



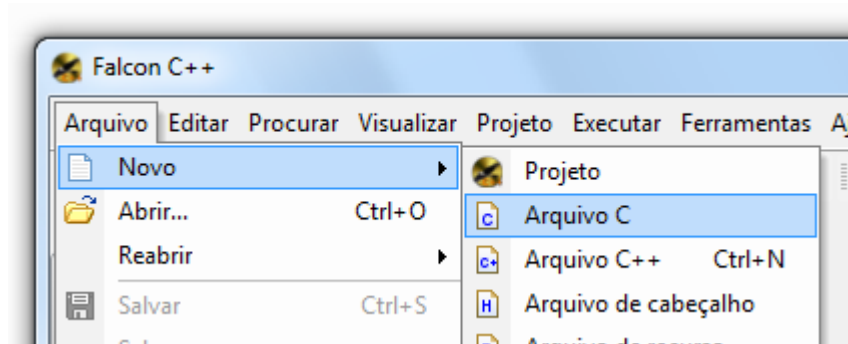
Falcon C++
C++ IDE easy and complete

Guia rápido com as principais funcionalidades

1. [Criando meu primeiro programa.](#)
2. [Compilando e executando.](#)
3. [Completando código e visualizando parâmetros de função.](#)
 - 3.1. [Completando includes, funções e variáveis.](#)
 - 3.2. [Visualizando parâmetros das funções.](#)
 - 3.3. [Inserindo modelos de códigos.](#)
4. [Navegando no código e localizando declarações.](#)
5. [Acompanhando a execução do programa.](#)
 - 5.1. [Inspecionando valor de variáveis.](#)
 - 5.2. [Resolvendo expressões.](#)
6. [Criando projetos.](#)
 - 6.1. [Escolhendo um modelo de projeto.](#)
 - 6.2. [Adicionando arquivos ao projeto.](#)
 - 6.3. [Criando cabeçalhos.](#)
 - 6.3.1. [Criando implementação inicial.](#)
 - 6.4. [Incluindo arquivos do projeto no código.](#)
 - 6.5. [Propriedades do projeto.](#)
 - 6.5.1. [Ajustando ícone e outras informações.](#)
 - 6.5.2. [Informações sobre a versão do executável.](#)
 - 6.5.3. [Escolhendo o compilador.](#)
 - 6.5.4. [Ligando bibliotecas.](#)
 - 6.5.5. [Adicionando diretórios e definindo macros.](#)
7. [Instalando pacotes.](#)
 - 7.1. [Verificando os pacotes instalados.](#)
 - 7.2. [Adicionando novos pacotes.](#)
 - 7.3. [Removendo pacotes.](#)
8. [Atualizando o Falcon C++.](#)
9. [Sites e página no facebook do Falcon C++.](#)

1. Criando meu primeiro programa.

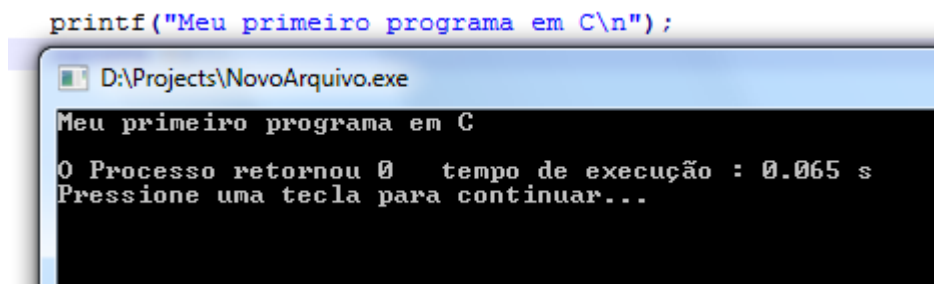
Para criar seu primeiro programa em C, abra o Falcon C++ e clique em Arquivo > Novo > Arquivo C




Agora digite o código abaixo e pressione o botão Compilar e Executar (F9).

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Meu primeiro programa em C\n");
6     return 0;
7 }
```


Executando.



2. Compilando e executando.

Para compilar e executar seu programa clique no botão  da barra de compilação ou pressione **F9**.

