

## **OBJETIVOS DO CAPÍTULO**

- Conceitos de: arquivo de entrada, arquivo interno, arquivo texto, arquivo binário, dispositivo e nome de arquivo genéricos
- Comandos do FORTRAN: uso avançado de OPEN, WRITE e SYSTEM
- No comando OPEN, usar os parâmetros FORMATTED, UNFORMATTED, REWIND e APPEND

Para inicializar as atividades deste capítulo, deve-se acessar o programa Fortran, no Windows, através de: **Start, Programs, Fortran PowerStation 4.0, Microsoft Developer Studio**

### **14.1 programa14a.f90**

- 1) Objetivos do programa: relembrar os conhecimentos atuais sobre
  - (a) o uso de janela DOS para entrar dados;
  - (b) escrever resultados num arquivo, com o nome do arquivo e o número do dispositivo sendo pré-definidos; e
  - (c) abrir o arquivo de resultados com o aplicativo Notepad.
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa14a**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa14a.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 14.1**.
- 5) Executar **Build, Compile** para compilar o programa.
- 6) Gerar o programa-executável fazendo **Build, Build**.
- 7) Executar o programa através de **Build, Execute com o valor 8**, por exemplo. O resultado deve ser o mostrado na Figura 14.1.
- 8) **Executar novamente o programa com outros dados e analisar os novos resultados.**
- 9) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**.

Tabela 14.1 Programa14a.f90

```
USE PORTLIB
IMPLICIT NONE
REAL*8 Pi
INTEGER UNIT, VER

Pi = DACOS(-1.0d0)

WRITE(*,*) "Entre com o valor de UNIT (inteiro) ="
READ(*,*) UNIT

OPEN(9,file="saida14a.txt")

WRITE(9,11) UNIT, Pi

CLOSE(9)

VER = SYSTEM("Notepad saida14a.txt")

11 FORMAT( 5X, "UNIT =", I4, 1/, &
          5X, "Pi   =", 1PE25.15 )

END
```

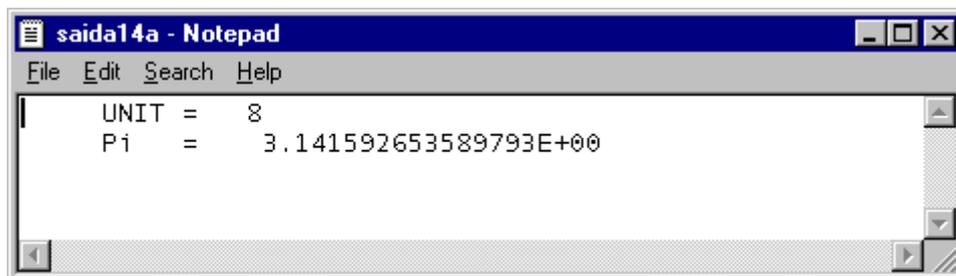


Figura 14.1 Resultado do programa14a.f90.

## 14.2 programa14b.f90

- 1) Objetivos do programa:
  - (a) Entrada de dados através de janela DOS;
  - (b) escrever resultados num arquivo cujo nome e o número do dispositivo são genéricos e definidos pelo usuário através dos dados do programa; e
  - (c) abrir o arquivo de resultados de nome genérico com o aplicativo Notepad.
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa14b**

- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa14b.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 14.2**.

Tabela 14.2 Programa14b.f90

```

USE PORTLIB
IMPLICIT NONE
REAL*8 Pi
INTEGER UNIT, VER
CHARACTER(50) SAIDA, TEXTO

Pi = DACOS(-1.0d0)

WRITE(*,*) "Unidade de saida (inteiro) ="
READ(*,*) UNIT

WRITE(*,*) "Nome do arquivo de saida (character) ="
READ(*,*) SAIDA

TEXTO = "Notepad " // SAIDA

OPEN(UNIT, file = SAIDA )

WRITE(UNIT,11) UNIT, Pi, SAIDA, TEXTO

CLOSE(UNIT)

11 FORMAT( 1/, 5X, "UNIT =", I4,      &
          1/, 5X, "Pi   =", 1PE25.15, &
          1/, 5X, "SAIDA = ", A50,    &
          1/, 5X, "TEXTO = ", A50    )

VER = SYSTEM( TEXTO )

END

