

# Sistemas de Computação 1.6/2

notação de tempo  $\rightarrow$  B.S  
(907.)

TPC (10%)

recursos de função

1q / tempo 1 / função  
1  
10 valores

$\rightarrow$  Níveis de um computador:

$\rightarrow$  é considerado computador a/distância do que não é um  
 $\rightarrow$  data-dependent instruction sequence computador:  
 $\rightarrow$  data-independent data selection.

$\rightarrow$  depende do dado a tempo de execução.

$\rightarrow$  dados independentes

depende de outro

dado. (ex. quando a

sema 7, os outros

7, vai depender do

dado inteiro da "7")

if (x == 1)

...

depende dos dados.

else

...

x = 3  $\Rightarrow$  depende dos dados.  
resolver o "else".

$\rightarrow$  é uma calculadora se não tiver 1 ou nenhum destes tópicos.

$\rightarrow$  programa: sequência de instruções que descrevem a ~~função~~  
função a ser executada.

ex: projeto: computadores que foi programado por uma tarefa específica  
para gerar um

programa em Haskell é transformado em programa para que  
o pc possa gerar.

Configurar piladas

o PC não

programa o código logo.



→ transformar o programa numa linguagem em formato que o PC entenda.

→ máquina que apenas executa apps

Level 6

User

5 Problem-oriented language → program  
4 Translation (compiler)

o que  
vai  
o  
processador  
de  
dados

assembly language → precisa de  
translation (assembler)

3 Operating system Machine → peça de software que a máquina  
precisa para funcionar  
2 partial interpretation (operating system)

1 instruction set architecture (ISA)

direct execution or interpretation (microprogram)

micro architecture

hardware

digital logic

0/1  
ou  
off/on  
ou  
0/1

utilizam

códigos de código binário

001 (códigos de bits  
constituídos de bits)

(cada instrução  
é 1 bit)

2 (1 bit)

0

→ códigos de

bits que o processador executa

Fronteira dos níveis 4 e 3.

O porque do código binário? A eletrónica flui por elétrons; e + fácil  
distinguir 2 níveis (0 e 1) 2 níveis.

↳ pensar em porta (ou ser aberta/fechada ⇒ não é mais aberta...)