

# Formação COBOL

Professor: Vagner Bellacosa

Disciplina: Linguagem de Programação COBOL

# COBOL



## Modulo 04.02 – Origens do COBOL

- 1) Vagner Bellacosa seu facilitador
- 2) Origens da Linguagem de Programação
- 3) Grace Hopper e o Flow-Matic
- 4) CODASYL e as Linguagens de Terceira Geração
- 5) Cobol 1968, 1974, 1985, 2002 e além
- 6) Folha de Codificação, papel perfurado
- 7) O maior computador de sempre
- 8) Mainframe e a Arquitetura de von Neumann
- 9) Processos Batch e Online
- 10) JCL e a compilação de programas
- 11) Enfim a codificação

# COBOL



## 1) Vagner Bellacosa seu facilitador

Analista programador desde 1989, onde comecei como auxiliar e tecnólogo em processamento de dados desbravando os primórdios da computação brasileira, um eterno aprendiz em processos Mainframe..

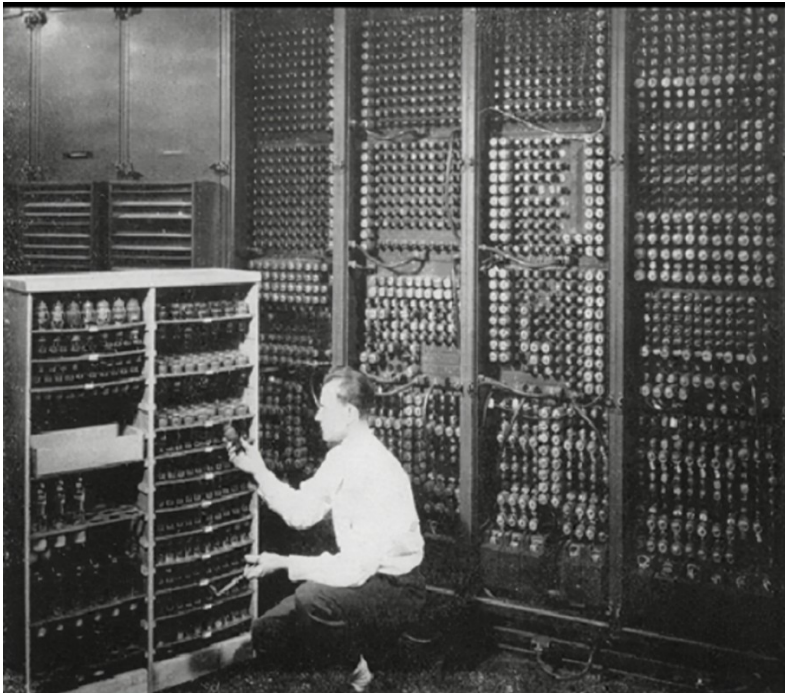
Desde então trabalhei em centenas de projetos, em 4 países e dezenas de instituições financeiras, ora como funcionário, consultor externo e freelancer.

A mais memorável foi o Banco REAL, uma verdadeira escola, onde aprendi muito e tive a oportunidade de participar no aliciante Projeto Y2K, o temível bug do milênio.

Particpei em projetos na CESP, Fundação CESP, Transbrasil, ABSA, Real Seguros, BPN, Skandia, DGITA, BES, CGD, BPI, Barclay, Skandia, IBM Italia, Sistemi Informativi, Unicredit, Zurich Assicurazione, Banco Safra e Banco Itaú.

Atualmente trabalho na Spread no Projeto BRB em Brasília, via remoto.

# COBOL



## 2) Origens da Linguagem de Programação

A necessidade é a mãe de todas as invenções.

O Censo Americano e o surgimento da IBM.

