

OBJETIVOS DO CAPÍTULO

- Conceitos de: linguagem de programação, FORTRAN, comando, instrução, projeto, programa-fonte, programa-objeto, programa-executável e compilação.
- Procedimentos básicos para programar em linguagem FORTRAN usando o aplicativo Fortran PowerStation 4.0.
- Comandos do FORTRAN a usar: WRITE e END

1.1 USO DESTES TEXTO EM DISCIPLINAS

O objetivo deste texto é ensinar a implementar programas com a linguagem FORTRAN 90 e, implicitamente, a desenvolver algoritmos para resolver problemas básicos de matemática e física. Os capítulos 1 a 12 deste texto são recomendados para disciplinas introdutórias de programação em cursos de engenharia. Já os capítulos 13 em diante podem ser usados em cursos avançados de programação em nível de graduação.

Cada capítulo foi estruturado para ser lecionado em aulas com duração aproximada de 1h30m. Uma disciplina de 30 horas foi ministrada para calouros, por mais de dez anos, no curso de graduação em engenharia mecânica da Universidade Federal do Paraná (UFPR). Esta disciplina tinha 15 aulas envolvendo os capítulos 1 a 12, uma prova simulada e mais duas provas valendo nota. Outra disciplina envolvendo os capítulos 1 a 20, com 45 horas, vem sendo lecionada há 20 anos em nível de pós-graduação, nos programas de engenharia mecânica (PGMec) e métodos numéricos em engenharia (PPGMNE) da UFPR. Em 2024 e 2025, a apostila foi revisada e acrescentados mais alguns capítulos.

1.2 BIBLIOGRAFIA

Não é objetivo deste texto entrar em detalhes sobre hardware e software de computadores. Aos interessados nestes aspectos recomenda-se, por exemplo, o livro de Ramalho (2000), que é adotado neste texto como referência para termos técnicos de informática em português. Este texto também não aborda teoria sobre algoritmos; isto será visto de forma implícita através dos programas-exemplo e exercícios. Dois textos recomendados sobre algoritmos são o livro de Forbellone e Eberspacher (2000) e a apostila de [Moraes \(2000\)](#). O livro de Adams et al. (1997) é um manual completo sobre a linguagem FORTRAN 95, padronizada pelas organizações ISO (Organização Internacional para Padronização) e ANSI (American National Standards Institute). Já o livro de Chivers e Sleightholme (2012) aborda diversas versões do FORTRAN: 90, 95, 2003, 2008 e 77.

1.3 FORTRAN

Existe uma infinidade de linguagens de programação, por exemplo, Assembly, Cobol, Visual Basic, Pascal, Java, C++ etc (Ramalho, 2000, p. 114 e 124-30). Elas são usadas para criar os programas, também denominados de softwares, códigos ou aplicativos. Os programas são usados para fazer os computadores realizarem tarefas. A utilidade das linguagens de programação pode ser percebida ao se lembrar para que são usados, por exemplo: os sistemas operacionais Windows, Linux e Android; os aplicativos Word, Excel, PowerPoint e Acrobat; os navegadores da internet Chrome e Firefox; os programas dos caixas automáticos dos bancos; e os aplicativos de transporte e alimentação.

A linguagem FORTRAN, contração de FORMula TRANslation (Tradução de Fórmula), ainda é a principal linguagem de programação usada em engenharia e aplicações científicas (Decyk et al., 2007) que demandam muito esforço computacional (memória e tempo de CPU), e é fácil de aprender e usar. Foi criada em 1954 por John Backus da IBM (International Business Machines). Sua primeira versão comercial surgiu em 1957, seguida por algumas outras não padronizadas.

As versões do FORTRAN padronizadas pela ANSI e ISO são:

- 1) Ano 1966 = FORTRAN 66 ou IV
- 2) Ano 1978 = FORTRAN 77
- 3) Ano 1992 = FORTRAN 90
- 4) Ano 1997 = FORTRAN 95
- 5) Ano 2004 = FORTRAN 2003
- 6) Ano 2010 = FORTRAN 2008
- 7) Ano 2018 = FORTRAN 2018
- 8) Ano 2023 = FORTRAN 2023

1.4 CONVENÇÕES

Para melhor entendimento deste texto, as seguintes convenções são adotadas:

- 1) São mostrados na cor azul links para arquivos disponíveis na internet.
- 2) Comandos de programas ou aplicativos que devem ser executados ao longo da leitura deste texto são mostrados em negrito. A vírgula é usada para separar uma sequência de comandos.
- 3) Os programas-fonte, que serão definidos ainda neste capítulo, são apresentados na cor vermelha, dentro de tabelas.
- 4) Qualquer referência ao sistema operacional Windows é abreviada apenas para Windows.
- 5) Qualquer referência ao aplicativo Fortran PowerStation 4.0, da Microsoft, é abreviada para Fortran, em minúsculo, exceto a primeira letra.
- 6) Qualquer referência à linguagem computacional FORTRAN é abreviada para FORTRAN, tudo em maiúsculo, para distinguir do aplicativo Fortran PowerStation 4.0.

- 7) Devido à grande quantidade de termos em inglês, não é feito qualquer distinção, em termos de grafia, para palavras em português.

1.5 INICIALIZAÇÃO DO APLICATIVO FORTRAN POWERSTATION 4.0

Para seguir este texto, não é necessário nenhum conhecimento anterior sobre linguagem de programação. É necessário apenas ter conhecimentos básicos sobre o sistema operacional Windows. Todos os programas em FORTRAN, apresentados neste texto, foram editados e compilados com o aplicativo Fortran PowerStation 4.0, da Microsoft, exceto se mencionado algo em contrário. Além de seus manuais, este aplicativo contém o livro de Hahn (1994) sobre FORTRAN 90 e o item Reference onde são descritos todos os comandos do FORTRAN, incluindo exemplos.

Conforme é mostrado na Figura 1.1, para iniciar o uso do aplicativo Fortran PowerStation 4.0, deve-se ativá-lo com o nome Microsoft Developer Studio. Para facilitar a edição dos programas, recomenda-se maximizar a janela principal do Fortran, clicando sobre o ícone central que fica no canto superior direito. Fazendo isso, o aspecto do Fortran deverá ser igual àquele mostrado na Figura 1.2.

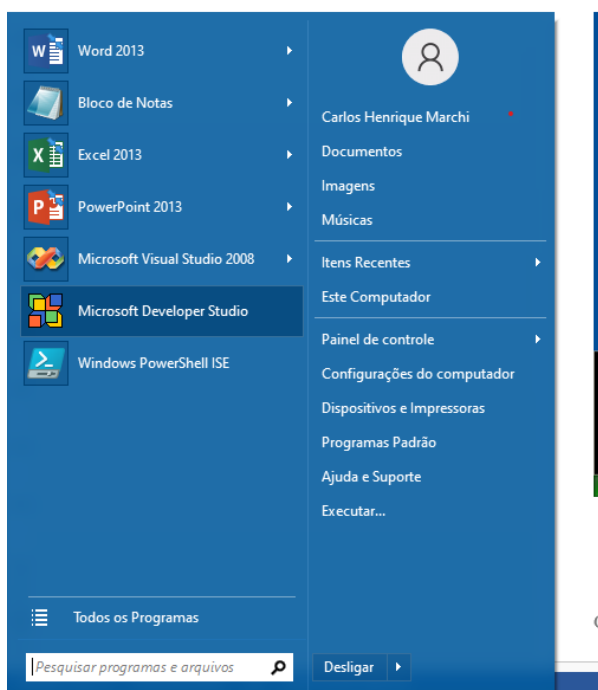


Figura 1.1 Inicialização do aplicativo Fortran PowerStation 4.0 no Microsoft Developer Studio.

