CEFET-MG / Campus Divinópolis

Introdução à Linguagem de Programação PDL2

(Parte 2)

MECATRÔNICA

CEFET-MG

Prof. Renato S. Dâmaso - 03/10/2022

Componentes da linguagem

PDL2 possui pré-definições para:

- Conjunto de caracteres

Letters:	a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Digits:	0 1 2 3 4 5 6 7 8 9
Symbols:	@ < > = / * + - _ , ; . # $ [] % {} \ : ! ()
Special Characters:	blank (space), tab

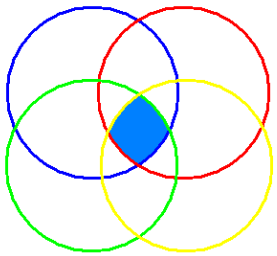
PDL2 não distingue maiúsculas de minúsculas.

CEFET-MG

Componentes da linguagem

PDL2 possui pré-definições para:

- Conjunto de caracteres
- Palavras, símbolos e operadores reservados



MECATRÔNICA

CEFET-MG

Componentes da linguagem

Palavras identificam: seções de um programa

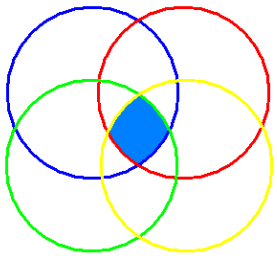
Ex: ROUTINE … BEGIN … END

tipos de dados

Ex: BOOLEAN, REAL, POSITION, TIME, etc

e palavras chave (*key words*)

Ex: VAR, VEC,



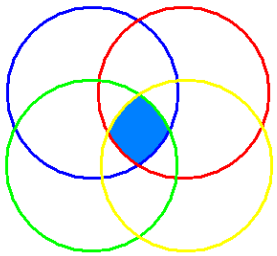
MECATRÔNICA

CEFET-MG

Componentes da linguagem

Símbolos usualmente apontam uma afirmação.

Ex: PLC, VAR,



Componentes da linguagem

Operadores indicam um cálculo ou operação.

Ex: NOT, AND, XOR, etc

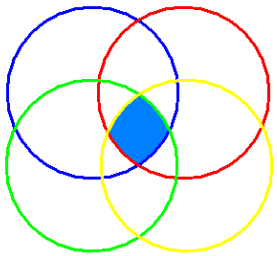
)	<	>	<=
>=	<>	=	+
-	*	/	**
,	.	..	:
::	:=	;	#
@	+=	-=	

Componentes da linguagem

PDL2 possui pré-definições para:

- Conjunto de caracteres
- Palavras, símbolos e operadores reservados
- Identificadores pré-definidos: constantes, variáveis, campos e rotinas que a linguagem já identifica.

  Variáveis pré-definidas são precedidas de $.

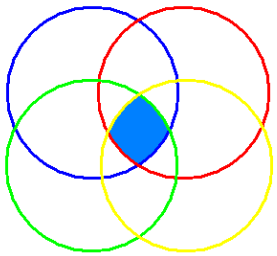


CEFET-MG

Componentes da linguagem

Identificadores definidos pelo usuário são nomes que o programador escolhe para identificar: programas; variáveis; constantes; rotinas; rótulos (labels); tipos; e campos.

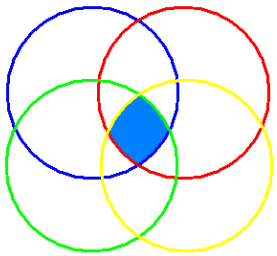
Comentários são separados por “– –”



CEFET-MG

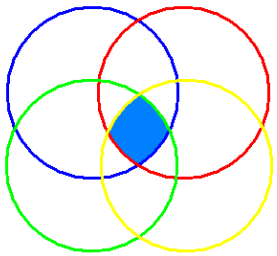
Unidades de medida usadas na PDL2

Distance:	millimeters(mm)
Time:	milliseconds(ms)
Angles:	degrees (°)
Linear Velocity:	meters/second (m/s)
Angular Velocity:	radians/second (rad/s)
Current:	ampere (A)
Encoder/Resolver Data:	revolutions (2 ms)



Tipos de dados na linguagem PDL2

–INTEGER
–REAL
–BOOLEAN
–ARRAY
–RECORD
–VECTOR
–POSITION
–JOINTPOS
–XTNDPOS
–NODE
–PATH
–SEMAPHORE



MECATRÔNICA

CEFET-MG

Tipos de dados na linguagem PDL2

- Record:

  Representa uma coleção de dados agrupados em um só nome.