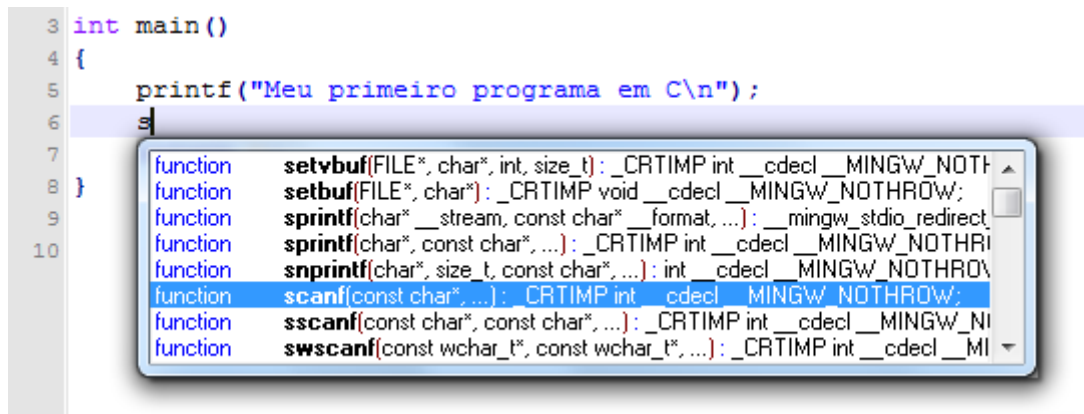
Caso queira apenas gerar o executável clique no botão  ou pressione **Ctrl + F9**.

Ou se preferir apenas executar um programa já compilado, pressione **F5** ou clique no botão .

3. Completando código e visualizando parâmetros de função.

Um recurso muito importante nas IDE's atuais é a completção de código, pois evita erros de digitação e aumenta nossa produtividade.

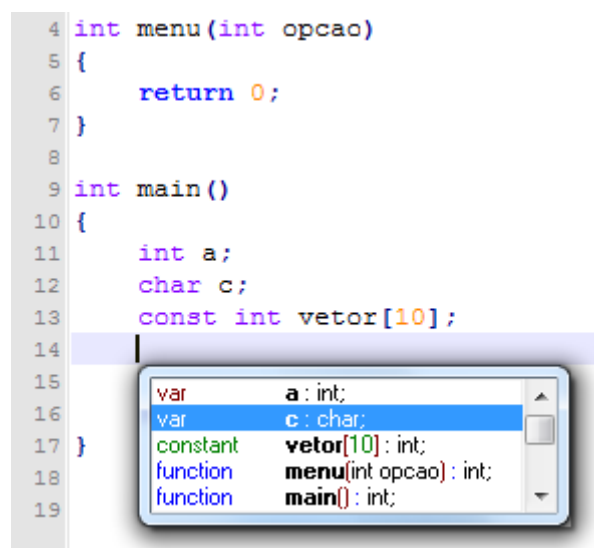
No Falcon C++ toda vez que você digita um `.` (Ponto) ou `->` ou pressione **Ctrl + Espaço** uma lista com todas as variáveis, funções e definições aparece, daí você pode escolher uma opção e pressionar **Enter** para completa-la. Na imagem abaixo é exibido a lista de completção.



3.1. Completando includes, funções e variáveis.

Quando você digitar `#include <` ou `#include "`, uma lista mostrará todos os arquivos que você poderá incluir no seu código fonte.

Quando você não lembrar o nome completo de uma função digite apenas a primeira letra e pressione **Ctrl + Espaço**, isso mostrará todas as declarações que iniciam com o que foi digitado.



3.2. Visualizando parâmetros das funções.

Quando não sabemos quais são e quantos são os parâmetros de uma função nós precisamos verificar a implementação da mesma. Isso atrasa muito no desenvolvimento de uma aplicação.

No Falcon C++ quando você digitar um abre parêntesis, todos os parâmetros da função são exibidos e em qual parâmetro você está no momento.


```
3 int max(int a, int b)
4 {
5     if(a > b)
6         return a;
7     return b;
8 }
9
10 int main()
11 {
12     max(10, 2);
```

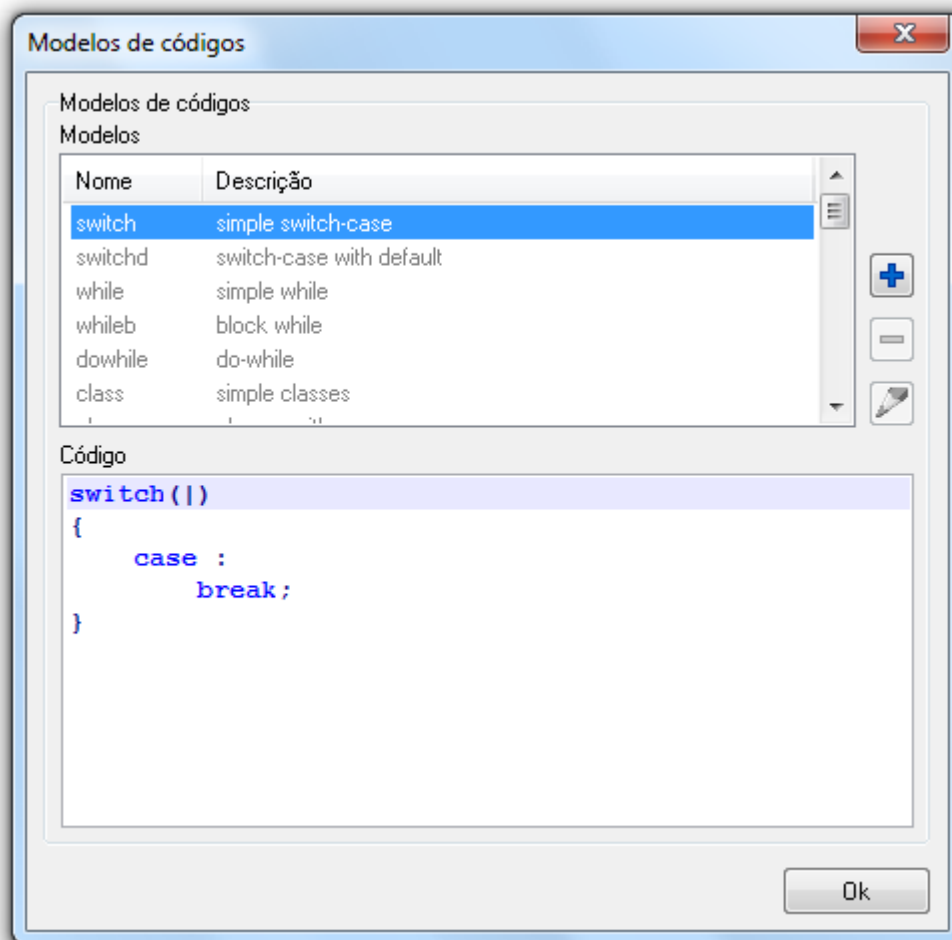
Caso a lista desapareça você pode pressionar **Ctrl + Shift + Espaço** para exibi-la novamente. Caso existam várias funções com o mesmo nome, serão exibidos os parâmetros de todas as funções.