```

- 5) Comentários sobre o programa:
  - (a) Na linha `OPEN(UNIT, file = SAIDA )`, deve-se notar que no comando OPEN, no lugar do número do dispositivo que identifica o arquivo, é usado uma variável chamada UNIT, que é um dado do programa. E o nome do arquivo, que aparece após a palavra-chave FILE, também é uma

variável, chamada SAIDA, que é definida pelo usuário através dos dados do programa; observa-se que não se deve usar apóstrofes ou aspas com esta variável, como era feito antes, por exemplo no programa14a.f90. Os nomes das variáveis UNIT e SAIDA podem ser outros quaisquer.

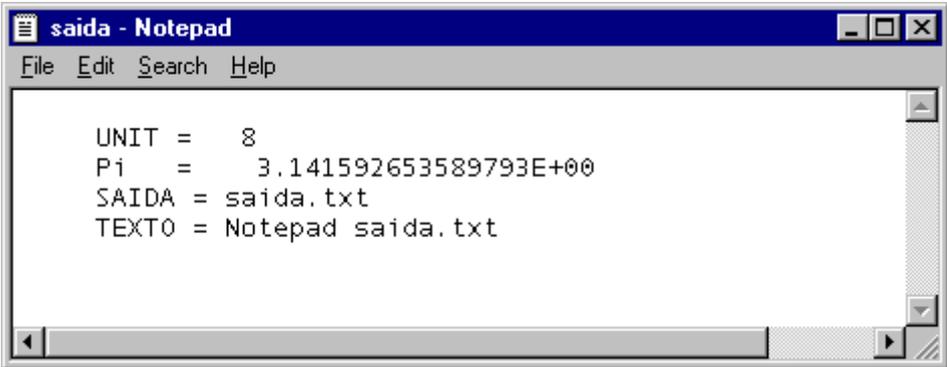
- (b) Na linha `WRITE(UNIT,11) UNIT, Pi, SAIDA, TEXTO`, deve-se notar que no comando WRITE, no lugar do número do dispositivo que identifica o arquivo, é usado uma variável chamada UNIT, que é um dado do programa.
- (c) Na linha `CLOSE(UNIT)`, deve-se notar que no comando CLOSE, no lugar do número do dispositivo que identifica o arquivo, é usado uma variável chamada UNIT, que é um dado do programa.
- (d) Na linha `VER = SYSTEM( TEXTO )`, deve-se notar que no comando SYSTEM, no lugar de um texto específico, como era feito antes, por exemplo no programa14a.f90, é usado uma variável chamada TEXTO.

- 6) Executar **Build, Compile** para compilar o programa.
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Executar o programa através de **Build, Execute** com:

**Unidade de saída = 8**

**Nome do arquivo de saída = saida.txt**

O resultado deve ser o mostrado na Figura 14.2.



```
saida - Notepad
File Edit Search Help
UNIT = 8
Pi = 3.141592653589793E+00
SAIDA = saida.txt
TEXTO = Notepad saida.txt
```

Figura 14.2 Resultado do programa14b.f90 para os dados do item 8.

- 9) Executar o programa através de **Build, Execute** com:  
**Unidade de saída = 9**  
**Nome do arquivo de saída = saida.txt**
- 10) Executar o programa através de **Build, Execute** com:  
**Unidade de saída = 9**  
**Nome do arquivo de saída = saida14b.txt**
- 11) Executar o programa através de **Build, Execute** com:

**Unidade de saída = 9**

**Nome do arquivo de saída = saída**

12) **Analisar os resultados da execução do programa com os dados dos itens 8 a 11.**

13) Alterar a linha `OPEN(UNIT, file = SAIDA )`

para `OPEN(UNIT, file = SAIDA, form = "formatted" )`

Em seguida, executar novamente os itens 6 a 8, acima. O resultado deve ser o mostrado na Figura 14.2. Isso ocorre porque o parâmetro “FORMATTED”, no comando OPEN, que aparece após a palavra-chave FORM, é a opção default. Ou seja, usando-o ou não, o resultado é o mesmo. Ele significa que o arquivo é do tipo texto, isto é, ao abri-lo, consegue-se ler o conteúdo, principalmente os números, escritos na base decimal.