Computadores e a grande guerra

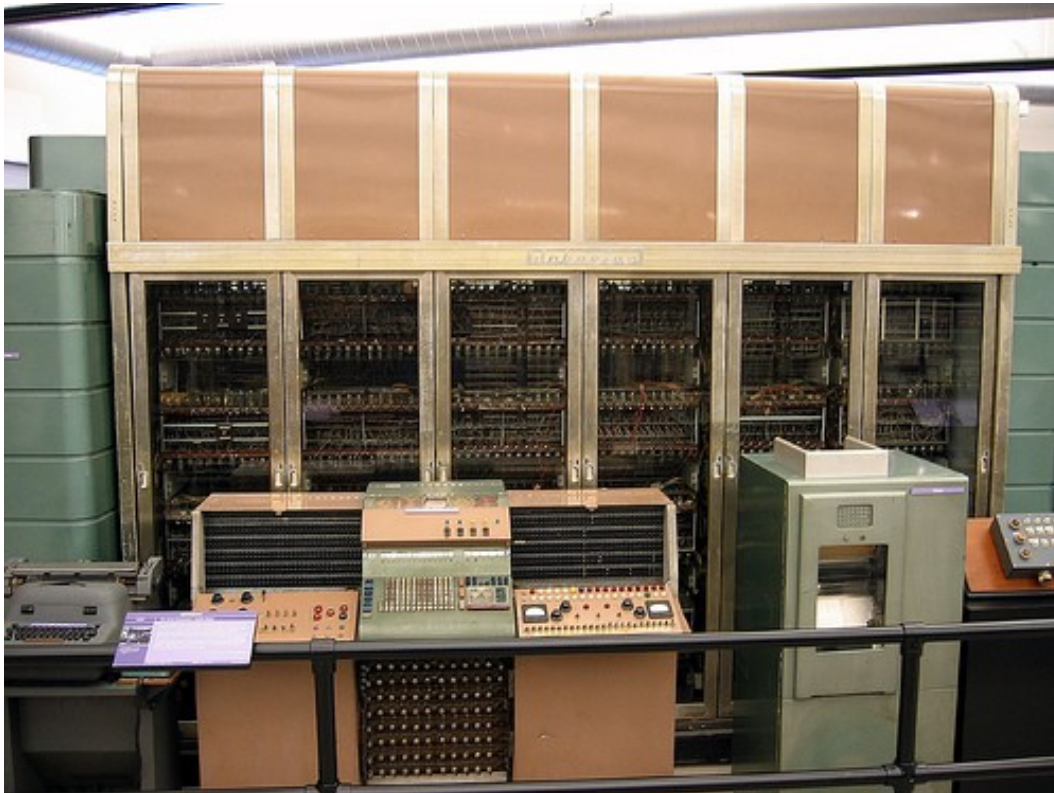
Pioneiros da Informática

Válvula e os bits e bytes, fluxos elétricos.

Painéis de controle e folhas perfuradas

Primeira Geração de Linguagem de Programação

# COBOL



Velhos mainframes



# COBOL



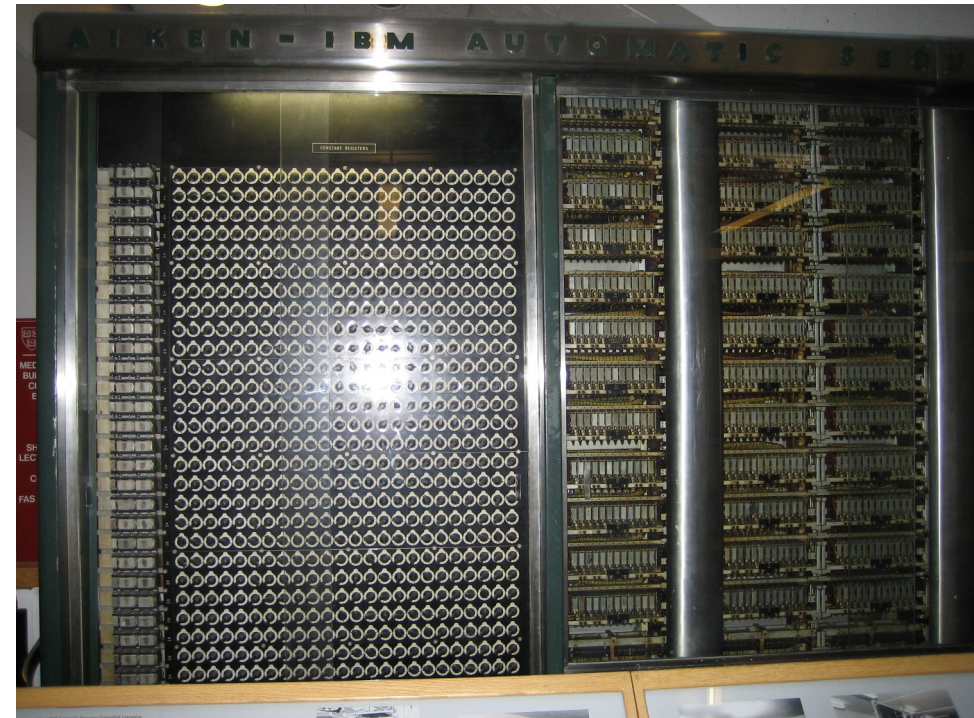
## 3) Grace Hopper e o Flow-Matic

A avó de todos os programadores e uma das maiores matemáticas do Século XX.

Ajudou nos esforços de guerra participando das equipes de computadores humanos, para calcular tabuas de coordenadas para artilharia.

Os canhões e a necessidade de processamento geográfico  
Colaboração e participação ativa na construção do primeiro computador MARK I  
Lendária criação do Flow-Matic

# COBOL



Grace Hopper e Mark I

# COBOL



## 4) CODASYL e as Linguagens de Terceira Geração

A Torre de Babel Informática, a difícil arte de especificar um programa, com as diferentes máquinas.

Falta de um padrão.

