

# EXPLORANDO A LINGUAGEM COBOL:

## Um Guia Rápido para Jovens Desenvolvedores

Este eBook é um guia introdutório para estudantes universitários e jovens profissionais de tecnologia que desejam aprender sobre a linguagem de programação COBOL.

Você descobrirá um pouco da história, a estrutura da linguagem, principais comandos e exemplos práticos para ajudar na compreensão do uso do COBOL em aplicações reais.



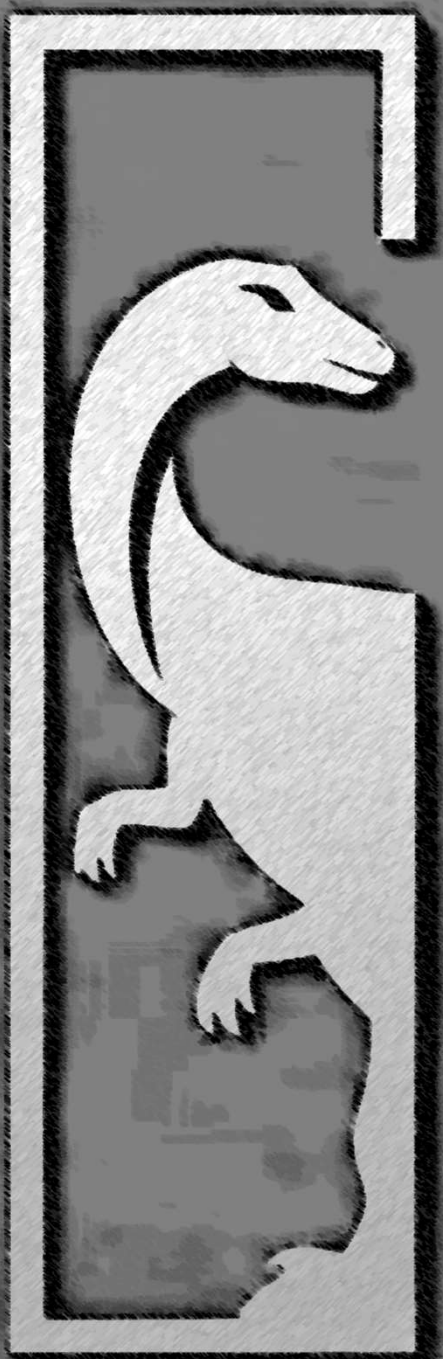
 por Carolina Ladeia

# HISTÓRIA DA LINGUAGEM COBOL

COBOL, abreviação de Common Business Oriented Language, foi criada em 1959 por um comitê de especialistas da indústria, liderado pelo Departamento de Defesa dos Estados Unidos. O objetivo era desenvolver uma linguagem de programação orientada a negócios que fosse fácil de ler e entender, com o intuito de padronizar a programação de computadores para diversas empresas e instituições.

A criação do COBOL foi impulsionada pela crescente necessidade de processar grandes quantidades de dados nas áreas de finanças, administração, comércio e outras áreas que exigiam sistemas de processamento de dados robustos e eficientes. Em seus primórdios, a programação era feita em linguagem de máquina, o que exigia conhecimentos específicos de cada tipo de computador e dificultava a portabilidade de códigos.

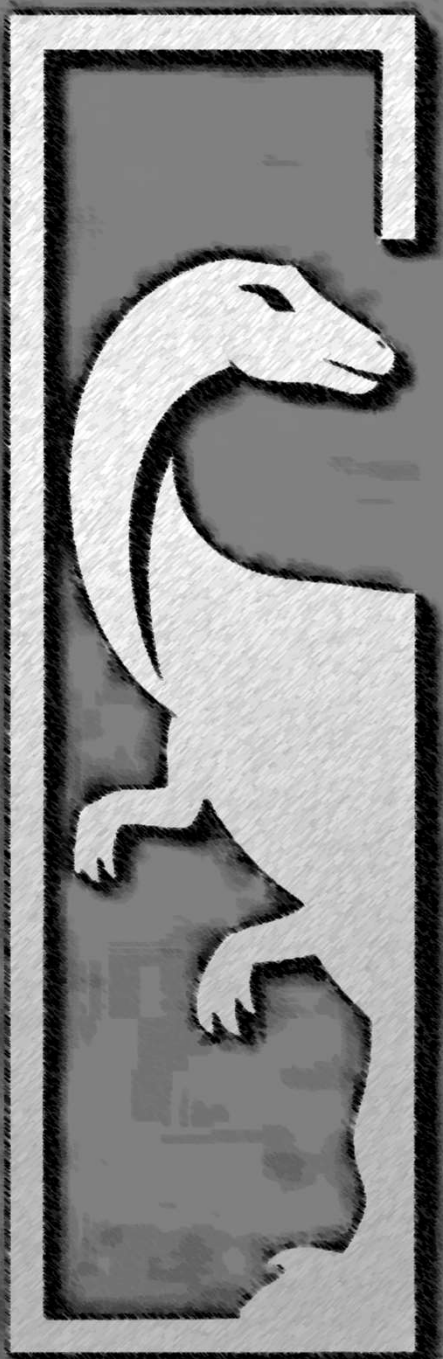
O COBOL revolucionou a programação de negócios ao introduzir um vocabulário baseado em inglês, que era mais amigável e intuitivo para os programadores da época. A linguagem se tornou rapidamente popular e foi adotada por diversas empresas, governos e instituições financeiras em todo o mundo.