14) Alterar a linha `OPEN(UNIT, file = SAIDA, form = "formatted" )`

para `OPEN(UNIT, file = SAIDA, form = "unformatted" )`

Alterar também a linha `WRITE(UNIT,11) UNIT, Pi, SAIDA, TEXTO`

para `WRITE(UNIT) UNIT, Pi, SAIDA, TEXTO`

Em seguida, executar novamente os itens 6 a 8, acima. O resultado deve ser o mostrado na Figura 14.3. O parâmetro “UNFORMATTED”, no comando OPEN, que aparece após a palavra-chave FORM, é usado para escrever resultados num arquivo do tipo binário. Isto é, ao abri-lo, não se consegue ler o conteúdo, de forma geral, principalmente os números, escritos na base binária. Para escrever em forma binária, no comando WRITE deve-se indicar apenas o número do dispositivo do arquivo, conforme exemplificado neste item 14. O formato binário é muito indicado para gravar grandes quantidades de resultados, pois precisa menos espaço em disco do que o formato texto.

15) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**.

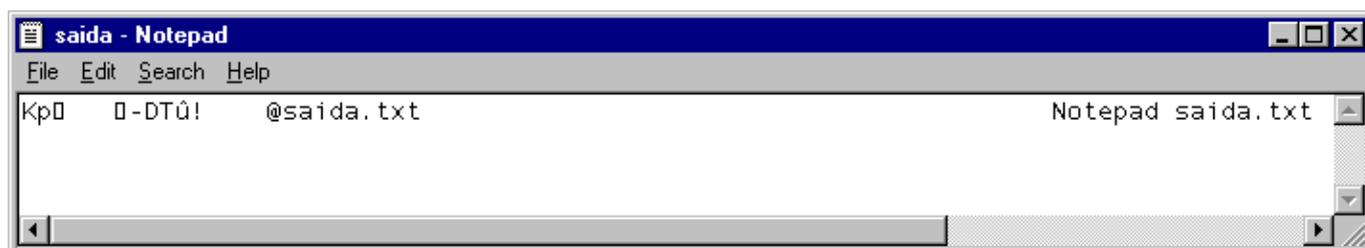


Figura 14.3 Resultado do programa14b.f90 para o item 14.

### **14.3 programa14c.f90**

1) Objetivo do programa: adicionar resultados num arquivo já existente.

2) No Fortran, seguindo o procedimento-padrão, **criar um projeto** com o nome **programa14c**