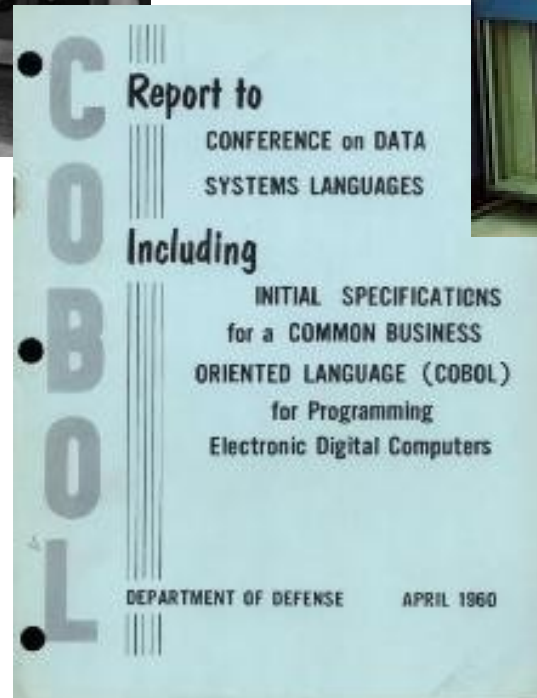
Não comunicação entre computadores de diferentes empresas e mesmo dentro da própria empresa.

A necessidade do concessão e os olhos nos gordos contratos com o Governo Americano.

CODASYL as Big 6: Burroughs, IBM, Honeywell, RCA, Sperry e Sylvania

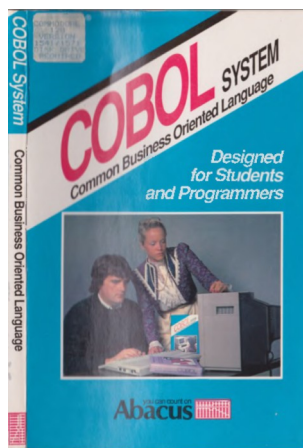
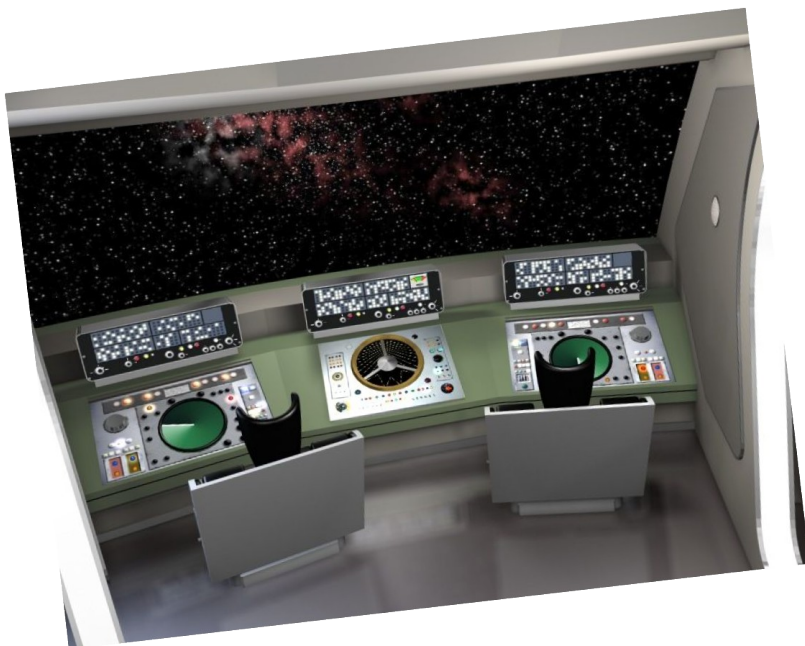


# COBOL



Assim surgiu a lenda.

# COBOL



## 5) Cobol 1968, 1974, 1985, 2002, 2014 e além

Após o longo braço de ferro em 1958 e a publicação dos primeiros apontamentos sobre o COBOL.

Ao longo das décadas foram surgindo novas implementações, com inúmeras melhorias, sempre devido ao aumento da capacidade de memória, velocidade de processamento, pedidos de clientes e ecossistema informático.

Uma linguagem de programação de Terceira Geração, facilmente lida por humanas, feita para humanos interagirem com as máquinas, sem a necessidade teórica de serem super matemáticos ou engenheiros.

O Homem comum ganhou seu espaço.

# COBOL



Standards

Sectors

About us

News

Taking part

Store

## ISO/IEC 1989:2023

Information technology

Programming languages, their environments and system software interfaces

Programming language COBOL

**COBOL 1959- Codasyl**

*COBOL 1968 - **ANSI***

*ISO 1989:1974*

*ISO 1989:1978*

*ISO 1989:1985*

**ISO 1989:1991- ISO**

**ISO/IEC 1989:2002**

**ISO/IEC 1989:2014**

***ISO/IEC 1989:2023***

<https://www.iso.org/standard/74527.html>

# COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID.      HELLOWORLD.

*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.

DATA DIVISION.
FILE SECTION.