Na janela do Fortran, mostrada na Figura 1.2, pode-se perceber

- 1) No título, o ícone do Fortran e o nome Microsoft Developer Studio.
- 2) O menu do Fortran, ou seja, as opções existentes para usar este aplicativo, que são: File, Edit, View, Insert, Build, Tools, Window e Help.
- 3) A janela do Fortran é dividida em três partes ou subjanelas. Se não estiver aparecendo a subjanela inferior, mostrada na Figura 1.2, deve-se executar: **View, Output**.

- 4) Na subjanela do lado esquerdo, tem-se uma lista dos manuais do Fortran, destacando-se: Reference, que é uma lista de comandos do FORTRAN 90, incluindo explicações e exemplos de uso; e Fortran 90 for Scientists and Engineers, que é um livro sobre a linguagem FORTRAN 90. Esta subjanela também é usada para mostrar os nomes dos programas editados na linguagem FORTRAN. Quando um programa estiver sendo editado, para acessar os manuais, deve-se clicar sobre a aba ? **InfoView**, mostrada na Figura 1.2, que fica na parte inferior da subjanela do lado esquerdo. E para voltar ao programa que está sendo editado, basta clicar sobre a aba **FileView**, que aparecerá quando algum programa estiver sendo editado (ver a Figura 1.9).
- 5) Na subjanela inferior são apresentadas informações sobre erros de programação e compilação. Detalhes serão mostrados no capítulo 2.
- 6) A subjanela do lado direito, a maior das três, é usada para apresentar o conteúdo dos manuais e editar programas em FORTRAN.

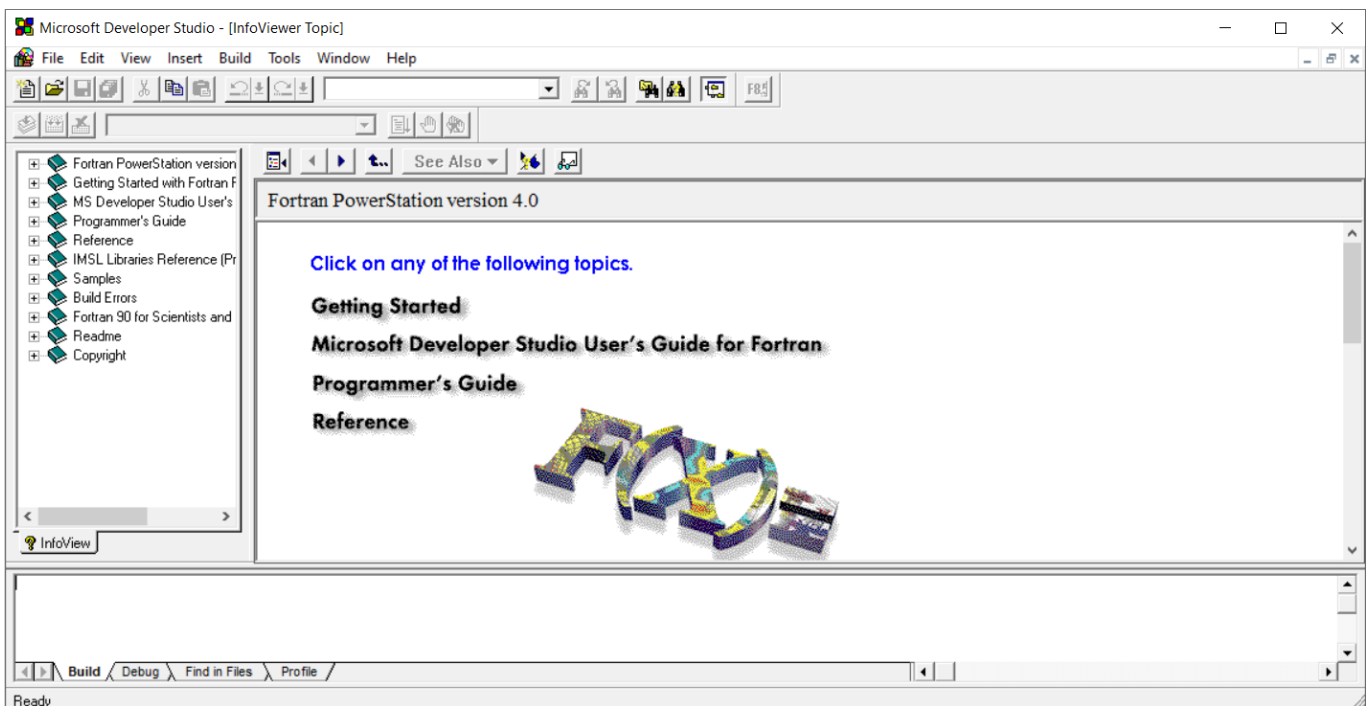


Figura 1.2 Janela do aplicativo Fortran PowerStation 4.0.

1.6 CRIAÇÃO DE UM PROJETO DO TIPO CONSOLE APPLICATION

O Fortran organiza todos os arquivos de um programa em uma entidade chamada projeto. Em termos práticos, a criação de um projeto resulta na criação de uma pasta no Windows. Existem diversos tipos de projetos que podem ser criados com o Fortran. A escolha do tipo depende da finalidade do programa e dos recursos do Windows que se deseja empregar. No caso deste texto, nos capítulos 1 a 22 são empregados apenas projetos do tipo Console Application, que é o mais simples.

Para criar um projeto do tipo Console Application, devem ser executados os seguintes passos:

- 1) No menu do Fortran, conforme é mostrado na Figura 1.3: **File, New**.

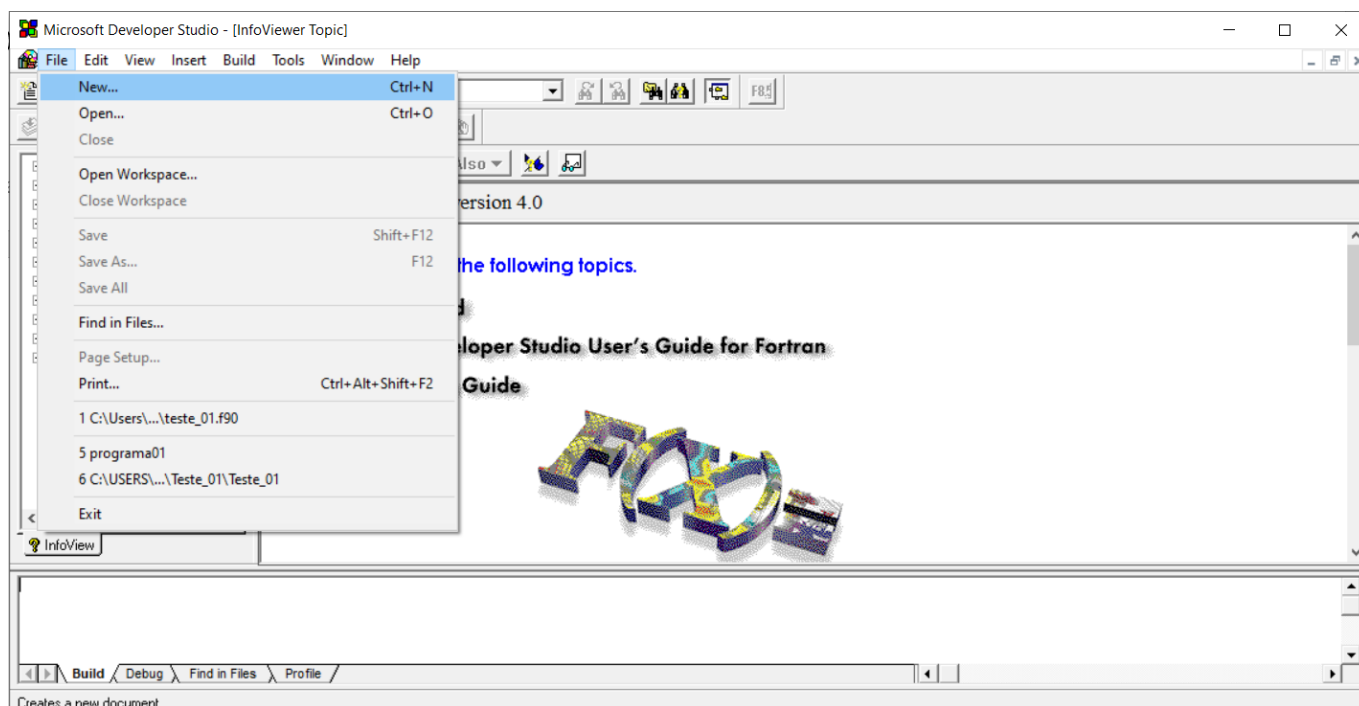


Figura 1.3 Início da criação de um projeto no Fortran.

