

# Linguagem COBOL

COBOL (Common Business-Oriented Language), criada em 1959, foi a primeira linguagem focada especificamente em soluções comerciais e administrativas. Sua sintaxe próxima ao inglês natural permite que profissionais de negócios compreendam o código com facilidade, criando uma ponte efetiva entre as áreas técnica e comercial.

Sua relevância persiste até hoje, com aproximadamente 70% das transações comerciais mundiais sendo processadas por sistemas COBOL, especialmente no setor financeiro. Bancos, seguradoras e instituições governamentais confiam em sua estrutura clara e capacidade excepcional de processar grandes volumes de dados com precisão.

A longevidade do COBOL deve-se à sua confiabilidade e eficiência comprovadas, tornando-a essencial para organizações que necessitam de sistemas estáveis e de alto desempenho em operações críticas.

*AriMart*



AriMar

# Breve história do COBOL

1

## Década de 1950

Surgiu a necessidade de uma linguagem de programação para processamento de dados.

---

2

**1959**

Criado o COBOL (Common Business Oriented Language).

---

3

## Década de 1960

COBOL se tornou a linguagem de programação dominante para negócios.

---

4

## Décadas de 1970 e 1980

COBOL se tornou a linguagem padrão para bancos e outras grandes empresas.

COBOL ainda é amplamente utilizado, sendo mantido e modernizado.

# Características principais do COBOL

## Linguagem Imperativa

COBOL é uma linguagem de programação imperativa. As instruções são executadas sequencialmente, instruindo o computador sobre como realizar tarefas específicas.

## Orientado a Processamento de Dados

Projetado para lidar com grandes volumes de dados e operações complexas. Ideal para aplicações de negócio e sistemas de informação.

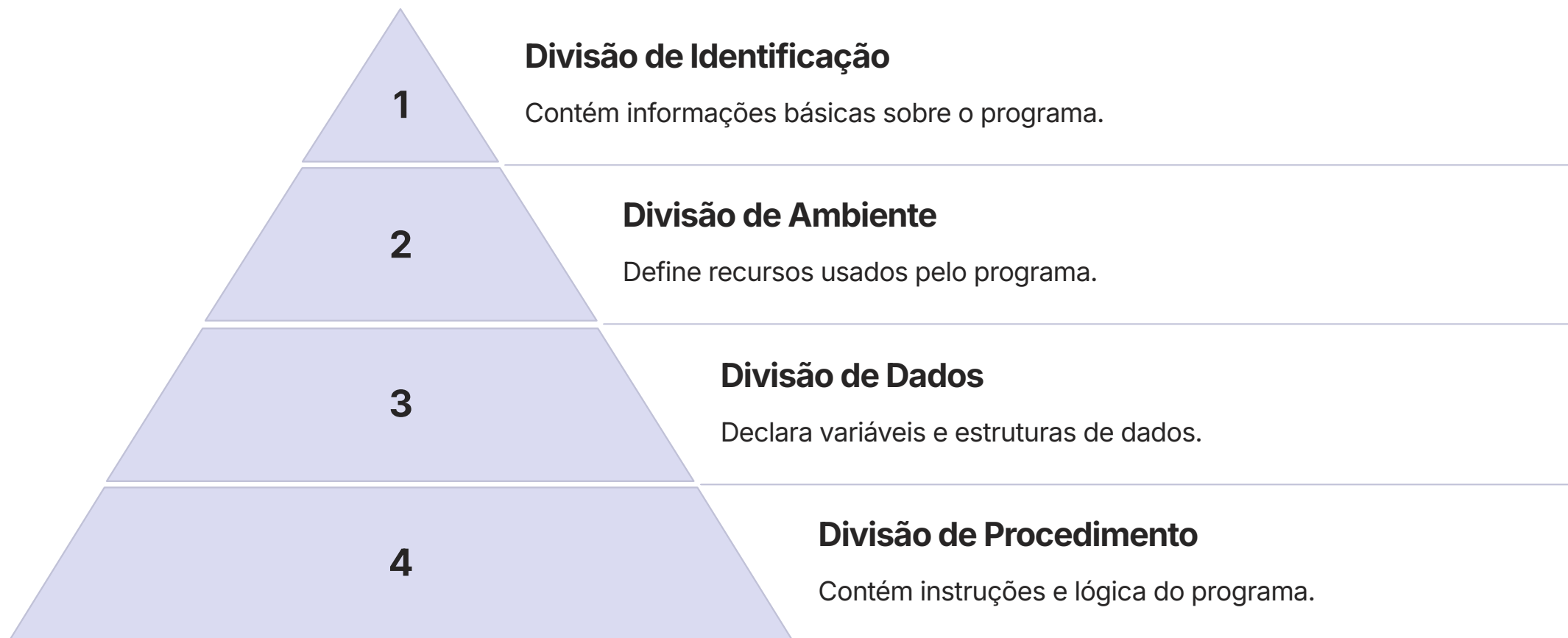
## Linguagem de Alto Nível

COBOL usa um vocabulário e sintaxe mais próximos da linguagem humana, tornando-o mais fácil de ler e escrever do que linguagens de baixo nível.

## Portátil e Robustos

COBOL é uma linguagem portátil, o que significa que o código pode ser compilado e executado em diferentes plataformas e sistemas operacionais.

# Estrutura básica de um programa em COBOL



A estrutura básica de um programa COBOL é organizada em quatro divisões, cada uma com um propósito específico. A divisão de identificação fornece informações sobre o programa, como nome e autor. A divisão de ambiente define os recursos externos usados pelo programa, como arquivos e dispositivos. A divisão de dados declara as variáveis e estruturas de dados usadas no programa. Finalmente, a divisão de procedimento contém as instruções e lógica que o programa executará.

# Declaração de dados em COBOL

## 1. Definindo Variáveis

A declaração de dados define as variáveis que serão usadas no programa.

## 3. Nomes das Variáveis

Os nomes das variáveis são atribuídos para identificar cada uma delas.

## 2. Tipos de Dados

Especifica o tipo de dado que cada variável pode armazenar, como números, texto ou datas.

## 4. Tamanho

Define o tamanho máximo de cada variável, que pode ser necessário para alocar memória.

# Tipos de dados em COBOL



## Numérico

O tipo de dados numérico é usado para representar valores numéricos, como inteiros e decimais.



## Alfabético

O tipo de dados alfabético é usado para representar caracteres alfabéticos, como letras e símbolos.



