

Programação para Mainframe



Curso de Análise de Desenvolvimento de Sistemas

Aula 05

Prof. Claudio Benossi

1. Unidade

Trabalhando com arquivos



Introdução ao COBOL

Nosso objetivo hoje é trabalhar utilizando arquivos sequenciais.

Em um **arquivo sequencial**, os registros são dispostos ordenadamente, obedecendo a sequência determinada por uma chave primária, chamada chave de ordenação e para **arquivos** sequências sem chave de ordenação, os registros estão ordenados um série, sendo que geralmente, cada novo registro é colocado no final do **arquivo**.





Comando OPEN

Todo arquivo antes de ser manipulado (lido ou gravado) deverá ser aberto pelo comando **OPEN**, que estabelece o contato com o dispositivo físico do arquivo e reserva na memória (**FILE SECTION**) áreas necessárias para a troca de dados com o dispositivo externo onde o arquivo reside ou residirá, indicando também se o arquivo será utilizado como entrada ou saída.





Comando OPEN

Regras do comando OPEN:

- ▶ **INPUT:** permite abrir o arquivo apenas para operações de leitura.
- ▶ **OUTPUT:** permite abrir o arquivo para operações de gravação / impressão.
- ▶ **EXTEND:** permite abrir um arquivo para operações de OUTPUT. Esta opção deve ser utilizada apenas para arquivo sequencial.





Comando OPEN

Regras do comando OPEN:

- nome-arquivo: nome lógico definido na cláusula **SELECT** e na **FD**.
- Permite a abertura de diversos arquivos em um mesmo comando.





Comando CLOSE

Após o uso do arquivo, independentemente de tratar-se de arquivo de entrada ou saída, o mesmo deverá ser fechado para liberá-lo para uso por outros programas de aplicação.





Comando CLOSE

Regras do comando CLOSE:

- nome-arquivo: nome lógico definido na cláusula **SELECT** e na **FD**.
- Permite o fechamento de diversos arquivos em um mesmo comando.





Comando CLOSE

Regras do comando CLOSE:

- ▶ Após execução do comando, a área associada ao arquivo na **FILE SECTION** será liberada e, não poderá ser acessada novamente.
- ▶ Se necessário reutilizar o arquivo um novo comando OPEN será necessário.





Comando READ

A instrução **READ** obtém o próximo registro lógico de um arquivo lido sequencialmente e aberto no modo **INPUT**.

```
READ ALUNOS INTO WS-DADOS-ALUNO
  AT END MOVE 'F' TO WS-STATUS-EOF
  NOT AT END
    DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO
END-READ
```





Comando READ

Regras do comando READ:

- nome-arquivo: nome lógico definido na clausula **SELECT** e na **FD**.
- **INTO**: O registro corrente é movido de uma área de **INPUT** para **nome-area-working**, de acordo com as regras da instrução **MOVE**.



Regras do comando READ:

- ▶ **AT END:** A instrução-imperativa do após o **AT END** será executada quando o programa detectar o final do arquivo (**EOF**).
- ▶ **NOT AT END:** A instrução-imperativa do após o **NOT AT END** será executada enquanto o programa não detectar o final do arquivo. Trabalha em conjunto com o **END-READ**.



Comando WRITE

A instrução **WRITE** grava um registro lógico em um arquivo sequencial.

Formato: **WRITE** nome-de-registro [FROM identificador]





Comando WRITE

Regras do comando WRITE:

- O arquivo associado à instrução **WRITE** foi aberto no modo **OUTPUT**.
- nome-de-registro: deve ser o nome de um registro lógico definido como nível **01** após a entrada da **FD** na **FILE SECTION** da **DATA DIVISION**.





Comando WRITE

Regras do comando WRITE:

- ▶ FROM identificador: O conteúdo do identificador é copiado para o **nome-deregistro** antes de ocorrer a gravação. Depois da execução da instrução **WRITE**, o registro continua disponível no identificador.





Comando FILE STATUS

O **FILE STATUS** é um recurso do COBOL utilizado para transferir ao programa de aplicação a decisão à adotar após a verificação do resultado da execução de operação de acesso a arquivos.

Os resultados esperados: **operação realizada com sucesso** ou **ocorrência de exceção na operação**.

A verificação do **FILE STATUS** é uma norma adotada como padrão nas instalações que trabalham com COBOL.



Comando FILE STATUS

Regras do uso do FILE STATUS:

- nome-arquivo e nome-externo: idem anterior.
- area-da-working: variável alfanumérica com duas posições.
- O uso do **FILE STATUS** é exclusivo, não permitindo que o programador utilize ao mesmo tempo comandos que não prevêem o uso do mesmo.



Comando FILE STATUS

Principais valores para o **FILE STATUS** e seu significado:

Valor	Significado
00	Operação realizada com sucesso
10	Fim de arquivo (EOF)
34	Tentativa de gravação após o limite final do arquivo
35	OPEN INPUT em arquivo não presente (FALTA O DD)
39	Erro no OPEN. Atributos do arquivo não condizem com os fornecidos.
41	OPEN em arquivo já aberto
42	CLOSE em arquivo não aberto
46	READ após final do arquivo
47	READ em arquivo não aberto como INPUT ou I-O
48	WRITE em arquivo não aberto como OUTPUT ou I-O
96	OPEN OUTPUT ou I-O em arquivo não presente (FALTA O DD).





Introdução ao COBOL

Para criação dos exemplos aplicados a aula de hoje vamos usar o OpenCobolIDE.



