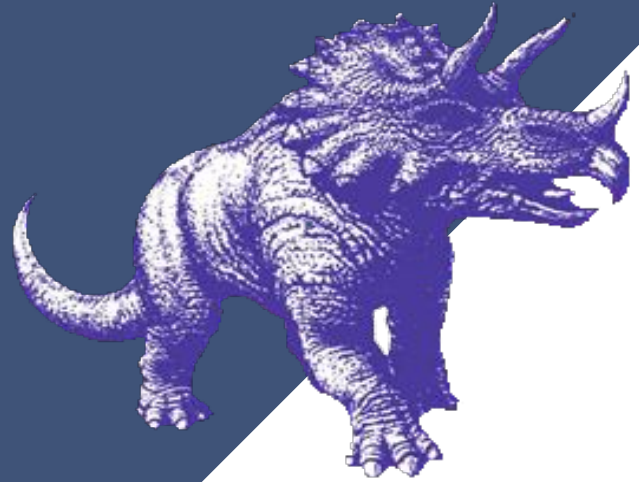


COBOL

COmmon Business Oriented Language





QUEM SOMOS



LUCAS COSTA

Sistemas de Informação
5º Período



PABLO HENRIQUE

Sistemas de Informação
6º Período



THIAGO APARECIDO

Sistemas de Informação
5º Período

Senhor do Destino

Super Herói





HISTÓRIA

- COBOL (sigla de COmmon Business Oriented Language) - Linguagem Comum Orientada para os Negócios
- Criada em 1959 por Grace Hopper, no comando do Projeto no departamento de defesa americano.
- Seu objetivo principal atender a sistemas comerciais, financeiros e administrativos para empresas e governos.





INFLUÊNCIAS





PARADIGMAS

- Imperativo
 - ▷ Expressam uma sequência de comandos para um computador
- Estruturado
 - ▷ Sequência: tarefa executada linearmente
 - ▷ Decisão: executa um trecho de código, a partir de teste lógico
 - ▷ Iteração: repete um trecho de código, a partir de um teste lógico



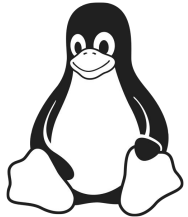
CURIOSIDADE

O tamanho do programa (a quantidade de linhas de código) é praticamente ilimitado. Existem programas em COBOL, que variam de 10.000 à 20.000 linhas de código.

Parece muito? Com o Cobol você pode chegar até um milhão de linhas.



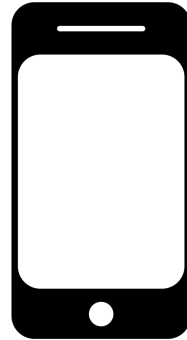
PORTABILIDADE



Linux™



UNIX





EMPRESAS QUE UTILIZAM O COBOL





START COBOL



OPENCOBOL IDE



HELLO WORLD

IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.

PROCEDURE DIVISION.
 DISPLAY "HELLO WORLD!"
 STOP RUN.

END PROGRAM HELLO-WORLD.

REGRAS DE UM PROGRAMA COBOL



REGRAS DE CODIFICAÇÃO

Colunas	Definição
1 a 6	Reservado para números de linha.
7	Ele pode ter asterisco (*) indicando comentários e hífen (-) continuação de literal não numérico.
8 a 11	Aqui vão as divisões, seções e parágrafos.
12 a 72	Os comandos propriamente ditos devem ficar nessa área.

Pode-se digitar até a coluna 80 mas tudo que for digitado entre 73 e 80, será considerado como um comentário.

DIVISÕES



IDENTIFICATION DIVISION

É a divisão de identificação do programa na qual se fornece informações sobre o programa, tais como nome do programa, autor, comentários do autor e informações de uso para o usuário final.



ENVIRONMENT DIVISION

É a divisão de ambiente, descreve o computador e os periféricos que serão utilizados pelo programa, fazendo ligação com o ambiente operacional onde o programa será executado.



DATA DIVISION

Essa é a divisão de dados, na qual se declaram as variáveis, as constantes e todos os tipos de dado que irão alocar memória no decorrer do processo.



PROCEDURE DIVISION

É a seção na qual se colocam os procedimentos que manipulam os dados contidos na DATA DIVISION. Esta divisão possui uma estrutura hierárquica dividida em seções, parágrafos, sentenças e comandos.

VARIÁVEIS



TIPOS DE DADOS

- Tipos de dados:
 - ▷ Numérico
 - ▷ Alfabético
 - ▷ Alfanumérico
- **PIC [PICTURE]** identifica o tipo e o tamanho da variável. Todas as variáveis e constantes utilizadas devem ser declaradas dentro da DATA DIVISION na seção WORKING-STORAGE.



DECLARANDO VARIÁVEIS

Sintaxe:

<nível> <identificador> <tipo> <valor>.

01 NOME PIC A(5) VALUE 'BRASIL'.

alfabética representado pelo caractere "A" de 5 posições.

01 NUM PIC 9(3) VALUE 123.

numérico representado pelo caractere "9" de 3 posições.

01 PALAVRA PIC X(6) VALUE 'A\$E&8b'.

alfanumérico representado pelo caractere "X" de 6 posições.



REGRAS DE FORMAÇÃO

As regras para nomes de variáveis são:

- No máximo 30 caracteres
- Não pode conter espaços ou brancos
- Pode começar com um número, mas precisa ter pelo menos um caractere alfabético
- Não podem ser utilizadas palavras reservadas da linguagem
- Não pode conter outros caracteres senão A-Z, 0-9 e o hífen (-)



NÍVEIS DE DADOS

Em COBOL, é possível hierarquizar as variáveis, a fim de facilitar a vida dos usuários do sistema. Os números de nível vão de 01 a 49 e os números 66, 77, 78 e 88 são níveis especiais que permitem um uso diferenciado para as variáveis.