## Alfanumérico

O tipo de dados alfanumérico é usado para representar uma combinação de caracteres alfabéticos e numéricos.



## Data

O tipo de dados data é usado para representar datas em um formato específico.

# Operações Aritméticas em COBOL

1

## Adição (+)

O operador de adição (+) soma dois operandos.

2

## Subtração (-)

O operador de subtração (-) subtrai o segundo operando do primeiro.

3

## Multiplicação (\*)

O operador de multiplicação (\*) multiplica dois operandos.

4

## Divisão (/)

O operador de divisão (/) divide o primeiro operando pelo segundo.

5

## Exponenciação(\*\*)

O operador de exponenciação (\*\*) eleva o primeiro operando à potência do segundo.



# Instruções de entrada e saída em COBOL



## Entrada de dados

O COBOL usa instruções como **ACCEPT** para receber dados do usuário ou de arquivos.



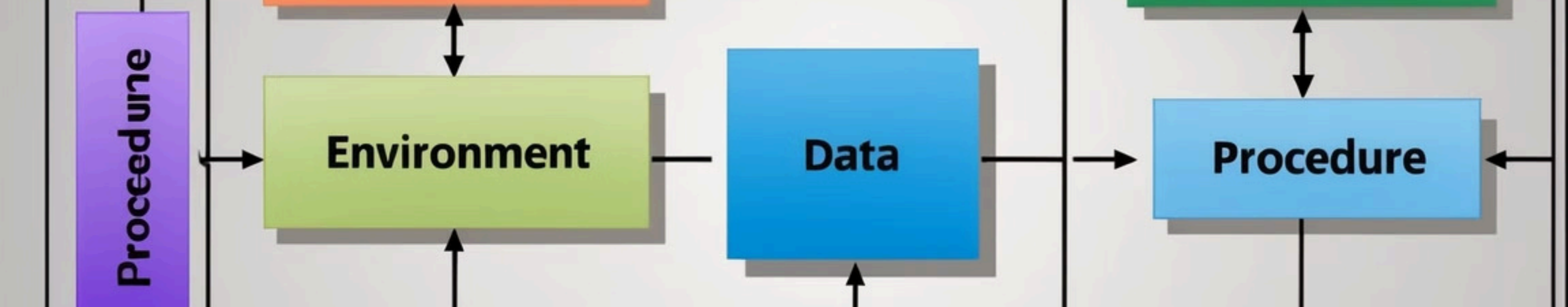
## Saída de dados

As instruções **DISPLAY** e **WRITE** permitem que o programa exiba dados na tela, em arquivos ou impressoras.



## Entrada/Saída de Arquivos

O COBOL fornece recursos para manipular arquivos, incluindo abertura, fechamento, leitura e escrita.



# Conceito de Divisão em COBOL

## 1 Organização Lógica

A estrutura de um programa COBOL é dividida em quatro seções lógicas, cada uma com responsabilidades específicas.

## 2 Sequência Predefinida

A ordem das divisões é fixa e define a estrutura do programa, com a Divisão de Identificação sendo a primeira e a Divisão de Procedimento sendo a última.

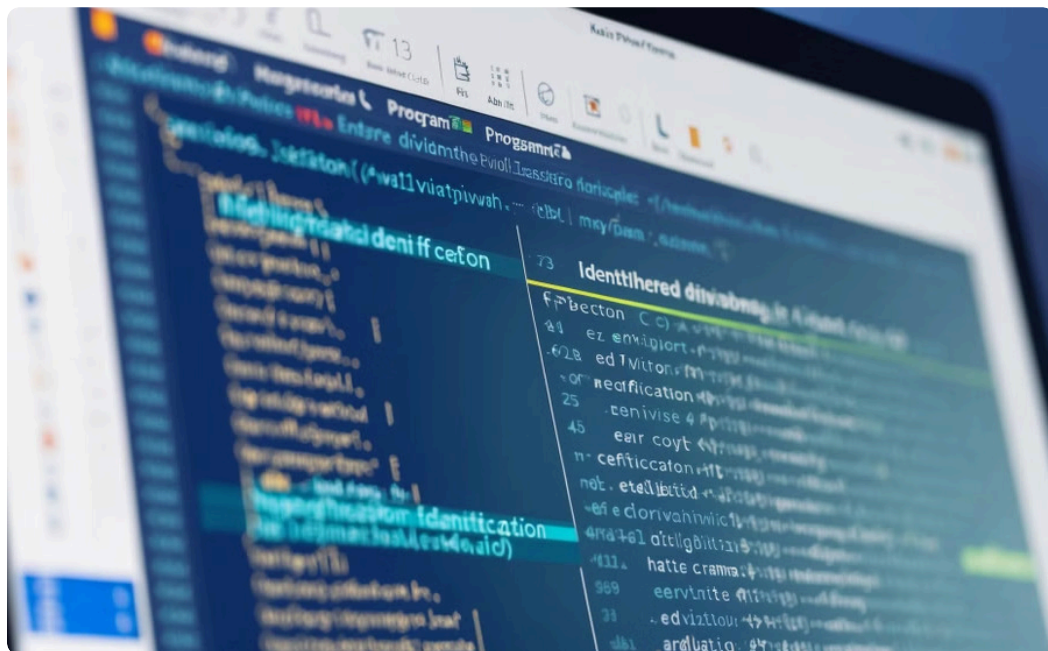
## 3 Declarações e Instruções

Cada divisão contém declarações e instruções específicas que definem o programa, como informações gerais, recursos usados, dados manipulados e a lógica de execução.