OpenCobolIDE

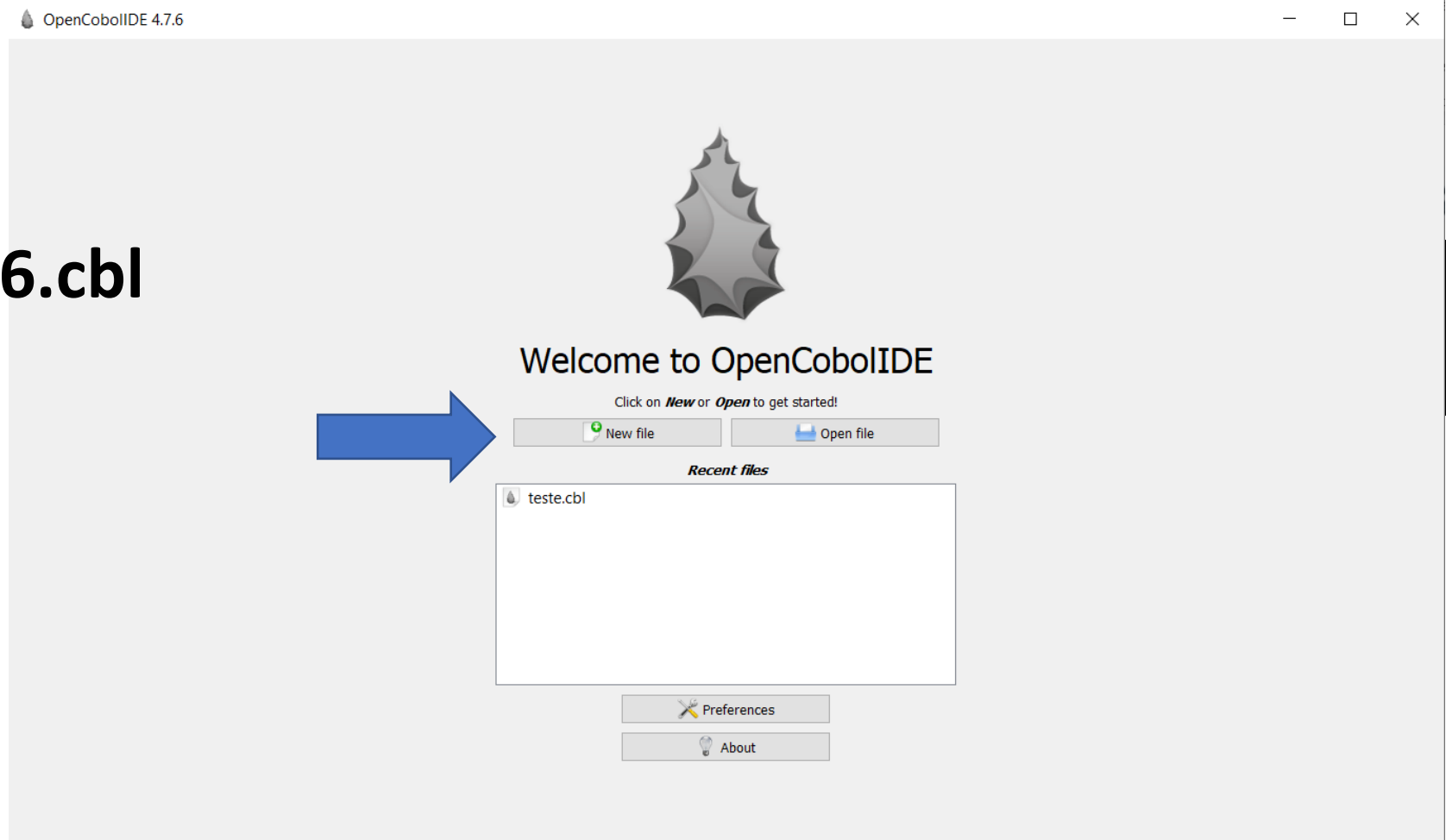
<https://launchpad.net/cobcide/+download>



Introdução ao COBOL

O OpenCobolIDE

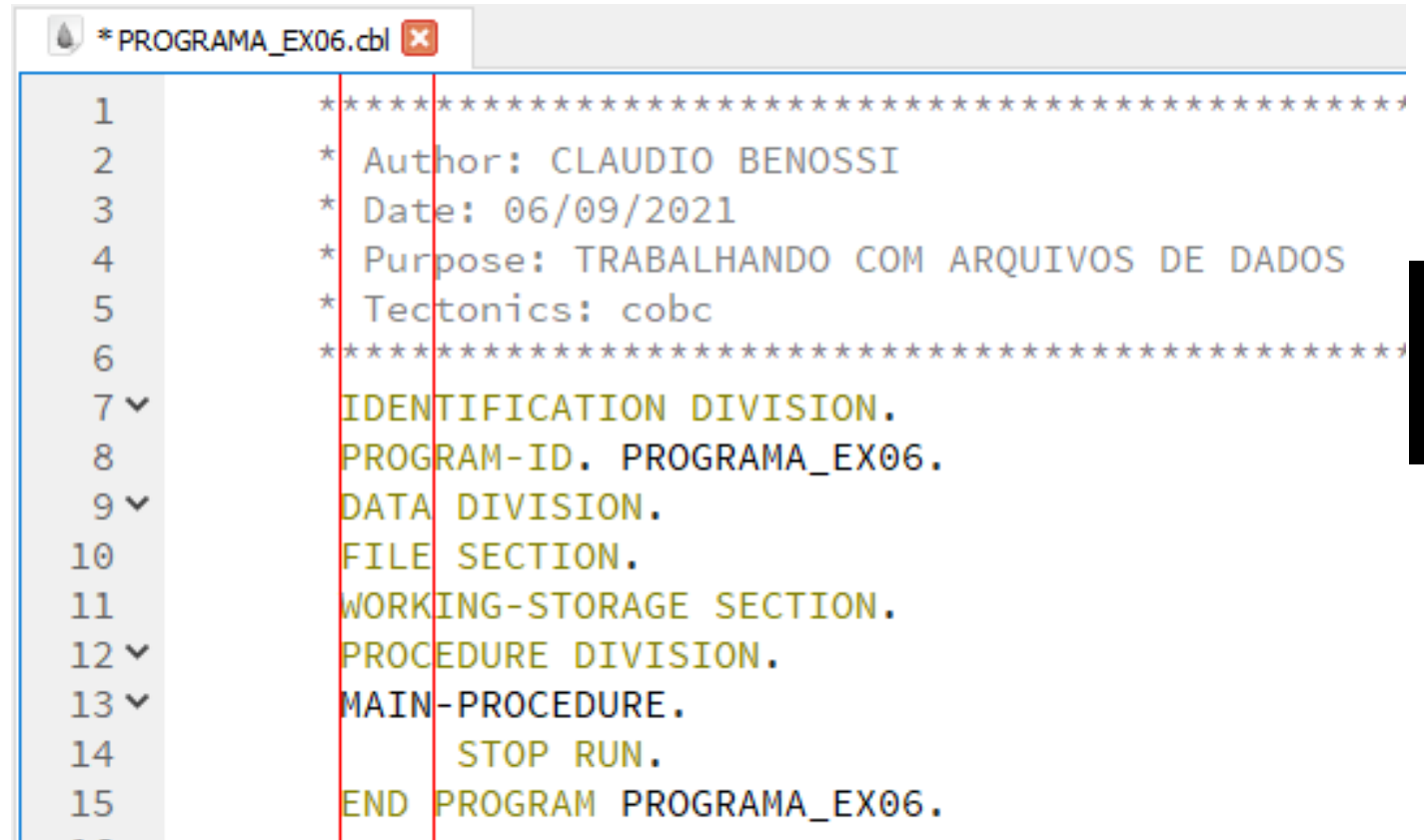
PROGRAMA_EX06.cbl



Introdução ao COBOL

Nosso ponto de partida:

PROGRAMA_EX06.cbl



```
* PROGRAMA_EX06.cbl
1 *****
2 * Author: CLAUDIO BENOSSE
3 * Date: 06/09/2021
4 * Purpose: TRABALHANDO COM ARQUIVOS DE DADOS
5 * Tectonics: cobc
6 *****
7 IDENTIFICATION DIVISION.
8 PROGRAM-ID. PROGRAMA_EX06.
9 DATA DIVISION.
10 FILE SECTION.
11 WORKING-STORAGE SECTION.
12 PROCEDURE DIVISION.
13 MAIN-PROCEDURE.
14     STOP RUN.
15 END PROGRAM PROGRAMA_EX06.
```



Introdução ao COBOL

Como vamos começar a trabalhar com arquivo de dados, precisamos fazer alguns ajustes em nosso código e vamos utilizar a seção INPUT-OUTPUT SECTION da ENVIROMENT DIVISION.





Introdução ao COBOL

ENVIROMENT DIVISION.

CONFIGURATION SECTION.

SPECIAL-NAME.

DECIMAL-POINT IS COMMA.



