exterior e que dentro dela contém as pastas "lib" e "include" além de outras. Após fornecer clique em Next>.



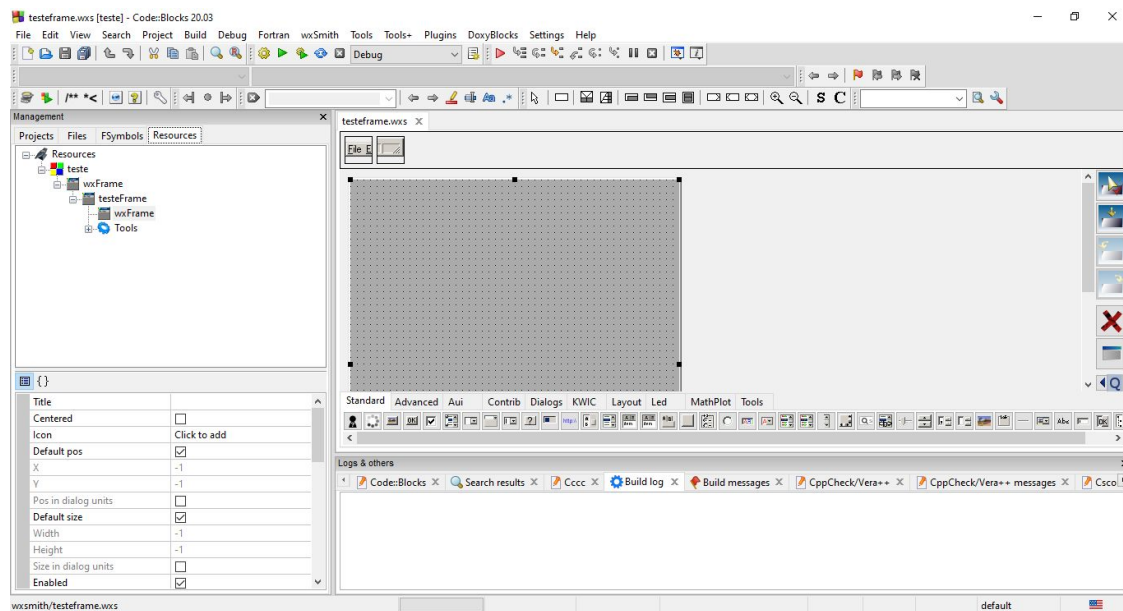
- A próxima tela especifica qual compilador C ++ queremos usar e se queremos uma compilação de depuração (com código extra para depuração) ou uma compilação de versão menor para uso por outros. Foi selecionado ambos para que sejam criadas em sub pastas separadas.



- A tela final pergunta se queremos usar a configuração padrão ou usar as opções avançadas. Vamos usar o padrão. Mas, na parte inferior, escolhemos “Criar e usar cabeçalhos pré-compilados” o que acelera as compilações após o primeiro. Após escolher clique em Next>.

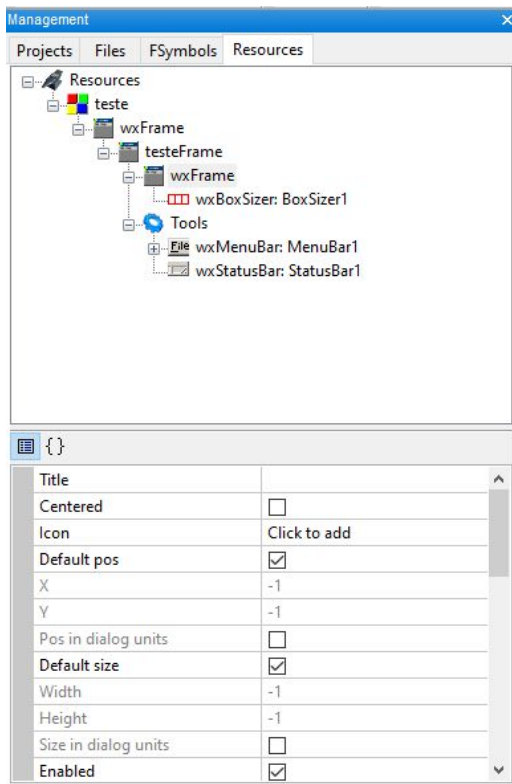


Após seguir todos esses passos você será apresentado a tela inicial do CodeBlocks.



Antes de partir para pôr a mão na massa vamos analisar a disposição de algumas funcionalidades.





Essa região de gestão de componentes. Onde está a parte de recursos que são utilizados na sua aplicação.

Logo abaixo está uma região com duas abas:

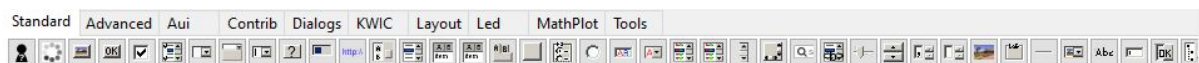
- uma aba que disponibiliza as propriedades dos componentes, que podem ser editados.
- e a outra aba que permite o usuário a escolher eventos para os componentes.

Logo abaixo de onde os arquivos de sua aplicação são apresentados existe uma barra que seria uma divisão dos componentes disponíveis:



Duas abas que serão a Standart e Layout onde:

Standart apresenta os componentes padrões que podem ser utilizados.



Os mais comuns são:

## wxButton



Este é o botão padrão, um componente muito comum. Seu objetivo é executar algum trecho de código quando o usuário clicar nele.

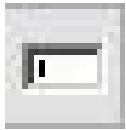
## wxStaticText





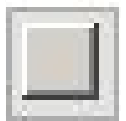
Este componente mostra apenas texto definido pelo usuário na janela. Não gera eventos.

## **wxTextCtrl**



Este componente permite que o usuário insira algum texto. Sua funcionalidade básica permite inserir texto de uma linha. Depois de alterar suas propriedades, ele pode se tornar um editor de várias linhas mais avançado, com suporte para diferentes fontes, cores e estilos de texto.

## **wxPanel**



Embora esse componente possa ser uma janela separada, é mais comumente adicionado como um componente em uma janela pai. É particularmente útil como pano de fundo para outros componentes. Tem algumas funcionalidades - pode ser usado como pano de fundo em notebooks e pode ser um local onde você coloca componentes sem calibradores. Mas também tem algumas desvantagens - wxWidgets tende a fazer alguns truques com o tamanho de wxPanel, então nem sempre é possível definir o tamanho que realmente queremos.

## **wxComboBox**



Este componente fornece uma lista suspensa onde o usuário pode selecionar um dos valores disponíveis ou inserir o seu próprio se o componente não estiver no estado Somente Leitura.

## **wxListBox**



Este componente fornece uma lista onde você pode selecionar um ou mais componentes. Mudar o componente ou clicar duas vezes nele gera um evento.

## **wxCheckBox**



As caixas de seleção podem ser usadas para mapear valores booleanos. Além disso, caixas de 2 estados também podem ser geradas (com valores: sim / não / não especificado). Alterar o valor emite um evento.