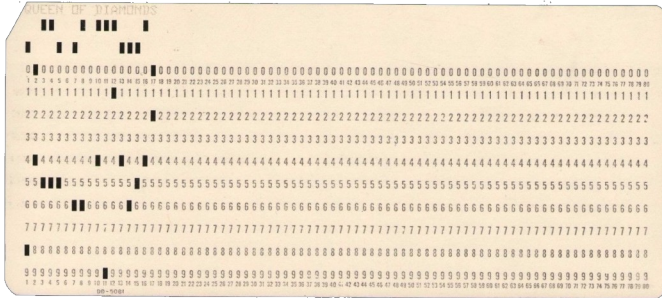
PROCEDURE DIVISION.
MAIN-LOGIC SECTION.
BEGIN.
DISPLAY "Hello world!" LINE 15 POSITION 10.
STOP RUN.
MAIN-LOGIC-EXIT.
EXIT.
```

COBOL : z/OS,  
AIX, AS400,  
Linux X86,  
Microfocus e o  
mítico  
MS Cobol 5.00





# COBOL



## 6) Folha de Codificação e o cartão perfurado

No início da programação, os engenheiros e matemáticos utilizavam-se de potenciômetros para codificar as instruções nos primeiros computadores.

Em outras situações utilizavam da genial ideia de tecelões belgas, reciclada por Holleryth: os cartões perfurados.

Não era nada fácil a vida dos pioneiros da informática, um calor infernal das válvulas, posteriormente transistores, até a revolução dos circuitos integrados.

Guardem bem, o próximo avanço foi a Folha de Codificação, intimamente relacionada ao COBOL até os dias de hoje.



# COBOL

COBOL STATEMENT										IDENTIFICATION									
PAGES	LINE	FUNCTION	AREA	STATEMENT	LINE	FUNCTION	AREA	STATEMENT	LINE	FUNCTION	AREA	STATEMENT	LINE	FUNCTION	AREA	STATEMENT	LINE	FUNCTION	AREA
00301				OPEN INPUT FILE-IN															
00302				OUTPUT FILE-OUT.															
00303				MOVE SPACES TO RECORD-OUT.															
00304				READ FILE-IN AT END MOVE 1 TO EOF.															
00305	*2.0			PROCESS															
00306				PERFORM GPA UNTIL EOF=1.															
00307	*3.0			STOP															
00308				CLOSE FILE-IN															
00309				FILE-OUT.															
00310				STOP RUN.															
00311				PROCESS GPA															
00312				IF STUDENT-CLASS-IN = 1															
00313				ADD GRADE-1, GRADE-2, GRADE-3, GRADE-4 GIVING TOTAL-GRD															
00314				DIVIDE TOTAL-GRD BY 3 GIVING AVERAGE-GRADE-OUT															
00315				MOVE STUDENT-NAME-IN TO STUDENT-NAME-OUT															
00316				WRITE RECORD AFTER ADVANCING 1 LINES.															
00317				ELSE															
00318				NEXT SENTENCE.															
00319				READ FILE-IN AT END MOVE 1 TO EOF.															
00320																			

# COBOL

The image shows a COBOL program structure in a terminal window. The program is named MATEKS.COBOL.SRCLIB(COB001) and is at version 01.00. The terminal shows the following structure:

```

VIEW      MATEKS.COBOL.SRCLIB(COB001) - 01.00      Columns 00001 00072
Command ==>
=COLS> ---+---1---+---2---+---3---+---4---+---5---+---6---+---7---
***** Top of Data *****
000100 000100
000200 000200
000300 000300
000400 000400
000500 000500
000600 000600
000700 000700
000800 000800
000900 000900 ENVIRONMENT DIVISION.
001000 001000 DATA DIVISION.
001100 001100 WORKING-STORAGE SECTION.
001200 001200
001300 001300 01 WS
001400 001400
001500 001500 PROCEDURE DIVISION.
  
```

Annotations explaining the structure:

- Colunas de 01 a 06 : Reservadas para numeração de linhas.** (Columns 01 to 06: Reserved for line numbering.)
- Coluna 7 - Indicator Area : Usada para \*, D e continuação** (Column 7 - Indicator Area: Used for \*, D and continuation)
- Colunas 8 a 11 - Area A** (Columns 8 to 11 - Area A)
- Colunas 73 a 80 ID do Programa não visível** (Columns 73 to 80: Program ID not visible)
- Colunas 12 a 72 - Area B** (Columns 12 to 72 - Area B)

Estrutura Física de um programa COBOL

# COBOL



## 7) O maior computador de sempre

Nossa mente e os seus periféricos principais LAPIS & PAPEL.

Faça programas e seus fluxos no papel antes de ir ao computador. Tenha uma previsão dos inputs, principais funções e seus outputs.

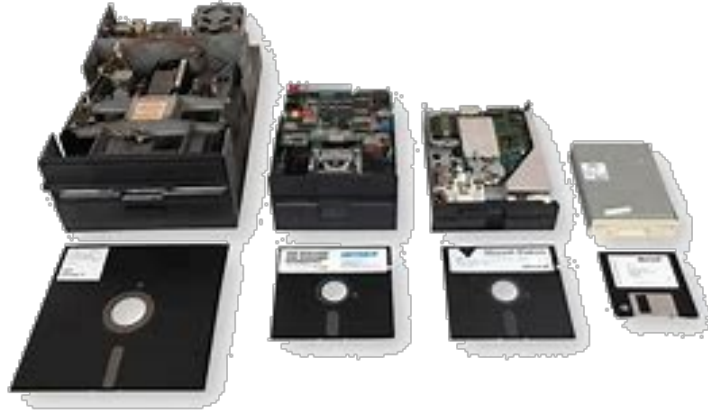
O Fluxograma, uma representação gráfica do programa.

Lista de tarefas e workflow, Logica Estruturada e o processamento procedural.

Programas Top Down



# COBOL



Velhas unidades de armazenamento físico, os famosos disquetes, fitas e cartuchos

# COBOL

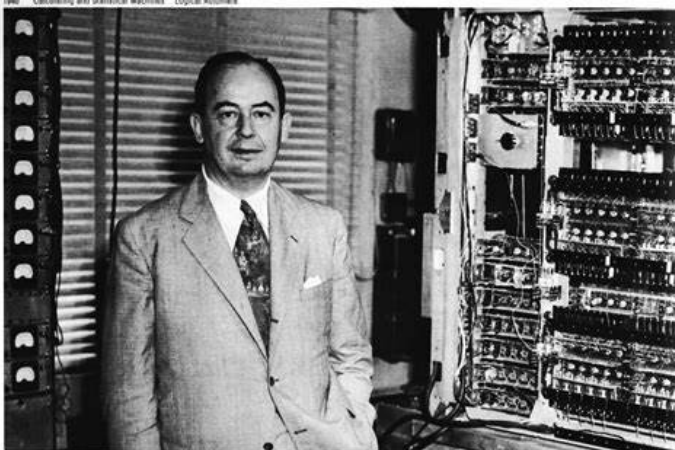


## 8) Mainframe e a Arquitetura de von Neumann

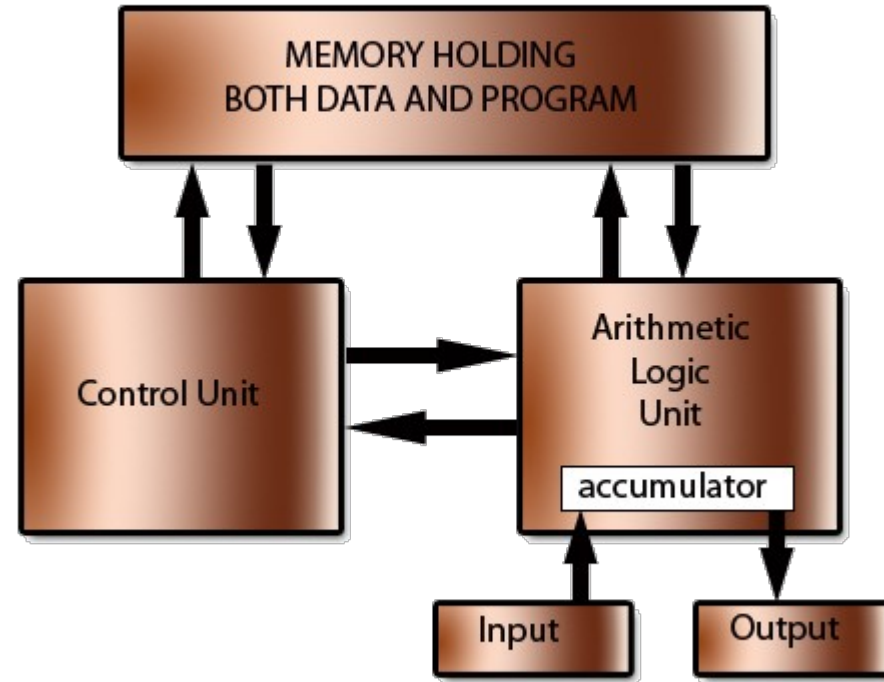
Input + Processamento = Output, como surgiu a ideia da computação, a arquitetura raiz de todos os computadores nutelas.