- 2) Na janela New, mostrada na Figura 1.4, clicar sobre a opção **Project Workspace**. Depois, clicar sobre o botão **OK**.

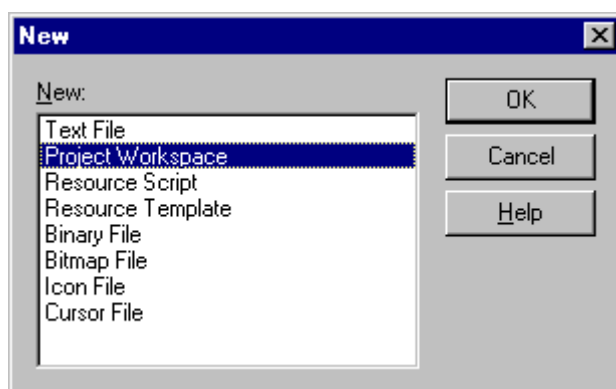


Figura 1.4 Segundo passo na criação de um projeto no Fortran.

- 3) Na janela New Project Workspace, mostrada na Figura 1.5, é necessário executar o que segue:
- Entre os seis tipos de projetos apresentados na opção Type, clicar sobre **Console Application** no caso deste tipo de projeto não estar selecionado (realçado em azul).
 - Na opção Location estará indicada uma pasta default, ou seja, uma pasta escolhida pelo próprio Fortran em função de sua configuração atual. É possível que não seja a pasta desejada. Neste caso, deve-se clicar sobre o botão **Browse** para indicar a pasta dentro da qual se pretende criar o projeto. Ao se fazer isso, surgirá a janela Choose Directory mostrada na Figura 1.6, na qual deve-se executar:
 - Na opção Drives, clicando sobre o triângulo invertido, escolher o drive desejado. No caso deste texto, será usado **C:**
 - Na opção Directory name, selecionar a pasta desejada. No caso deste texto, será usado **C:\MSDEV\Projects**, conforme é mostrado na Figura 1.7.

- Clicar sobre o botão **OK**.
- c) Estará de volta a janela New Project Workspace, mostrada na Figura 1.5. Clicar dentro do espaço de edição da opção Name. Escrever o nome do projeto, por exemplo, **programa01**. Depois disso, a janela New Project Workspace deverá ficar conforme a Figura 1.8. Deve-se perceber que o nome do projeto, digitado na opção Name, aparece automaticamente na opção Location. Em seguida, deve-se clicar sobre o botão **Create**. Após isso, o Fortran criará uma pasta com o nome do projeto indicado, dentro da pasta já selecionada, conforme é mostrado na Figura 1.9.

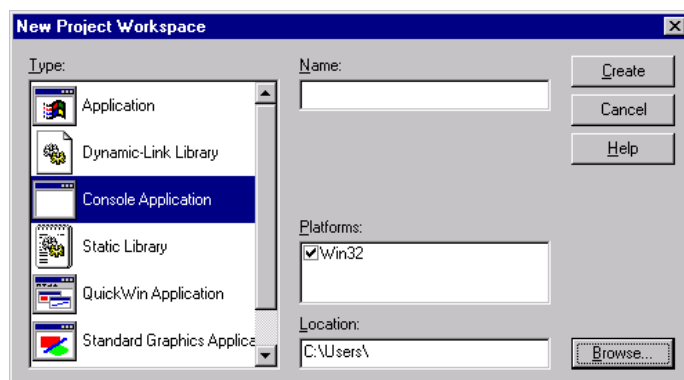


Figura 1.5 Janela para definição do tipo, pasta e nome do projeto.

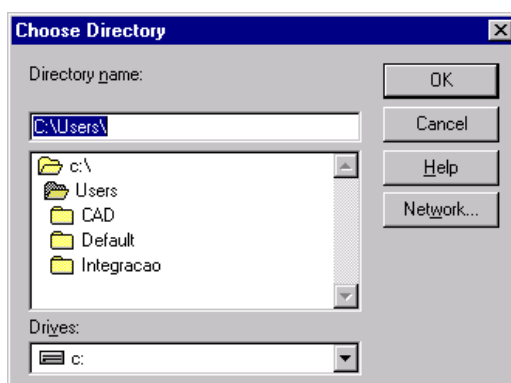


Figura 1.6 Definição do drive do projeto.

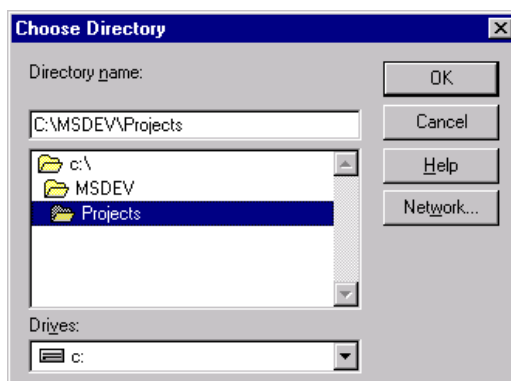


Figura 1.7 Definição da pasta para criação do projeto.

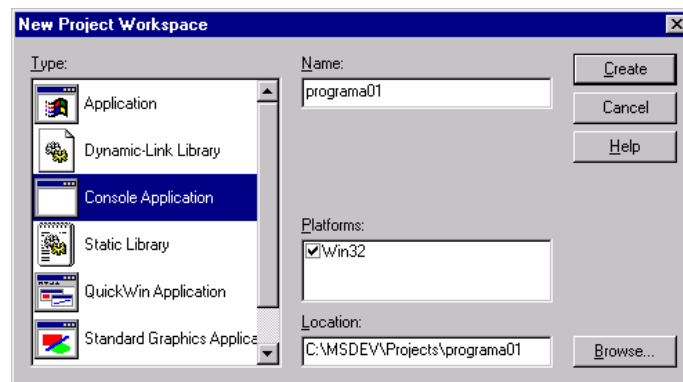


Figura 1.8 Tipo, nome e pasta do projeto já definidos.