## 4 Facilidade de Manutenção

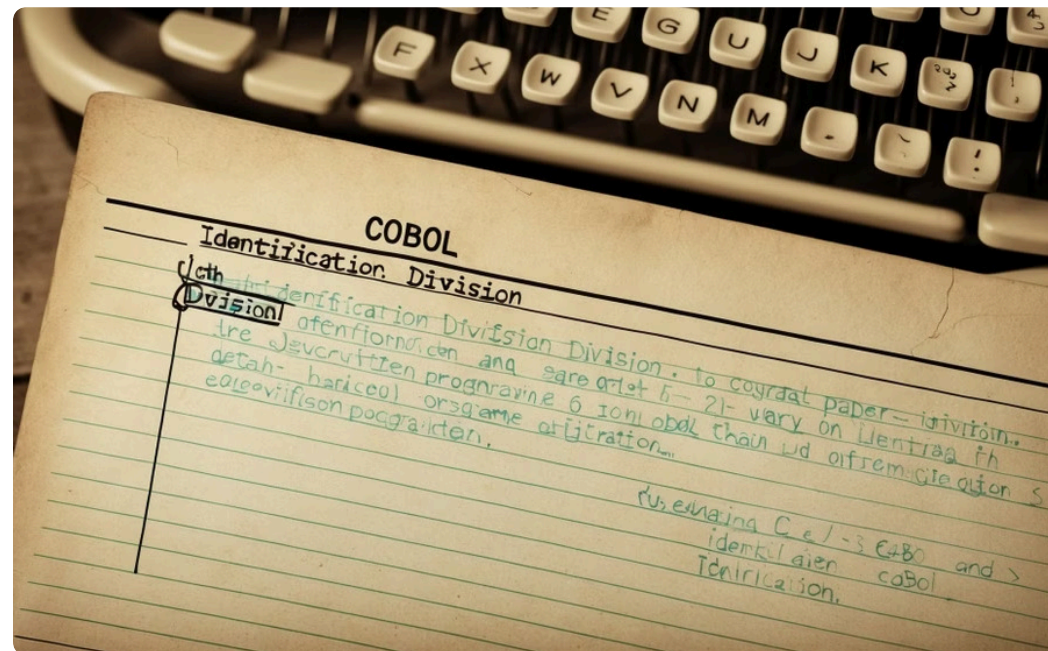
A divisão em seções facilita a leitura, compreensão e manutenção do código, pois cada parte tem um propósito definido.

# Divisão de Identificação em COBOL



## Programação em COBOL

A Divisão de Identificação é a primeira seção de um programa COBOL. Ela contém informações essenciais sobre o programa, como seu nome, autor, data de criação e outros detalhes relevantes.



## Identificação do Programa

A Divisão de Identificação é obrigatória em todos os programas COBOL. O programador fornece informações que ajudam na organização e gerenciamento do código-fonte.

# Divisão de Ambiente em COBOL

## Configuração do Ambiente

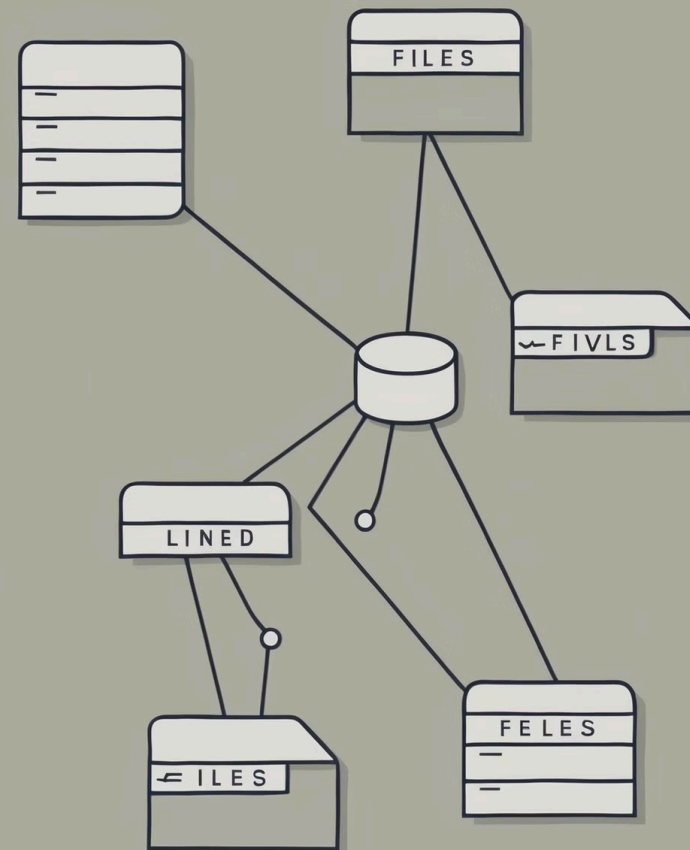
Especifica os recursos e configurações do ambiente de execução do programa, como dispositivos de entrada e saída e arquivos.

## Definição de Arquivos

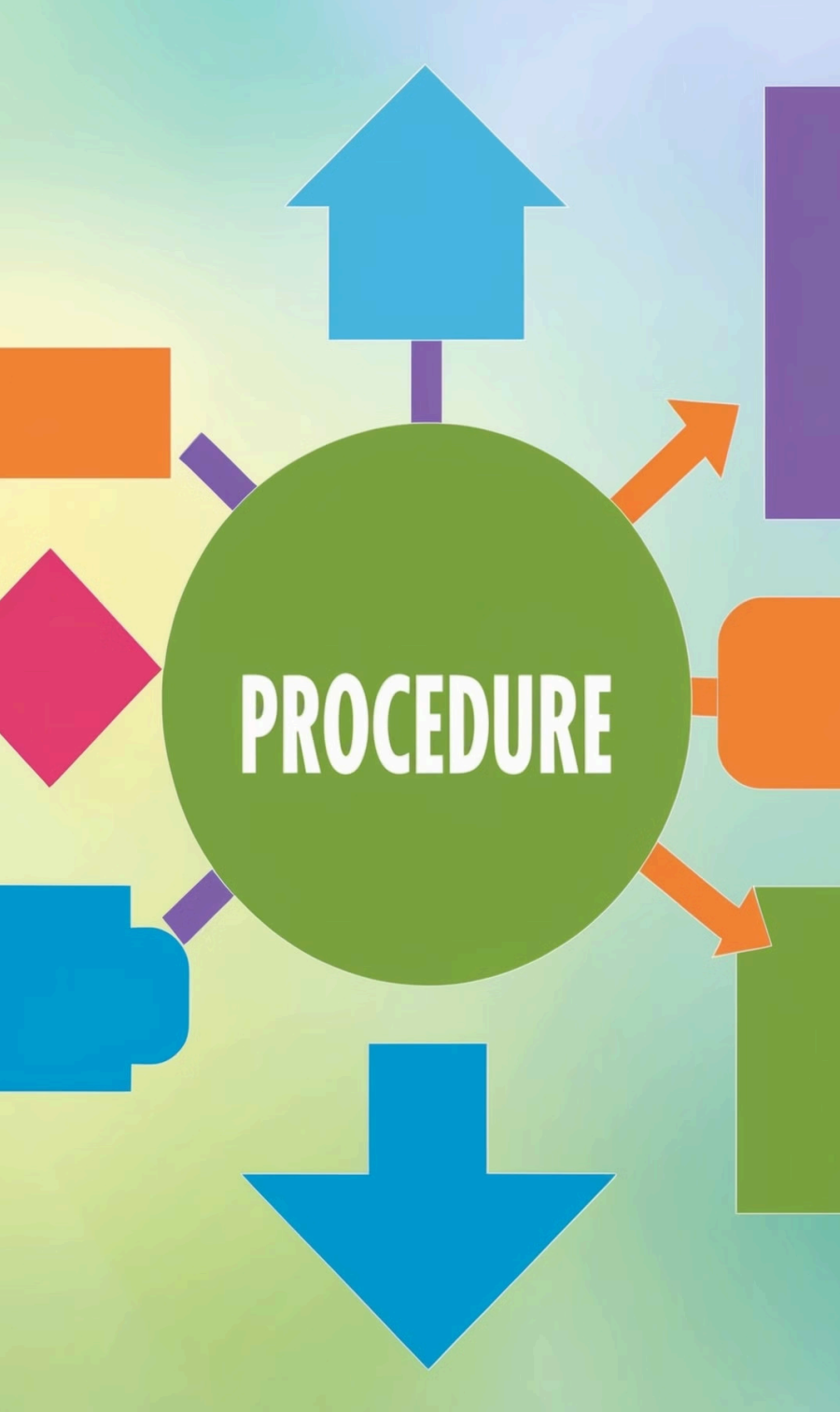
Define os arquivos que serão usados pelo programa, incluindo nomes, tipos e atributos.