## **wxRadioButton**



O radio button funciona de maneira semelhante às caixas de seleção. A diferença é que neste apenas uma caixa de rádio em um grupo pode ser selecionada.. Alteração selecionada emite um evento.

## wxSlider



Um Slider é um controle com uma alça que pode ser puxada para frente e para trás para alterar o valor. Gera eventos.

E a aba Layout que representa como novos componentes serão dispostos em um Panel ou Frame por exemplo:

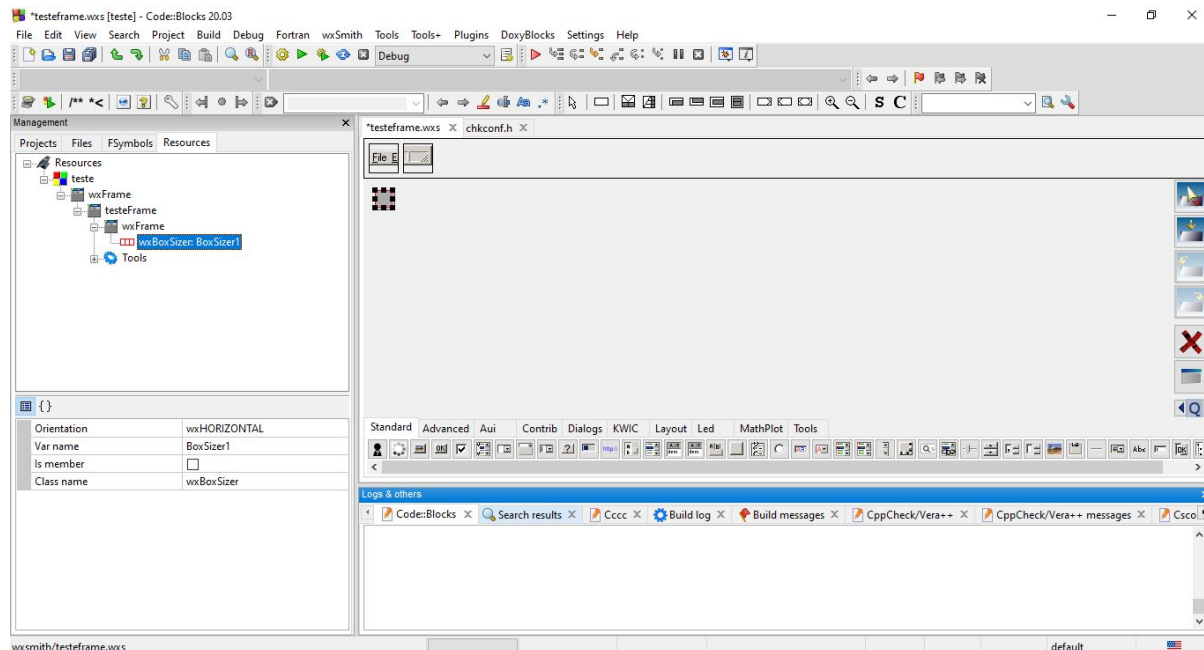


Ao lado direito você tem os modos de se inserir os componentes onde:

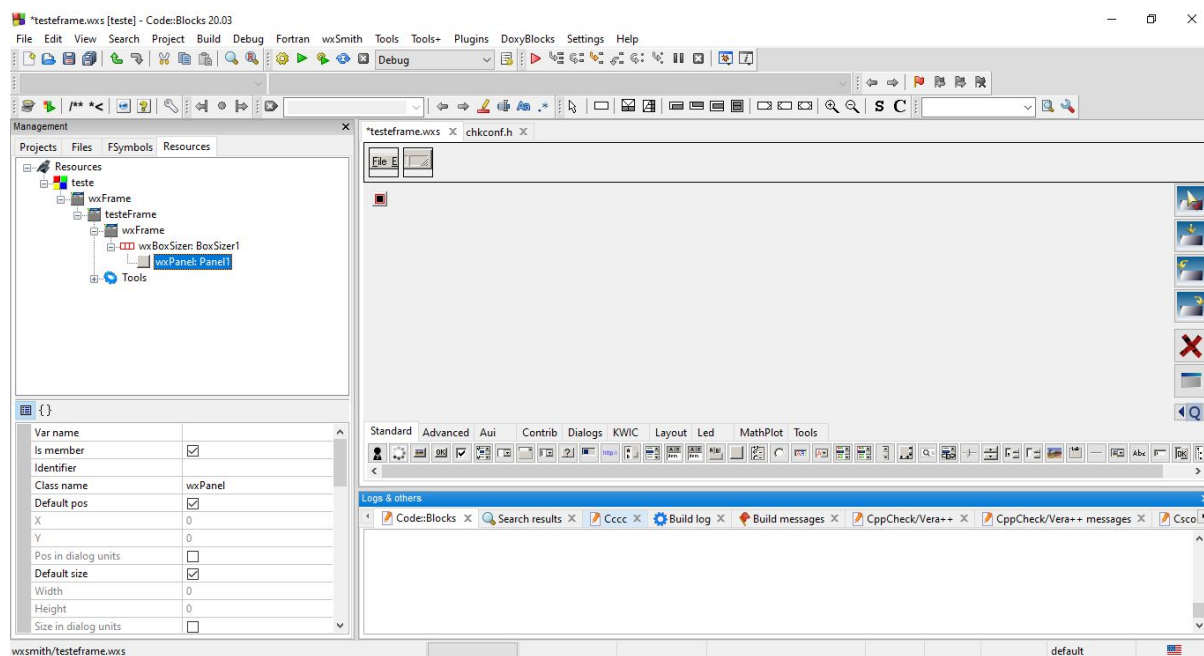


- Essa opção é para se inserir um componente apontando com o mouse.
- Essa opção é para inserir um componente na região que já está selecionada.
- Essa opção é para inserir um componente antes da região que já está selecionada.
- Essa opção é para inserir um componente depois da região que já está selecionada.
- Essa opção é para deletar um componente já selecionado.
- Essa opção é para dar uma ideia de como a interface está.
- Essa opção é para se abrir o painel com as opções.

Como visto ao se criar o projeto é adicionado automaticamente um Frame que será a base da interface. Para começarmos a criar uma interface utilizaremos um componente do Layout que no caso será o wxBoxSizer.