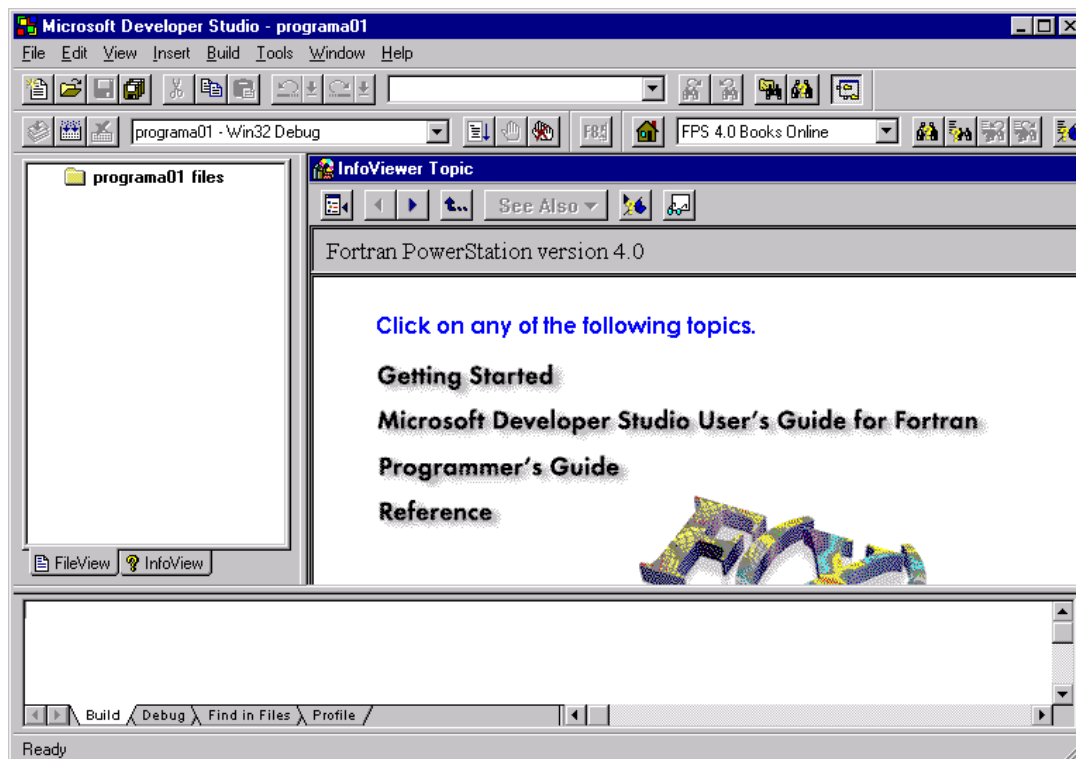


Figura 1.9 Fortran com o nome da pasta criada para o projeto especificado.

1.7 CRIAÇÃO E INSERÇÃO DO PROGRAMA-FONTE DENTRO DO PROJETO

O processo de edição ou criação de um programa é semelhante à redação de uma carta com o aplicativo Word, que é um editor de textos. São necessários dois componentes: um editor de textos para escrever o programa computacional e um arquivo para guardar o texto no disco rígido (HD) do computador ou outro dispositivo. No caso dos programas computacionais, a carta é chamada de programa-fonte e pode ser aberta, editada ou redigida dentro do Fortran ou em vários outros editores de texto.

Ao longo deste texto, as palavras edição, digitação, escrita e implementação serão usadas como sinônimos. Para iniciar a edição ou escrita de um programa-fonte em linguagem FORTRAN é necessário criar e inserir um arquivo dentro do projeto. Isso é feito através dos seguintes passos:

- 1) No menu do Fortran, conforme é mostrado na Figura 1.10, executar: **Insert, Files into Project...**
- 2) Na janela Insert Files into Project, mostrada na Figura 1.11, executar o seguinte:

a) Na opção File Name, digitar o nome do programa-fonte a ser criado; por exemplo, como mostrado na Figura 1.12, **programa01.f90**

Sempre deve-se usar f90 como extensão dos nomes dos arquivos do tipo programa-fonte. Ele indica que o programa está ou será escrito na linguagem FORTRAN 90.

b) Clicar sobre o botão **OK**

3) Na janela Microsoft Developer Studio, mostrada na Figura 1.13, clicar sobre o botão **Sim**

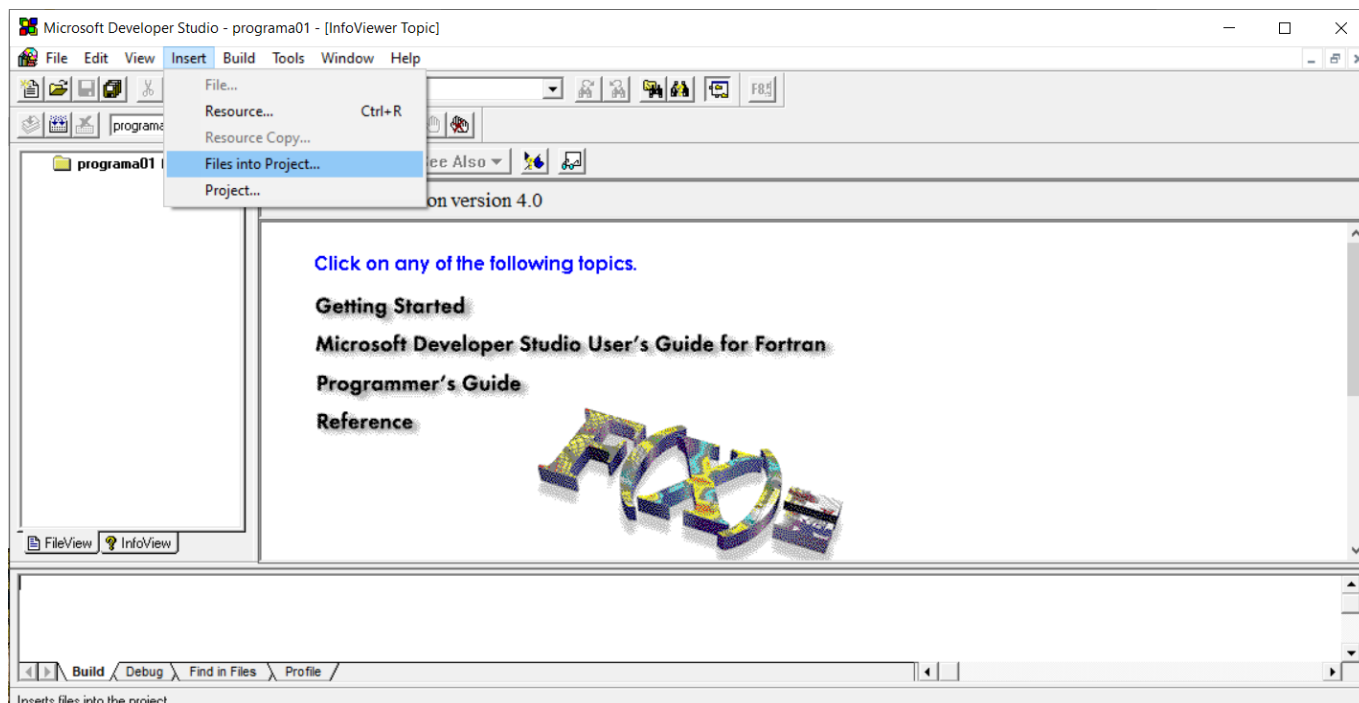


Figura 1.10 Início da criação do programa-fonte.

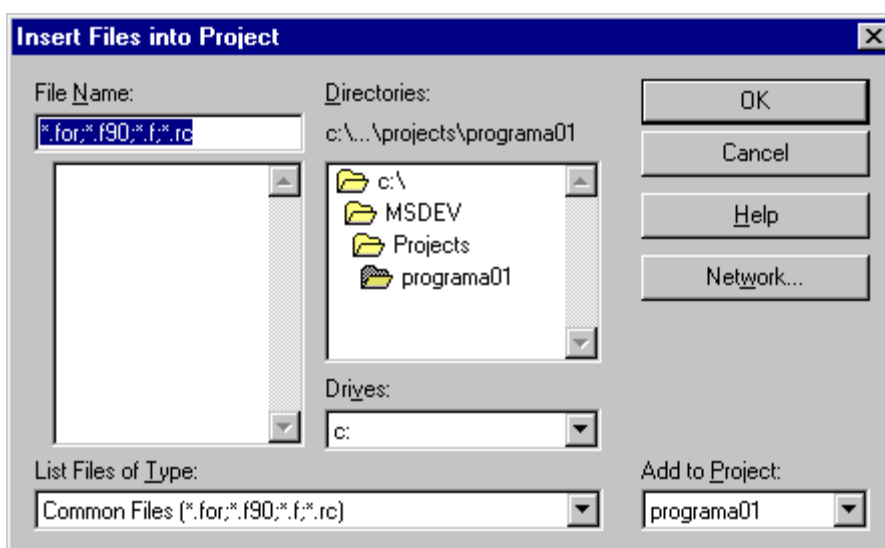


Figura 1.11 Janela para definição do nome do programa-fonte a inserir no projeto.