## Mapeamento de Dispositivos

Associa os arquivos a dispositivos físicos, como impressoras, terminais ou discos.







# Divisão de Procedimento em COBOL



## Lógica do Programa

A divisão de procedimento contém as instruções que determinam o fluxo de execução do programa COBOL.



## Sequência de Instruções

As instruções COBOL são processadas na ordem em que aparecem nesta divisão, a menos que instruções de controle de fluxo sejam usadas.



## Blocos de Código

Esta divisão pode conter vários blocos de código, cada um com um propósito específico, como entrada, processamento ou saída de dados.

# Declaração PROCEDURE DIVISION em COBOL

## Ponto de Entrada

A PROCEDURE DIVISION é o ponto de entrada para a execução do programa COBOL.

## Lógica do Programa

Contém as instruções que definem o comportamento e as ações do programa COBOL.

## Sequência de Instruções

As instruções na PROCEDURE DIVISION são executadas sequencialmente, a menos que sejam modificadas por instruções de controle de fluxo.



# Declaração MOVE em COBOL



## Movimento de Dados

A instrução MOVE copia o conteúdo de um campo de dados para outro.



## Sintaxe

MOVE **valor-fonte** TO **valor-destino**;



## Tipos de Dados

- Numérico
- Alfanumérico
- Data

# Declaração IF-THEN-ELSE em COBOL

## Sintaxe da Declaração

A declaração IF-THEN-ELSE permite que um programa COBOL execute blocos de código com base em condições específicas. Essa estrutura de controle de fluxo é crucial para a tomada de decisões dentro de um programa COBOL.

## Exemplo

Um exemplo clássico de uso da declaração IF-THEN-ELSE é verificar se um valor é maior ou menor que outro. Se a condição for verdadeira (THEN), um bloco de código será executado; caso contrário (ELSE), outro bloco de código será executado.

# Declaração PERFORM em COBOL

## 1 Repetição de Blocos de Código

A declaração PERFORM permite executar um bloco de código várias vezes, simplificando a escrita de loops.

## 3 Eficiência

A declaração PERFORM é fundamental para automatizar tarefas repetitivas, tornando o código mais eficiente e legível.

## 2 Controle de Fluxo

Permite controlar o fluxo de execução de um programa, repetindo um conjunto de instruções até que uma condição seja satisfeita.

## 4 Flexibilidade

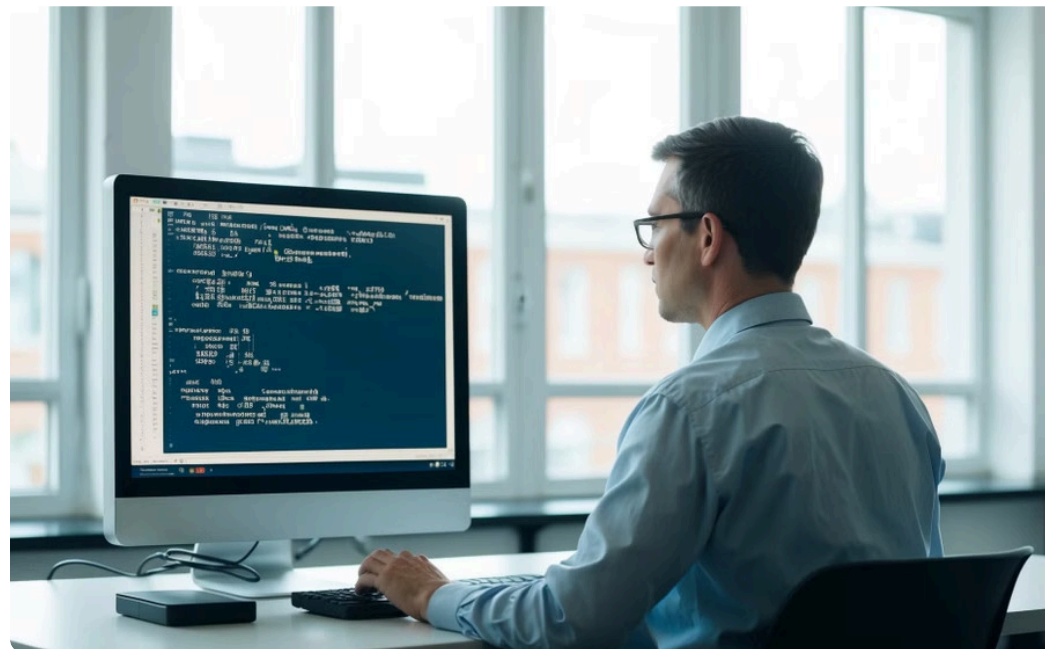
A declaração PERFORM oferece diferentes variações, permitindo controle sobre o número de repetições e condições de término.