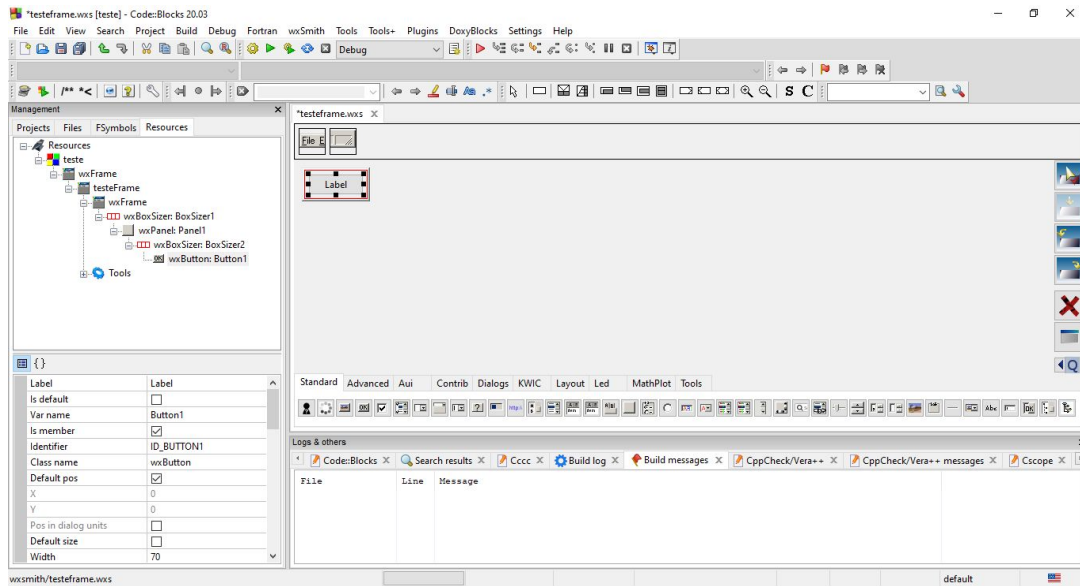
Logo depois disso é adicionado o Panel, Como dito para adicionar um componente você pode utilizar as opções na região direita da IDE, nesse caso a opção escolhida foi a de apontar com o mouse.



## COMPONENTES BÁSICOS

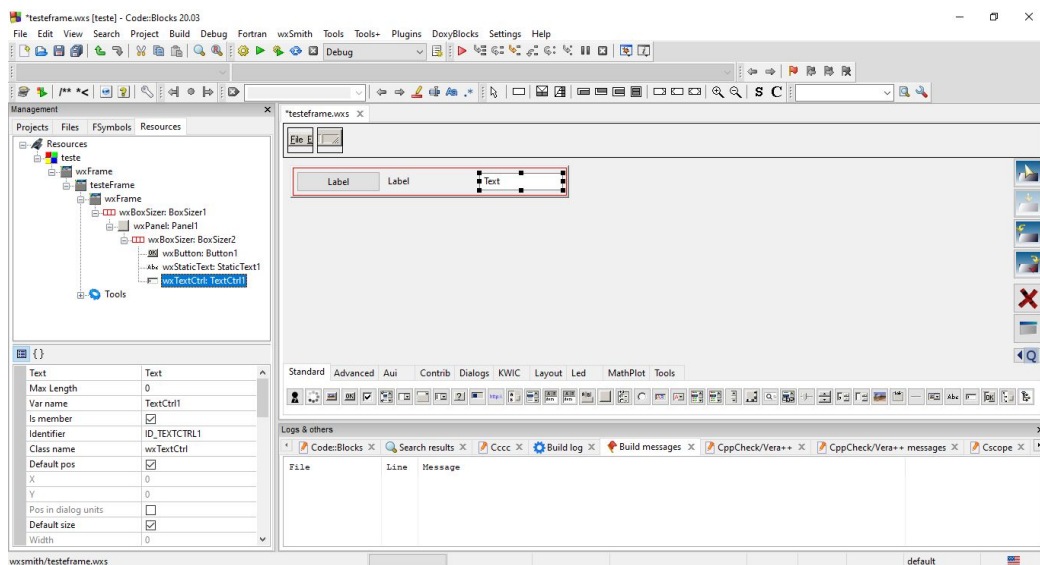
- Botão

Para se adicionar um botão na aplicação você aperta o wxButton na aba Standart e o arrasta, se estiver na opção de apontar com o mouse, e insere num outro componente que o suporte, nesse caso um Panel..



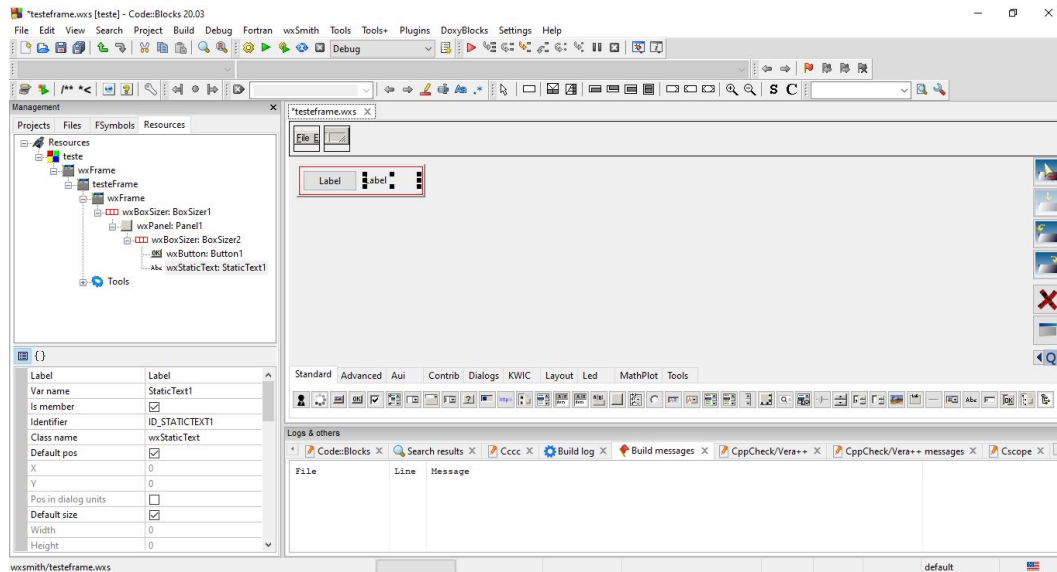
- Caixa para entrada de texto

Para se adicionar uma caixa para entrada de texto em um componente aperta o wxTextCtrl na aba Standart e o arrasta, se estiver na opção de apontar com o mouse, e insere num outro componente que o suporte, nesse caso um Panel.



