C.O.B.O.L. - Common Business Oriented Language - Linguagem de Programação Comum Orientada à negócios. Um programa COBOL é composto de 4(quatro) divisões:

1)IDENTIFICATION DIVISION - divisão que serve para identificar o programa-fonte, dados sobre o autor,data em que foi escrito,observações sobre o que o programa faz,e sobre a segurança. Sintaxe:

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  EXEMPLO.
AUTHOR. WILSON PEDRO CARLI.
DATE-WRITTEN. 01/08/1996.
REMARKS. ESTE PROGRAMA EH UM EXEMPLO.
SECURITY. NÃO EXECUTAR SEM TESTAR ANTES.
```

2)ENVIRONMENT DIVISION - esta divisão fornece informações relativas aos meios externos,ou seja, arquivos e equipamento.Define os arquivos a serem utilizados no programa,a sua organização,meio de acesso,chaves primárias e/ou secundárias.

Sintaxe:

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.  DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
   SELECT  nome-arquivo  ASSIGN TO   {DISK,PRINTER}

       [ ORGANIZATION  IS  { SEQUENTIAL,INDEXED,LINE SEQUENTIAL}

       [ACCESS  MODE  IS { SEQUENTIAL,DYNAMIC}

       [ RECORD KEY IS nome-chave-primaria ]

       [ ALTERNATE RECORD KEY IS nome-chave-secundaria
             [WITH  DUPLICATES ] ]

       [ FILE STATUS IS  nome-campo-file-status ]  .
```

Para cada arquivo a ser utilizado pelo programa, deverá haver uma cláusula SELECT.

ACCESS- define o método de acesso aos registros do arquivo.

Sequential=>leitura e gravação dos registros a partir do primeiro registro do arquivo até o final do mesmo.

Dynamic=> leitura,regravação,exclusão,gravação de registros através de um chave primária, ou secundária definida dentro do lay-out do arquivo.

ORGANIZATION- especifica a estrutura lógica do arquivo. A organização do arquivo é estabelecida no momento em que o arquivo é criado, e não pode ser modificada

subsequentemente.Quando a organização não é especificada, a organização sequencial é assumida na compilação. SEQUENTIAL=> os registros são criados através de programas cobol e armazenados na ordem em que foram gerados(disco,impressora,fita). LINE SEQUENTIAL => os registros são criados por editor de texto e armazenados na ordem em que foram criados. INDEXED=> os registros são identificados pelo conteúdo dos campos denominados “chaves”(primária ou secundária), e são armazenados em qualquer ordem na criação do registro, mas na leitura, obedece a ordem da chave e o método de acesso.
RECORD KEY - especifica a chave primária de um arquivo com organização indexada. É um campo definido no lay-out do registro do arquivo e deverá ser alfanumérica, e pode chegar até 250 bytes.
ALTERNATE KEY - especifica a chave secundária do arquivo, que é opcional para os arquivos com organização indexada. É também um campo definido no lay-out do registro do arquivo, com até 250 bytes alfanuméricos, mas tem a opção de aceitar valores duplicados.
FILE STATUS - campo alfanumérico de 2 bytes definido na WORKING_STORAGE da DATA DIVISION, que a cada operação efetuada com o arquivo, demonstra o valor da operação efetuada.

Valor	Descrição
00	Comando executado com sucesso
02	Comando arquivo c/chave duplicada executado com sucesso
10	Fim do arquivo
21	Erro de sequencia na gravação de arquivo sequencial
22	Chave duplicada não definida.
23	Chave não encontrada.
24	Espaço em disco está cheio
30	Arquivo não encontrado
91	Erro na estrutura do arquivo
94	Registro ou arquivo sendo utilizado por outro programa.

3) DATA DIVISION - armazena todos os dados a serem processados ou manipulados pelo programa, durante o processamento, podendo ser internos ou externos.Pode ser dividida em até seis seções, mas usualmente são utilizadas 3 seções:

FILE SECTION- seção que define a estrutura dos arquivos de dados. Esta definição envolve a descrição do arquivo e seus respectivos registros. Para cada SELECT definido temos uma definição de arquivo na FILE SECTION.

Sintaxe da File Section
DATA DIVISION.
FILE SECTION.
FD nome-arquivo
[RECORD CONTAINS nn CHARACTERS]

[LABEL RECORD IS { OMITTED, STANDARD }]

[VALUE OF FILE-ID valor-identificação-arquivo] .
01 nome-de-registro-arquivo .

[nro-nivel
[nome-campo ou FILLER]
[REDEFINES nome-de-dado]
[PIC tipo(tamanho)]
[OCCURS nro-inteiro TIMES]] .