  Cria uma definição de tipo de dado que é avaliado no sistema inteiro.

Tipos de dados na linguagem PDL2

- Vector:

  É um tipo de dado que possui módulo e direção.

  É formado por três valores reais, que representam um deslocamento numa determinda direção no espaço cartesiano.

Tipos de dados na linguagem PDL2

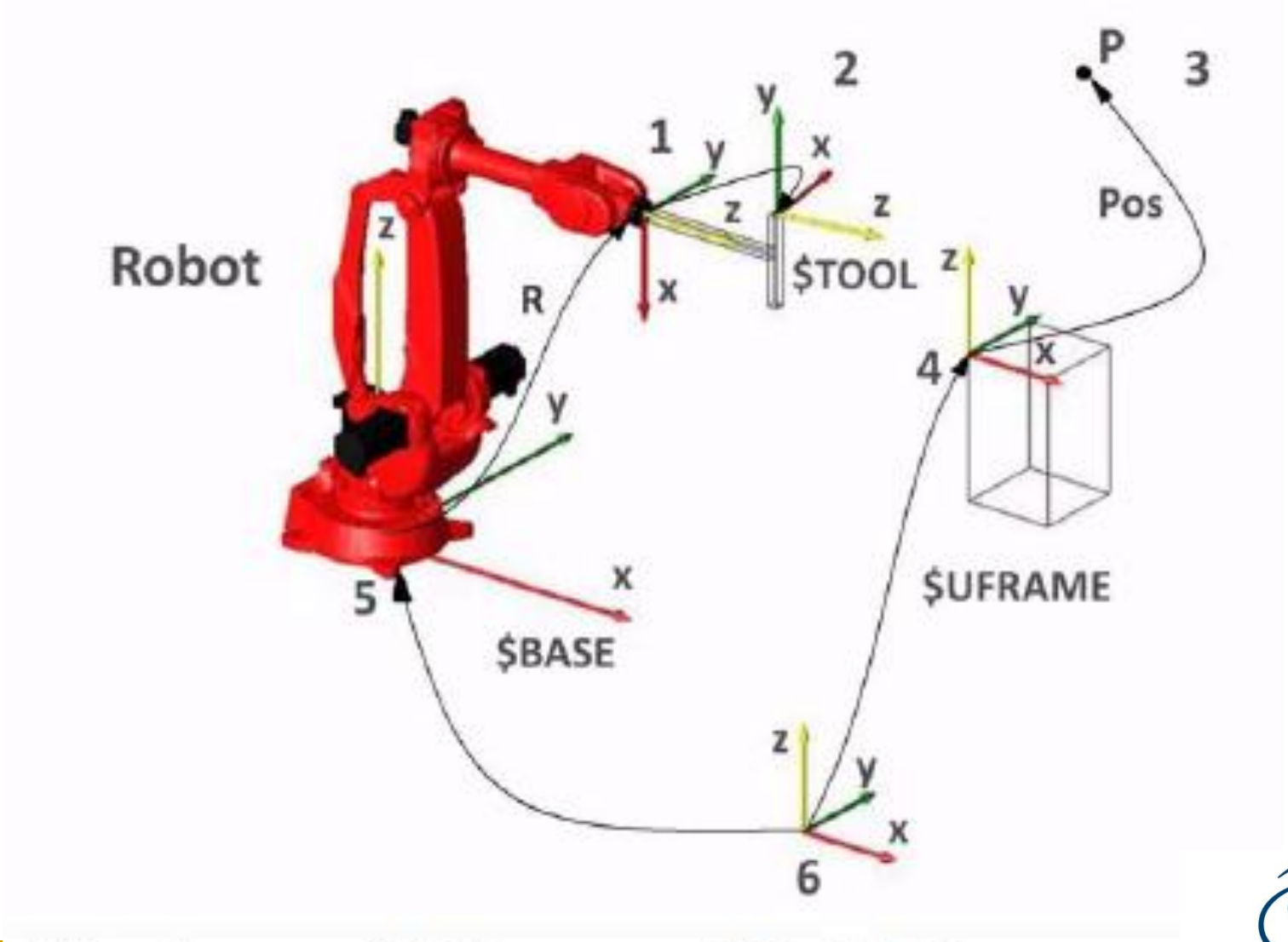
- Position:

  Descreve a posição e orientação de um sistema cartesiano em relação a outro de referência, chamado de *frame* de partida (ou inicial).