Introdução ao COBOL

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT **nome-do-arquivo** ASSIGN TO {**DISK, PRINTER**}

[ORGANIZATION IS {**SEQUENTIAL, INDEXED, LINE SEQUENTIAL, RELATIVE** }

[ACCESS MODE IS {**SEQUENTIAL, RANDOM, DYNAMIC** }

[RECORD KEY IS **nome-chave-primaria**]

[ALTERNATE RECORD KEYS IS **nome-chave-secundaria**_[WITH DUPLICATES]]

[FILE STATUS IS **nome-campo-status**] .



Introdução ao COBOL

SELECT nome-do-arquivo ASSIGN TO {**DISK, PRINTER**}

Aqui nos vamos indicar o nome do arquivo que vamos usar e a forma que vamos apresentar o resultado que pode ser armazenado **DISK** ou que pode ser somente impresso **PRINTER**.

No nosso exemplos vamos usar **DISK**.





Introdução ao COBOL

[ORGANIZATION IS {**SEQUENTIAL, INDEXED, LINE SEQUENTIAL, RELATIVE** }

A forma como meu arquivo será organizado.

Sequencial um após o outro.

Indexado, criando um index de localização (Semelhante ao conceito de chave primária)

Linha sequencial – Uma linha para cada registro (Tuplas) de forma sequencial.

Relativo – que pode uma “mescla” entre eles.





Introdução ao COBOL

[ACCESS MODE IS {**SEQUENTIAL, RANDOM, DYNAMIC** }

O modo como vamos acessar os dados, lendo de forma sequencial um após o outro, de forma randômica que pode ser usado para fazer sorteios, pegar uma amostra, etc., ou de modo dinâmico indo direto como se fosse uma dependência .

Recomenda-se que o modo acesso siga o modo como ele foi organizado.





Introdução ao COBOL

[RECORD KEY IS nome-chave-primaria]

Chave primária garante que o arquivo terá uma identificação única para que não exista registro duplicados.





Introdução ao COBOL

[ALTERNATE RECORD KEYS IS nome-chave-secundaria_[WITH DUPLICATES]]

Chave secundária, esta chave tem como finalidade estabelecer o relacionamento entre arquivos.

E nesse caso posso criar uma condição (WITH DUPLICATES), em situações onde apresente informações duplicadas.





Introdução ao COBOL

[FILE STATUS IS nome-campo-status].

Demonstra o status do nosso arquivo, pode ser aberto, em edição ou fechado.



Introdução ao COBOL

Importante falar que além de indicar os dados do arquivo, precisamos fazer uma descrição dos campos desse arquivo.



Introdução ao COBOL

DATA DIVISION.

FILE-SECTION.

FD nome-do-arquivo

[RECORD CONTAINS nn CHARACTERS]

[BLOCK CONTAINS nn RECORDS]

[LABEL RECORD IS { OMITTED, STANDARD }]

[VALUE OF FILE-ID nome-chave-primaria].

01 REG-ARQUIVO.

03 FD-CAMPO-01

03 FILLER

03 FS-DIA-NASCIMENTO.

05 FD-DIA-NASCIMENTO

05 FD-MES-NASCIMENTO

05 FD-AMO-NASCIMENTO

PIC X(004).

PIC X(005).

PIC 9(002).

PIC 9(002).

PIC 9(004).

Indica quantos caracteres
cada registro tem nesse
arquivo

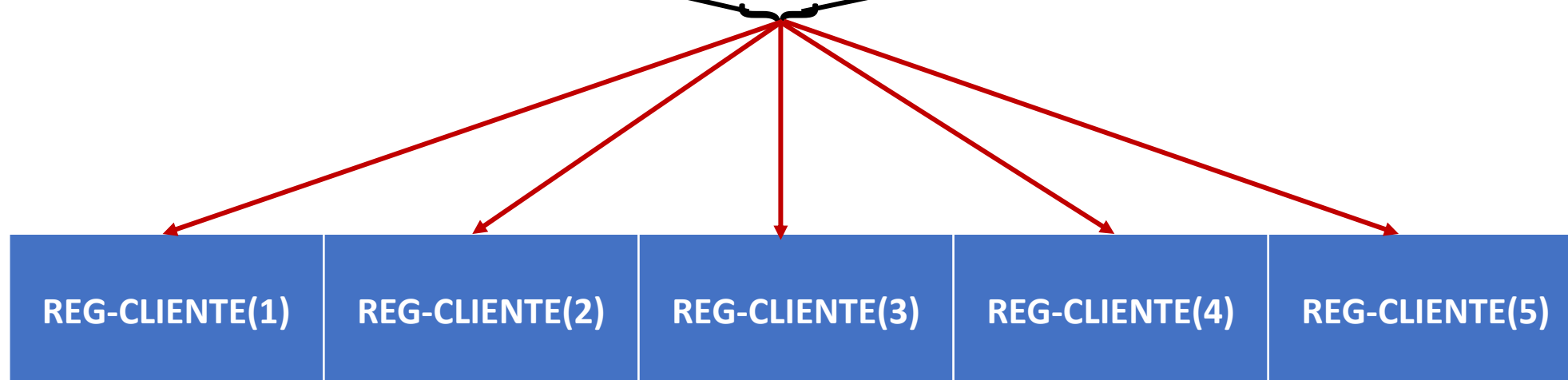
Indica quantos registro
tem nesse arquivo

O tipo de identificação do
arquivo

Identificação da chave
primária do arquivo

Registro Lógico

COD-CLIENTE	NOME	ENDERECO	BAIRRO	CIDADE	ESTADO	CEP
-------------	------	----------	--------	--------	--------	-----



Registro Físico

(Neste caso composto por 5 registros lógicos)

BLOCK CONTAINS 5 RECORD



Introdução ao COBOL

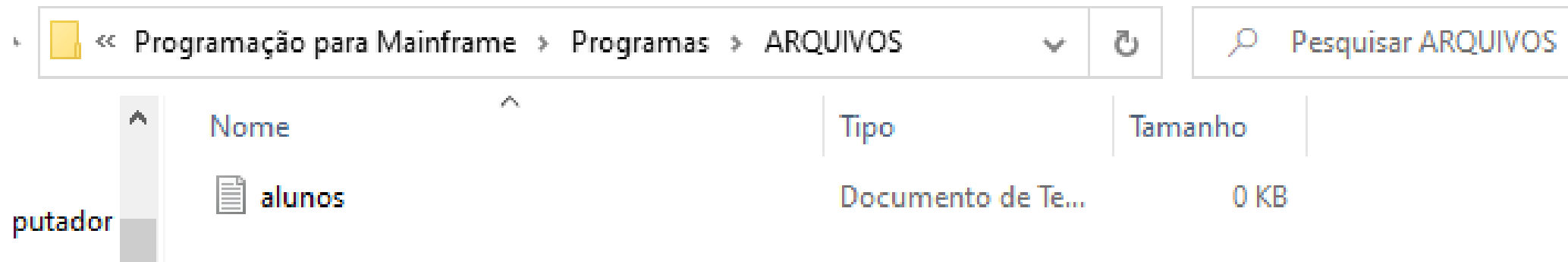
Antes de começar nosso código vamos criar uma pasta onde será armazenado nosso arquivo.