A CPU, memoria, portas e periféricos os primeiros computadores com reles até o surgimento do CICS e MVS.

Os inúmeros projetos de computadores e a vitoria da IBM no mercado informático.

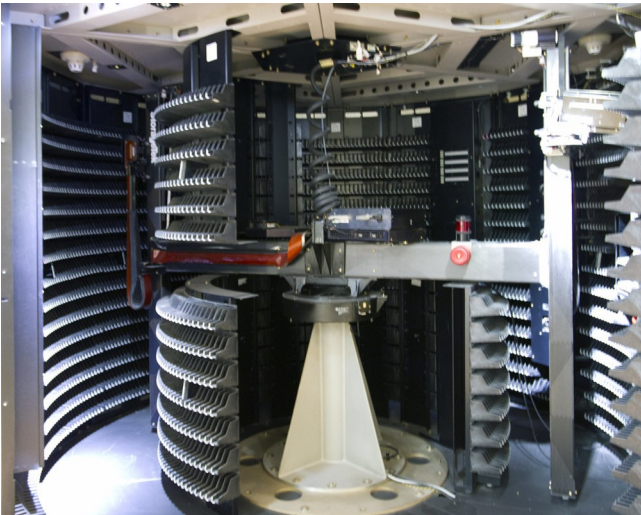


# COBOL



A arquitetura Von Neumann para armazenamento de dados e processamento de programas.

# COBOL



## 9) Processos Batch e Online

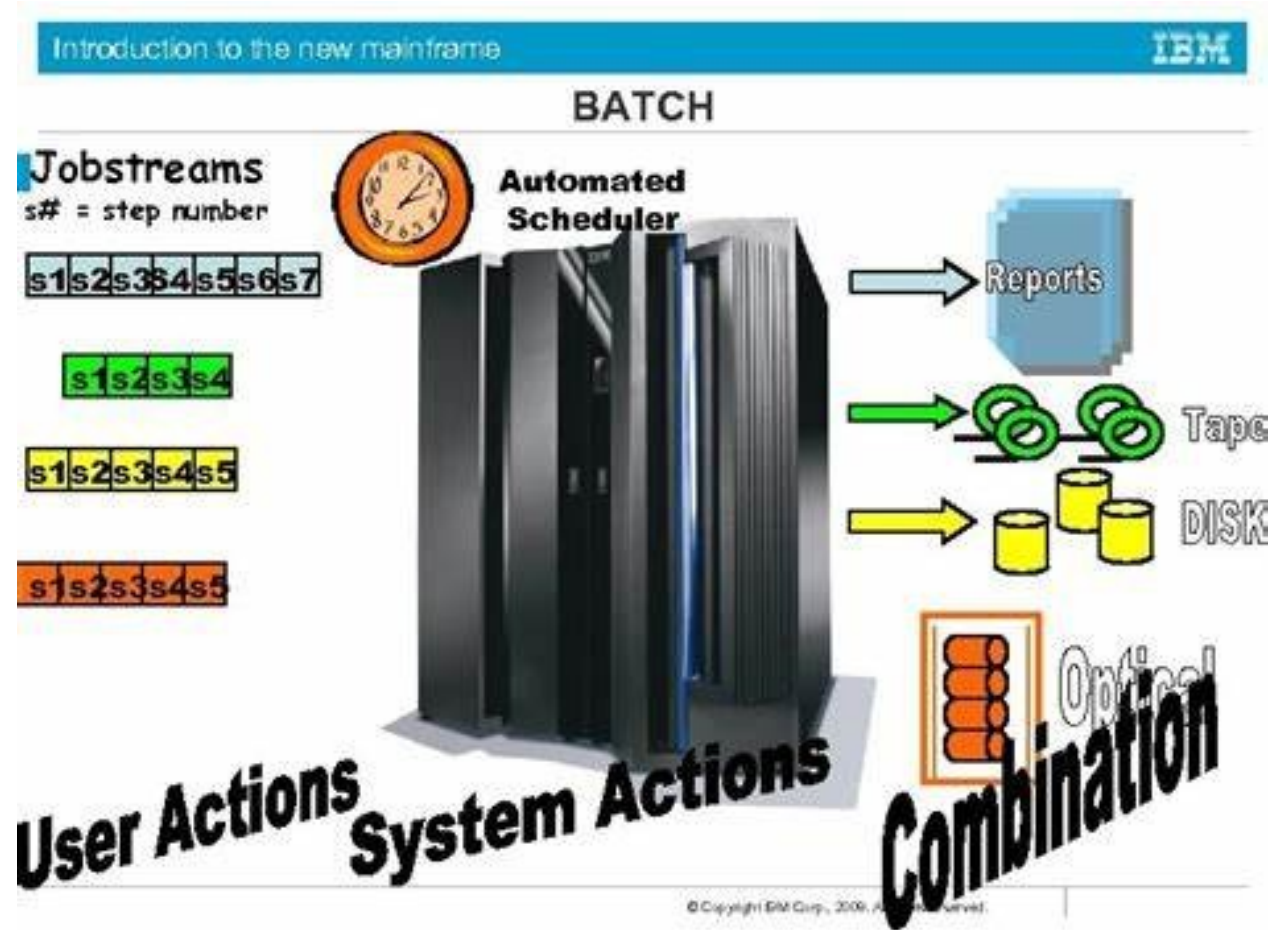
Falamos do CICS e veremos os dois tipos de processos no Mainframe e o famoso terminal 3270.

Batch usando procedures, jobs e jcls. Processos encadeados com mínima interferência do operador.

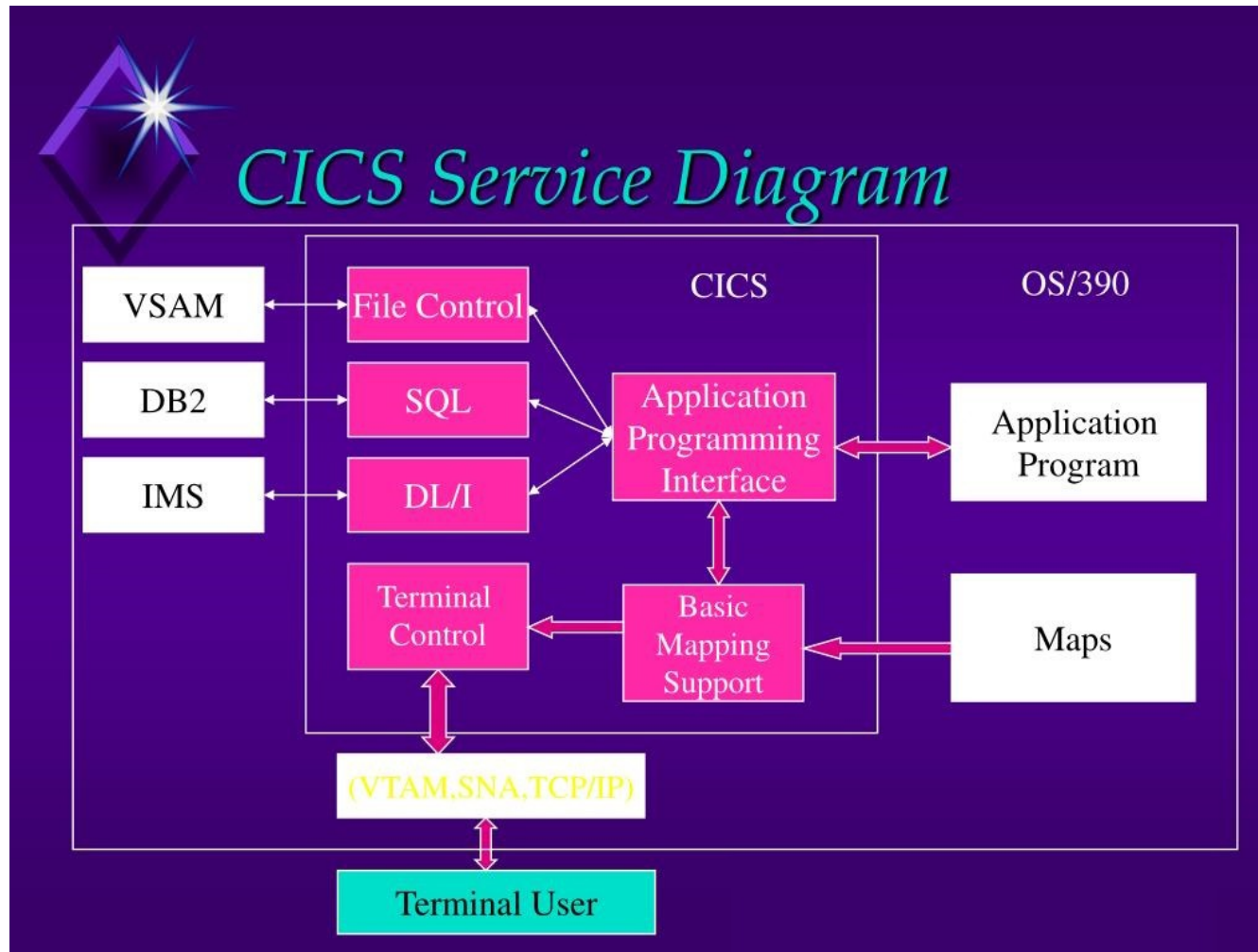
Online através de uma interface gerenciada pelo CICS, tem a função pseudo-conversacional com o sistema, executando transações previamente cadastradas e alocadas no sistema operacional: MVS, zOs e Zowe



# COBOL



# COBOL



# COBOL