  Geralmente é usado para definir o ponto final para uma função MOVE, que é o ponto a ser alcançado pela ferramenta na extremidade do robô em relação ao *frame* do usuário.

CEFET-MG

Linguagem PDL2 – Frames



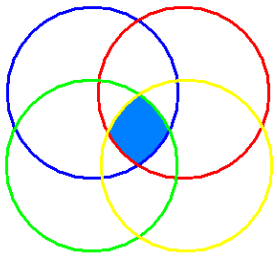
1 - Flange Frame
2 - Tool Frame
3 - Taught Position
4 - User Frame
5 - Base Frame
6 - World Frame

CEFET-MG

Tipos de dados na linguagem PDL2

- Position:

```
PROGRAM postest
VAR
  pos_var : POSITION
BEGIN
  pos_var := POS(294, 507, 1492, 13, 29, 16, )
END pos
```

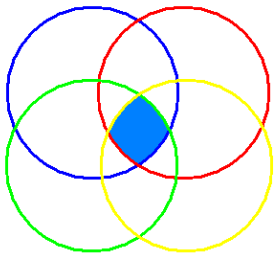


MECATRÔNICA

CEFET-MG

Linguagem PDL2 – Estrutura Básica

```
PROGRAM name <attributes>
 <import statements>
 <constant, variable, and type declarations>
  <routine declarations>
BEGIN <CYCLE>
  <executable statements>
END name
```