# ESTRUTURA BÁSICA DO COBOL

Um programa COBOL é dividido em quatro divisões principais, que representam as diferentes etapas da construção de um programa:

- **IDENTIFICATION DIVISION:** Contém informações de identificação do programa, como nome, autor, data de criação e outras informações relevantes.
- **ENVIRONMENT DIVISION:** Define o ambiente de hardware e software onde o programa será executado, incluindo o tipo de computador, sistema operacional e outros recursos necessários.
- **DATA DIVISION:** Declara todas as variáveis que serão utilizadas no programa, definindo seus nomes, tipos de dados, tamanhos e valores iniciais.
- **PROCEDURE DIVISION:** Contém o código-fonte do programa, ou seja, as instruções que o programa irá executar para realizar as tarefas desejadas. O COBOL segue uma estrutura hierárquica e utiliza uma linguagem de programação procedural, o que significa que as instruções são executadas de forma sequencial, de cima para baixo.



# EXEMPLO SIMPLES DE UM PROGRAMA COBOL

IDENTIFICATION DIVISION.

PROGRAM-ID. ExemploBasico.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 mensagem PIC X(30) VALUE 'Hello, COBOL!'.

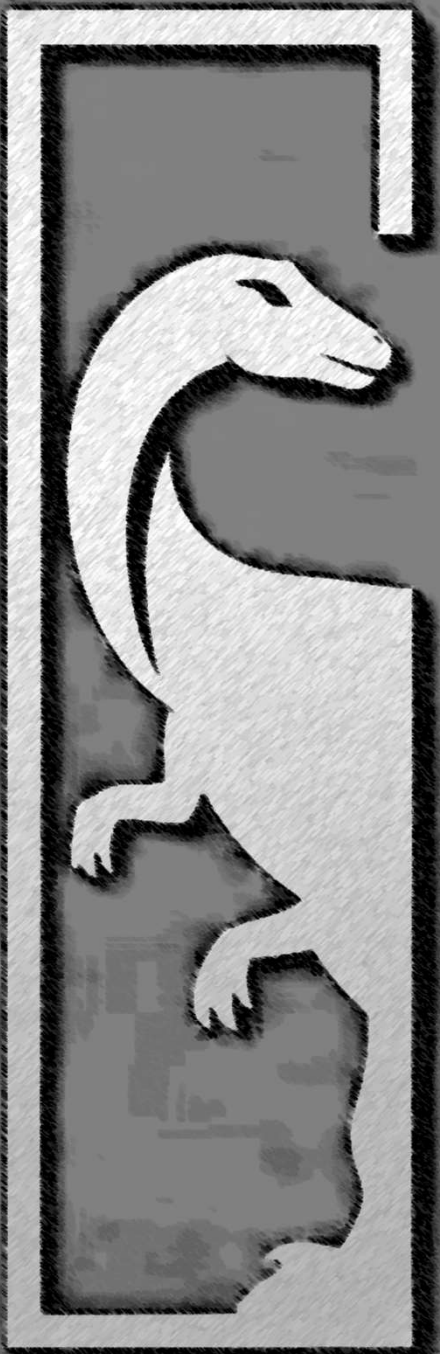
PROCEDURE DIVISION.

DISPLAY-MENSAGEM.

DISPLAY mensagem.

STOP RUN.

Este exemplo demonstra a estrutura básica de um programa COBOL. A IDENTIFICATION DIVISION define o nome do programa, a ENVIRONMENT DIVISION especifica o ambiente de execução, a DATA DIVISION declara a variável "mensagem" e a PROCEDURE DIVISION contém as instruções para exibir a mensagem na tela.