# Declaração ACCEPT e DISPLAY em COBOL



## Entrada de Dados

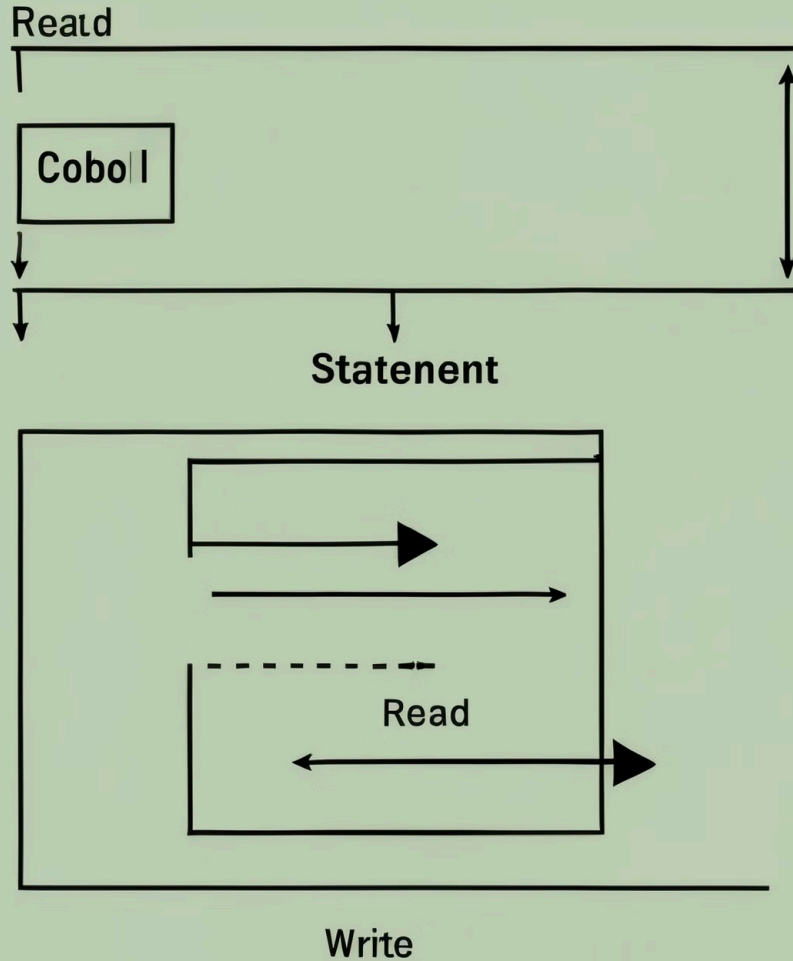
A instrução ACCEPT é usada para receber dados do usuário ou de um dispositivo de entrada, como um teclado.



## Saída de Dados

A instrução DISPLAY exhibe dados na tela ou em outro dispositivo de saída, como uma impressora.

## Flow of Cobol Program



# Declaração READ e WRITE em COBOL

1

## 1. Leitura de Dados

A instrução READ é utilizada para ler dados de um arquivo de entrada para uma área de armazenamento na memória.

2

## 2. Especificação de Arquivo

O arquivo de entrada é especificado no comando READ, incluindo o nome do arquivo e o tipo de acesso.

3

## 3. Escrita de Dados

A instrução WRITE é usada para gravar dados de uma área de armazenamento na memória para um arquivo de saída.

4

## 4. Formatação de Saída

É possível formatar a saída para atender a requisitos específicos, como espaçamento e alinhamento, utilizando instruções adicionais.

# Declaração OPEN e CLOSE em COBOL

## OPEN

A declaração OPEN prepara um arquivo para leitura ou gravação. Isso permite que o programa acesse os dados armazenados nesse arquivo.

A declaração OPEN especifica o modo de acesso, como entrada (INPUT) para leitura ou saída (OUTPUT) para gravação.

## CLOSE

A declaração CLOSE encerra o acesso a um arquivo. Isso libera os recursos do sistema associados ao arquivo e garante a consistência dos dados.

Depois de fechar um arquivo, ele não pode ser acessado novamente até que seja aberto novamente.



# Declaração CALL em COBOL



## Chamando Subprogramas

A instrução CALL invoca um subprograma ou módulo COBOL separado.



## Reutilização de Código

Permite a modularização do código, promovendo organização e reutilização de funcionalidades.



## Parâmetros

Parâmetros podem ser passados para o subprograma, permitindo a comunicação de dados entre módulos.



# Declaração COPY em COBOL

## Reutilização de Código

A instrução COPY permite incluir o conteúdo de outro programa COBOL no programa atual.

## Modularidade

Essa funcionalidade facilita a criação de bibliotecas de código reutilizáveis, promovendo modularidade e organização do código.

## Eficiência

Reduz a redundância de código, simplificando a manutenção e a atualização do código.

# Comentários em COBOL



## Melhorando a Leitura

Comentários ajudam a tornar o código mais compreensível para outros programadores, além de facilitar a própria manutenção do código.



## Documentação do Código

Comentários devem ser usados para explicar o propósito de cada seção de código ou de trechos mais complexos, descrevendo a lógica por trás da implementação.



## Padrões de Estilo

Existem convenções de estilo para comentários em COBOL, incluindo a localização, a formatação e a linguagem utilizada.

# Convenções de nomenclatura em COBOL

## Nomes de programas

Programas COBOL são geralmente nomeados com nomes descritivos que refletem sua função. Nomes de programas devem ser significativos e fáceis de entender, com até 30 caracteres.

## Nomes de variáveis

Variáveis COBOL devem seguir regras específicas de nomenclatura. O nome de uma variável deve começar com uma letra e pode incluir letras, números e o caractere '\_'. Nomes de variáveis devem ser descritivos e refletir o tipo de dados que armazenam.

## Nomes de arquivos

Os nomes dos arquivos devem ser descritivos e refletir o conteúdo do arquivo. Eles devem ter até 8 caracteres. Em COBOL, os arquivos são declarados no DATA DIVISION e o nome do arquivo é usado para identificar o arquivo.

## Nomes de parágrafos

Os nomes dos parágrafos devem refletir a lógica que eles implementam. Eles devem ser concisos e fáceis de entender. Eles devem começar com uma letra e podem incluir letras, números e o caractere '\_'.