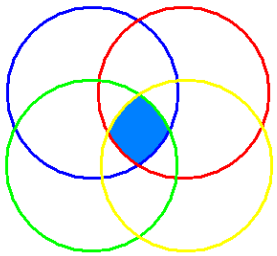


MECATRÔNICA

CEFET-MG

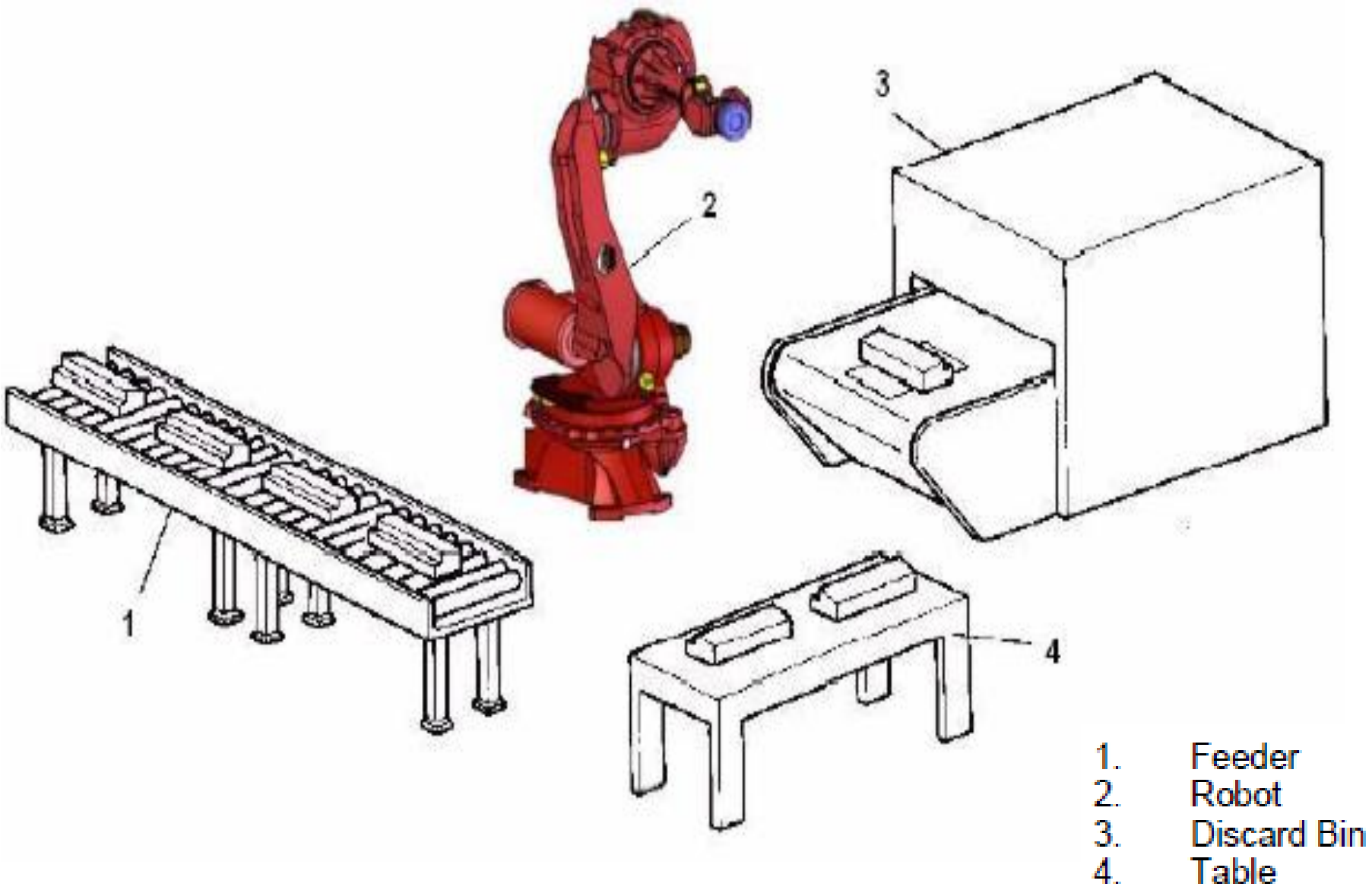
Linguagem PDL2 – Estrutura Básica

- O identificador de programa, que é usado também para nomear o arquivo que armazena o programa.
- Esse identificador deve vir após PROGRAM e, no fim do programa, após END.
- Variáveis podem ser declaradas entre PROGRAM e BEGIN.
- Rotinas devem vir antes do programa principal.



CEFET-MG

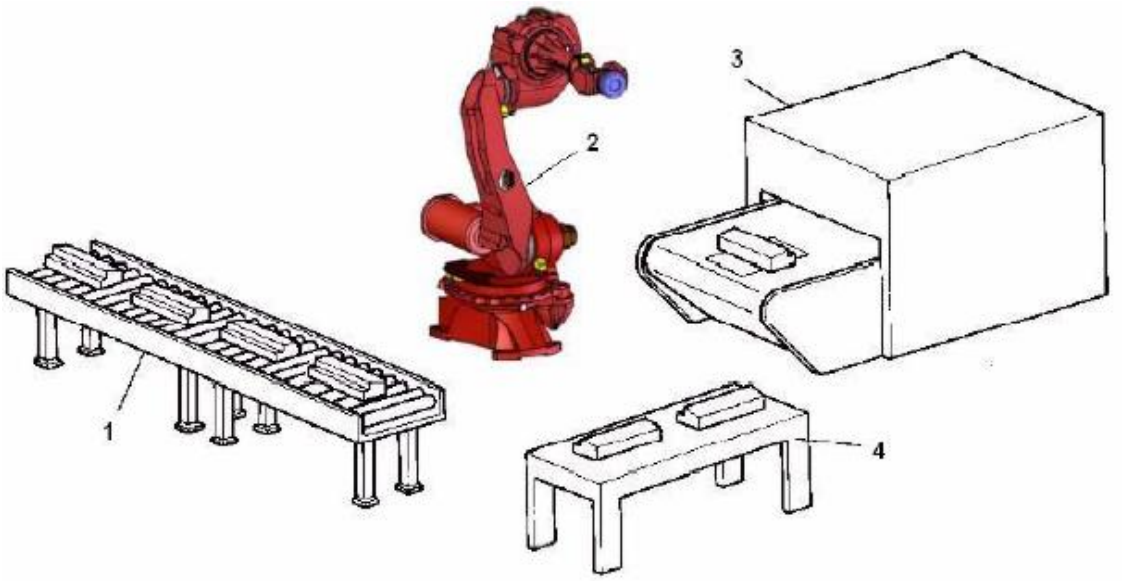
Exemplo de programa PDL2



1. Feeder
2. Robot
3. Discard Bin
4. Table

Exemplo de programa PDL2

```
PROGRAM pack
VAR
   perch, feeder, table, discard  :  POSITION
BEGIN CYCLE
   MOVE TO perch
   OPEN HAND 1
   WAIT FOR $DIN[1] = ON
-- signals feeder ready
   MOVE TO feeder
   CLOSE HAND 1
   IF $DIN[2] = OFF THEN
-- determines if good part
      MOVE TO table
   ELSE
      MOVE TO discard
   ENDIF
   OPEN HAND 1
-- drop part on table or in bin
END pack
```



1. Feeder
2. Robot
3. Discard Bin
4. Table

CEFET-MG