4) Na subjanela do lado esquerdo do Fortran, clicar sobre o sinal + ao lado do nome do projeto; deverá aparecer o nome do programa-fonte que foi inserido no projeto; no caso, programa01.f90

5) Dar duplo clique sobre o nome do programa-fonte que foi inserido

- 6) Na janela Microsoft Developer Studio, mostrada na Figura 1.14, clicar sobre o botão **Sim**
- 7) O programa-fonte inserido no projeto já está pronto para ser editado na subjanela do lado direito do Fortran, conforme mostrado na Figura 1.15; os dois avisos de erro que aparecem podem ser desconsiderados.

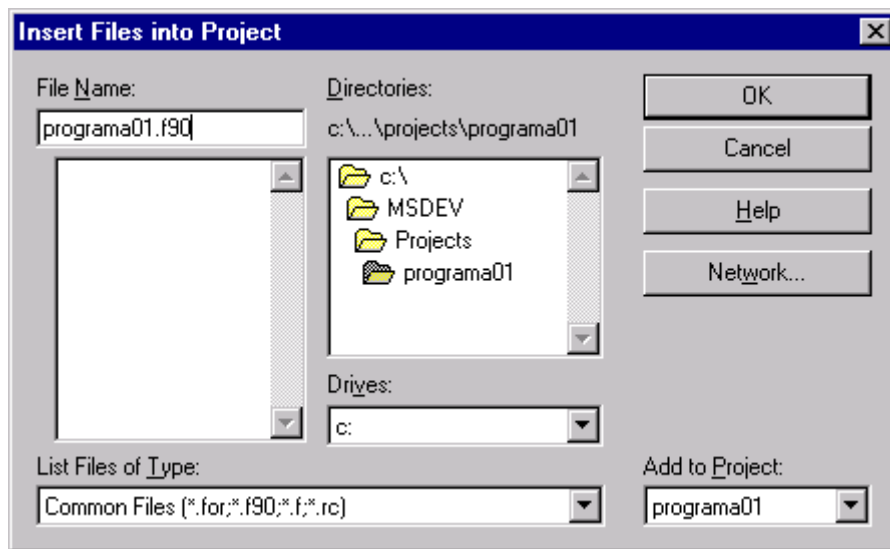


Figura 1.12 Nome do programa-fonte já definido.

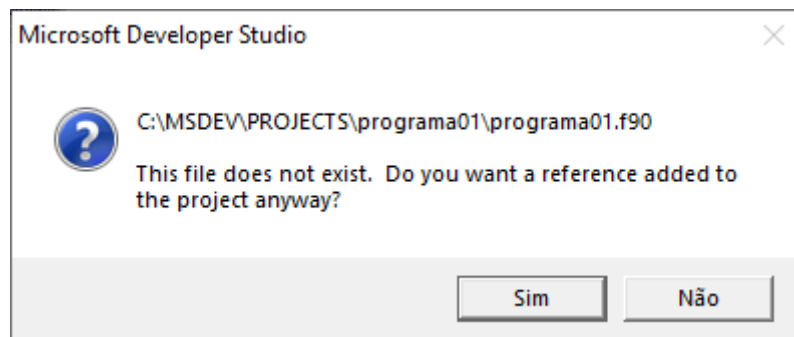


Figura 1.13.

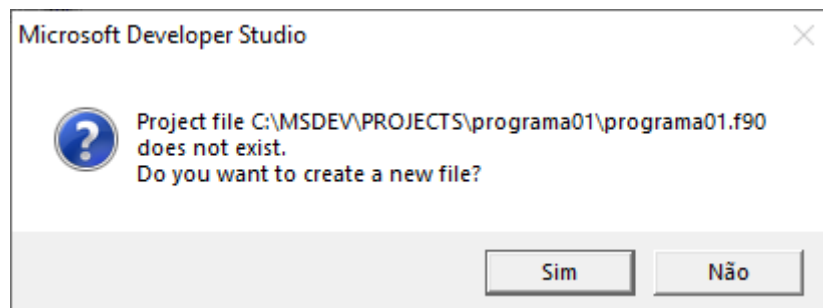


Figura 1.14.

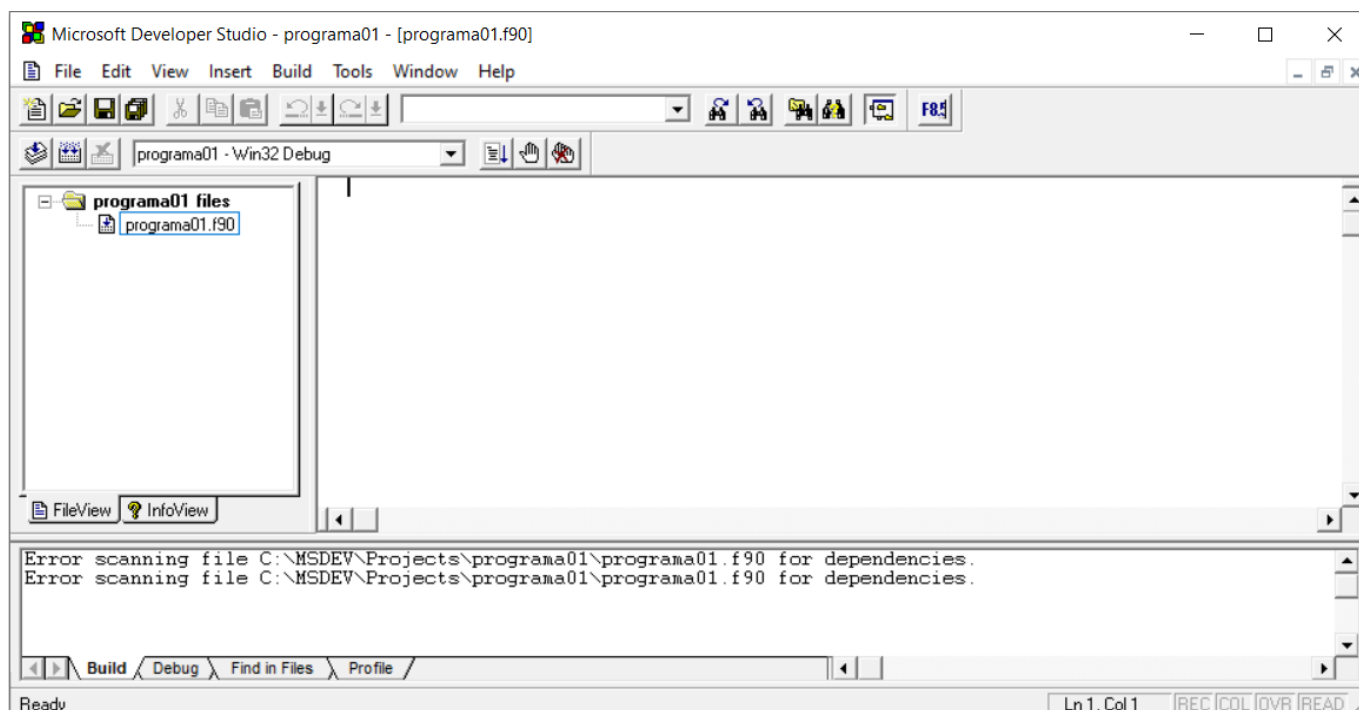


Figura 1.15 Fortran pronto para edição do programa-fonte.