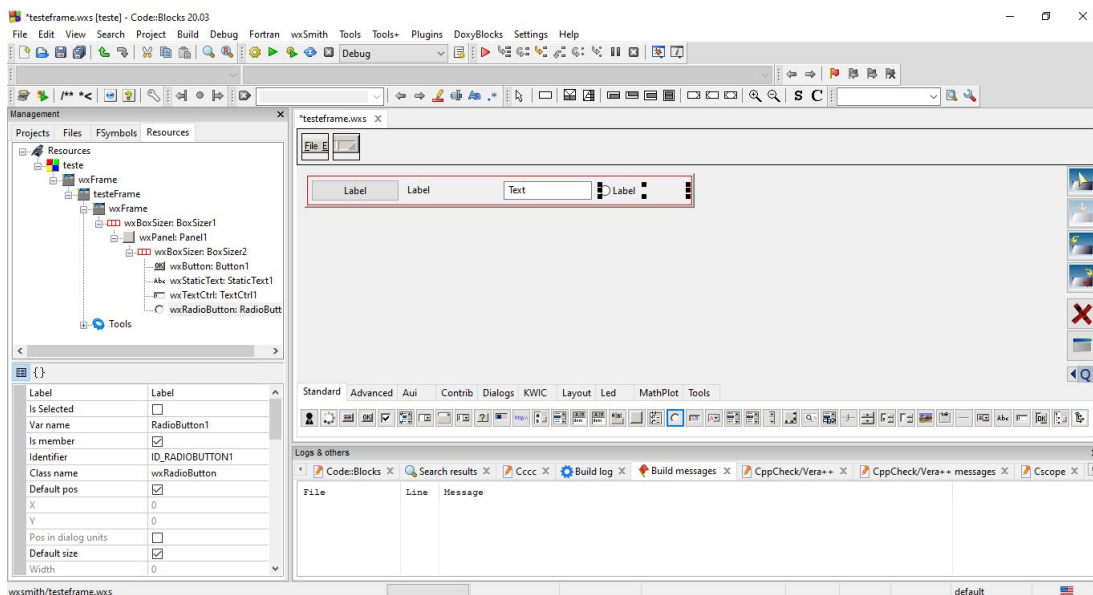
- Label de texto

Para se adicionar um Label de texto na aplicação você aperta o wxStaticText na aba Standart e o arrasta, se estiver na opção de apontarcom o mouse, e insere num outro componente que o suporte, nesse caso um Panel.



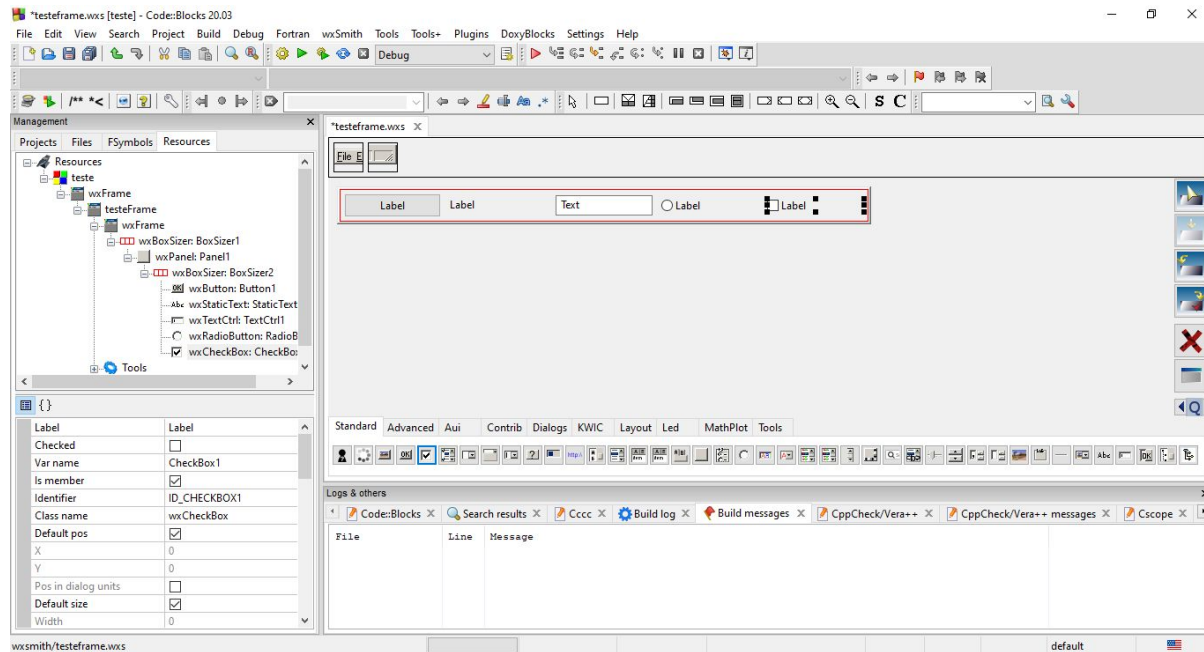
- Radio button

Para se adicionar um Radio Button na aplicação você aperta o wxRadioButton na aba standart e o arrasta, se estiver na opção de apontarcom o mouse, e insere num outro componente que o suporte, nesse caso um Panel.



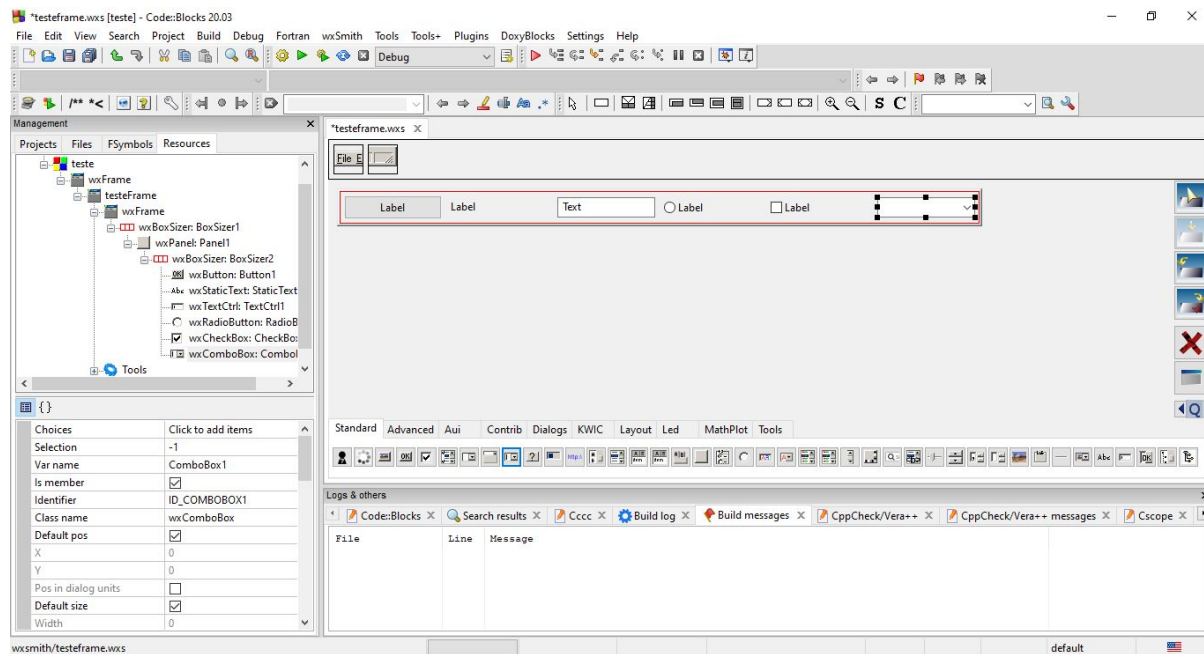
- Checkbox

Para se adicionar uma checkBox na aplicação você aperta o wxCeckBox na aba Standart e o arrasta, se estiver na opção de apontarcom o mouse, e insere num outro componente que o suporte, nesse caso um Panel.