- 3) No Fortran, seguindo o **procedimento-padrão**, **criar e inserir** no projeto o programa-fonte **programa14c.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 14.2**, que é o programa14b.f90.
- 5) Alterar a linha `OPEN(UNIT, file = SAIDA )`  
para `OPEN(UNIT, file = SAIDA, position = "rewind" )`
- 6) Comentários sobre o programa:  
O parâmetro “REWIND”, no comando OPEN, que aparece após a palavra-chave POSITION é a opção default. Ou seja, usando-o ou não, o resultado é o mesmo. Ele é usado para fazer com que resultados sejam escritos a partir do início de um arquivo, seja o arquivo novo ou já existente.
- 7) Executar **Build, Compile** para compilar o programa.
- 8) Gerar o programa-executável fazendo **Build, Build**.
- 9) Executar o programa através de **Build, Execute** com:  
**Unidade de saída = 8**  
**Nome do arquivo de saída = saida.txt**  
O resultado deve ser o mostrado na Figura 14.2.
- 10) Executar o programa através de **Build, Execute** com:  
**Unidade de saída = 9**  
**Nome do arquivo de saída = saida.txt**  
O novo resultado difere do mostrado na Figura 14.2 apenas em UNIT = 9.
- 11) Alterar a linha `OPEN(UNIT, file = SAIDA, position = "rewind" )`  
para `OPEN(UNIT, file = SAIDA, position = "append" )`  
O parâmetro “APPEND”, no comando OPEN, que aparece após a palavra-chave POSITION é usado para fazer com que resultados sejam escritos a partir do fim de um arquivo já existente. Se o arquivo for novo, os resultados serão escritos a partir do seu início.
- 12) Executar novamente os itens 7 e 8, acima.
- 13) Executar novamente o item 9, acima. O resultado da execução do programa14c.f90 deve ser o mostrado na Figura 14.4. Deve-se perceber que ele representa a soma das execuções dos itens 10 e 13.
- 14) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**.

#### **14.4 programa14d.f90**

- 1) Objetivos do programa:
  - (a) ler os dados do programa de um arquivo, em vez de usar a janela DOS; e

(b) abrir o arquivo de dados com o aplicativo Notepad.

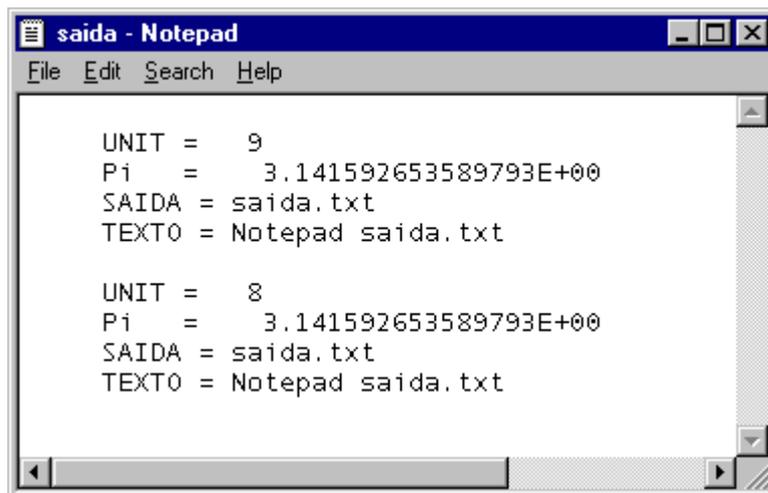


Figura 14.4 Resultado do programa14c.f90 para o item 13.

- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa14d**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa14d.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 14.3**.
- 5) Comentários sobre o programa:
  - (a) O programa14b.f90 é praticamente igual ao programa14d.f90. A diferença está na forma de entrar os dados do programa. No primeiro é usada a janela DOS e, no segundo, um arquivo de dados.
  - (b) Na linha **OPEN(1, file = "dados.txt" )** do programa14d.f90, define-se o número 1 como o dispositivo associado ao arquivo de dados chamado "dados.txt".

Tabela 14.3 Programa14d.f90

```
USE PORTLIB
IMPLICIT NONE
REAL*8 Pi
INTEGER UNIT, VER
CHARACTER(50) SAIDA, TEXTO

VER = SYSTEM("Notepad dados.txt" )

OPEN(1, file = "dados.txt" )

READ(1,*) UNIT
```

```

READ(1,*) SAIDA

CLOSE(1)

Pi = DACOS(-1.0d0)

TEXTO = "Notepad " // SAIDA

OPEN(UNIT, file = SAIDA )

WRITE(UNIT,11) UNIT, Pi, SAIDA, TEXTO

CLOSE(UNIT)

11 FORMAT( 1/, 5X, "UNIT =", I4,      &
          1/, 5X, "Pi   =", 1PE25.15, &
          1/, 5X, "SAIDA = ", A50,    &
          1/, 5X, "TEXTO = ", A50    )

VER = SYSTEM( TEXTO )

END