1.8 EDIÇÃO DO PROGRAMA-FONTE

Um exemplo de edição de programa-fonte é apresentado a seguir:

- 1) Dentro do espaço de edição do Fortran, na subjanela maior, copiar exatamente o texto mostrado abaixo em vermelho.

Tabela 1.1 programa01.f90

```
WRITE(*,*) "Meu primeiro programa em FORTRAN 90"
WRITE(*,*) "Curso da UFPR"
END
```

- 2) Para salvar este texto, no menu do Fortran, executar: **File, Save**. Em seguida, na extremidade inferior da janela do Fortran, surgirá uma mensagem informando todo o caminho da pasta do projeto e os nomes do projeto e do programa-fonte que foi salvo, como mostrado na Figura 1.16. Quando há um asterisco à direita do nome do programa-fonte na extremidade superior da janela do Fortran, significa que a versão atual do programa-fonte não está salva.
- 3) Comentários sobre este programa:
 - a) Dentro do editor de textos do Fortran, os comandos da linguagem FORTRAN são mostrados na cor azul, conforme pode ser visto na Figura 1.16. E o que é criado ou definido pelo programador é mostrado na cor preta. Comandos são palavras-chave que têm uma função específica e devem ser usados exatamente de acordo com suas normas de aplicação; estas normas são chamadas de sintaxe do comando.
 - b) Existem dezenas de comandos na linguagem FORTRAN. No caso do primeiro programa-fonte, mostrado na Figura 1.16, são usados apenas dois comandos: WRITE e END.

- c) Como será visto após a execução deste programa, ele simplesmente irá mostrar, dentro de uma janela, duas linhas de texto: na primeira, Meu primeiro programa em FORTRAN 90; e, na segunda, Curso da UFPR. Isso é conseguido com o comando WRITE. Qualquer texto colocado entre aspas após o comando WRITE(*,*) é escrito na janela de saída. As aspas podem ser substituídas por apóstrofes.
- d) O comando END é necessário para indicar o fim do programa.
- e) Cada linha do programa-fonte é chamada de instrução. Uma instrução executa uma parte do algoritmo do programa.

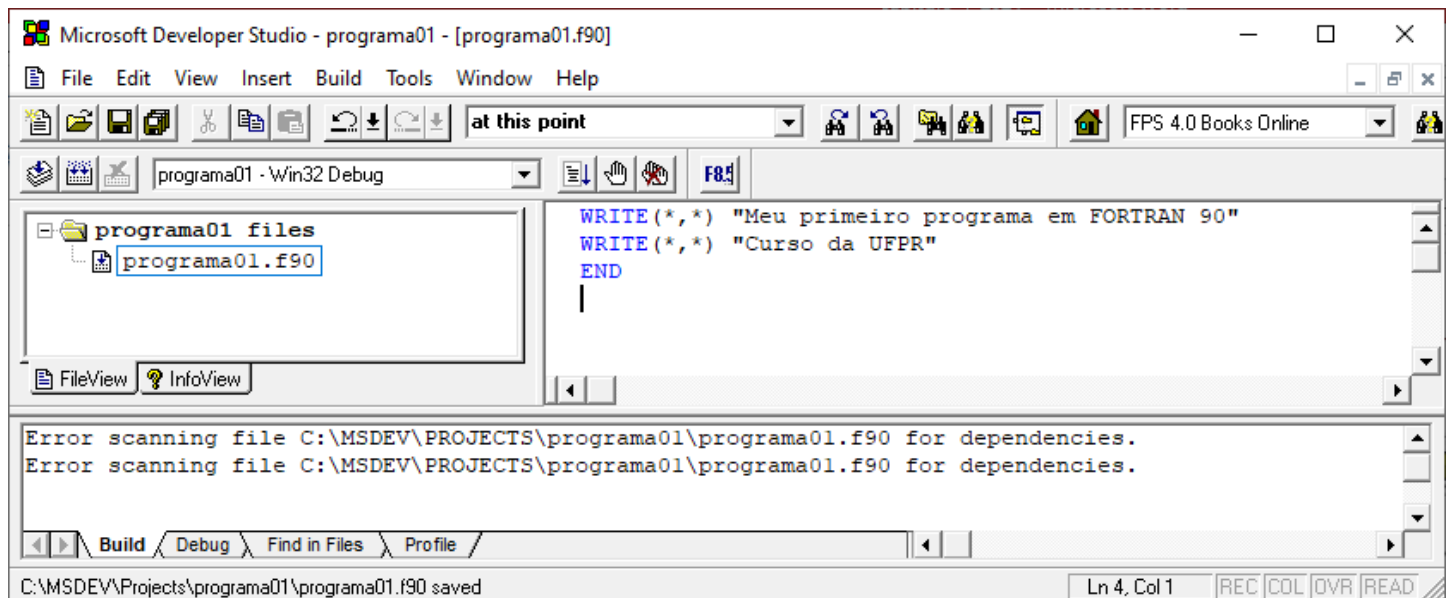


Figura 1.16 Programa-fonte escrito na linguagem FORTRAN.

1.9 COMPILAÇÃO E GERAÇÃO DO PROGRAMA-OBJETO

Após concluir a edição de um programa-fonte, ele deve ser compilado. O processo de compilação traduz o programa-fonte (que é um arquivo do tipo texto, isto é, que as pessoas conseguem ler naturalmente, como o programa01.f90) no chamado programa-objeto, que é um arquivo do tipo binário (escrito na linguagem dos computadores, usando apenas os algarismos zero-0 e um-1).

Uma linguagem de programação de alto nível (Ramalho, 2000), como o FORTRAN, é uma forma “amigável” ou relativamente fácil de se escrever programas. O aplicativo Fortran PowerStation 4.0, além do editor de programas, também tem junto um compilador de programas. Para compilar um programa-fonte que está aberto no editor de textos do Fortran, inserido em um projeto, deve-se executar o seguinte no menu do Fortran:

Build, Compile

Após a compilação, conforme é mostrado na Figura 1.17, aparecerão três linhas com mensagens na subjanela da parte inferior do Fortran:

- 1) A primeira informa que o Fortran está compilando (Compiling) o programa-fonte.
- 2) A segunda informa o nome do programa-fonte que está sendo compilado e a pasta na qual ele se encontra.

3) Finalmente, a terceira informa o nome do programa-objeto (no caso, programa01.obj) que foi gerado e o número de erros (errors) e de avisos (warnings) relativos ao programa-fonte compilado. Se houver algum erro, o programa-objeto não é gerado. Avisos não impedem a geração do programa-objeto. O nome do programa-objeto é igual ao nome do programa-fonte mas com extensão obj.