FATEC > Programação para Mainframe > Programas					Pesquisar Programas	
Nome		Data de modificação	Tipo	Tamanho		
ARQUIVOS		24/09/2021 10:30	Pasta de arquivos			
bin		24/09/2021 09:10	Pasta de arquivos			
PROGRAMA_EX01		08/09/2021 17:27	Cobol source code	3 KB		
PROGRAMA_EX02		10/09/2021 16:03	Cobol source code	3 KB		
PROGRAMA_EX03		13/09/2021 18:06	Cobol source code	3 KB		
PROGRAMA_EX04		19/09/2021 10:25	Cobol source code	3 KB		
PROGRAMA_EX05		24/09/2021 09:11	Cobol source code	4 KB		
PROGRAMA_EX06		24/09/2021 10:30	Cobol source code	1 KB		



Introdução ao COBOL

Dentro desta pasta vou criar um arquivo chamado de **alunos.txt**.



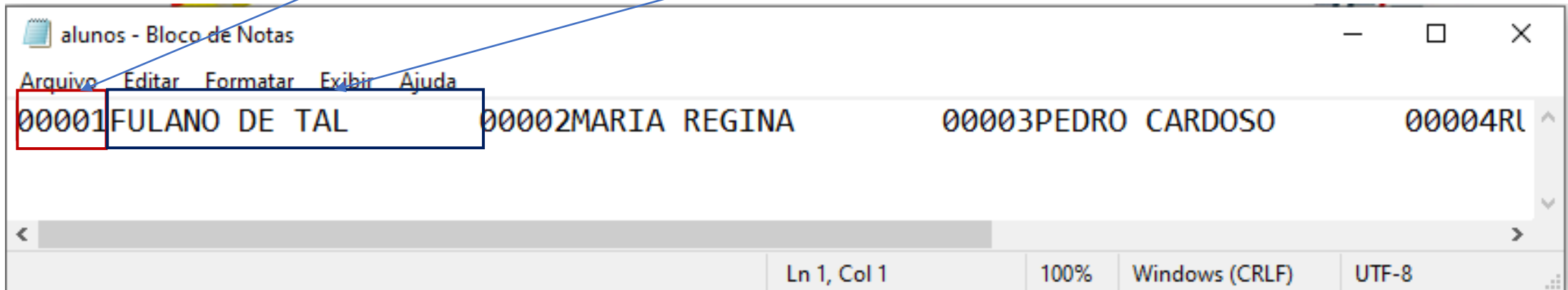
Vou deixar este arquivo para download no Teams.



Introdução ao COBOL

Nosso arquivo **alunos.txt** terá informações dos alunos, nesse exemplo vamos ter o **RGM** (Registro de Matricula) que é uma informação numérica de 5 dígitos e o **Nome** do aluno que é uma informação alfabética com 20 caracteres.

36

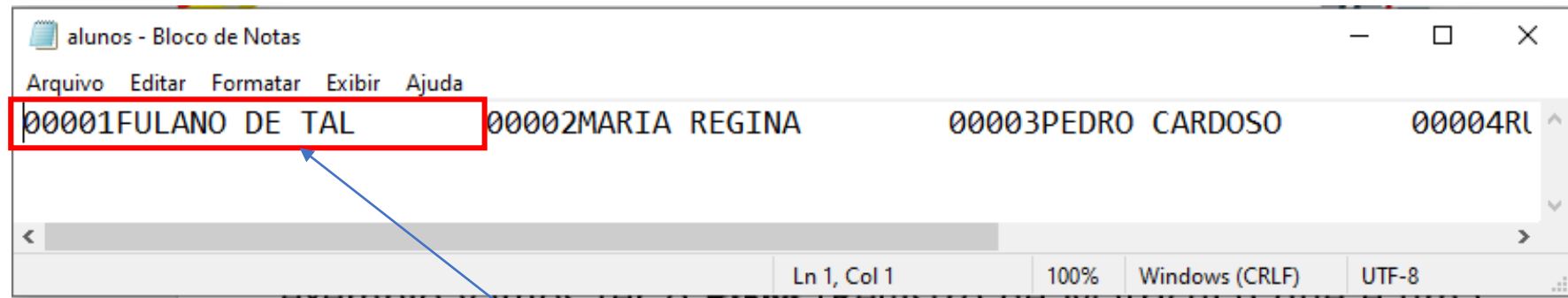


```
alunos - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
00001FULANO DE TAL      00002MARIA REGINA      00003PEDRO CARDOSO      00004RI
Ln 1, Col 1      100%      Windows (CRLF)      UTF-8
```

Introdução ao COBOL

Vamos configurar a nossa ENVIROMENTO DIVISION.

```
PROGRAMA_EX06.cbl
1 *****
2 * Author: CLAUDIO BENOSSI
3 * Date: 06/09/2021
4 * Purpose: TRABALHANDO COM ARQUIVOS DE DADOS
5 * Tectonics: cobc
6 *****
7 IDENTIFICATION DIVISION.
8 PROGRAM-ID. PROGRAMA_EX06.
9
10 ENVIRONMENT DIVISION.
11 INPUT-OUTPUT SECTION.
12 FILE-CONTROL.
13 SELECT ALUNOS ASSIGN TO 'C:\programas\ARQUIVOS\alunos.txt'
14 ORGANIZATION IS SEQUENTIAL.
15
```



Agora a DATA DIVISION.

PROGRAMA_EX06.cbl

```
16 DATA DIVISION.  
17 FILE SECTION.  
18 ***** DESCRIÇÃO DO ARQUIVO *****  
19 FD ALUNOS.  
20 01 ALUNOS-FILE.  
21 03 RGM-ALUNO PIC 9(05).  
22 03 NOME-ALUNO PIC A(20).  
23  
24 WORKING-STORAGE SECTION.  
25 ***** CRIAR UMA VARIÁVEL QUE CONTÉM A NOSSA "TRIPA DE DADOS" *****  
26 01 WS-DADOS-ALUNO PIC X(25) VALUE SPACES.  
27 01 FILLER REDEFINES WS-DADOS-ALUNO.  
28 03 WS-RGM-ALUNO PIC 9(05).  
29 03 WS-NOME-ALUNO PIC A(20).  
30  
31 ***** VARIÁVEL QUE VAI AJUDAR A CONTROLAR STATUS DO ARQUIVO *****  
32 77 WS-STATUS-EOF PIC A VALUE SPACE.  
33
```