```

- (c) Nas linhas `READ(1,*) UNIT` e `READ(1,*) SAIDA` os dois dados são lidos, variáveis UNIT e SAIDA. Deve-se notar que são dois comandos READ, cada um sendo usado para ler uma única variável. Assim, após o programa ler o conteúdo da variável UNIT na primeira linha do arquivo “dados.txt”, tudo que estiver nesta linha após o valor de UNIT não é considerado pelo programa. Portanto, podem ser inseridos comentários nos arquivos de dados para ficar claramente definido ao que corresponde cada dado. O mesmo se aplica ao valor da variável SAIDA na segunda linha do arquivo “dados.txt”; o que está após este dado, na mesma linha, não é considerado pelo programa.
- (d) Recomenda-se sempre fazer isso: um dado por linha e, em seguida, na mesma linha, um comentário para informar o que significa a variável. Além disso, nunca se deve usar formato para ler dados; este é o motivo do asterisco nos dois comandos READ. Os formatos devem ser usados apenas para escrever resultados.
- (e) Todos os comentários já feitos nos capítulos anteriores e neste sobre o comando OPEN, para arquivos de resultados ou de saída, também valem para arquivos de entrada ou de dados.
- 6) Executar **Build, Compile** para compilar o programa.
- 7) Gerar o programa-executável fazendo **Build, Build**.

8) Antes de executar um novo programa que use arquivo de dados, é necessário criar o arquivo de dados e inserir nele os respectivos dados. No caso do programa14d.f90, é necessário **criar o arquivo “dados.txt” e inserir os dois dados que correspondem às variáveis UNIT e SAIDA. Para fazer isso, executar o seguinte:**

(a) Start, Programs, Accessories, Notepad

(b) Dentro do espaço de edição do Notepad, editar:

```
8           Unidade de saida (inteiro)
saida.txt  Nome do arquivo de saida (character)
```

(c) File, Save

(d) Na janela “Save As”, em “File Name”, colocar “dados.txt”; em “Save in:”, selecionar o diretório do projeto. Em seguida, clicar sobre o botão “Save”. Finalmente, selecionar File, Exit.

O resultado deve ser o mostrado na Figura 14.5.

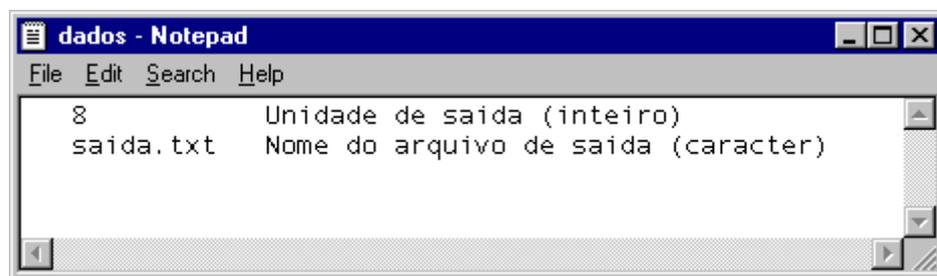


Figura 14.5 Arquivo de dados “dados.txt” do programa14d.f90.

9) Executar o programa através de **Build, Execute**. O resultado deverá ser:

(a) A abertura do arquivo “dados.txt” com o programa Notepad. Se o arquivo não existir no diretório do projeto, ocorrerá um erro. Se o usuário quiser, poderá alterar o conteúdo das variáveis no arquivo “dados.txt”. Depois, deve-se gravar o arquivo alterado com File, Save. Finalmente, fechar o arquivo com File, Exit.

(b) Em seguida, ocorrerá a abertura do arquivo “saida.txt” com o programa Notepad. O conteúdo deverá ser o mesmo mostrado na Figura 14.2. Fechar o arquivo com File, Exit.