NÍVEL 88

Nível 88 é utilizado para definição de condição de variáveis, sua utilização é opcional, mas costuma tornar o código muito mais legível.

Exemplo com Nível 88:

```
77 HORA PIC 9(2).
```

```
88 HORA-OK VALUES 0 THRU 23.
```

```
IF NOT HORA-OK
```

```
    DISPLAY "HORA INVALIDA !!"
```

```
END-IF
```

Exemplo sem Nível 88:

```
77 HORA PIC 9(2).
```

```
IF HORA > 23
```

```
    DISPLAY "HORA INVALIDA !!"
```

```
END-IF
```



COMANDOS BÁSICOS

Comandos	Descrição
ACCEPT	Entrada de dados do usuário
MOVE	Atribuição de valores
DISPLAY	Exibir saída de um programa

PALAVRAS RESERVADAS



CONSTANTES FIGURATIVAS

São literais geradas pelo compilador e usados como palavras reservadas.

ZERO, ZEROS, ZEROES - Representa o valor numérico "zero" ou uma ou mais ocorrências do caracter 0.

SPACE, SPACES - espaço em branco.

HIGH-VALUE, HIGH-VALUES - Representa um ou mais caracteres com valores-altos. Usualmente é o hexadecimal "FF".

LOW-VALUE, LOW-VALUES Representa um ou mais caracteres com valores-baixos. Usualmente é o binário 0.

ALL Literal - Representa um conjunto de caracteres pré definido.



CONDIÇÕES DE CLASSE

A condição de classe é usada para testes onde se deseja saber se uma variável é formada ou não de um tipo particular de dados.

NUMERIC Numérico, caracteres de 0 a 9.

ALPHABETIC Alfabético, caracteres de A - Z, de a - z e espaços.

ALPHABETIC-UPPER Alfabético, caracteres de A - Z, e espaços.

ALPHABETIC-LOWER Alfabético, caracteres de a - z, e espaços.

COMANDO ARITMÉTICOS



ADIÇÃO - ADD

Utilizado para adicionar um valor a uma variável.

Sintaxe:

ADD [nome-de-dado] **TO** (nome-de-dado-n).

ADD [nome-de-dado-1], [nome-de-dado-2] **GIVING** (nome-de-dado-n).



SUBTRAÇÃO - SUBTRACT

Utilizado para subtração de valores.

Sintaxe:

SUBTRACT [nome-de-dado] **FROM** (nome-de-dado-n).

SUBTRACT [nome-de-dado] **FROM** [nome-de-dado-n] **GIVING** (nome-de-dado-n)



MULTIPLICAÇÃO - MULTIPLY

Utilizado para multiplicação de valores.

Sintaxe:

MULTIPLY [nome-de-dado] **BY** (nome-de-dado-n).

MULTIPLY [nome-de-dado] **BY** [nome-de-dado] **GIVING** (nome-de-dado-n)



DIVISÃO - DIVIDE

Utilizado para divisão de valores. O comando DIVIDE divide dois valores numéricos e armazena o quociente.

Sintaxe:

DIVIDE {nome-de-dado} {**BY/INTO**} {nome-de-dado} **GIVING** {nome-de-dado-n}



COMPUTE

Utilizado para cálculos formados por expressões matemáticas de diferentes complexidades, podendo-se utilizar parênteses para priorizar cálculos e organizar a expressão.

Os operadores são: '+' para adição, '-' para subtração, '*' para multiplicação, '/' para divisão e '**' para exponenciação.

Sintaxe:

COMPUTE [nome-de-dado] = nome-de-dado [símbolos] nome-de-dado-n

ESTRUTURAS DE CONTROLE



CONDICIONAL - IF

IF <Condição>

<Comandos se a condição for **verdadeira**>.

ELSE

<Comandos se a condição for **falsa**>.

END-IF.

1 - Todo **"IF"** precisa ter um **"END-IF"**.

2 - Um **"IF"** não precisa ter um **"ELSE"**.

3 - Um **"ELSE"** vale pro último **"IF"** sem **"ELSE"** dentro de um **"END-IF"** dentro do ninho.



CONDICIONAL - EVALUATE WHEN

EVALUATE <variável>

WHEN <condição 1> <sentença>.

WHEN <condição 2> <sentença>.

WHEN <condição 3> <sentença>.

END-EVALUATE.



ITERAÇÃO - PERFORM UNTIL

PERFORM VARYING <variável> **FROM** <valor1> **TO** <valor2> **UNTIL** <condição>
<sentença>
END-PERFORM.



ITERAÇÃO - PERFORM TIMES

Nessa estrutura o parágrafo será executado “n” vezes.

Exemplo:

A.

PERFORM B 3 **TIMES**.

STOP RUN.

B.

DISPLAY 'FlashCLIP'.

DESAFIOS





EMPRESAS PATROCINADORAS



Centro de
Reabilitação Nossa
Senhora de Lourdes
Saúde Física, Intelectual e Social



Gráfica Rápida
CENTRO DIGITAL
A impressão que fica!

(63) **3224-4434**



INFORMÁTICA

FLASH CLIP 2018



CURSOS DE LINGUAGENS DE PROGRAMAÇÃO

27 / OUTUBRO

10 / NOVEMBRO

17 / NOVEMBRO