- Lista suspensa

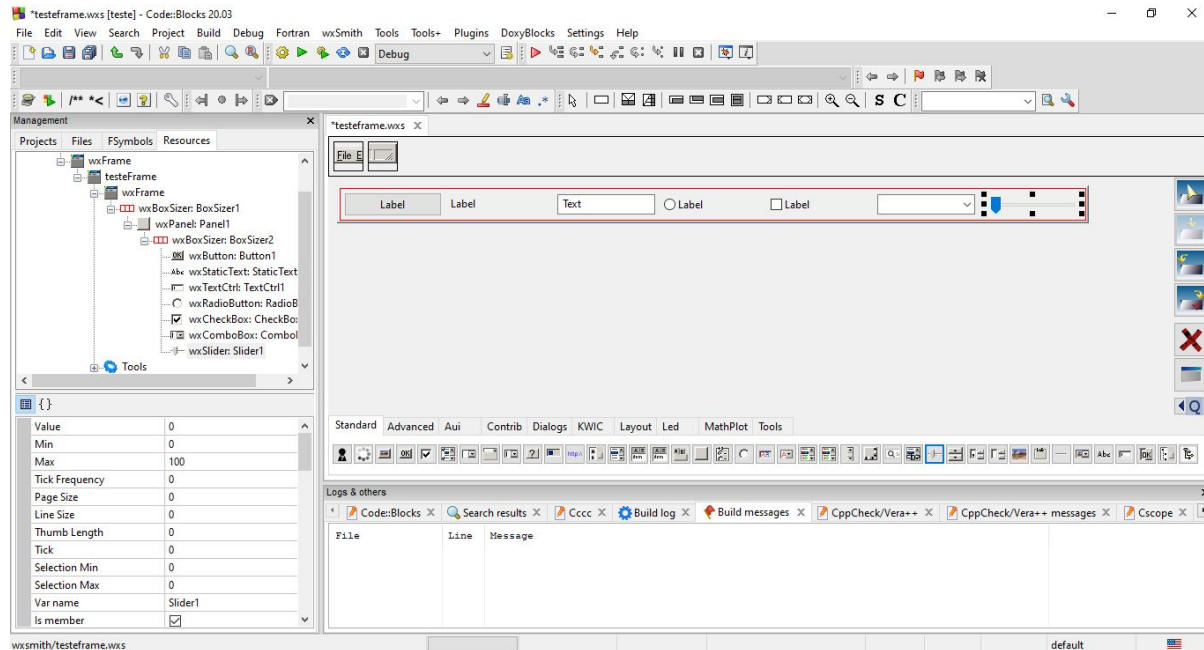
Para se adicionar uma lista suspensa na aplicação você aperta o wxComboBox na aba Standart e o arrasta, se estiver na opção de apontar com o mouse, e insere num outro componente que o suporte, nesse caso um Panel.



- Slider



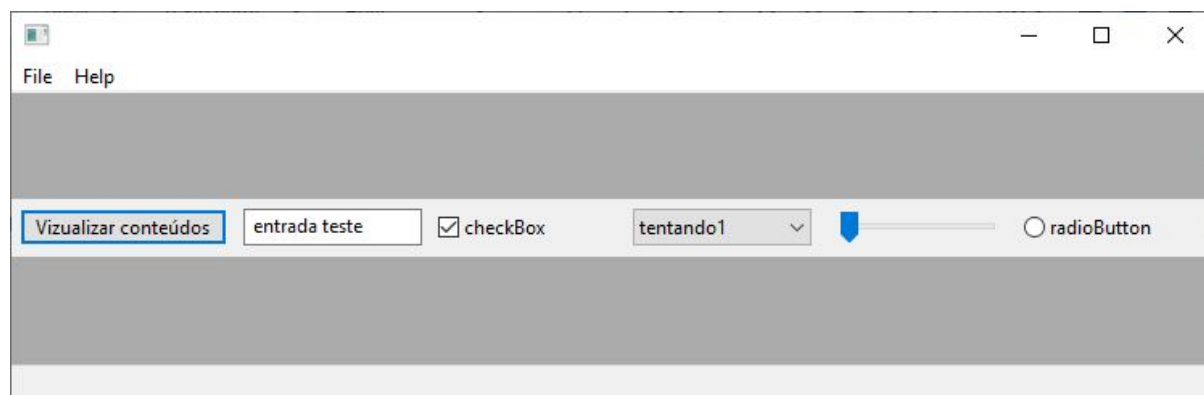
Para se adicionar um slider na aplicação você aperta o wxSlider na aba Standart e o arrasta, se estiver na opção de arrastar com o mouse, e insere num outro componente que o suporte, nesse caso um Panel.



\*\*\*Uma observação interessante é que quando já existe um componente, por exemplo um botão num Panel e você tentar adicionar outro componente como um label. Conforme o label se aproxima o botão fica azul claro à direita e então o label ficará a direita do botão e se o brilho for a esquerda o label ficará a esquerda.

## Obtenção de atributos

Para exemplificar como se obtém o conteúdo de um componente será utilizado a seguinte interface:





E para mostrar os conteúdos foi utilizado o seguinte código:

```
void testeFrame::OnButton1Click(wxCommandEvent& event)
{
    ///captura do conteúdo da caixa de texto.
    wxMessageBox("Conteúdo da caixa de texto: " + TextCtrl1->GetValue());

    ///captura do conteúdo da checkBox.
    wxMessageBox(CheckBox1->GetValue()==true?"checkbox marcada":"checkbox desmarcada");

    ///captura do conteúdo do radioButton.
    wxMessageBox(RadioButton1->GetValue()==true?"radio button marcado":"radio button desmarcado");

    ///captura do conteúdo do slider.
    wxMessageBox(wxString::Format("Valor do Slider: %d",Slider1->GetValue()));

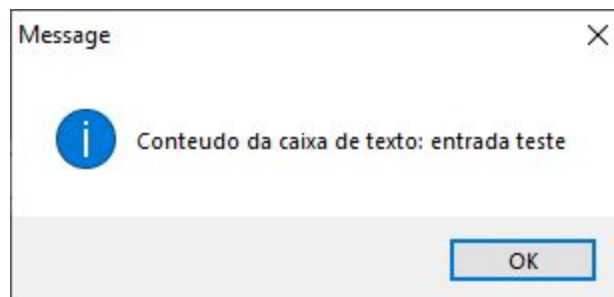
    ///captura do conteúdo da lista suspensa.
    wxMessageBox(wxString::Format("Opção da lista suspensa: %s",Choice1->GetString(Choice1->GetSelection())));
}
```

Onde foi utilizado o evento de clique no botão para assim ser mostrado os conteúdos de todos os componentes. E para exibir uma caixa de mensagem com o conteúdo é utilizado o wxMessageBox().