Para visualizar informações de uma declaração como o nome do arquivo, quais os parâmetros, tipo de retorno e em qual linha está declarado um função, variável ou definição, posicione o ponteiro do mouse sobre um termo e espere 1 segundo, será mostrada uma dica de código como mostra na figura abaixo.

```
1 #include <stdio.h>
2 #include <stdio.h>
3
4 int max(int a, int b)
5 {
6     if(a > b)
7         return a;
8     return b;
9 }
10
11 int main()
12 {
13     max(10, 2);
14     func max(int a, int b): int - NovoArquivo.c(4)
15 }
```

3.3. Inserindo modelos de códigos.

Às vezes quando estamos programando digitamos códigos que se repetem em vários lugares. Para facilitar a codificação temos os modelos de códigos, que podemos inserir grandes blocos de códigos com apenas um nome e um comando (**Ctrl + J**). Para criar ou visualizar um modelo de código vá em Ferramentas > Opções do Editor na aba Recursos do código clique no botão  na área Modelos de códigos. A seguinte janela será mostrada.



Existem dois tipos de modelos, os modelos padrões e os que você pode criar. Ao criar um novo modelo você pode informar onde o cursor ficará usando o | (Pipe) e então salve clicando em  e informando um nome e uma descrição.

Usando um modelo: digite parte ou todo o nome de um modelo e pressione **Ctrl + J**.




4. Navegando no código e localizando declarações.

Você pode navegar entre funções de um mesmo arquivo pressionando **Ctrl + Alt + Up** para ir para a função logo acima ou **Ctrl + Alt + Down** para ir para a função logo abaixo.

Para localizar uma declaração segure a tecla **Ctrl** e **clique** sobre o termo a ser encontrado. O termo ficará azul como um link de internet.

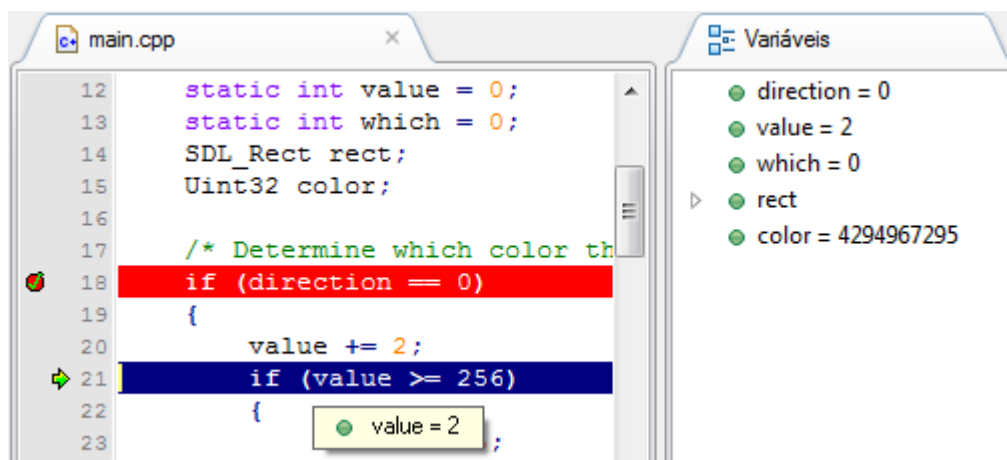
5. Acompanhando a execução do programa.

Programas de computador nem sempre estão livres de erros, um lixo de memória pode causar vários problemas na execução de um programa.