Descrição do formato do arquivo

Introdução ao COBOL

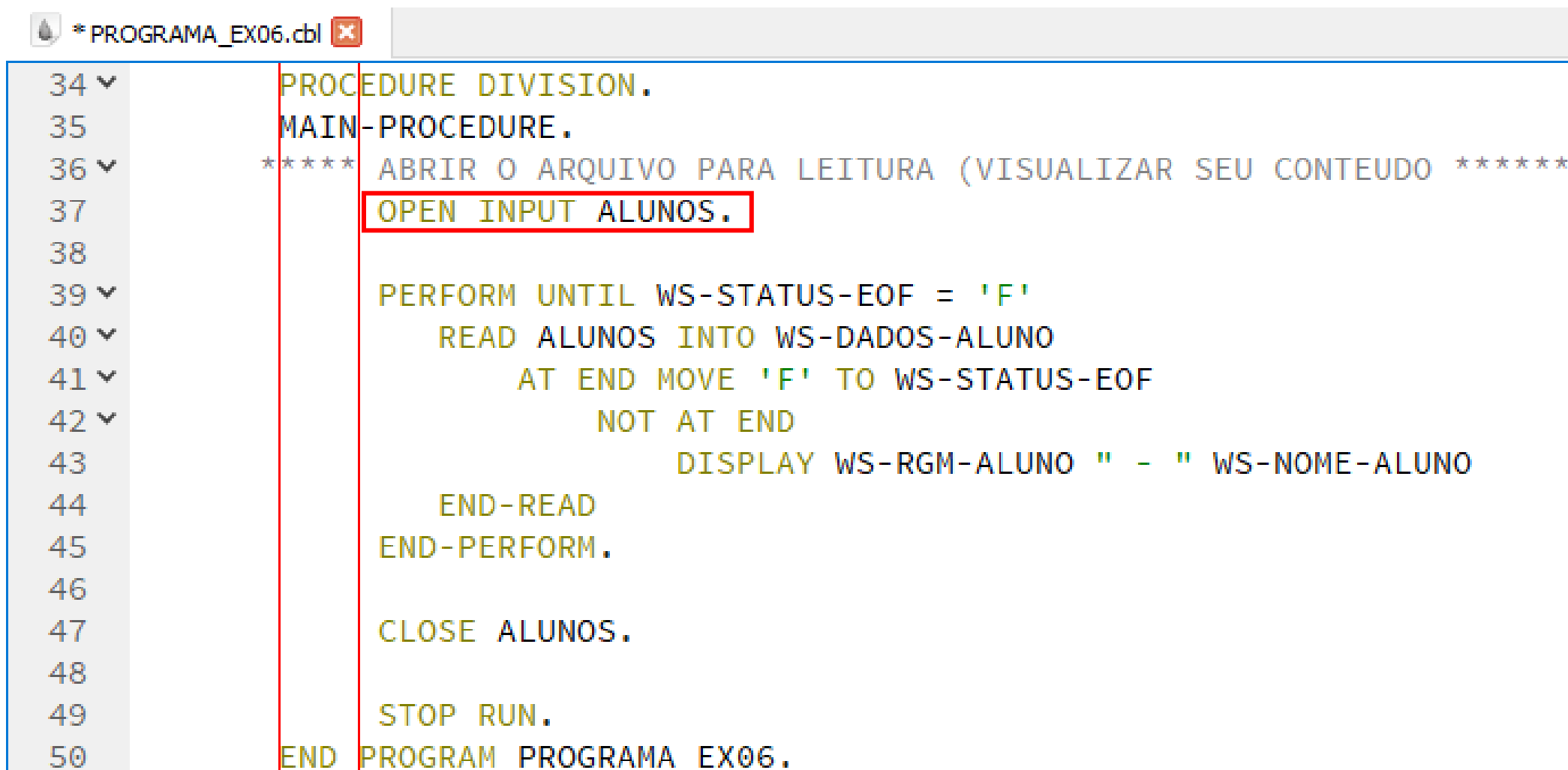
Vamos inserir nossa lógica na PROCEDURE DIVISION.

* PROGRAMA_EX06.cbl

```
34  PROCEDURE DIVISION.  
35  MAIN-PROCEDURE.  
36  ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****  
37  OPEN INPUT ALUNOS.  
38  
39  PERFORM UNTIL WS-STATUS-EOF = 'F'  
40      READ ALUNOS INTO WS-DADOS-ALUNO  
41      AT END MOVE 'F' TO WS-STATUS-EOF  
42      NOT AT END  
43          DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO  
44      END-READ  
45  END-PERFORM.  
46  
47  CLOSE ALUNOS.  
48  
49  STOP RUN.  
50  END PROGRAM PROGRAMA_EX06.
```

Introdução ao COBOL

Abrir o arquivo, para que possa ser feita a leitura.



```
* PROGRAMA_EX06.cbl
34  PROCEDURE DIVISION.
35  MAIN-PROCEDURE.
36  ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****
37  OPEN INPUT ALUNOS.
38
39  PERFORM UNTIL WS-STATUS-EOF = 'F'
40    READ ALUNOS INTO WS-DADOS-ALUNO
41    AT END MOVE 'F' TO WS-STATUS-EOF
42    NOT AT END
43      DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO
44  END-READ
45  END-PERFORM.
46
47  CLOSE ALUNOS.
48
49  STOP RUN.
50  END PROGRAM PROGRAMA_EX06.
```


Introdução ao COBOL

Ler o arquivo e inserir as informações na variável

* PROGRAMA_EX06.cbl

```
34  ▼ PROCEDURE DIVISION.  
35  MAIN-PROCEDURE.  
36  ▼ ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****  
37  OPEN INPUT ALUNOS.  
38  
39  ▼ PERFORM UNTIL WS-STATUS-EOF = 'F'  
40  ▼ READ ALUNOS INTO WS-DADOS-ALUNO  
41  ▼ AT END MOVE 'F' TO WS-STATUS-EOF  
42  ▼ NOT AT END  
43  DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO  
44  END-READ  
45  END-PERFORM.  
46  
47  CLOSE ALUNOS.  
48  
49  STOP RUN.  
50  END PROGRAM PROGRAMA_EX06.
```

Introdução ao COBOL

Pergunta se é o final do arquivo se sim inserir 'F' na variável de controle

* PROGRAMA_EX06.cbl

```
34  ▼ PROCEDURE DIVISION.  
35  MAIN-PROCEDURE.  
36  ▼ ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****  
37  OPEN INPUT ALUNOS.  
38  
39  ▼ PERFORM UNTIL WS-STATUS-EOF = 'F'  
40  ▼     READ ALUNOS INTO WS-DADOS-ALUNO  
41  ▼     AT END MOVE 'F' TO WS-STATUS-EOF  
42  ▼     NOT AT END  
43         DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO  
44     END-READ  
45     END-PERFORM.  
46  
47     CLOSE ALUNOS.  
48  
49     STOP RUN.  
50 END PROGRAM PROGRAMA_EX06.
```

Introdução ao COBOL

Caso não seja o fim do arquivo, exibir o conteúdo que foi transferido para as variáveis

* PROGRAMA_EX06.cbl

```
34  ▼ PROCEDURE DIVISION.
35      MAIN-PROCEDURE.
36  ▼      ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****
37          OPEN INPUT ALUNOS.
38
39  ▼          PERFORM UNTIL WS-STATUS-EOF = 'F'
40  ▼              READ ALUNOS INTO WS-DADOS-ALUNO
41  ▼              AT END MOVE 'F' TO WS-STATUS-EOF
42  ▼              NOT AT END
43                  DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO
44          END-READ
45      END-PERFORM.
46
47      CLOSE ALUNOS.
48
49      STOP RUN.
50  END PROGRAM PROGRAMA_EX06.
```