RECORD CONTAINS => especifica o tamanho do registro de dados. O tamanho do registro é determinado pelo soma do número de caracteres de todos os itens elementares subordinados ao registro.
LABEL RECORD=> especifica se existe rótulo presente no arquivo. Omitted especifica que não existe rótulo explícito(arquivos de impressão). Standard especifica que existem rótulos e estão conforme as especificações do sistema operacional(disco).
VALUE OF FILE-ID=> identifica o nome do arquivo no meio externo.
NRO-NIVEL=> são números entre 01 e 49 que permitem a estruturação de um registro lógico, pela subdivisão deste registro. Uma vez que uma subdivisão tenha sido especificada(item de grupo), ela pode ser ainda mais subdividida(itens elementares), para permitir uma referencia mais detalhada. Item elementar é a subdivisão fundamental de um registro, que não é mais subdividido. Um registro pode ser constituido de uma sequencia de itens elementares ou pode ser somente um item elementar. Um item de grupo é uma sequencia de um ou mais itens elementares ou também de um ou mais itens de grupo.Uma descrição de um registro sempre começa pelo número de nível 01.
NOME-CAMPO=> nome definido pelo programador que não pode ser repetido dentro do fonte do programa, pode ter até 30(trinta) caracteres e não pode ser igual a alguma palavra reservada da sintaxe do COBOL.
FILLER=> palavra reservada do COBOL que serve para reservar uma determinada quantidade de bytes em um arquivo ou na memória.
REDEFINES=> cláusula utilizada para redefinir um item de grupo e/ou item elementar em partes menores ou em uma imagem diferente.

OCCURS=> cláusula que define a repetição do itens que estão após a definição da mesma. Um numero inteiro define quantas vezes o campo se repete contiguamente.
PIC=> cláusula que define a imagem do campo que está sendo descrito num item elementar. Os tipos existentes para a descrição de dados é a seguinte:
Tipo numérico- pode conter os símbolos 9,V, e S. O número de digitos permitidos varia de 01 até 18, sem considerar o sinal e a casa decimal.O símbolo V determina a separação dos inteiros dos decimais. O símbolo S determina a presença de sinal.
Tipo alfabético- pode conter o símbolo A. Somente para campos alfabéticos.
Tipo Alfanumérico- contém o símbolo X. Serve para a descrição de campos que contém a combinação de letras e números.
Tamanho - define quantos bytes ocupa o campo definido anteriormente. Se for do tipo numérico poderá ter no máximo 18 bytes.
WORKING-STORAGE SECTION -seção que descreve e armazena numa área de memória todos os dados,informações,variáveis e constantes, com valores definidos

ou não, a serem manipulados pelo programa. É composta de itens de grupo e itens elementares. Sintaxe:

WORKING-STORAGE SECTION.
01 nome-de-registro-arquivo .
[nro-nivel
[nome-campo ou FILLER]
[REDEFINES nome-de-dado]
[PIC tipo(tamanho)]
[OCCURS nro-inteiro TIMES]] .

A descrição das cláusulas acima são idênticas as descritas na FILE SECTION. O que muda são os tipos de pictures,que em alguns casos podem ser usadas em várias combinações:
Item alfanumérico editado- os tipos podem ser A - edição só de letras;
X- edição de itens alfanuméricos; B- insere um espaço no local indicado;
0 - insere um zero no local indicado; / - insere uma barra no local;
, (vírgula)- insere um vírgula no local ; . (ponto)- insere um ponto no local
Item numérico editado -
Z - representa números, e insere espaços à esquerda quando o digito contiver zero.
- (hífen)- representa números com sinal à esquerda e insere espaços à esquerda quando o digito contiver o valor zero.
Outras representações devem ser pesquisadas em livros e/ou manuais.

SCREEN SECTION
Extensão da DATA DIVISION que possui recursos para facilitar a formatação e descrição dos itens de tela e sua posterior manipulação na PROCEDURE DIVISION. Serve para :
— Especificar a posição exata na tela de determinados campos;
— Entrar com dados digitados em posições específicas;
— Mostrar valores literais em posições predeterminadas;
— Definir atributos de tela e controlar recursos de teclado.

FORMATO GERAL:
SCREEN SECTION.
01 nome-tela. CADA DESCRIÇÃO DE TELA DEVE INICIAR COM UM NÍVEL 01 E TER UM NOME ESPECIFICADO. OS DEMAIS ITENS INICIAM COM O NRO. DE NÍVEL 02 ATÉ 49.