No Falcon C++ você pode acompanhar passo a passo a execução de um programa e encontrar os possíveis problemas caso existam, para isso clique na margem à esquerda na linha em que você deseja acompanhar a execução a partir da li, a linha toda ficará vermelha e um 🍎 será mostrada na margem, então pressione **F9** para iniciar a depuração. Quando a linha que você marcou for executada a execução irá pausar e você verá uma seta verde 🟢, para avançar a execução entrando nas funções clique no botão  ou pressione **F8**, caso não queira entrar nas funções clique no botão  ou pressione **F7**, esses dois comandos avançam linha por linha, caso você queira sair da função em que está executando clique no botão  ou pressione **F6**, e para continuar a execução parando apenas quando surgir outro ponto de parada pressione **F9**.

5.1. Inspeccionando valor de variáveis.

Para visualizar o valor de uma variável basta posicionar o ponteiro do mouse sobre um termo e esperar 1 segundo ou visualizar diretamente no painel à direita os valores de todas as variáveis atuais.



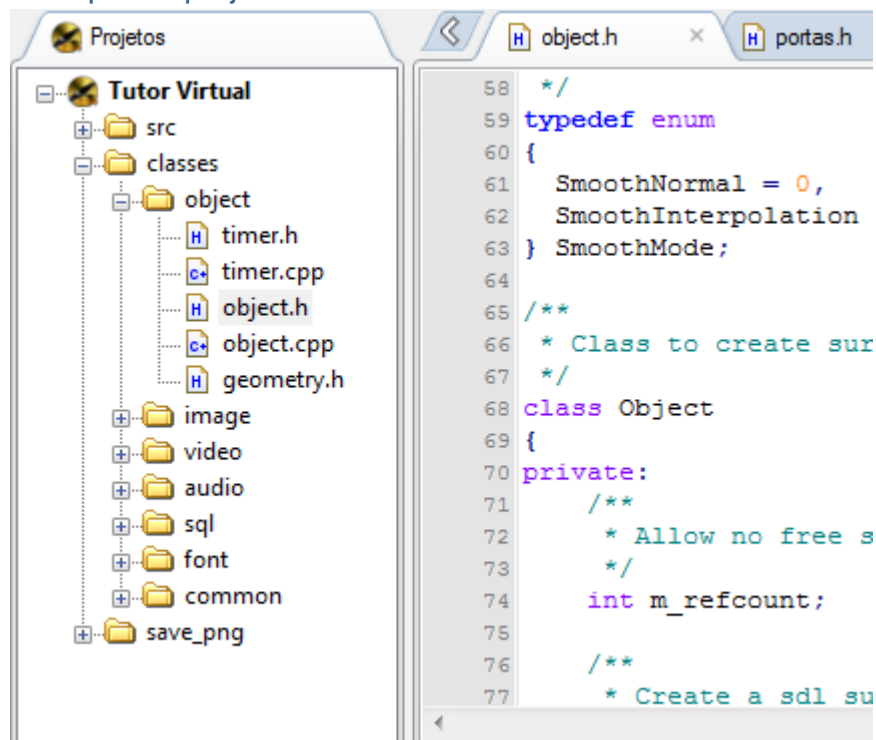
5.2. Resolvendo expressões.

Para visualizar o resultado de uma expressão basta selecioná-la e posicionar o ponteiro do mouse sobre ela e esperar 1 segundo. É possível resolver expressões que executam funções.

6. Criando projetos.

Quando desejamos criar programas maiores, mais planejados e organizados, surge a necessidade de criarmos um projeto, que é um conjunto de arquivos de código fonte que quando compilado formam um executável.