(c) Na janela do DOS, o resultado deverá ser apenas a frase “Press any key to continue”

10) Executar novamente o programa através de **Build, Execute** com:

**Unidade de saída = 9**

**Nome do arquivo de saída = saida.txt**

**Analisar** o novo resultado.

11) **Executar** novamente o programa com outros dados e **analisar** o novo resultado.

12) Encerrar a sessão seguindo o [procedimento-padrão](#).

## 14.5 programa14e.f90

- 1) Objetivo do programa: usar arquivos internos.
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa14e**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa14e.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 14.4**.
- 5) Comentários sobre o programa:
  - (a) Até agora o dispositivo associado a um arquivo de resultados ou de dados era explicitamente um número ou uma variável do tipo inteiro, por exemplo: a linha `WRITE(9,11) UNIT, Pi` do programa14a.f90, na qual o dispositivo é o número 9; e a linha `WRITE(UNIT,11) UNIT, Pi, SAIDA, TEXTO` do programa14b.f90, na qual o dispositivo é a variável inteira UNIT. O dispositivo associado a um arquivo de resultados ou de dados também pode ser uma variável do tipo caracter.
  - (b) Na linha `WRITE(TEXT02,*) UNIT` do programa14e.f90, define-se a variável TEXT02, do tipo caracter, como o dispositivo no qual será escrito o conteúdo da variável UNIT, que é do tipo inteiro.
  - (c) Na linha `WRITE(TEXT03,*) Pi` do programa14e.f90, define-se a variável TEXT03, do tipo caracter, como o dispositivo no qual será escrito o conteúdo da variável Pi, que é do tipo real dupla.
  - (d) Na linha `WRITE(TEXT04,*) TEXTO // TEXT03` do programa14e.f90, define-se a variável TEXT04, do tipo caracter, como o dispositivo no qual será escrito o conteúdo da variável TEXT0 concatenada com o conteúdo da variável TEXT03, sendo ambas do tipo caracter.
  - (e) Na linha `READ(TEXT03,*) TEXTO5` do programa14e.f90, define-se a variável TEXT03, do tipo caracter, como o dispositivo do qual será lido o conteúdo da variável TEXT05, que é do tipo caracter.

Tabela 14.4 Programa14e.f90

```
USE PORTLIB
IMPLICIT NONE
REAL*8 Pi
INTEGER UNIT, VER
CHARACTER(30) SAIDA, TEXTO, TEXT02, TEXT03, TEXT05
CHARACTER(70) TEXT04
```

```

Pi = DACOS(-1.0d0)

WRITE(*,*) "Unidade de saida (inteiro) ="
READ(*,*) UNIT

WRITE(*,*) "Nome do arquivo de saida (character) ="
READ(*,*) SAIDA

TEXT0 = "Notepad " // SAIDA

WRITE(TEXT02,*) UNIT

WRITE(TEXT03,*) Pi

WRITE(TEXT04,*) TEXT0 // TEXT03

READ(TEXT03,*) TEXT05

OPEN(UNIT, file = SAIDA )

WRITE(UNIT,11) UNIT, Pi, SAIDA, TEXT0, TEXT02, TEXT03, TEXT04, TEXT05

CLOSE(UNIT)

11 FORMAT( 1/, 5X, "UNIT   =", I4,      &
          1/, 5X, "Pi     =", 1PE25.15, &
          1/, 5X, "SAIDA  = ", A30,     &
          1/, 5X, "TEXT0  = ", A30,     &
          1/, 5X, "TEXT02 = ", A30,     &
          1/, 5X, "TEXT03 = ", A30,     &
          1/, 5X, "TEXT04 = ", A70,     &
          1/, 5X, "TEXT05 = ", A30      )

VER = SYSTEM( TEXT0 )

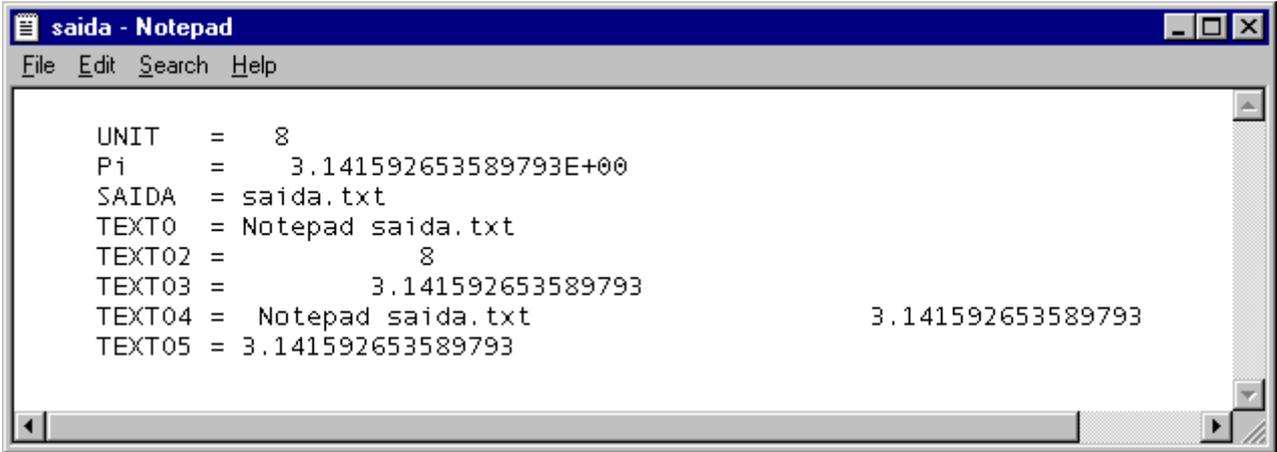
END