[BLANK SCREEN] LIMPA TODA A TELA E POSICIONA O CURSOR NO INICIO
[BLANK LINE] LIMPA A LINHA DE ONDE SE ENCONTRA O CURSOR ATÉ O FINAL.
[BELL] DISPARA SOM DE ALARME DURANTE O ACCEPT DO CAMPO
[BLINK] FAZ COM QUE O CONTEÚDO DO CAMPO FIQUE PISCANTE
[HIGHLIGHT] FAZ COM QUE O CONTEÚDO DO CAMPO FIQUE BRILHANTE
[UNDERLINE] FAZ COM QUE O CONTEÚDO DO CAMPO APAREÇA SUBLINHADO
[BLANK WHEN ZERO] MOVE ESPAÇOS PARA UM ITEM NUMÉRICO QUANDO O SEU VALOR FOR IGUAL A ZEROS

[JUST RIGHT] SOMENTE PARA ITENS ELEMENTARES ALFANUMÉRICOS NÃO EDITADOS, POIS PROVOCA O ALINHAMENTO DO CONTEÚDO DO CAMPO DA DIREITA PARA A ESQUERDA QUANDO O

CAMPO RECEPTOR FOR MAIOR DO QUE O CAMPO FONTE. QUANDO O CAMPO RECEPTOR FOR MENOR TRUNCA O CAMPO EMISSOR A ESQUERDA.

[REVERSE-VÍDEO] TROCA A COR DA LETRA PELO FUNDO DA TELA E VICE-VERSA.
[AUTO] TERMINA AUTOMATICAMENTE UMA OPERAÇÃO DE ENTRADA DE DADOS NA PROCEDURE DIVISION QUANDO A ÚLTIMA POSIÇÃO DO CARACTERE É PREENCHIDA.

[SECURE] NÃO APARECEM OS CARACTERES NO MOMENTO DA DIGITAÇÃO
[REQUIRED] OBRIGA O CAMPO A SER TOTALMENTE PREENCHIDO.
[FULL] O ITEM DA TELA DEVE ESTAR TOTALMENTE PREENCHIDO OU VAZIO

[LINE número is [{ PLUS} { + } { — } { inteiro}]] ESPECIFICA A LINHA DA TELA NA QUAL O ITEM DEVERÁ INICIAR. O NÚMERO DE LINHA DEVE ESTAR SEMPRE ENTRE 1 E 255. CASO O NÚMERO SEJA OMITIDO SERÁ ASSUMIDO O VALOR DEFAULT=01. PLUS OU + DEFINE A POSIÇÃO CORRENTE DO CURSOR ACRESCIDO DE HUM OU MAIS QUANDO O INTEIRO FOR DEFINIDO. SE FOR — PROCEDE A SUBTRAÇÃO.
[COLUMN número IS [{PLUS} { + } { — } { inteiro}]] ESPECIFICA A COLUNA DA TELA NA QUAL O ITEM DEVERÁ INICIAR. O NÚMERO DEVE ESTAR SEMPRE ENTRE 1 E 255. CASO O NÚMERO SEJA OMITIDO SERÁ ASSUMIDO O VALOR DEFAULT=01. PLUS OU + DEFINE A POSIÇÃO CORRENTE DO CURSOR ACRESCIDO DE HUM OU MAIS QUANDO O INTEIRO FOR DEFINIDO. SE FOR — PROCEDE A SUBTRAÇÃO.

[PIC IS descrição [FROM { literal } {campo}] [TO campo] [USING campo]
Descrição — informar o tipo e tamanho da picture naquela posição da tela;
FROM — valor do literal ou campo é mostrado na tela;
TO — valor digitado será repassado para o campo ;
USING — demonstra o valor do campo e também atualiza o conteúdo do mesmo após a digitação.

PROCEDURE DIVISION - SINTAXE DOS COMANDOS:

ACCEPT - COMANDO PARA PEGAR AS INFORMAÇÕES VIA TECLADO
ACCEPT campo FROM {DAY} . FORMATO AADDD
{DATE}. FORMATO AAMMDD
{TIME}. FORMATO HHMMSSCC
ACCEPT campo.

ADD - SOMA VALORES
ADD campo [campo-2] [...] TO campo-x [...] [ROUNDED] [ON SIZE ERROR procedimento] .
ADD campo [...] GIVING campo-x [ROUNDED] [ON SIZE ERROR procedimento] .

COMPUTE - CALCULA UMA EXPRESSÃO ARITMÉTICA
COMPUTE campo [ROUNDED] = expressão aritmética.
OPERANDOS => ( ) PARÊNTESES; + ADIÇÃO; - SUBTRAÇÃO
* MULTIPLICAÇÃO ; / DIVISÃO; ** EXPONENCIAÇÃO
ORDEM DAS OPERAÇÕES= 1) ENTRE PARÊNTESES; 2) EXPONENCIAÇÃO;
3) MULTIPLICAÇÃO E DIVISÃO; 4) ADIÇÃO E SUBTRAÇÃO.

DISPLAY - DEMONSTRA MENSAGEM E/OU CAMPOS NO VÍDEO
DISPLAY {literal} [campo] [,...] .