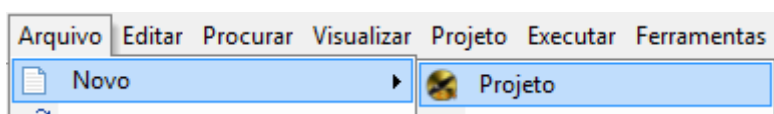
Exemplo de projeto:



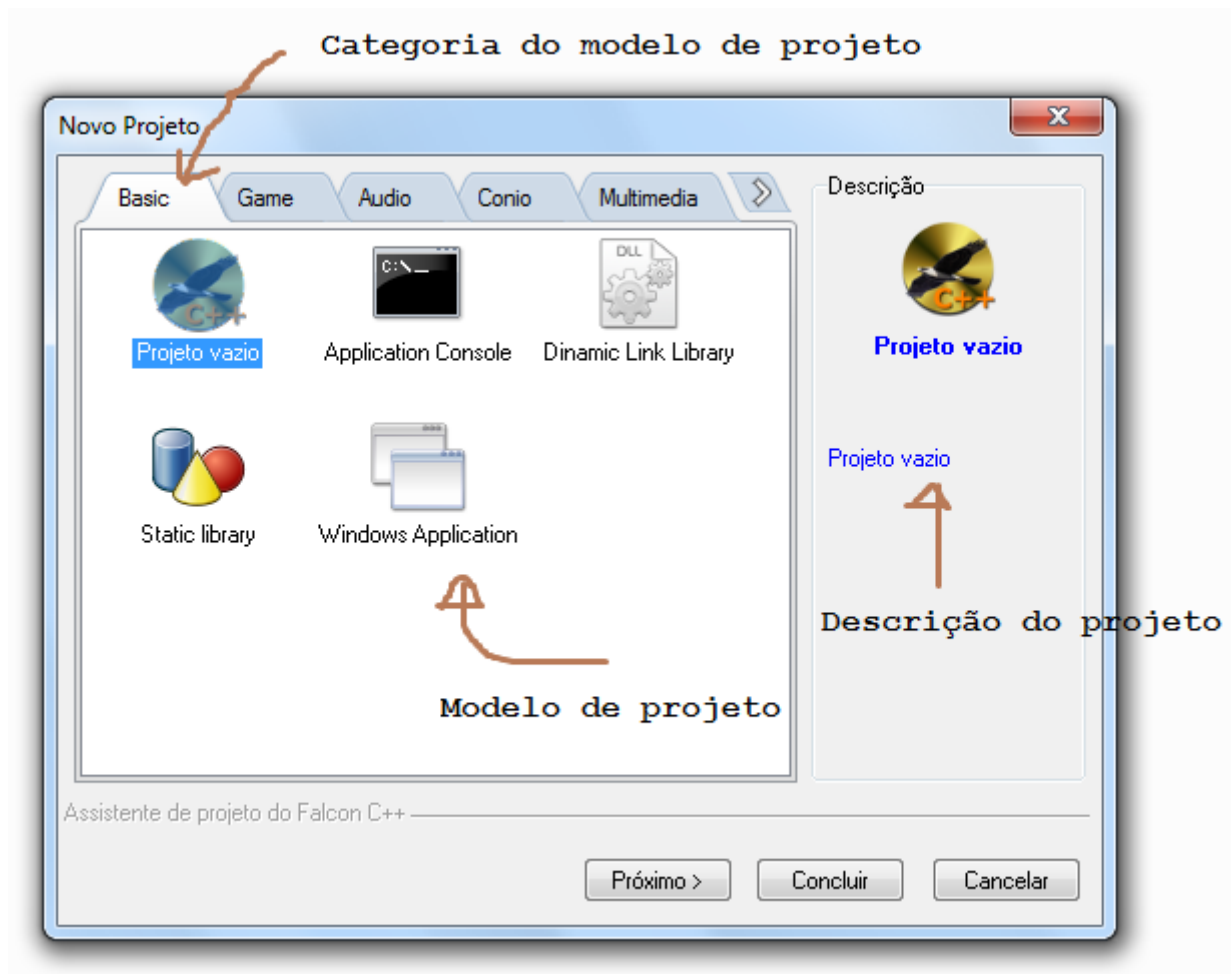
6.1. Escolhendo um modelo de projeto.

Você pode criar um projeto sem nenhum arquivo a princípio ou utilizar um modelo de projeto.

Para criar um novo projeto clique em Arquivo > Novo > Projeto no menu principal.

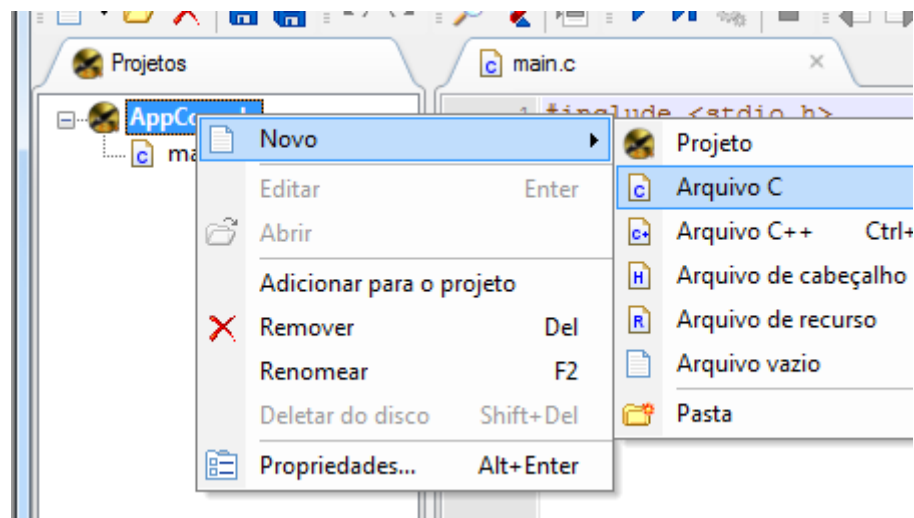


A seguinte janela será exibida:



6.2. Adicionando arquivos ao projeto.

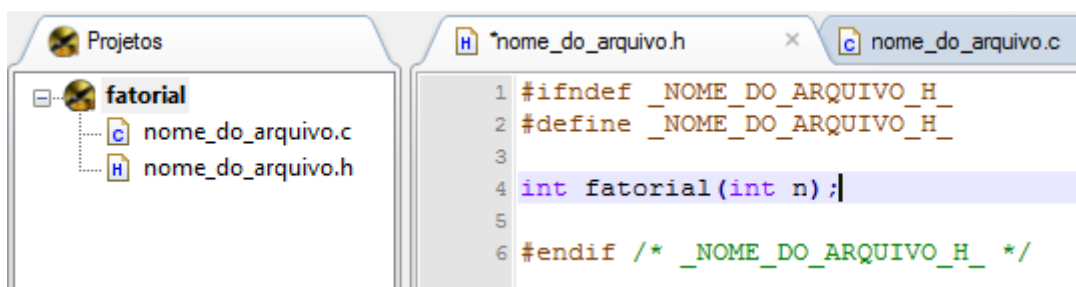
Para adicionar arquivos ao projeto basta selecionar o projeto na aba Projetos e criar um arquivo fonte ou arrastar um arquivo existente em uma pasta e soltar sobre o projeto.



6.3. Criando cabeçalhos.

Quando você usa a função `printf` você precisa incluir o arquivo `stdio.h` isso porque o protótipo da função `printf` foi declarada lá para que possamos usa-la.

Assim como nas bibliotecas padrão nossos projetos também precisam dos arquivos de cabeçalho para conter nossos protótipos de funções. Então você pensa porque não incluir os nossos arquivos.c diretamente? O problema é que assim não podemos incluí-lo em mais de um arquivo porque o código fonte ficaria duplicado. Nos cabeçalhos ou arquivos.h nós devemos incluir as diretivas de não repetição de código. Segue abaixo um modelo.



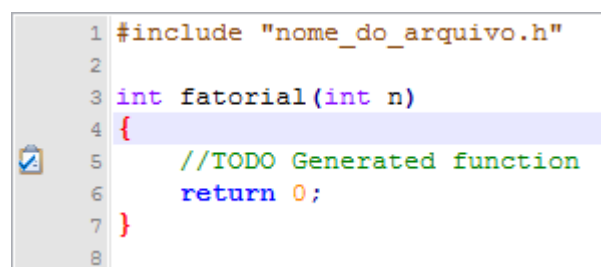
Os códigos em marrom evitam que ao incluir esse arquivos várias vezes, não seja repetido tudo que tem entre essas declarações.

6.3.1. Criando implementação inicial.

O sucesso de um bom projeto é o planejamento logo devemos pensar em quais estruturas e funções devemos implementar. Tudo deve ser feito primeiro no arquivo .h, para só então ser de fato implementado.

O Falcon C++ ajuda na criação do arquivo .c após a criação e elaboração do arquivo .h, para isso abra o arquivo.h e pressione **F11** se o arquivo de implementação(.c) não existir uma pergunta será exibida, responda sim para criar uma implementação inicial.

Exemplo de implementação inicial.




Observe que do lado esquerdo apareceu uma prancheta mostrando que você deve completar a implementação.

6.4. Incluindo arquivos do projeto no código.

Para incluir cabeçalhos use aspas duplas " em vez do < >.
Exemplo:

```
1 #include "meu_arquivo.h"
```

6.5. Propriedades do projeto.

Você pode renomear o projeto, mudar o ícone do executável, informar que a aplicação necessita de permissão de administrador para executar, escolher entre a linguagem C ou C++, entre outros. Para isso selecione seu projeto e clique em  ou pressione **Alt + Enter**.

6.5.1. Ajustando ícone e outras informações.

Você pode deixar sua aplicação mais bonita adicionando um ícone a ela, para isso acesse as propriedades do projeto e na aba Aplicação clique em Mudar o ícone..., escolha um ícone de sua preferência e pressione o botão Ok.

6.5.2. Informações sobre a versão do executável.

Aplicações profissionais exibem informações de versão e descrição do aplicativo, você também pode adicionar essas informações no seu programa, para isso acesse a aba Informações de versão nas propriedades do seu projeto e marque a caixa onde está escrito "Incluir informações de versão no projeto" e em seguida preencha os campos de versão e descrição.

6.5.3. Escolhendo o compilador.


Na aba Compilador você pode definir qual a linguagem do seu projeto escolhendo C, C++ ou RC em Tipo de compilador, RC é um compilador de recursos usado somente para criar arquivos que contêm dados como imagens, janelas e documentos para que possa ser adicionado ao executável.