```

- 6) Executar **Build, Compile** para compilar o programa.
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Executar o programa através de **Build, Execute** com:
  - Unidade de saída = 8**
  - Nome do arquivo de saída = saida.txt**

O resultado deve ser o mostrado na Figura 14.6.

**Analisar** o resultado.



```
saida - Notepad
File Edit Search Help

UNIT = 8
Pi = 3.141592653589793E+00
SAIDA = saida.txt
TEXT0 = Notepad saida.txt
TEXT02 = 8
TEXT03 = 3.141592653589793
TEXT04 = Notepad saida.txt 3.141592653589793
TEXT05 = 3.141592653589793
```

Figura 14.6 Resultado do programa14e.f90, com o item 8.

- 9) Executar novamente o programa através de **Build, Execute** com outros dados. **Analisar** o novo resultado.
- 10) Encerrar a sessão seguindo o [procedimento-padrão](#).

## **14.6 EXERCÍCIOS**

### **Exercício 14.1**

Seguindo o algoritmo do programa14d.f90, Tabela 14.3, refazer o exercício 13.1 atendendo:

- (a) um arquivo para entrar os dados do programa;
- (b) no início da execução do programa, abrir automaticamente o arquivo de dados com o aplicativo Notepad;
- (c) um arquivo para escrever os resultados do programa, com o nome do arquivo sendo definido pelo usuário;
- (d) no final da execução do programa, abrir automaticamente o arquivo de resultados com o aplicativo Notepad.

### **Exercício 14.2**

Seguindo o algoritmo do programa14d.f90, Tabela 14.3, refazer o exercício 13.2 atendendo aos mesmos 4 itens do exercício 14.1.

### **Exercício 14.3**

Seguindo o algoritmo do programa14d.f90, Tabela 14.3, refazer o exercício 9.3 atendendo aos mesmos 4 itens do exercício 14.1.

### **Exercício 14.4**

Seguindo o algoritmo do programa14d.f90, Tabela 14.3, refazer o programa da seção 12.4, Tabela 12.7, atendendo aos mesmos 4 itens do exercício 14.1.

### **Exercício 14.5**

Implementar uma nova versão do programa14d.f90 para que:

- (a) o nome do arquivo de dados seja definido pelo usuário; e
- (b) no início da execução do programa, o arquivo de dados seja aberto automaticamente com o aplicativo Notepad.