DIVIDE - EXECUTA A DIVISÃO DE VALORES
DIVIDE {campo} {inteiro} INTO campo-2 [ROUNDED] [ON SIZE ERROR procedimento] .
DIVIDE {campo} {inteiro} BY {campo-2} {inteiro} GIVING campo-x [ROUNDED]
[REMAINDER campo-resto] [ON SIZE ERROR procedimento] .

IF - TESTA CONDIÇÕES DENTRO DO PROGRAMA
IF condição {procedimento} THEN SIGNIFICADO

[NEXT SENTENCE]
[ELSE procedimento]
[ELSE NEXT SENTENCE] .

= equal to (igual)
< less than (menor do que)
> greater than (maior do que)
NOT = not equal to (diferente-não igual)
NOT < greater than or equal to (maior ou igual)
NOT > less than or equal to (menor ou igual)
OR ou
AND e

GO TO - TRANSFERE CONTROLE DO PROGRAMA PARA O PARÁGRAFO ESPECIFICADO.
GO TO nome-parágrafo.

MOVE - MOVE DADOS DE UMA ÁREA PARA OUTRA DA MEMÓRIA.
MOVE {inteiro} {literal} {campo} TO campo [campo-2,...] .

MULTIPLY - EFETUA A MULTIPLICAÇÃO DE VALORES.
MULTIPLY {campo} {inteiro} BY campo-x [ROUNDED] [ON SIZE ERROR procedimento].
MULTIPLY {campo} {int} BY campo-2 GIVING campo-x [ROUNDED] [ON SIZE ERROR proc.] .

PERFORM-EXECUTA PARTES SEPARADAS DO CORPO PRINCIPAL DO PROGRAMA.
PERFORM {paragrafo} {seção} [TRHU {parag-2} {seção-2}]
PERFORM {paragrafo} {seção} {inteiro} {campo} TIMES.
PERFORM {paragrafo} {seção} UNTIL condição.
PERFORM {paragrafo} {seção} VARYING {campo} {indexador}
FROM {inteiro} {indexador} {campo} BY {inteiro} {campo} UNTIL condição
[AFTER VARYING {indx} {campo} FROM {int} {campo} {indx} BY {int} {campo} UNTIL cond]
[AFTER VARYING {indx} {campo} FROM {int} {campo} {indx} BY {int} {campo} UNTIL cond] .

STOP - PARA A EXECUÇÃO DO PROGRAMA.
STOP {RUN} finaliza a execucao do programa.
STOP { literal} para a execução e demonstra mensagem, aguardando decisão do operador.

SUBTRACT - SUBTRAI VALORES
SUBTRACT {inteiro} {campo} FROM campo-2 [ROUNDED] [ON SIZE ERROR procedimento] .
SUBTRACT {int.} {cmp} FROM {int.-2} {cmp-2} GIVING cmp-x [ROUNDED] [ON SIZE ERROR proc] .

COMANDOS PARA TRATAMENTO DE ARQUIVOS

OPEN - ABERTURA DE ARQUIVO DE ACORDO COM O MODO DE ACESSO.
OPEN {INPUT } nome-arquivo [,...] .=> SO LEITURA
{OUTPUT } => SO GRAVAÇÃO
{I-O } => LEITURA,GRAVAÇÃO,EXCLUSÃO E REGRAVAÇÃO.

CLOSE - FECHA OS ARQUIVOS: CLOSE nome-arquivo [,...] .

READ - PROCEDE A LEITURA DO PRÓXIMO REGISTRO LÓGICO DISPONÍVEL.
ARQUIVO SEQUENCIAL: READ arquivo [NEXT] RECORD [AT END procedimento] .
ARQUIVO INDEXADO: READ arquivo RECORD [KEY IS nome-chave-sec] [INVALID KEY [proc.] .

WRITE - LIBERA O REGISTRO LÓGICO PARA A GRAVAÇÃO FÍSICA.
ARQUIVO DE IMPRESSÃO
WRITE nome-registro [FROM campo] [{BEFORE} {AFTER} ADVANCING {PAGE} {inteiro}].
ARQUIVO SEQUENCIAL : WRITE nome-registro [FROM campo].
ARQUIVO INDEXADO : WRITE nome-registro [FROM campo] [INVALID KEY procedimento].

REWRITE - REGRAVA O ÚLTIMO REGISTRO LÓGICO LIDO.
REWRITE nome-registro [FROM campo] [INVALID KEY procedimento] .

DELETE - REMOVE O ÚLTIMO REGISTRO LIDO NO ARQUIVO.
DELETE nome-arquivo [INVALID KEY procedimento].

START - POSICIONA ARQUIVO P/ LEITURA DE ACORDO COM O VALOR
DA CHAVE ESPECIFICADA
START nome-arquivo [KEY {IS EQUAL TO} {IS GREATER THAN }
{IS NOT LESS THAN} nome-chave] [INVALID KEY procedimento].