- Conteúdo de uma caixa de texto

Para se obter o conteúdo de uma caixa de texto é utilizado um método presente na própria classe wxTextCtrl chamado GetValue().

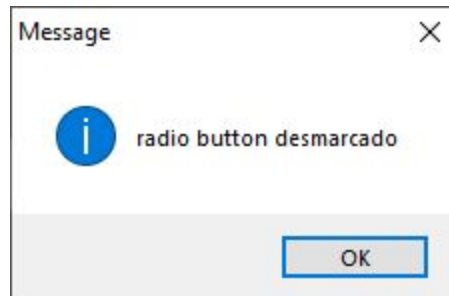
```
///captura do conteúdo da caixa de texto.
wxMessageBox("Conteúdo da caixa de texto: " + TextCtrl1->GetValue());
```



- Opção selecionada radioButton

Para se obter a opção selecionada num radioButton existe um método presente na classe wxRadioButton chamado GetValue() que permite verificar se a opção está marcada ou não. Em caso afirmativo é retornado true e caso contrário é retornado false.

```
///captura do conteúdo do radioButton.
wxMessageBox(RadioButton1->GetValue()==true?"radio button marcado":"radio button desmarcado");
```



- Opção selecionada checkbox

Para se obter a opção selecionada numa checkBox existe um método presente na classe `wxCheckBox` chamado `GetValue()` que permite verificar se a opção está marcada ou não. Em caso afirmativo é retornado `true` e caso contrário é retornado `false`.

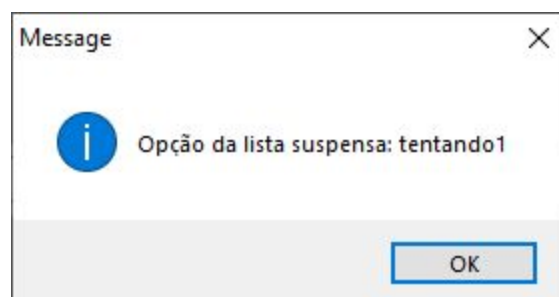
```
///captura do conteúdo da checkBox.  
wxMessageBox(CheckBox1->GetValue() == true ? "checkbox marcada" : "checkbox desmarcada");
```



- Opção selecionada lista suspensa

Para se obter a opção selecionada numa lista suspensa ou combobox existe um método presente na classe `wxChoice` chamado `GetString()` que recebe um inteiro como parâmetro representando uma posição da lista. Esta posição pode ser acessada com outro método da classe `wxChoice` chamado `GetSelection()`.

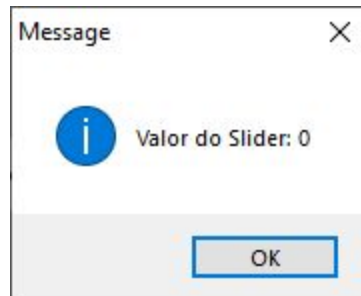
```
///captura do conteúdo da lista suspensa.  
wxMessageBox(wxString::Format("Opção da lista suspensa: %s", Choice1->GetString(Choice1->GetSelection())));
```



- Valor marcado no slider

Num slider também existe um `getValue()` mas esse retorna um inteiro que é o que foi selecionado pelo usuário.

```
///captura do conteúdo do slider.
wxMessageBox(wxString::Format("Valor do Slider: %d",Slider1->GetValue()));
```

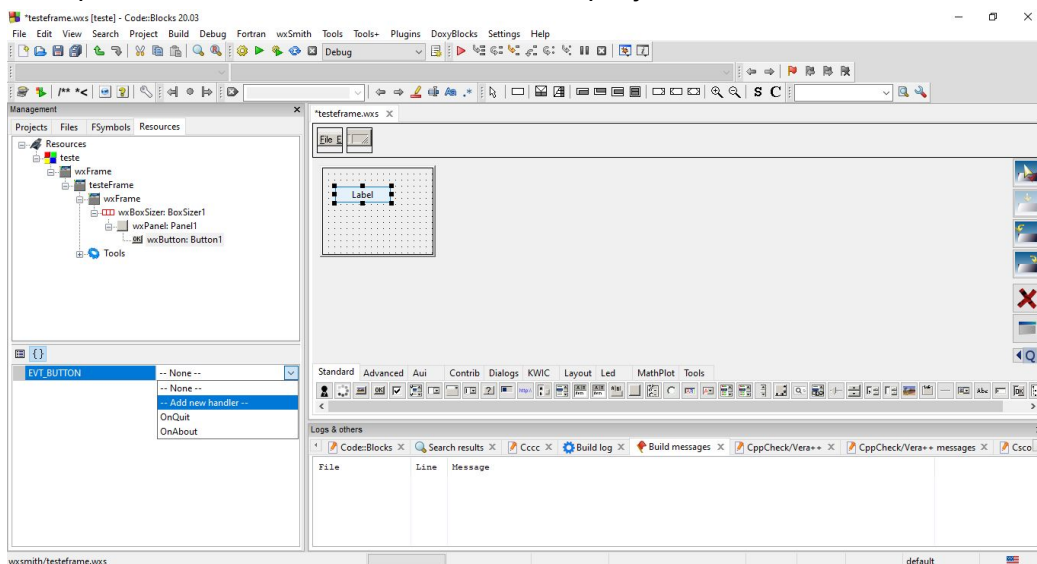


## Tratamento básico de eventos

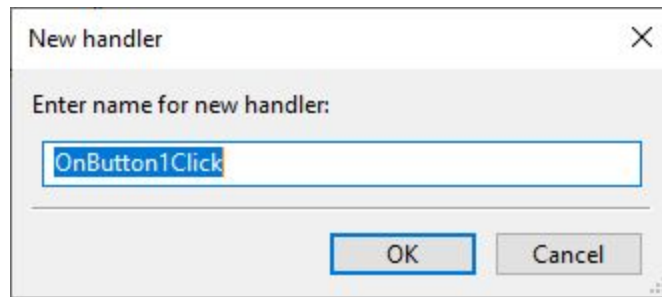
Para se tratar eventos é utilizado a região à direita da IDE chamada de events para assim o usuário fazer ou um novo evento ou utilizar eventos existentes.

- Clique no botão

Você pode criar um novo evento ou usar os que já existem.



Ao selecionar qual evento será utilizado você escolherá o nome do mesmo no código.