```
000100 //IEFBRI41 JOB 'AK99',AVINASH,NOTIFY=&SYSUID
000200 //*****
000300 //*THIS UTILITY USED TO ALLOCATE OR DEALLOCATE A DATASET
000400 //*CREATES BOTH PS AND PDS FILES ALSO DELETES SAME
000410 //*****
000420 //MYPROC PROC ----->MYPROC IS PROCEDURE NAME THEN PROC THIS IS SYNTAX
000500 //DELETE EXEC PGM=IEFBRI4
000600 //SYSOUT DD SYSOUT=*
000700 //SYSPRINT DD SYSOUT=*
000800 //SYSUT1 DD DSN=AK999K.AVINASH.PS4,
000900 // DISP=(NEW,CATLG,DELETE),
000910 // DCB=(LRECL=80,BLKSIZE=8000,RECFM=FB),
000920 // SPACE=(TRK,(2,3),RLSE)
000930 // PEND ----->END OF PROC.
000940 //*****
000950 //STEP2 EXEC MYPROC ----->HERE I AM CALLING MY INSTREAM PROC IN SAME JCL
001000 //SYSIN DD DUMMY
001100 //
```

```
DD DSN=RACFID.SAMPLE.INPUT,DISP=SHR
DD DSN=RACFID.SAMPLE.OUTPUT,
DISP=(NEW,CATLG,CATLG),DATACLAS=DEVLOPS
DD SYSOUT=*
DD SYSOUT=C
```

## 10) JCL e a compilação de programas

TSO e a edição de programas

Arquivos sequenciais Fixos e Particionados

Biblioteca de programas

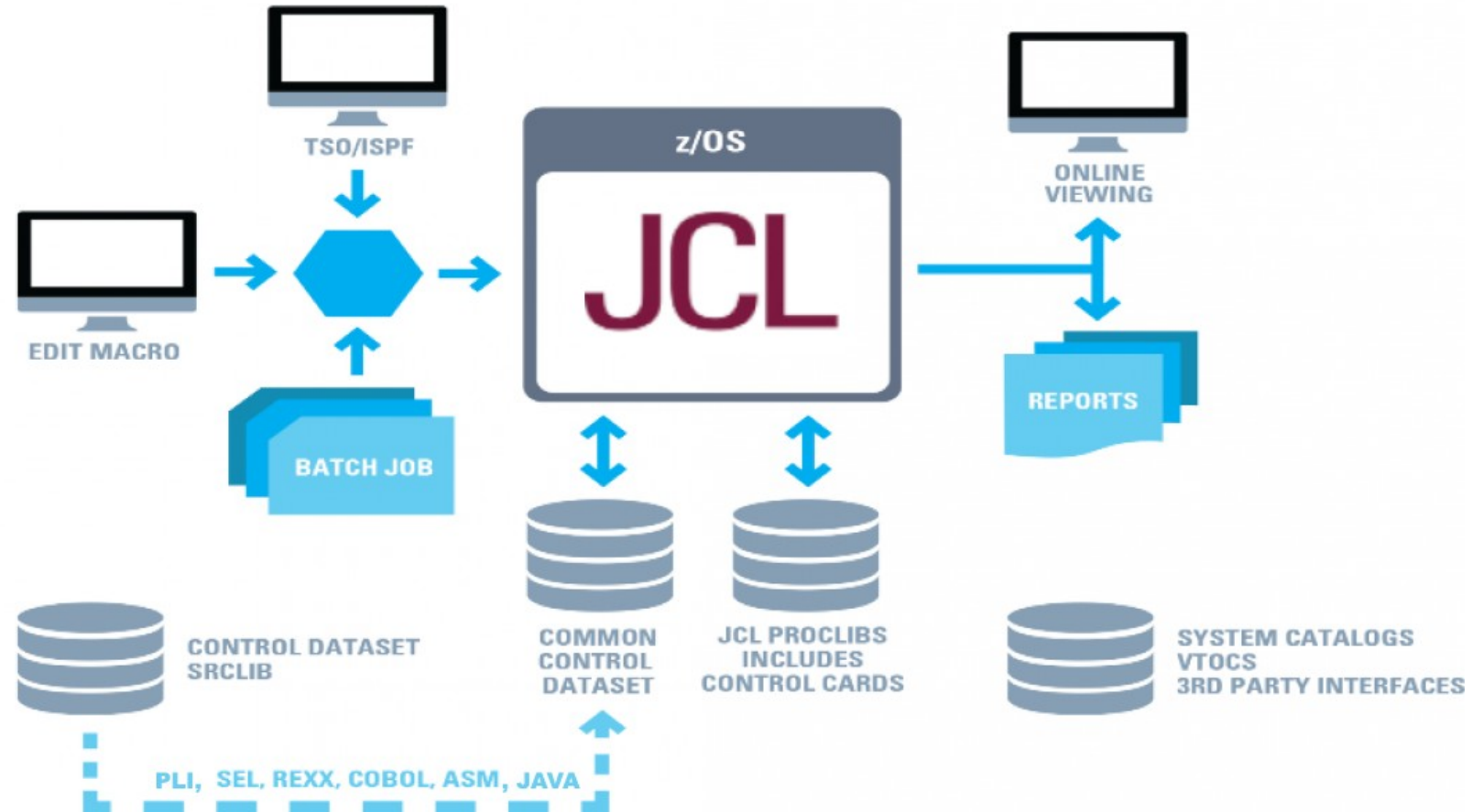
As tais 80 colunas e herança da Folha de Codificação

Áreas A, B e Especial; coluna de números e colunas de comentários

Sub no Job.

Expansão, Compilação, Linkedição e Binario.

# COBOL





# COBOL



## 11) Enfim a codificação

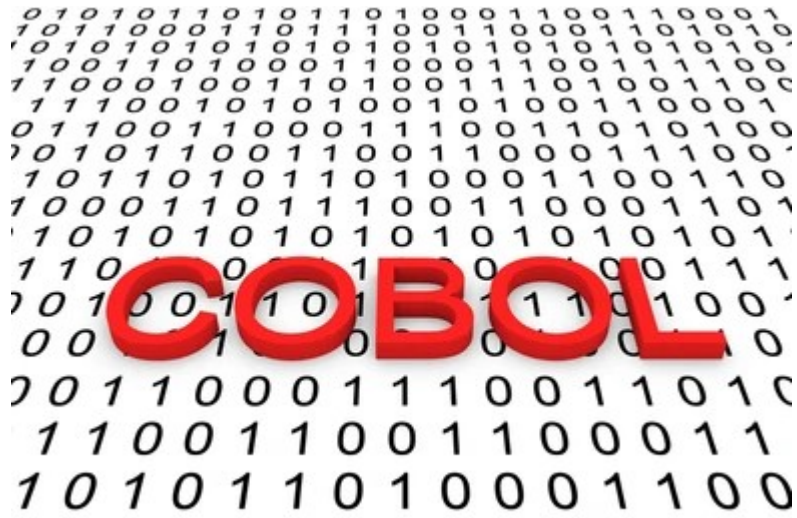
Nossa jornada esta apenas começando.

Foi uma chuva de conteúdo, muitos mnemônicos, palavras estranhas e siglas.

Mas veras que o COBOL é uma linguagem super fácil de aprender, ela é apenas manhosa e caprichosa, as vezes nos passa uma rasteira, porém com o tempo dominamos os comandos.

Conceitos extra: Copybook, CopyProcedure, Funções Intrínsecas e Parágrafos

# COBOL



shutterstock.com · 474702454

Duvidas???