Introdução ao COBOL

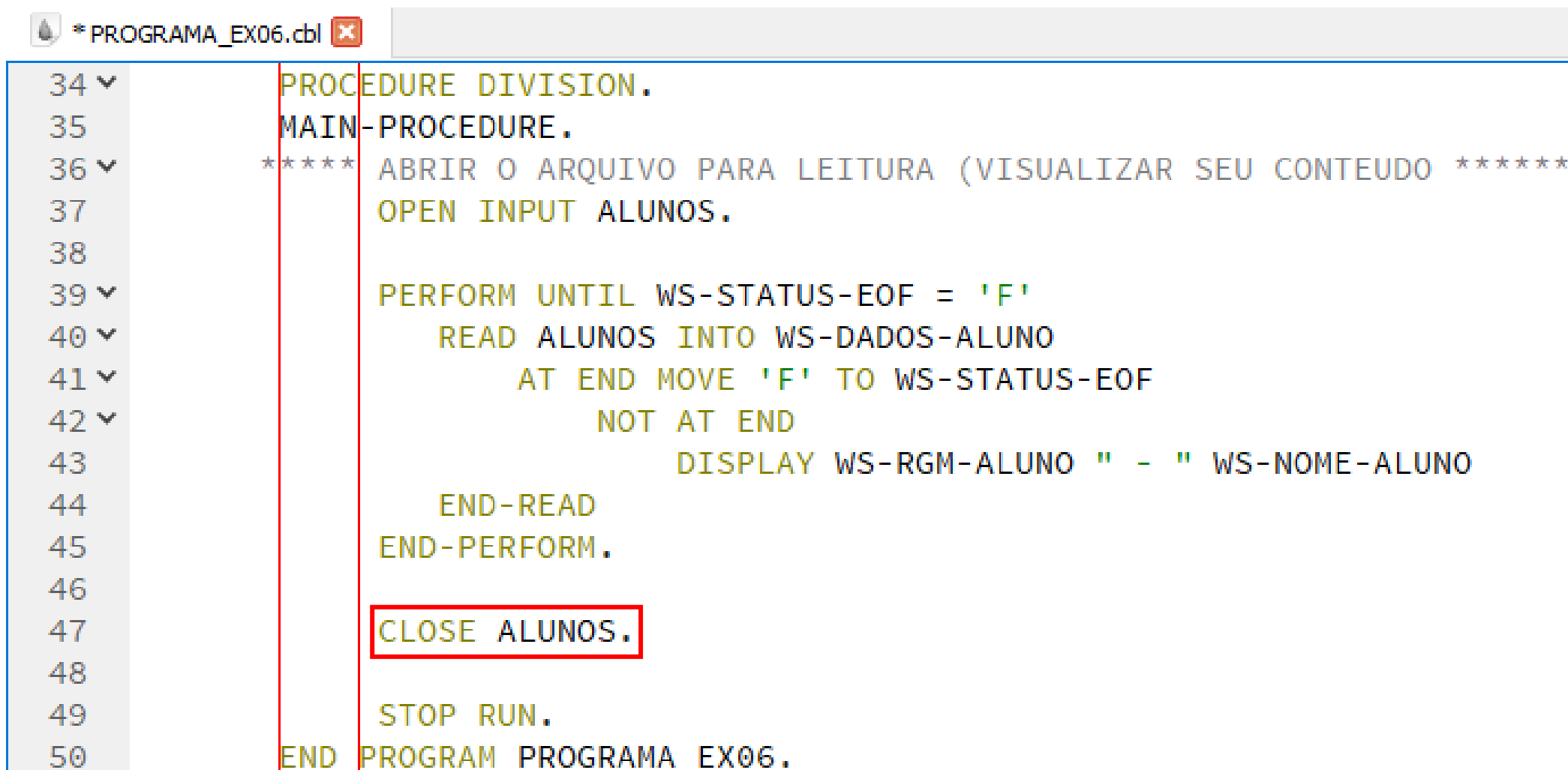
Encerra a leitura do arquivo

* PROGRAMA_EX06.cbl

```
34  ▼ PROCEDURE DIVISION.  
35      MAIN-PROCEDURE.  
36  ▼      ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****  
37          OPEN INPUT ALUNOS.  
38  
39  ▼          PERFORM UNTIL WS-STATUS-EOF = 'F'  
40  ▼              READ ALUNOS INTO WS-DADOS-ALUNO  
41  ▼              AT END MOVE 'F' TO WS-STATUS-EOF  
42  ▼              NOT AT END  
43                  DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO  
44          END-READ  
45          END-PERFORM.  
46  
47          CLOSE ALUNOS.  
48  
49          STOP RUN.  
50      END PROGRAM PROGRAMA_EX06.
```

Introdução ao COBOL

Fecha o arquivo, evitando que ele seja corrompido.



The screenshot shows a COBOL program editor window titled '* PROGRAMA_EX06.cbl'. The program code is as follows:

```
34  PROCEDURE DIVISION.  
35  MAIN-PROCEDURE.  
36  ***** ABRIR O ARQUIVO PARA LEITURA (VISUALIZAR SEU CONTEUDO) *****  
37  OPEN INPUT ALUNOS.  
38  
39  PERFORM UNTIL WS-STATUS-EOF = 'F'  
40      READ ALUNOS INTO WS-DADOS-ALUNO  
41      AT END MOVE 'F' TO WS-STATUS-EOF  
42      NOT AT END  
43          DISPLAY WS-RGM-ALUNO " - " WS-NOME-ALUNO  
44      END-READ  
45  END-PERFORM.  
46  
47  CLOSE ALUNOS.  
48  
49  STOP RUN.  
50  END PROGRAM PROGRAMA_EX06.
```

The line `CLOSE ALUNOS.` on line 47 is highlighted with a red rectangular box.

Introdução ao COBOL

OK funcionou!