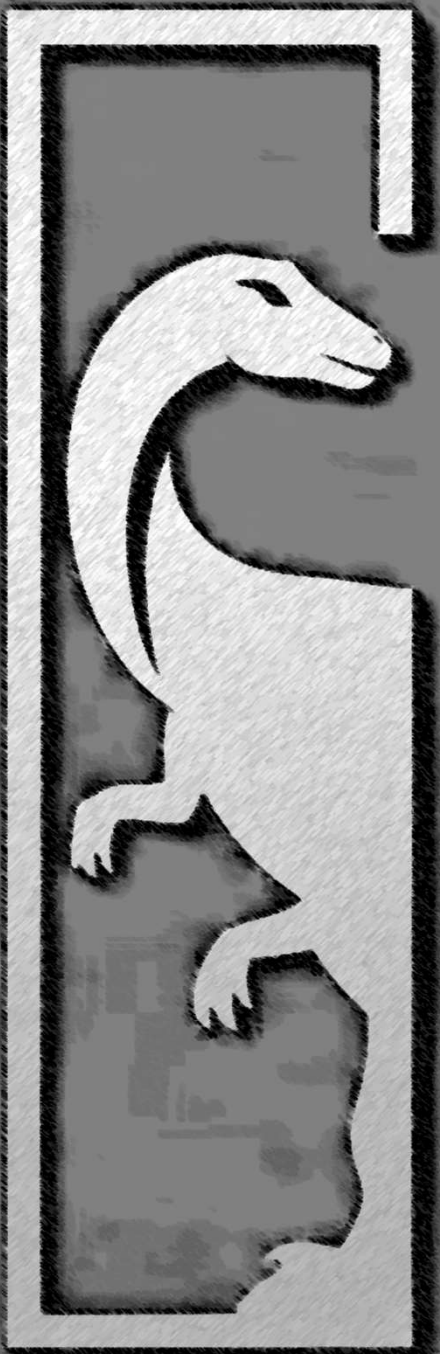


# ALGUNS COMANDOS COBOL

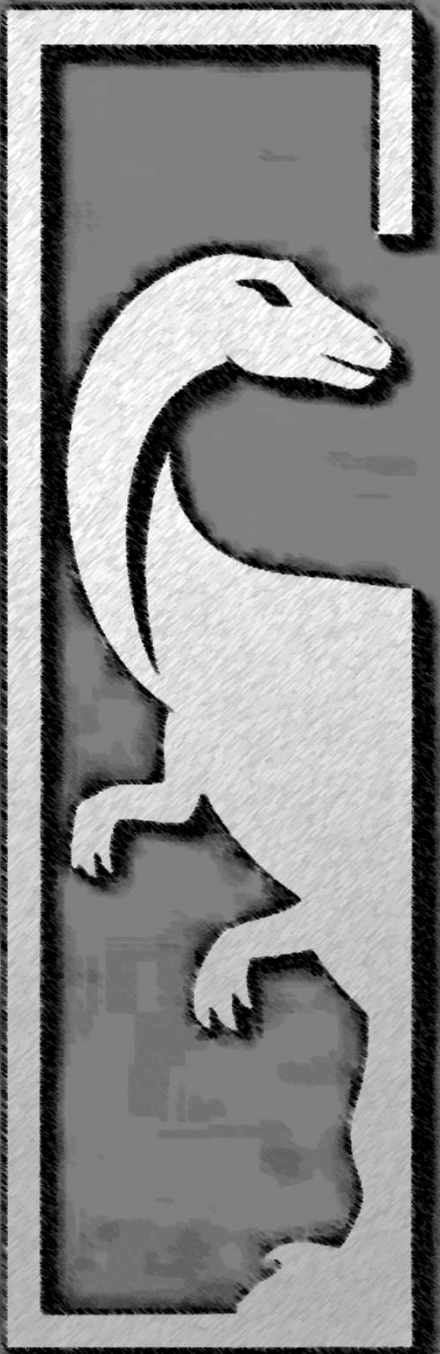
O COBOL oferece uma variedade de comandos para realizar operações de entrada, saída, processamento de dados e controle de fluxo. Alguns dos principais comandos incluem:

- **DISPLAY:** Usado para exibir mensagens ou valores de variáveis na tela.
- **ACCEPT:** Usado para receber entrada do usuário, como digitação de dados pelo teclado.
- **MOVE:** Usado para atribuir valores a variáveis, como mover dados de uma variável para outra.
- **IF:** Realiza comparações e condições para executar instruções específicas com base no resultado da comparação.
- **PERFORM:** Executa um bloco de código ou uma seção repetidamente, ideal para loops e repetições.

Esses comandos são as ferramentas básicas para construir programas COBOL que manipulam dados, interagem com o usuário e realizam tarefas específicas.