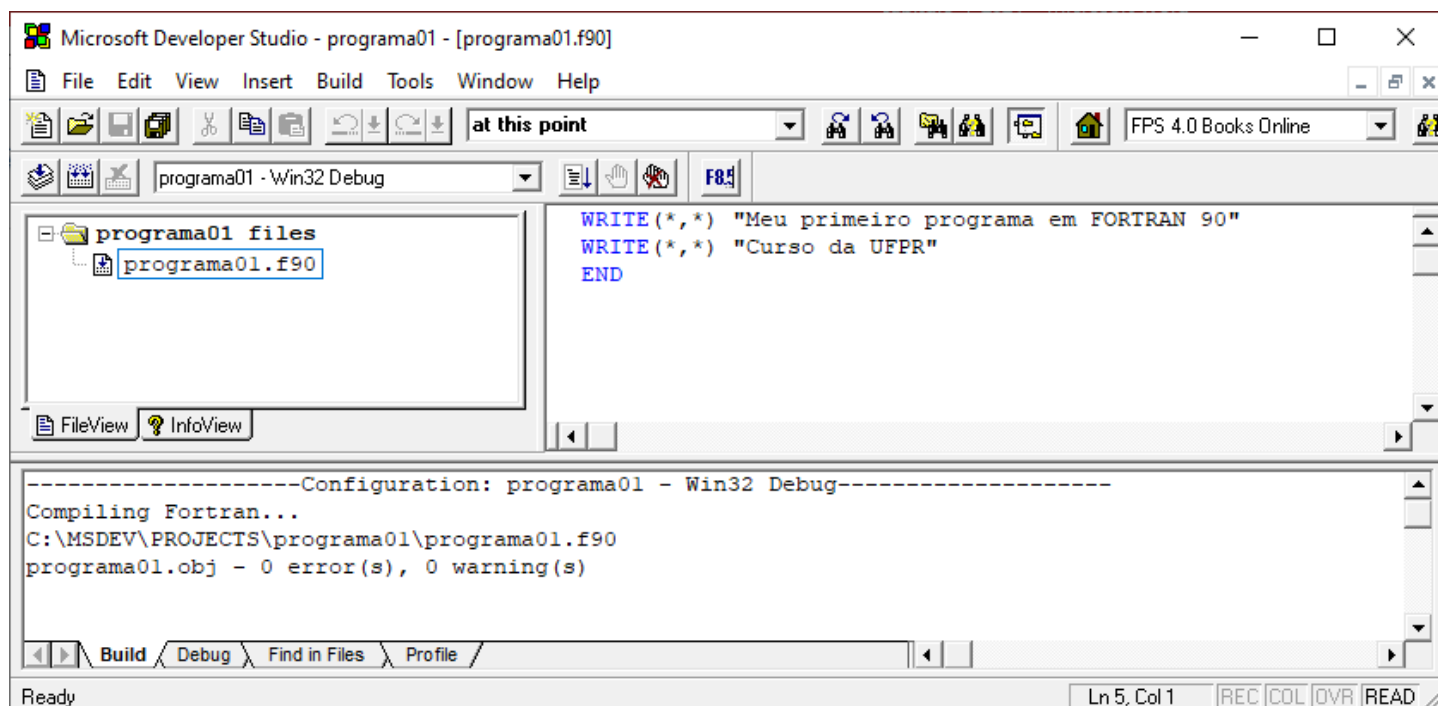


Figura 1.17 Compilação do programa-fonte.

A compilação cria automaticamente uma subpasta chamada Debug dentro da pasta do projeto. Dentro desta subpasta é que se encontra o programa-objeto bem como outros arquivos auxiliares. Pode-se ver isso com o aplicativo Windows Explorer, que é o navegador de arquivos do Windows.

1.10 GERAÇÃO DO PROGRAMA-EXECUTÁVEL

Após o processo de compilação, é necessário gerar o programa-executável, que é um arquivo possível de ser executado ou “rodado” pelo computador, como qualquer outro programa ou aplicativo; por exemplo, os aplicativos Word, Excel, PowerPoint e Acrobat. O programa-executável resulta da união do programa-objeto, que foi gerado na compilação do programa-fonte, com outros programas-objeto da própria linguagem FORTRAN, que contêm a tradução dos comandos do FORTRAN, como o WRITE e END usados no primeiro programa-fonte (Figura 1.17).

Para gerar o programa-executável, deve-se fazer o seguinte no menu do Fortran: **Build, Build**. Em seguida, conforme é mostrado na Figura 1.18, aparecerão duas linhas com mensagens na subjanela da parte inferior do Fortran:

1) A primeira informa que o Fortran está gerando o programa-executável, processo chamado de linkagem (linking).

- 2) A segunda informa o nome do programa-executável (no caso, programa01.exe) que foi gerado e o número de erros (errors) e de avisos (warnings) relacionados a ele. O nome do programa-executável é igual ao nome do projeto mas com extensão exe.

O programa-executável é salvo automaticamente dentro da pasta Debug, que fica dentro da pasta do projeto, além de outros arquivos auxiliares. Isso pode ser visto com o aplicativo Windows Explorer, como mostrado na Figura 1.19. No Windows Explorer a extensão ou tipo do programa-executável aparece como Aplicativo.

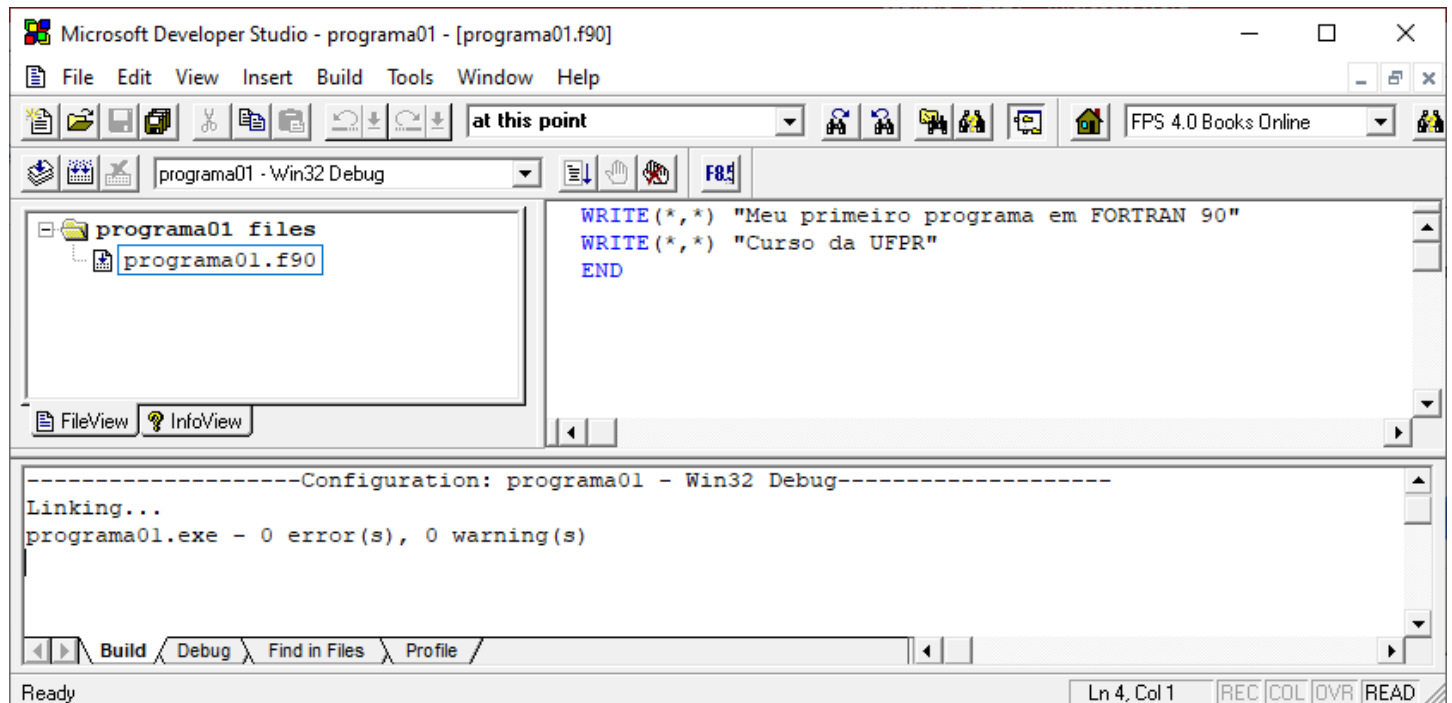


Figura 1.18 Geração do programa-executável.