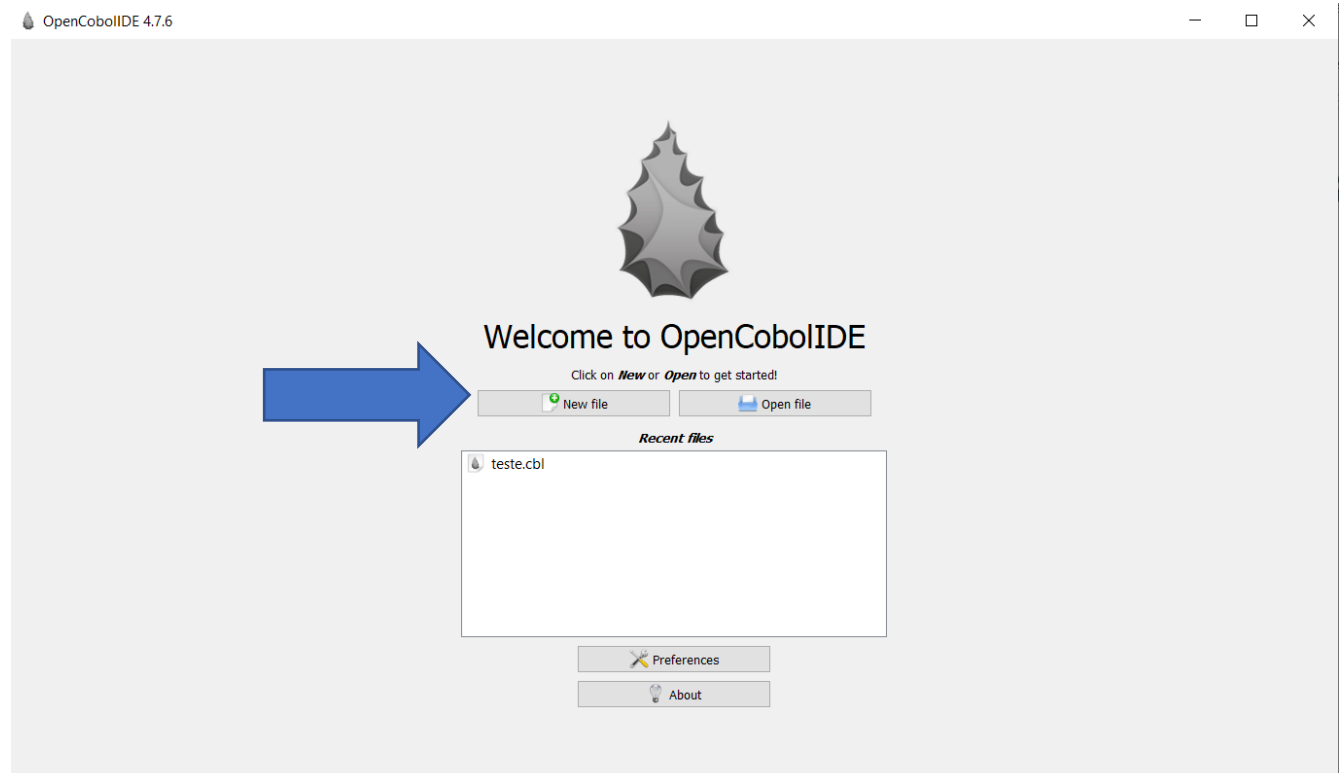


Introdução ao COBOL

Agora que já conseguimos fazer a leitura dos dados de um arquivo, vamos criar outro programa para poder criar o arquivo, caso ele não exista ainda e inserir dados nesse arquivo.

Então vamos criar no
OpenCobolIDE

PROGRAMA_EX07.cbl



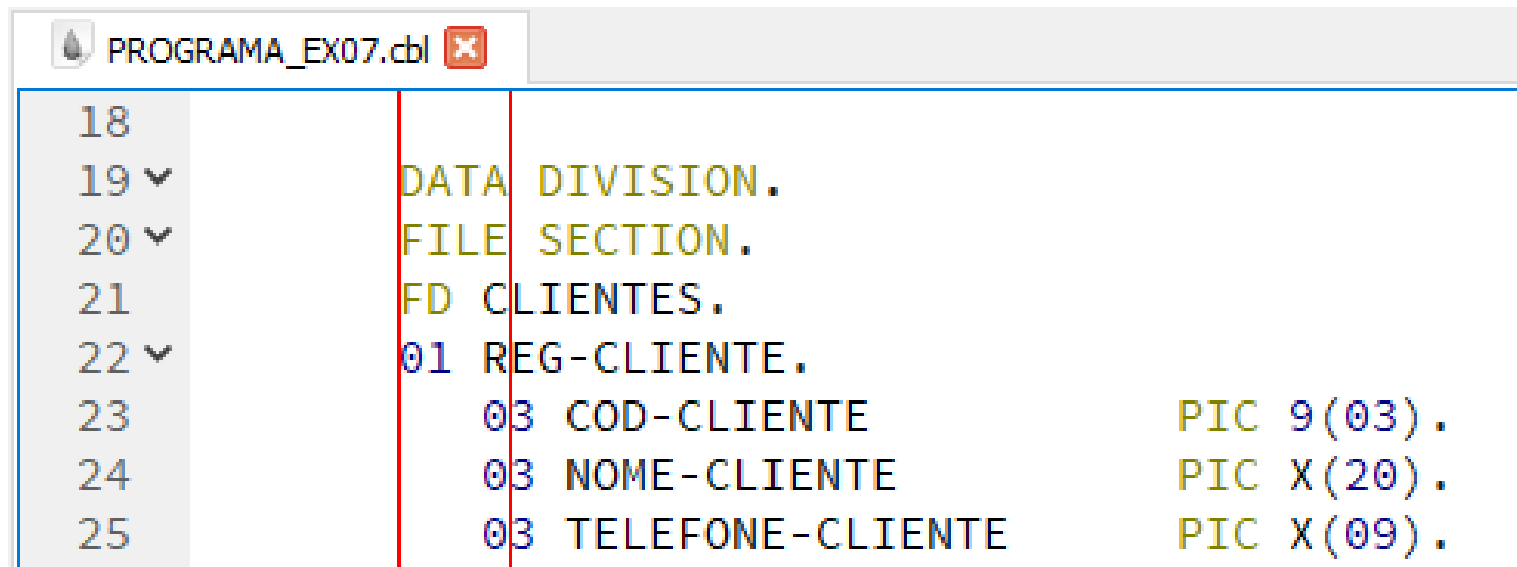
Introdução ao COBOL

Desta vez não vamos criar o arquivo e nem inserir dados manualmente, só vamos indicar o endereço e o nome do arquivo.

```
PROGRAMA_EX07.cbl x
1 *****
2 * Author: CLAUDIO BENOSSI
3 * Date: 06/09/2021
4 * Purpose: TRABALHANDO COM ARQUIVOS DE DADOS
5 * Tectonics: cobc
6 *****
7 IDENTIFICATION DIVISION.
8 PROGRAM-ID. PROGRAMA_EX07.
9
10 ENVIRONMENT DIVISION.
11 INPUT-OUTPUT SECTION.
12 FILE-CONTROL.
13 SELECT CLIENTES ASSIGN TO
14     'C:\programas\ARQUIVOS\clientes.txt'
15 ORGANIZATION IS SEQUENTIAL
16 ACCESS MODE IS SEQUENTIAL
17 FILE STATUS IS WS-FS.
18
```


Introdução ao COBOL

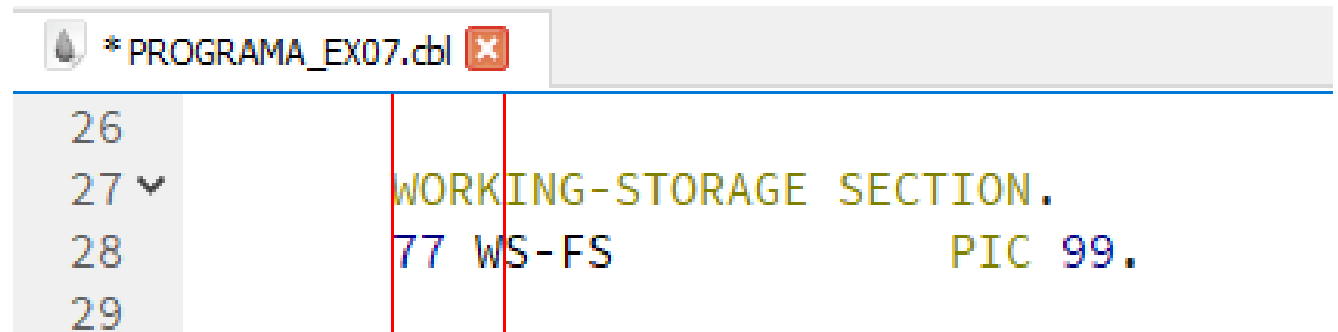
Aqui temos a
descrição do nosso
arquivo.



```
18  
19  DATA DIVISION.  
20  FILE SECTION.  
21  FD CLIENTES.  
22  01 REG-CLIENTE.  
23     03 COD-CLIENTE          PIC 9(03).  
24     03 NOME-CLIENTE         PIC X(20).  
25     03 TELEFONE-CLIENTE     PIC X(09).
```

Introdução ao COBOL

Na WORKING-STORAGE SECTION, vamos criar uma variável que vai assumir o valor do **FILE STATUS**.



```
*PROGRAMA_EX07.cbl  
26  
27  
28 WORKING-STORAGE SECTION.  
29 77 WS-FS PIC 99.
```