6.5.4. Ligando bibliotecas.


Às vezes precisamos criar aplicações mais complexas que exigem muito conhecimento e tempo. É aí que surge a necessidade de usarmos bibliotecas, para não termos que reinventar a roda são disponibilizadas várias bibliotecas para os variados fins.



Biblioteca para criação de jogos.

Para adicionar uma biblioteca insira na caixa de texto da aba Bibliotecas que fica localizado em Propriedades do projeto aba Compilador, o prefixo `-l` e em seguida o nome da biblioteca por exemplo: `-ISDL` ou clique em  e escolha o arquivo da biblioteca que deseja.

6.5.5. Adicionando diretórios e definindo macros.

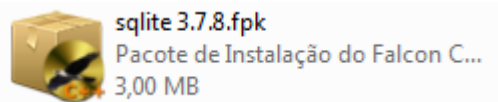
Você pode definir macros pelas propriedades do projeto e incluir pastas que contém arquivos de cabeçalhos que são incluídos no seu projeto, para isso digite o prefixo `-D` e em seguida digite o nome da macro e seu valor, para incluir pastas clique em  e escolha a pasta que deseja adicionar. Tudo isso pode ser feito na aba Incluir diretórios localizado em Propriedades do projeto na aba Compilador.

7. Instalando pacotes.

Pacote é um conjunto de arquivos composto por cabeçalhos, bibliotecas e arquivos de ajuda que fornecem um abstração de funcionalidades complexas de serem implementadas.

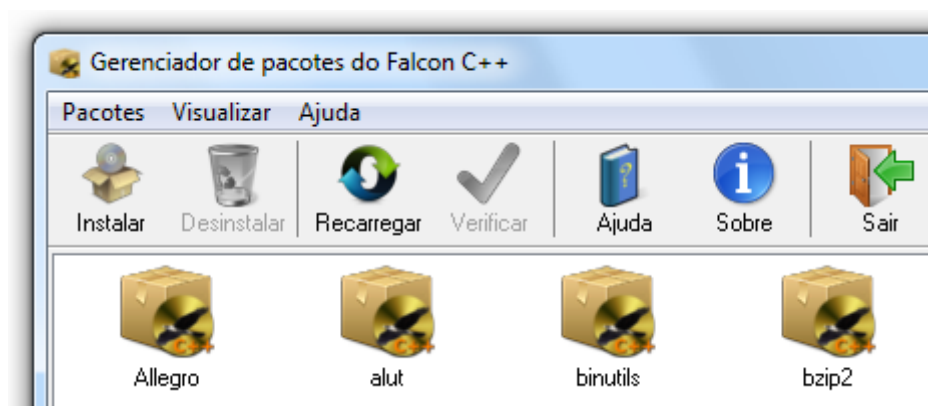
O Falcon C++ aceita dois tipos de pacotes, os DevPack que são pacotes feito para a IDE Dev-C++ e os pacotes FPK que são pacotes nativos da IDE.

Pacote do Falcon C++



7.1. Verificando os pacotes instalados.

Para verificar se um determinado pacote está instalado acesse Ferramentas > Pacotes, será mostrado uma lista com todos os pacotes instalados.

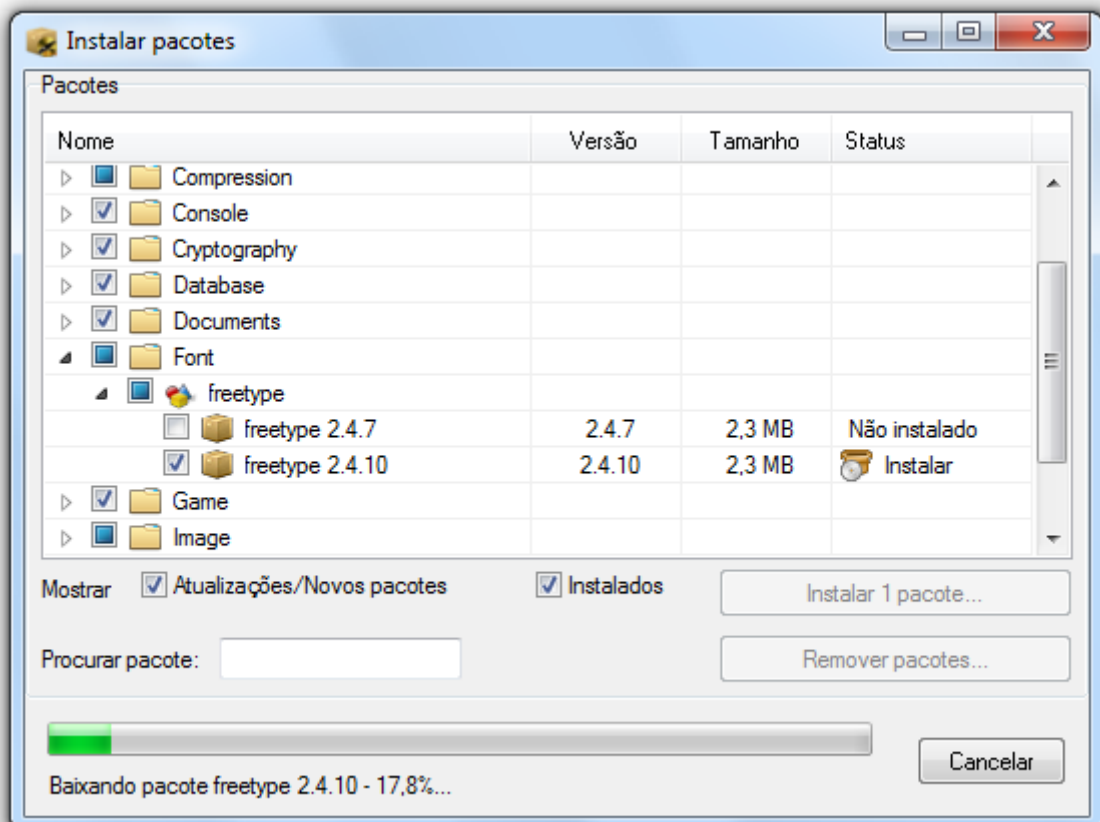