1.11 EXECUÇÃO DO PROGRAMA

Após gerar o programa-executável, para executá-lo ou “rodá-lo”, isto é, para colocá-lo em funcionamento, basta fazer o seguinte no menu do Fortran: **Build, Execute**. Como resultado deste comando, deve ocorrer o seguinte:

- 1) Aparecer uma janela do tipo DOS, conforme a Figura 1.20. Isso ocorre porque na seção 1.6 foi criado um projeto do tipo Console Application.
- 2) No título desta janela, aparece o nome do programa que está sendo executado e a pasta dentro da qual ele se encontra, no caso C:\MSDEV\Projects\programa01\Debug\programa01.exe
- 3) Dentro desta janela, com fundo em preto, aparece o resultado da execução do programa em amarelo; as cores do fundo desta janela e do texto dependem da configuração de cada computador ou dispositivo. No caso, as frases “Meu primeiro programa em FORTRAN 90” e “Curso da UFPR”. A ordem das duas frases é a mesma em que elas foram colocadas no programa-fonte. Isso ocorre, porque o programa é executado do topo para baixo, linha por linha, do programa-fonte.

- 4) Além disso, também aparece a frase “Press any key to continue”. Como diz este aviso, basta clicar em qualquer tecla para continuar. Ao fazer isso, a execução do programa é encerrada.

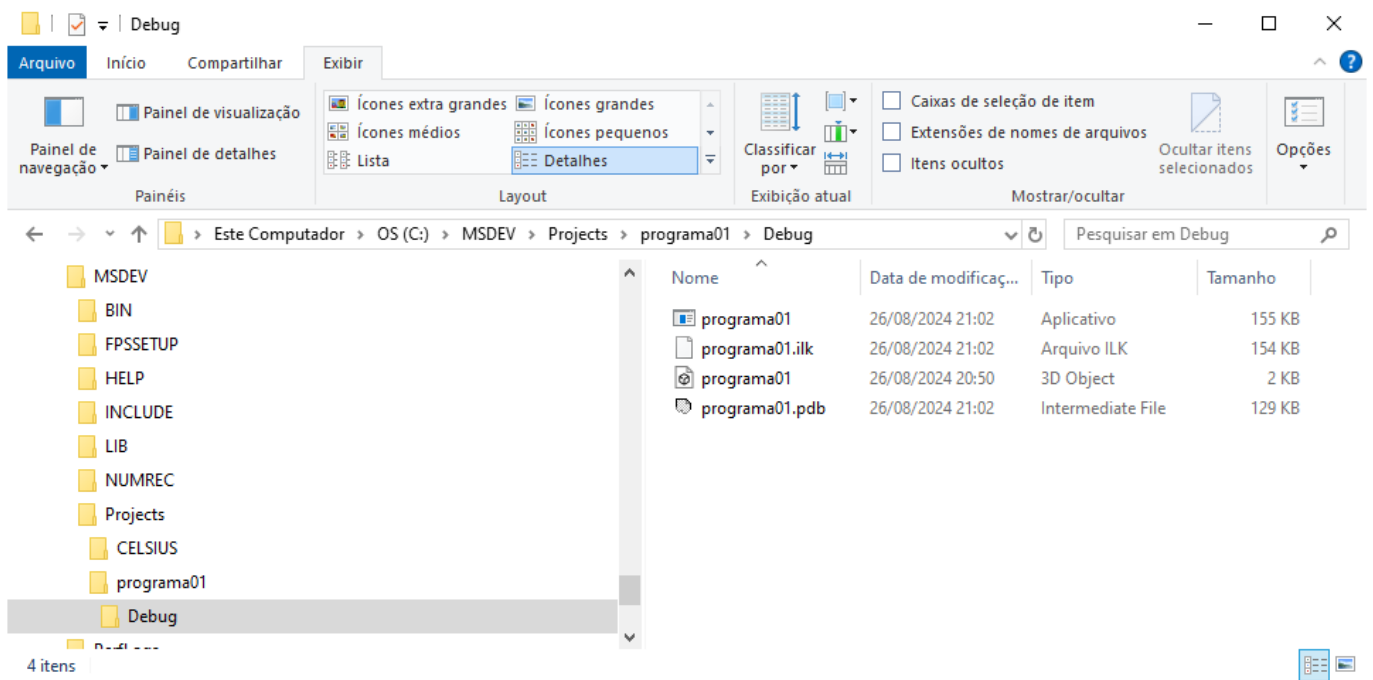


Figura 1.19 Pasta do projeto programa01 e sua subpasta Debug.

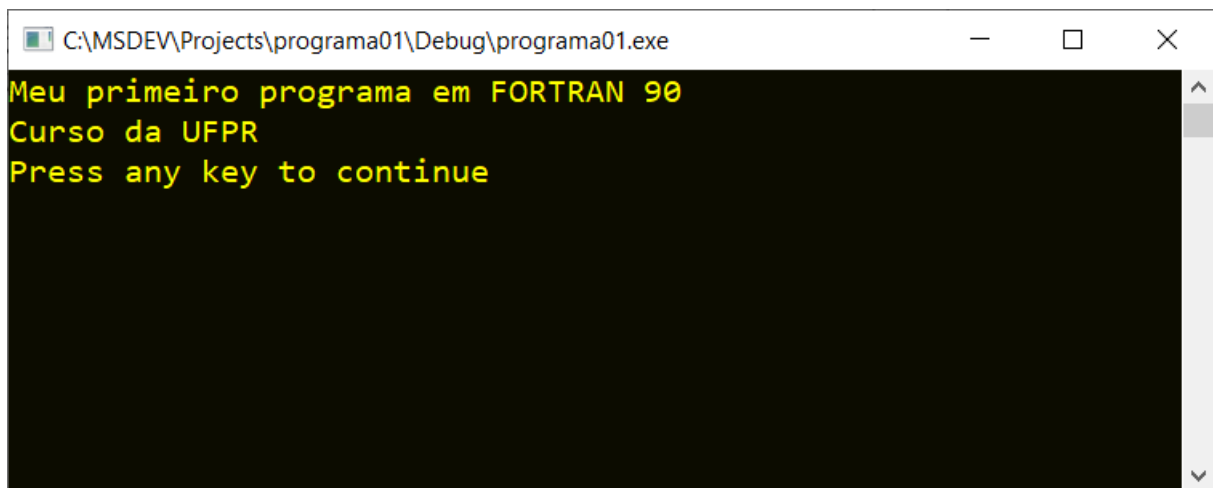


Figura 1.20 Resultado da execução do programa01.

1.12 FIM DA SESSÃO

Para concluir o uso do Fortran, deve-se executar o seguinte:

- 1) Para encerrar as atividades com um projeto, basta executar no menu do Fortran: **File, Close Workspace**
- 2) Na janela Microsoft Developer Studio, clicar sobre o botão **Sim**
- 3) Para fechar o aplicativo Fortran, basta fazer o seguinte em seu menu: **File, Exit**

1.13 EXERCÍCIOS

Exercício 1.1

- 1) Criar um projeto do tipo Console Application com o nome **exercicio_1_1**
- 2) Editar um programa-fonte em FORTRAN para escrever a data atual.
- 3) Compilar o programa-fonte
- 4) Gerar o programa-executável
- 5) Executar o programa

Exercício 1.2

- 1) Criar um projeto do tipo Console Application com o nome **exercicio_1_2**
- 2) Repetir os itens 2 a 5 do Exercício 1.1 para escrever, em linhas diferentes, cada um dos seguintes itens: um nome de pessoa, nome de alguma cidade e um número de telefone.

Exercício 1.3

- 1) Criar um projeto do tipo Console Application com o nome **exercicio_1_3**
- 2) Repetir os itens 2 a 5 do Exercício 1.1 para escrever com um único WRITE, em uma mesma linha, cada um dos seguintes itens: um nome de pessoa, nome de alguma cidade e um número de telefone.