# Debugging em COBOL

## Técnicas de depuração

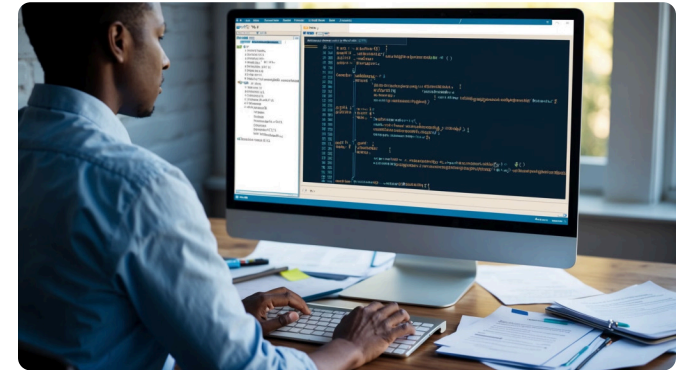
O processo de encontrar e corrigir erros em programas COBOL é essencial para garantir a qualidade do software.

As técnicas de depuração podem incluir a análise de registros de erros, pontos de interrupção e a utilização de ferramentas de depuração.

## Ferramentas de depuração

Existem ferramentas de depuração específicas para COBOL que permitem analisar o fluxo de execução do programa, inspecionar variáveis e rastrear o estado do programa.

Essas ferramentas podem ajudar a identificar a causa de erros e facilitar o processo de correção.



# Portabilidade de código COBOL



## Migração de ambientes

A portabilidade do COBOL possibilita a migração de sistemas legados para plataformas modernas, garantindo a continuidade das operações.



## Integração com sistemas

O código COBOL pode ser integrado a outras linguagens de programação e plataformas, ampliando a funcionalidade e o alcance dos sistemas.



## Atualização para plataformas

A portabilidade do COBOL permite a adaptação de sistemas a novas plataformas, garantindo a atualização e o bom funcionamento.

# Vantagens do COBOL

## 1. Legado e Maturidade

COBOL é uma linguagem com décadas de uso, o que significa que possui um código-base robusto e confiável, e um grande número de profissionais experientes.

## 3. Segurança e Conformidade

COBOL tem um histórico comprovado de segurança e conformidade com regulamentações, o que é fundamental para aplicações críticas que lidam com dados sensíveis.

## 2. Processamento de Dados massivos

O COBOL foi projetado para lidar com grandes volumes de dados, o que o torna ideal para aplicações financeiras, de seguros e outras áreas que exigem processamento intensivo de dados.

## 4. Custo-Benefício

O COBOL é uma linguagem relativamente barata para manter e operar, pois possui um grande número de ferramentas e recursos de suporte disponíveis.

# Desvantagens do COBOL

## Sintáxe Verbose

COBOL é conhecido por sua sintaxe extensa e detalhada, o que pode tornar o código mais longo e complexo.

Essa verbosidade pode resultar em maior tempo de desenvolvimento e manutenção.

## Curva de Aprendizagem

A sintaxe e os conceitos de COBOL podem ser desafiadores para iniciantes.

A familiarização com a linguagem requer tempo e esforço, o que pode ser um obstáculo para novos desenvolvedores.

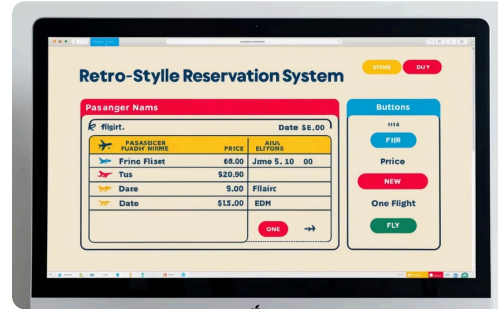
# Aplicações típicas do COBOL



## Serviços Financeiros

O COBOL é amplamente utilizado em sistemas bancários, seguros e investimentos.

Gerenciamento de contas, transações e processamento de pagamentos são algumas de suas aplicações.



## Indústria de Viagens

O COBOL desempenha um papel crucial em sistemas de reservas aéreas e hotéis.

Ele gerencia reservas, roteiros de voo e controle de estoque.



## Governos e Instituições Públicas

O COBOL é usado em sistemas governamentais para gerenciar impostos, benefícios e registros populacionais.

É também utilizado em áreas de saúde, educação e justiça.



## Logística e Distribuição

O COBOL é usado em sistemas de gerenciamento de estoque, controle de cadeia de suprimentos e logística.

Ele gerencia o movimento de produtos, rastreia inventários e otimiza processos.

# Tendências futuras do COBOL

## 1 Modernização e Integração

A linguagem COBOL está sendo modernizada para se integrar a tecnologias modernas e oferecer um desenvolvimento mais ágil.

## 3 Nuvem e Contêineres

A migração de aplicações COBOL para a nuvem e o uso de contêineres estão crescendo para oferecer maior escalabilidade e flexibilidade.

## 2 Automação e IA

O uso de ferramentas de automação e inteligência artificial está se expandindo para otimizar processos e reduzir o tempo de desenvolvimento.

## 4 Segurança Aprimorada

O foco na segurança de aplicações COBOL está se intensificando para garantir a proteção de dados e a conformidade com regulamentos.



# Ferramentas e Ambientes de Desenvolvimento COBOL



## Editores de Código

Editores de código especializados em COBOL oferecem recursos como realce de sintaxe, depuração e autocompletar.



## Compiladores COBOL

Os compiladores convertem o código COBOL em linguagem de máquina, permitindo que o programa seja executado.



## Ambientes de Desenvolvimento Integrados (IDEs)

IDEs fornecem um ambiente unificado para edição, compilação, depuração e execução de programas COBOL.



## Ferramentas de Gerenciamento de Banco de Dados (DBMS)

DBMS como DB2, IMS e Oracle são usados para gerenciar dados em aplicações COBOL.



# Integração do COBOL com outras linguagens

## Interoperabilidade

COBOL pode interagir com outras linguagens, permitindo a integração com sistemas modernos e plataformas.

## APIs e Web Services

Através de APIs e web services, COBOL pode se comunicar com aplicações em outras linguagens, como Java, Python ou JavaScript.

## Integração de Dados

A integração com bancos de dados modernos, como PostgreSQL ou MySQL, é facilitada pelo uso de ferramentas e bibliotecas.

# Boas práticas de programação em COBOL

## Nomenclatura Consistente

Utilize nomes descritivos e significativos para variáveis, procedimentos e arquivos.