7.2. Adicionando novos pacotes.

Para adicionar um novo pacote você deve estar conectado à internet e abrir o Gerenciador de pacotes (PkgManager) que pode ser acessado em Ferramentas > Pacotes ou pelo menu do sistema. Em seguida clique



em **Instalar** espere até que a lista de pacotes disponíveis seja baixada e escolha um ou mais pacotes para a instalação, em seguida clique em **Instalar** e aguarde até o fim da instalação.



7.3. Removendo pacotes.

Para remover um pacote abra o gerenciador de pacotes, selecione o



pacote que deseja remover e clique em **Desinstalar**, vale ressaltar que um pacote pode ser dependência de outro pacote, sendo assim não será possível remove-lo.

Outra forma de remover um pacote com suas dependências é pelo

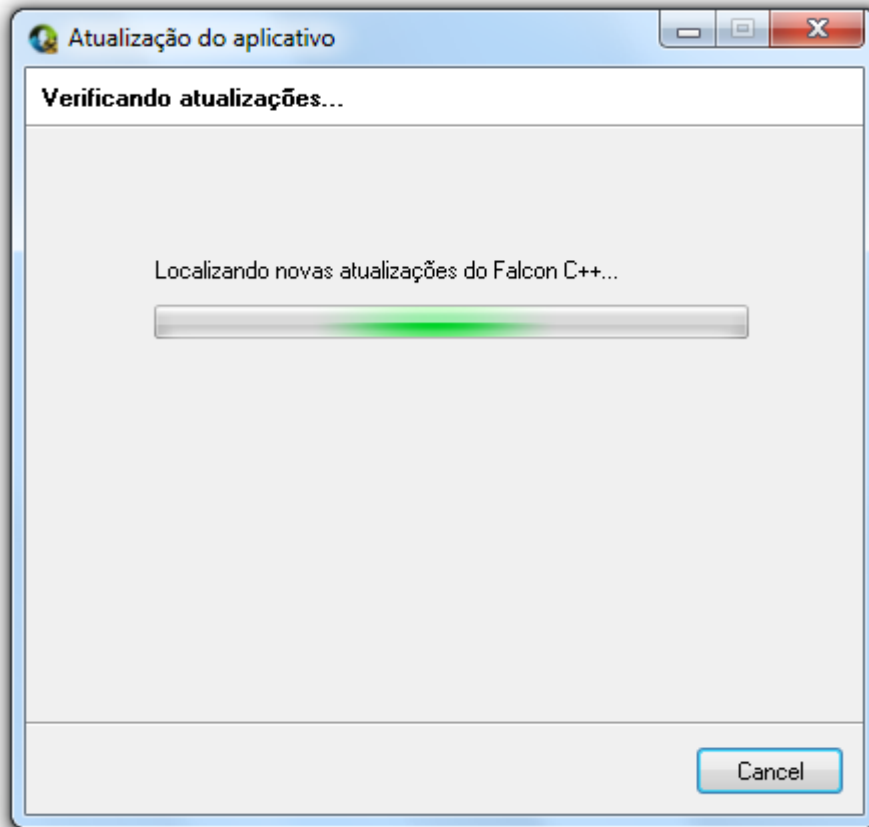


próprio instalar **Instalar**, desmarcando e clicando em remover pacotes.

8. Atualizando o Falcon C++.

Quando uma nova atualização estiver disponível uma janela irá aparecer e então será possível realizar a atualização.

Tela do Updater:



9. Sites e página no facebook do Falcon C++.

Você pode acessar o site do Falcon C++ em falconcpp.sourceforge.net e curtir a página do facebook <http://www.facebook.com/falconcpp> além de tirar suas dúvidas, fazer comentários, elogios ou críticas.