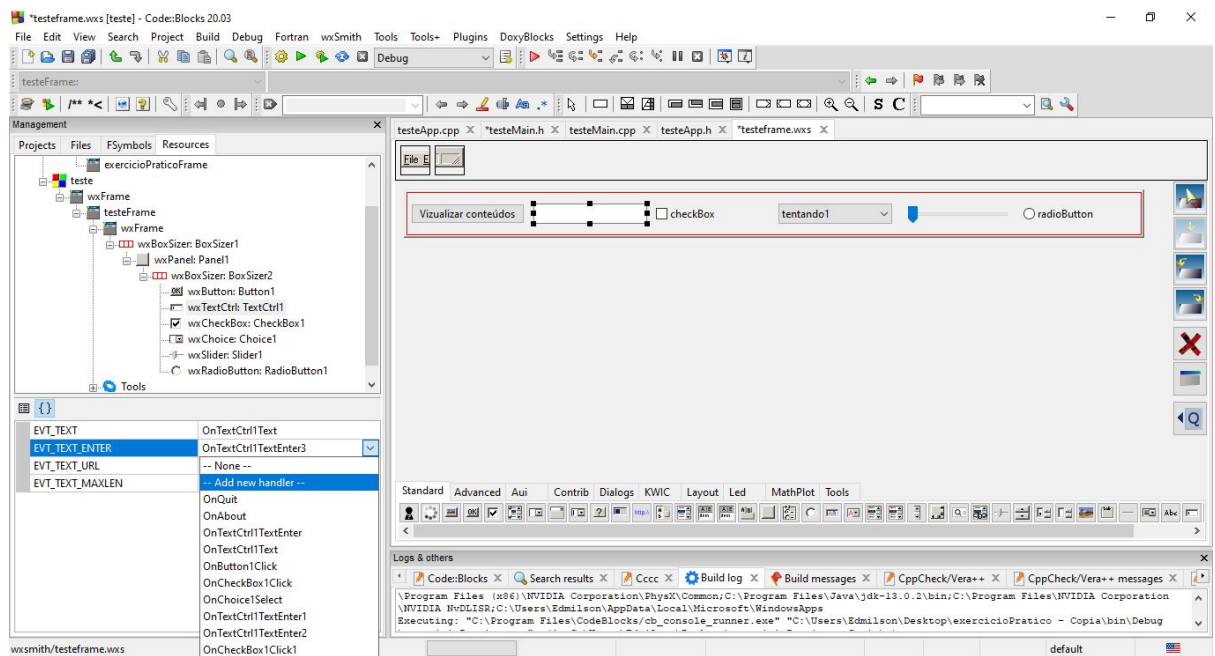


Então você será levado ao código para que você possa escrever o que ocorrerá.

```
102 {
103     //(*Destroy(testeFrame)
104     //*)
105 }
106
107 void testeFrame::OnQuit(wxCommandEvent& event)
108 {
109     Close();
110 }
111
112 void testeFrame::OnAbout(wxCommandEvent& event)
113 {
114     wxString msg = wxbuildinfo(long_f);
115     wxMessageBox(msg, _("Welcome to..."));
116 }
117
118 void testeFrame::OnButton1Click(wxCommandEvent& event)
119 {
120 }
121
122 }
```

- <enter> no input

Você pode criar um novo evento ou usar os que já existem.

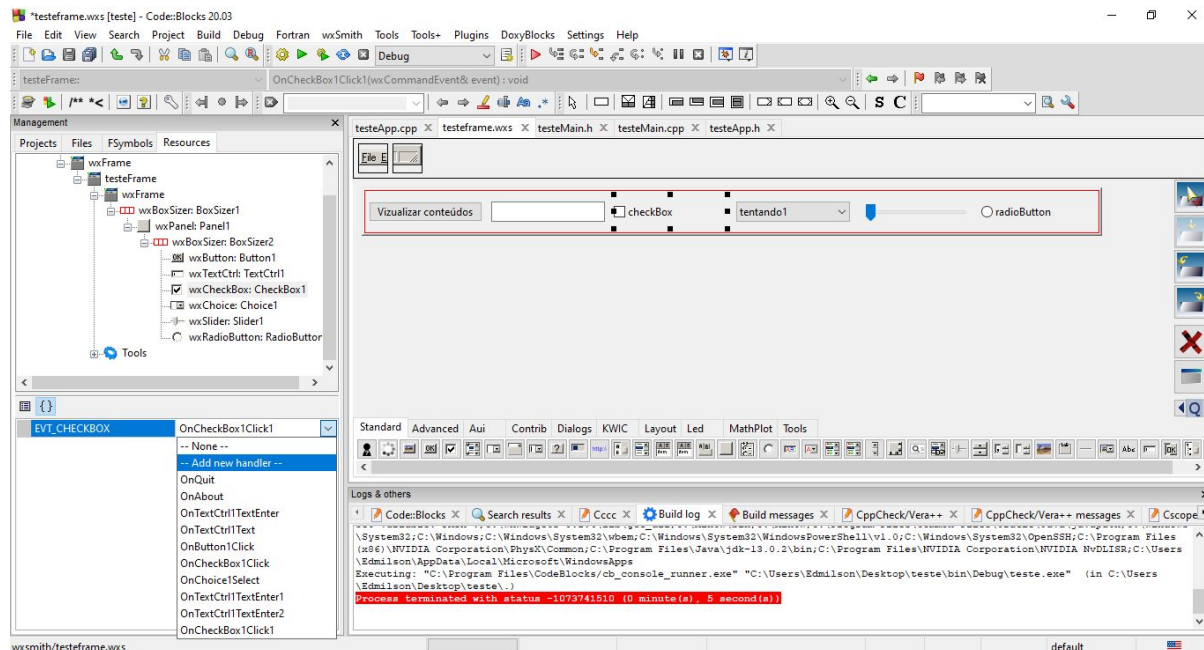


Então você será levado ao código para que você possa escrever o que ocorrerá.

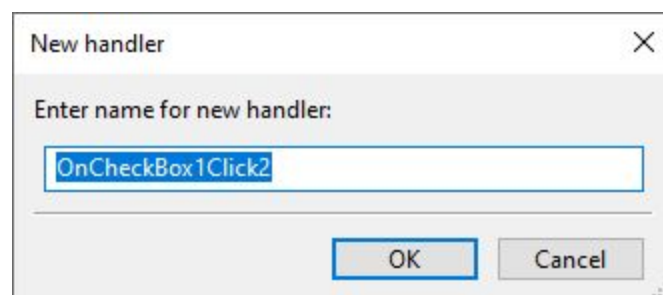
```
// evento ao dar enter no campo de texto.
void testeFrame::OnTextCtrl1TextEnter3(wxCommandEvent& event)
{
    // mensagem com o valor do campo.
    wxMessageBox("Valor no input:" + TextCtrl1->GetValue());
}
```

- Marcar/desmarcar radio/checkbox

Você pode escolher um dos eventos já gerados ou criar um novo.



Ao selecionar qual evento será utilizado você escolherá o nome do mesmo no código.



Será gerado um código para que se possa escolher a ação que ocorrerá ao se marcar ou desmarcar o radioButton ou a checkBox. Nesse caso ele verifica se caso o usuário marcar a box a mensagem “Marcou a box” aparece numa caixa de mensagem e caso o usuário desmarcá-la será exibida a mensagem “Desmarcou a box”.