Mantenha uma convenção de nomenclatura padronizada para facilitar a leitura e a manutenção do código.

## Documentação Adequada

Insira comentários concisos e informativos para explicar o propósito e a lógica do código.

Documente as variáveis, os procedimentos e as seções importantes para facilitar a compreensão do código.

## Teste Completo

Implemente testes unitários e de integração abrangentes para garantir a qualidade do código.

Execute testes de regressão para verificar se alterações no código não introduzem novos erros.

## Revisão de Código

Promova revisões de código regulares para identificar e corrigir erros.

Peça a outros programadores para revisar o código e fornecer feedback construtivo.

# Segurança em aplicações COBOL



## Autenticação e Autorização

O COBOL oferece recursos para implementar mecanismos de segurança robustos, como autenticação e autorização, protegendo o acesso a dados e funcionalidades.



## Controle de Acesso

O controle de acesso baseado em papéis, funções e perfis, limita o acesso a dados e funções sensíveis, assegurando a integridade do sistema.



## Criptografia de Dados

O COBOL suporta algoritmos de criptografia para proteger dados confidenciais durante o armazenamento, transmissão e processamento.



## Prevenção de Ataques

As aplicações COBOL devem ser protegidas contra vulnerabilidades comuns, como injeção de código SQL, XSS, CSRF, garantindo a segurança da aplicação.

# Gerenciamento de Dados em COBOL



## Armazenamento de Dados

COBOL oferece recursos para definir, organizar e gerenciar dados em arquivos, tabelas e bancos de dados.



## Organização de Arquivos

Permite a criação, leitura, escrita e atualização de arquivos sequenciais, indexados e de acesso direto.



## Gerenciamento de Banco de Dados

Suporta a interação com sistemas de gerenciamento de banco de dados (DBMS) para acesso e manipulação de dados.



## Validação de Dados

Fornecer ferramentas para validar a integridade dos dados, garantindo a precisão e consistência.



# Processamento em Lote em COBOL

## Definição

O processamento em lote é uma técnica em que as operações são realizadas em um conjunto de dados de forma simultânea, sem interação do usuário durante o processamento.

## Vantagens

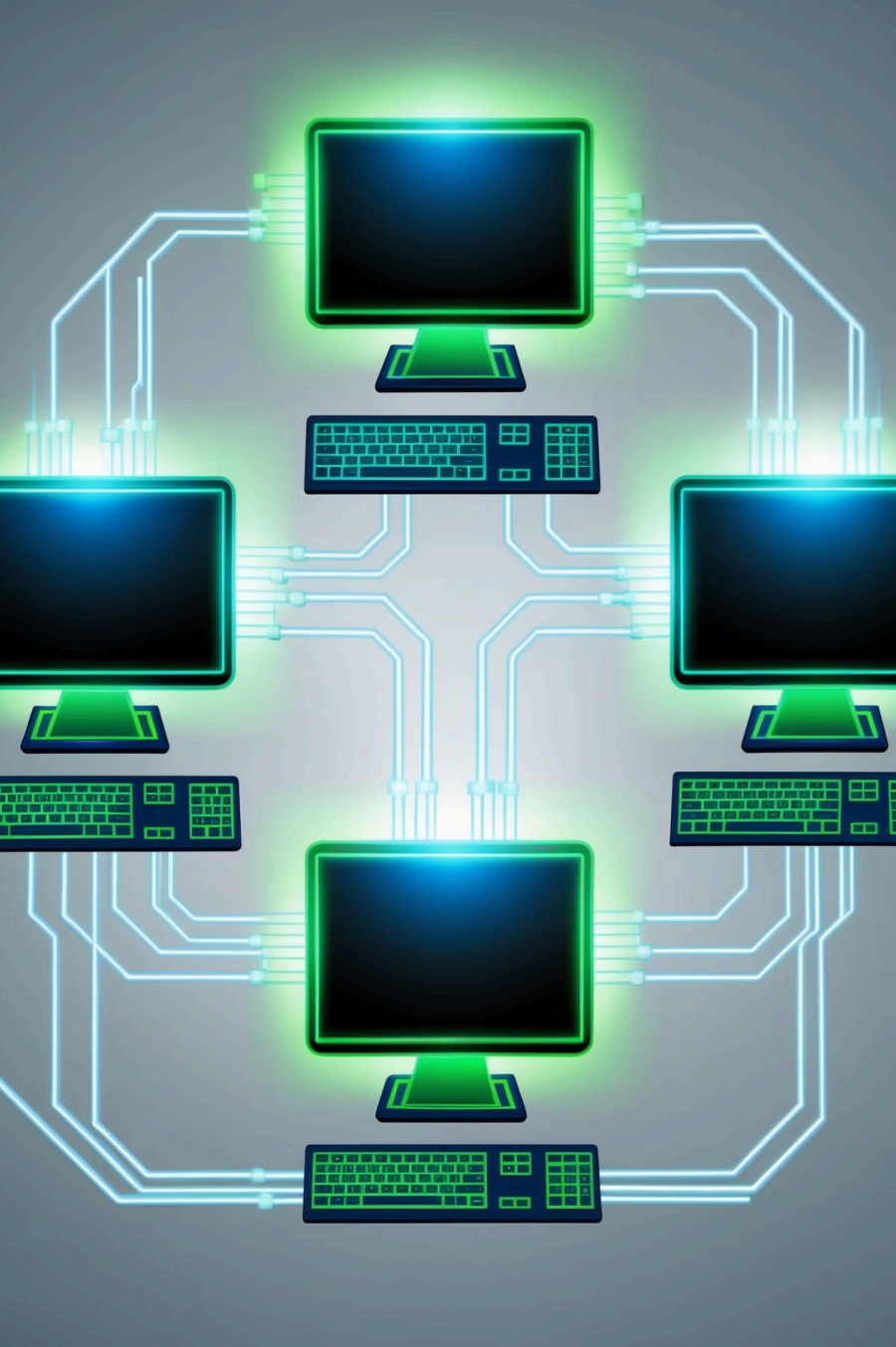
Eficiência, pois permite o processamento de grandes volumes de dados com tempo de resposta otimizado.

## Aplicações

Ideal para tarefas repetitivas que não exigem interação do usuário, como processamento de folha de pagamento, geração de relatórios e faturamento.

## COBOL

O COBOL oferece recursos para implementar o processamento em lote, como a instrução SORT para ordenar dados e a instrução WRITE para gravar saída em arquivos.