# EXEMPLO PRÁTICO COM CONDIÇÕES E SAÍDA



IDENTIFICATION DIVISION.  
PROGRAM-ID. ExemploCondicional.

DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 idade PIC 9(2).

PROCEDURE DIVISION.  
INICIO.  
DISPLAY 'Digite sua idade:'.  
ACCEPT idade.  
IF idade > 17  
DISPLAY 'Você é maior de idade.'  
ELSE  
DISPLAY 'Você é menor de idade.'  
STOP RUN.

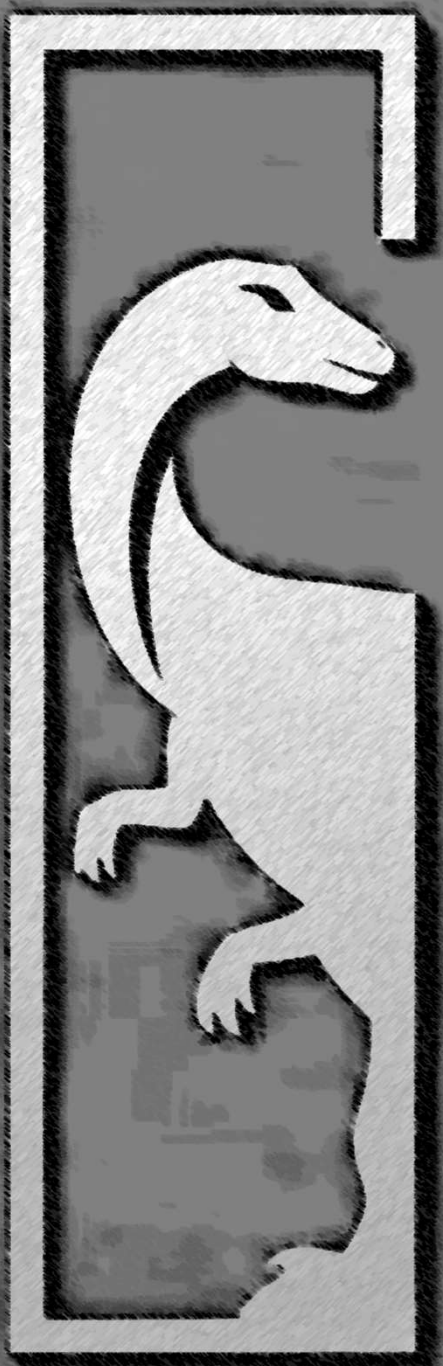
Neste exemplo, o programa solicita a idade do usuário, armazena na variável "idade" e utiliza um comando IF para verificar se a idade é maior que 17. Em caso afirmativo, exibe a mensagem "Você é maior de idade", caso contrário, exibe "Você é menor de idade."

# APLICAÇÕES EM COBOL: CONTEXTOS REAIS

O COBOL ainda é amplamente utilizado em sistemas legados de bancos, seguradoras, empresas de telecomunicações, governos e outras organizações que lidam com grandes volumes de dados e transações. Apesar de sua idade, o COBOL continua relevante em vários setores devido à sua robustez, performance e capacidade de gerenciar sistemas complexos e críticos.

Muitos sistemas que foram construídos há décadas com base em COBOL ainda estão em operação, processando transações e gerenciando dados de forma eficiente. As empresas que utilizam COBOL em seus sistemas legados geralmente o fazem devido à sua estabilidade e desempenho, especialmente em cenários com grande número de transações simultâneas.

O conhecimento em COBOL pode ser um diferencial importante para profissionais que desejam trabalhar em áreas que dependem de sistemas antigos ou em empresas que possuem um legado histórico em COBOL.





# CONCLUSÃO

O COBOL, apesar de ter sido criado há décadas, continua a ser uma linguagem relevante em muitos setores, especialmente em áreas que dependem de sistemas legados com grandes volumes de dados e transações. A familiaridade com o COBOL pode ser uma vantagem para profissionais que desejam trabalhar em áreas que utilizam essa linguagem, como bancos, seguradoras e órgãos governamentais.

Este eBook introduziu os conceitos básicos do COBOL, sua história, estrutura, comandos importantes e exemplos práticos. É um ponto de partida para aqueles que desejam explorar o mundo do COBOL e descobrir as oportunidades que essa linguagem ainda oferece.

Com este conhecimento inicial, você pode iniciar sua jornada de aprendizado, explorar recursos adicionais, praticar seus conhecimentos com exercícios e exemplos práticos, e quem sabe, no futuro, contribuir para a modernização de sistemas legados ou desenvolver novos projetos que utilizam COBOL.