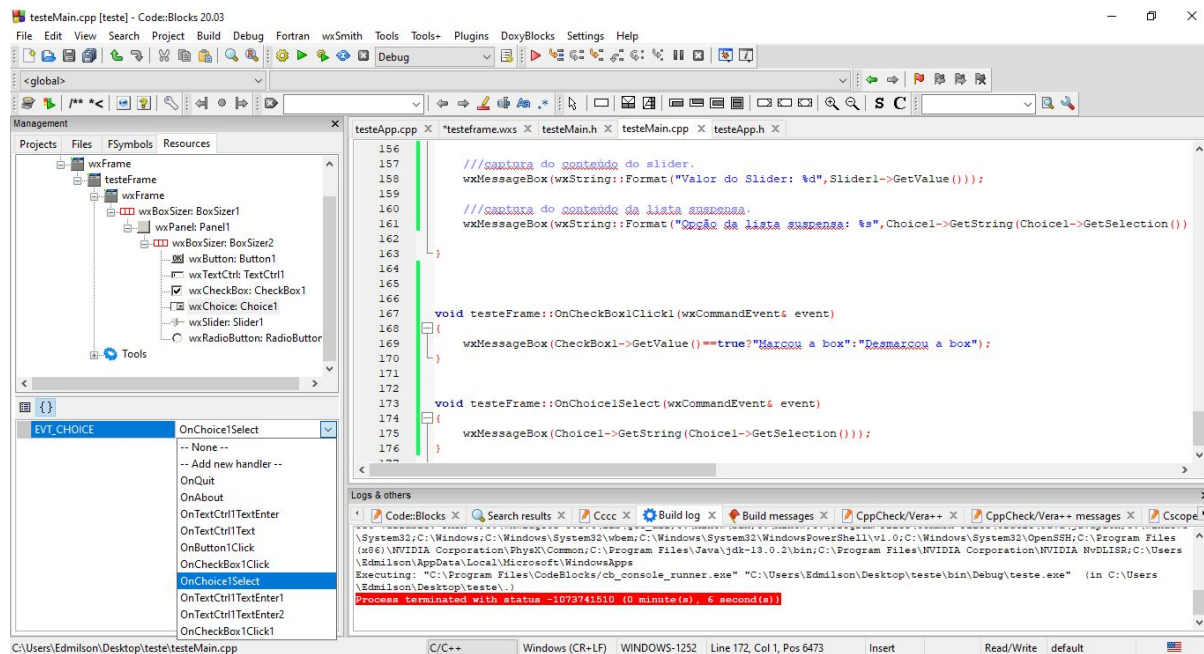
```

void testeFrame::OnCheckBox1Click1(wxCommandEvent& event)
{
    wxMessageBox(CheckBox1->GetValue() == true ? "Marcou a box" : "Desmarcou a box");
}

```

- Alteração do valor da lista suspensa

O usuário pode escolher tanto entre eventos que já existem ou criar um novo.



Para variar um pouco foi escolhido um evento já existente. Agora você pode criar alguma ação no momento que uma nova opção é selecionada na lista suspensa. Nesse caso a função exibe uma caixa de diálogo cuja mensagem é o item que o usuário selecionou.

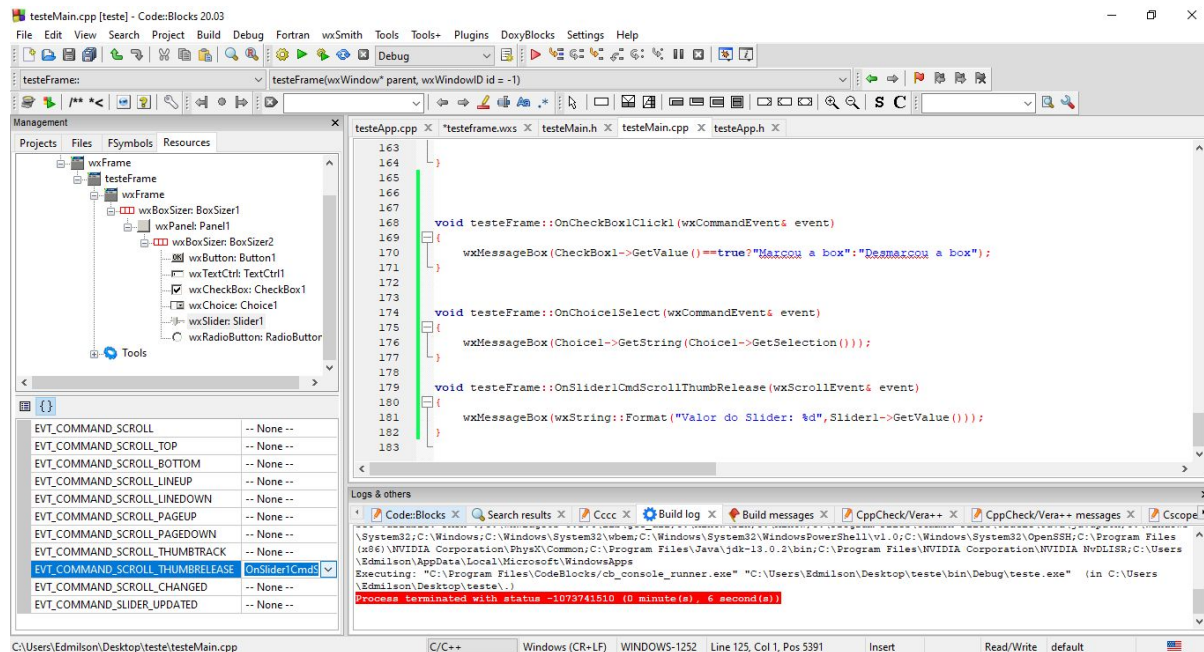
```

void testeFrame::OnChoice1Select(wxCommandEvent& event)
{
    wxMessageBox(Choice1->GetString(Choice1->GetSelection()));
}

```

- Alteração do valor do slider

Você pode escolher eventos que já existem e criar uma determinada ação ou criar uma função nova.



Nesse exemplo foi escolhido um evento que ocorre ao soltar o slider, e a ação escolhida para ocorrer é que uma caixa de diálogo com o valor que foi deixado é exibida.

```
void testeFrame::OnSlider1CmdScrollThumbRelease(wxScrollEvent& event)
{
    wxMessageBox(wxString::Format("Valor do Slider: %d", Slider1->GetValue()));
}
```

## Exemplo prático básico

- Janela gráfica com um slider de 0 a 100 e um botão;
- O usuário marca no slider a nota que ele acredita que um teste de usabilidade System Usability Scale irá dar (hipótese);
- Depois, clica no botão e seleciona o CSV gerado por um formulário de entrevista do SUS;
- O score SUS é calculado pelo programa, e é exibido na interface o resultado, e o quanto a hipótese (marcada no slider) se aproximou do mesmo.

O exercício prático pode ser acessado nesse link:

<https://github.com/EdmilsonOSJr/exercicioPratico>