Introdução ao COBOL

- Vamos iniciar a variável como FILE STATUS E ATRIBUIR O VALOR DE 0;
- Solicitar a abertura do arquivo;
 - Verificar se o arquivo não existe vamos cria-lo.

```
* PROGRAMA_EX07.cbl
31  ▾  PROCEDURE DIVISION.
32  ▾  MAIN-PROCEDURE.
33      DISPLAY "----- CADASTRO DE CLIENTE -----"
34
35      SET WS-FS      TO 0.
36
37      OPEN EXTEND CLIENTES
38
39  ▾  IF WS-FS EQUAL 35 THEN
40      OPEN OUTPUT CLIENTES
41  END-IF
42
```

Introdução ao COBOL

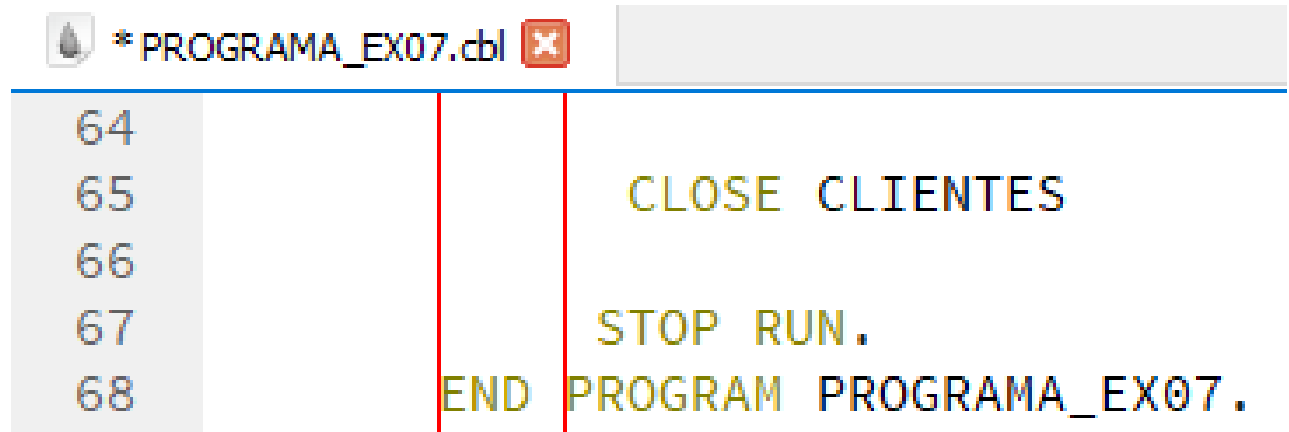
- Se o FILE STATUS = 0 inserir os dados e gravar no arquivo;
- Caso o FILE STATUS seja diferente de 0 envia mensagem de ERRO;

```

*PROGRAMA_EX07.cbl
43  IF WS-FS EQUAL ZEROS
44      DISPLAY 'INFORME O CODIGO DO CLIENTE:'
45      ACCEPT COD-CLIENTE
46      DISPLAY 'INFORME O NOME DO CLIENTE'
47      ACCEPT NOME-CLIENTE
48      DISPLAY 'INFORME O TELEFONE DO CLIENTE'
49      ACCEPT TELEFONE-CLIENTE
50      ***** COMANDO PARA INSERIR DADOS NO ARQUIVO *****
51      WRITE REG-CLIENTE
52
53      ***** VERIFICANDO SE O REGISTRO FOI REALIZADO *****
54      IF WS-FS NOT EQUAL ZEROS
55          DISPLAY 'ERRO - NÃO FOI POSSIVEL GRAVAR O REGISTRO'
56          DISPLAY 'FILE STATUS: ' WS-FS
57      ELSE
58          DISPLAY 'REGISTRO GRAVADO COM SUCESSO!'
59      END-IF
60      ELSE
61          DISPLAY 'ERRO AO CRIAR O ARQUIVO'
62          DISPLAY 'FILE STATUS: ' WS-FS
63      END-IF
    
```

Introdução ao COBOL

Fechar o Arquivo;



```
64  
65      CLOSE CLIENTES  
66  
67      STOP RUN.  
68  END PROGRAM PROGRAMA_EX07.
```

```

1 *****
2 * Author: CLAUDIO BENOSSE
3 * Date: 06/09/2021
4 * Purpose: TRABALHANDO COM ARQUIVOS DE DADOS
5 * Tectonics: cobc
6 *****
7 IDENTIFICATION DIVISION.
8 PROGRAM-ID. PROGRAMA_EX07.
9 ENVIRONMENT DIVISION.
10 INPUT-OUTPUT SECTION.
11 FILE-CONTROL.
12 SELECT CLIENTES ASSIGN TO
13     'C:\programas\ARQUIVOS\clientes.txt'
14     ORGANIZATION      IS SEQUENTIAL
15     ACCESS MODE       IS SEQUENTIAL
16     FILE STATUS       IS WS-FS.
17 DATA DIVISION.
18 FILE SECTION.
19 FD CLIENTES.
20 01 REG-CLIENTE.
21     03 COD-CLIENTE          PIC 9(03).
22     03 NOME-CLIENTE         PIC X(20).
23     03 TELEFONE-CLIENTE     PIC X(09).
24 WORKING-STORAGE SECTION.
25 77 WS-FS                   PIC 99.

```

```

26 PROCEDURE DIVISION.
27 MAIN-PROCEDURE.
28     DISPLAY "----- CADASTRO DE CLIENTE -----"
29     SET WS-FS      TO 0.
30     OPEN EXTEND CLIENTES
31     IF WS-FS EQUAL 35 THEN
32         OPEN OUTPUT CLIENTES
33     END-IF
34     IF WS-FS EQUAL ZEROS
35         DISPLAY 'INFORME O CODIGO DO CLIENTE:'
36         ACCEPT COD-CLIENTE
37         DISPLAY 'INFORME O NOME DO CLIENTE'
38         ACCEPT NOME-CLIENTE
39         DISPLAY 'INFORME O TELEFONE DO CLIENTE'
40         ACCEPT TELEFONE-CLIENTE
41 ***** COMANDO PARA INSERIR DADOS NO ARQUIVO *****
42 WRITE REG-CLIENTE
43 ***** VERIFICANDO SE O REGISTRO FOI REALIZADO *****
44     IF WS-FS NOT EQUAL ZEROS
45         DISPLAY 'ERRO - NÃO FOI POSSIVEL GRAVAR O REGISTRO'
46         DISPLAY 'FILE STATUS: ' WS-FS
47     ELSE
48         DISPLAY 'REGISTRO GRAVADO COM SUCESSO!'
49     END-IF
50     ELSE
51         DISPLAY 'ERRO AO CRIAR O ARQUIVO'
52         DISPLAY 'FILE STATUS: ' WS-FS
53     END-IF
54     CLOSE CLIENTES
55     STOP RUN.
56 END PROGRAM PROGRAMA_EX07.

```

Introdução ao COBOL

OK funcionou!





Introdução ao COBOL

Vamos praticar.

Desenvolva um programa que leia as informações dos fornecedores (Código do Fornecedor, Nome e Telefone e exiba uma listagem com esses dados.



"Coragem é ir de falha
em falha sem perder o
entusiasmo"



Winston Churchill

Obrigado!

Se precisar ...

Prof. Claudio Benossi

claudio.benossi@fatec.sp.gov.br