# Concorrência e paralelismo em COBOL

## 1 Multitarefa

O COBOL permite que programas executem várias tarefas simultaneamente, otimizando o uso de recursos do sistema.

## 2 Paralelismo

Através de recursos como a instrução "CALL", o COBOL possibilita a execução paralela de diferentes blocos de código.

## 3 Gerenciamento de Recursos

É possível gerenciar os recursos do sistema, como memória e tempo de processamento, para otimizar a execução de tarefas concorrentes.

# Interoperabilidade do COBOL com sistemas modernos

## Integração com APIs modernas

O COBOL pode se integrar a APIs RESTful e SOAP, permitindo que sistemas legados se conectem a serviços modernos na nuvem.

APIs modernas facilitam o acesso a dados e serviços em tempo real, tornando o COBOL mais flexível e adaptável a novas tecnologias.

## Conectividade com bancos de dados

O COBOL pode interagir com bancos de dados modernos, como PostgreSQL e MongoDB, por meio de drivers e bibliotecas específicas.

Esta conectividade permite que sistemas COBOL acessem dados em tempo real e aproveitem as vantagens de tecnologias modernas de gerenciamento de dados.

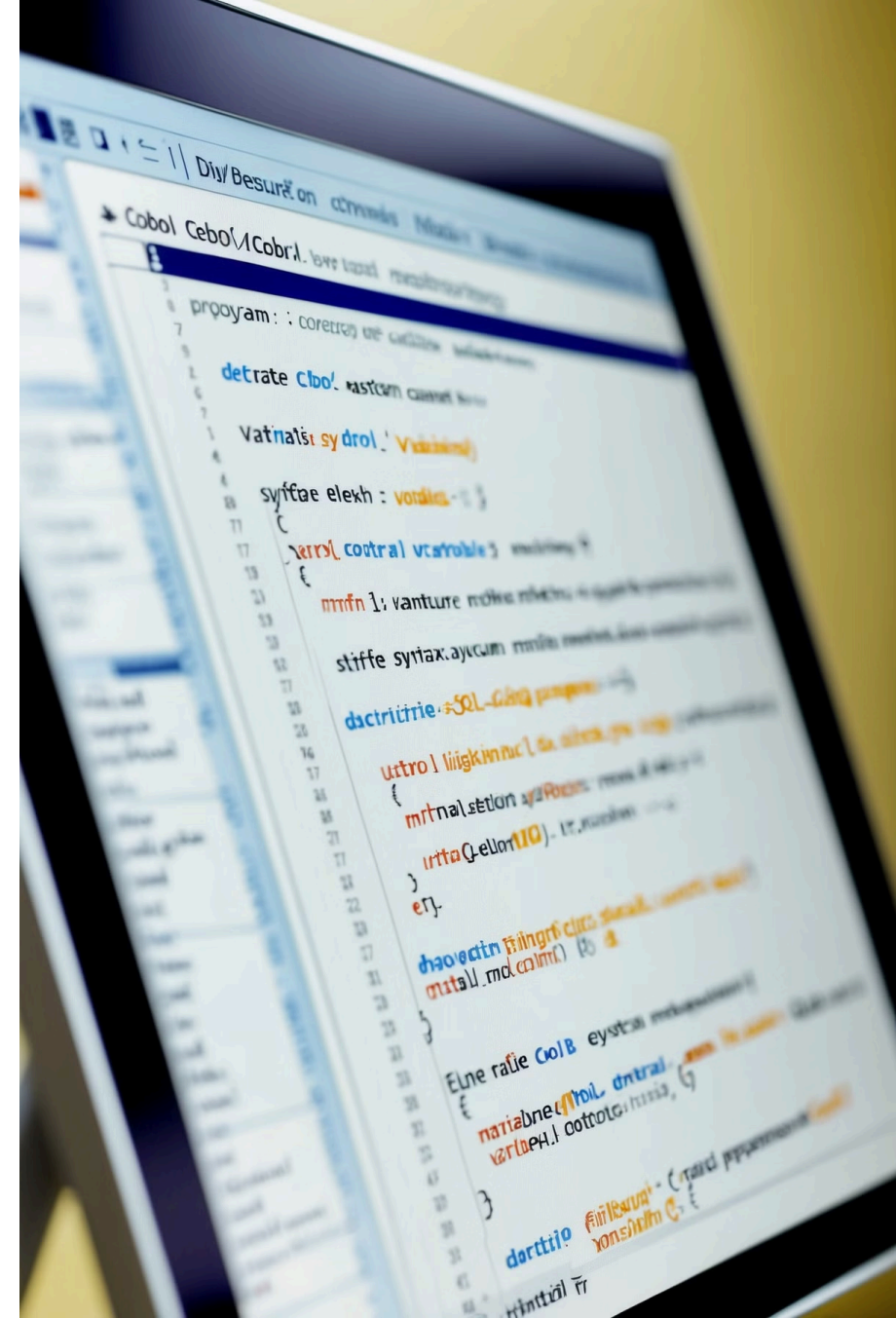
# Exemplos de códigos COBOL

Aqui estão alguns exemplos de código COBOL simples para ilustrar os conceitos básicos.

O primeiro exemplo demonstra a declaração de variáveis e operações aritméticas básicas.

O segundo exemplo apresenta uma estrutura de controle de fluxo, como a instrução IF-THEN-ELSE.

Esses exemplos servem como uma introdução ao código COBOL e como ponto de partida para o desenvolvimento de programas mais complexos.



# Sobre a Obra

Este conteúdo foi desenvolvido com o auxílio de Inteligência Artificial, passando por um rigoroso processo de edição e revisão humana para garantir máxima qualidade e precisão das informações apresentadas.

A ideia é proporcionar aqueles que buscam conhecimento através de um resumo claro e objetivo sobre o tema, contudo, a nossa visão poderá divergir e até mesmo se opor a obra especificada. De qualquer modo, a nossa missão é despertar o interesse no aprofundamento sobre tal tema e a busca por recursos complementares noutras obras pertinentes.

As imagens utilizadas são exclusivamente ilustrativas, selecionadas com propósito didático, e seus direitos autorais pertencem aos respectivos proprietários. As imagens podem não representar fielmente os personagens, eventos ou situações descritas.

Este material pode ser livremente reinterpretado, integral ou parcialmente, desde que citada a fonte e mantida a referência ao Canal.

*AriMart*

**11/2024 - 1676**