

Introdução à Programação

Fernanda Barbosa

Fernando Birra

Luís Caires

Armanda Rodrigues

Licenciatura em Engenharia Informática FCT UNL

Introdução à Programação 2010 - 2011

Docentes:



Fernanda Barbosa



João Araújo



Miguel Goulão



Miguel Monteiro



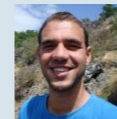
André Guerreiro



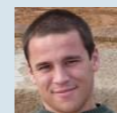
Miguel Lourenço



Nuno Cunha



Ricardo Alves



Vasco Pessanha

Licenciatura em Engenharia Informática FCT UNL

O que é a Programação?

Programação

- A **programação** é a actividade mais importante da Engenharia Informática
- Todos os sistemas informáticos são constituídos por alguns componentes **físicos** (hardware) e por alguns componentes **lógicos** ou **virtuais** (software)
- Hardware: computador, iPod, calculadora, Xbox, automóvel, terminal GPS, telemóvel, ...
- Software: Windows, Messenger, Firefox, SAP, Compilador de Java, Word/Excel, Need for Speed,...
- A **programação** é a actividade que tem por objectivo a construção de software.

Tipos de software

- Para além do hardware, um sistema informático necessita de múltiplos componentes de software:
- Software de sistema, por exemplo:
 - O sistema operativo (Windows, Linux, Unix, MacOS), que auxilia os utilizadores a utilizar os recursos da máquina ...
 - Componentes especiais destinados a lidar com extensões ao sistema hardware básico (por exemplo, um “driver” para permitir a utilização de uma placa gráfica especial) ...
 - Compiladores de linguagens de programação ...
- Software aplicativo, por exemplo:
 - Aplicações de uso pessoal (editores de texto, editores gráficos) ...
 - Sistemas de suporte a bases de dados ...
 - Aplicações de comunicação e uso da internet ...
 - Aplicações de suporte a sistema de informação empresarial, ...
 -

Estrutura dos Sistema de Software

- Qualquer sistema de software é constituído pela composição de múltiplos programas, que colaboram entre si para garantir a funcionalidade pretendida.
 - Ordenar informação
 - Seleccionar informação
 - Armazenar informação
 - Efectuar cálculos
 - Visualizar informação
- Uma ideia chave na informática é a possibilidade de decomposição de uma tarefa complicada em várias tarefas mais simples.
- Um dos aspectos importantes na programação é a capacidade de fazer boas decomposições de tarefas.

O que é um programa ?

- Um programa é uma descrição precisa de uma tarefa, de forma a poder ser executável por uma máquina.
- Tipicamente essa máquina é um computador.
- Às vezes o computador está escondido: leitor de CDs, automóvel, leitor de MP3, telemóvel...
- Um programa descreve minuciosamente todas as acções básicas assim como todas as decisões necessárias para executar a dita tarefa.
- O computador executa as instruções descritas num programa de forma cega, tendo o programa de ser muito completo e preciso, prevendo todas as situações.

Exemplo: Robot Programável

Operações reconhecidas pelo Robot:

– FORWARD

O robot dá um passo em frente

– LEFT

O robot vira 90° para a esquerda

– RIGHT

O robot vira 90° para a direita

– BACK

O robot dá um passo à retaguarda

– HELLO

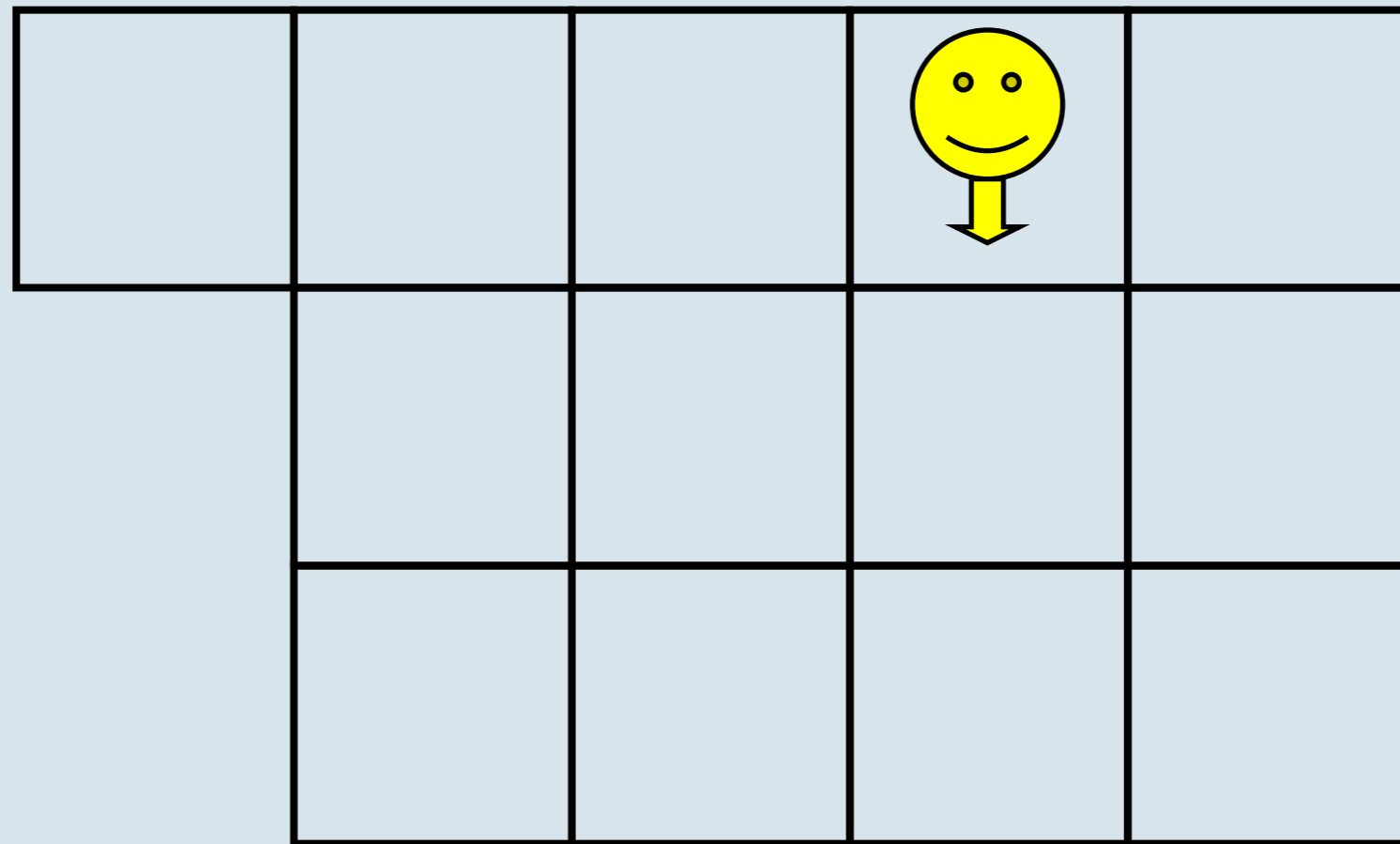
O robot diz “Olá”.



Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)

- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- HELLO

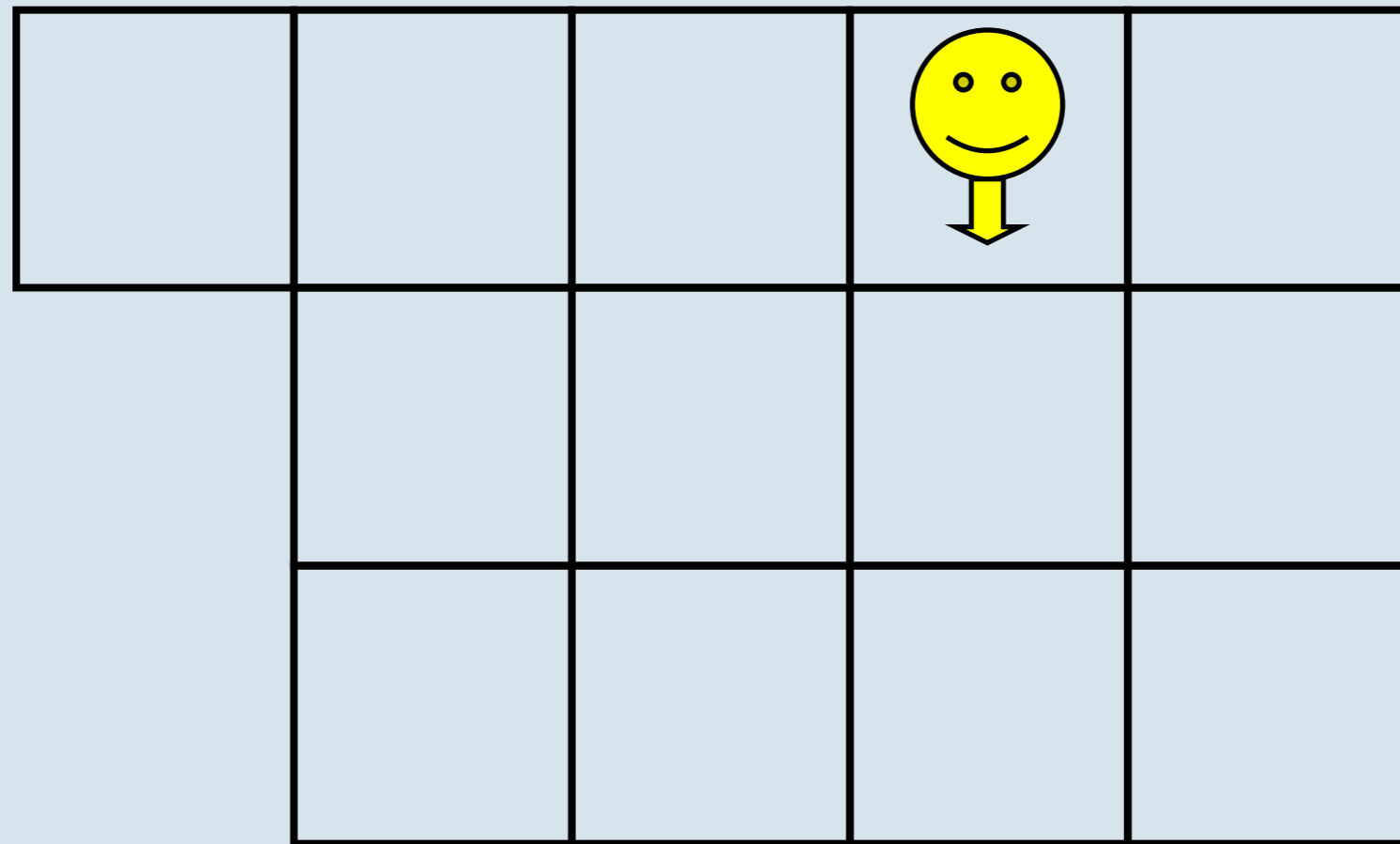


Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)




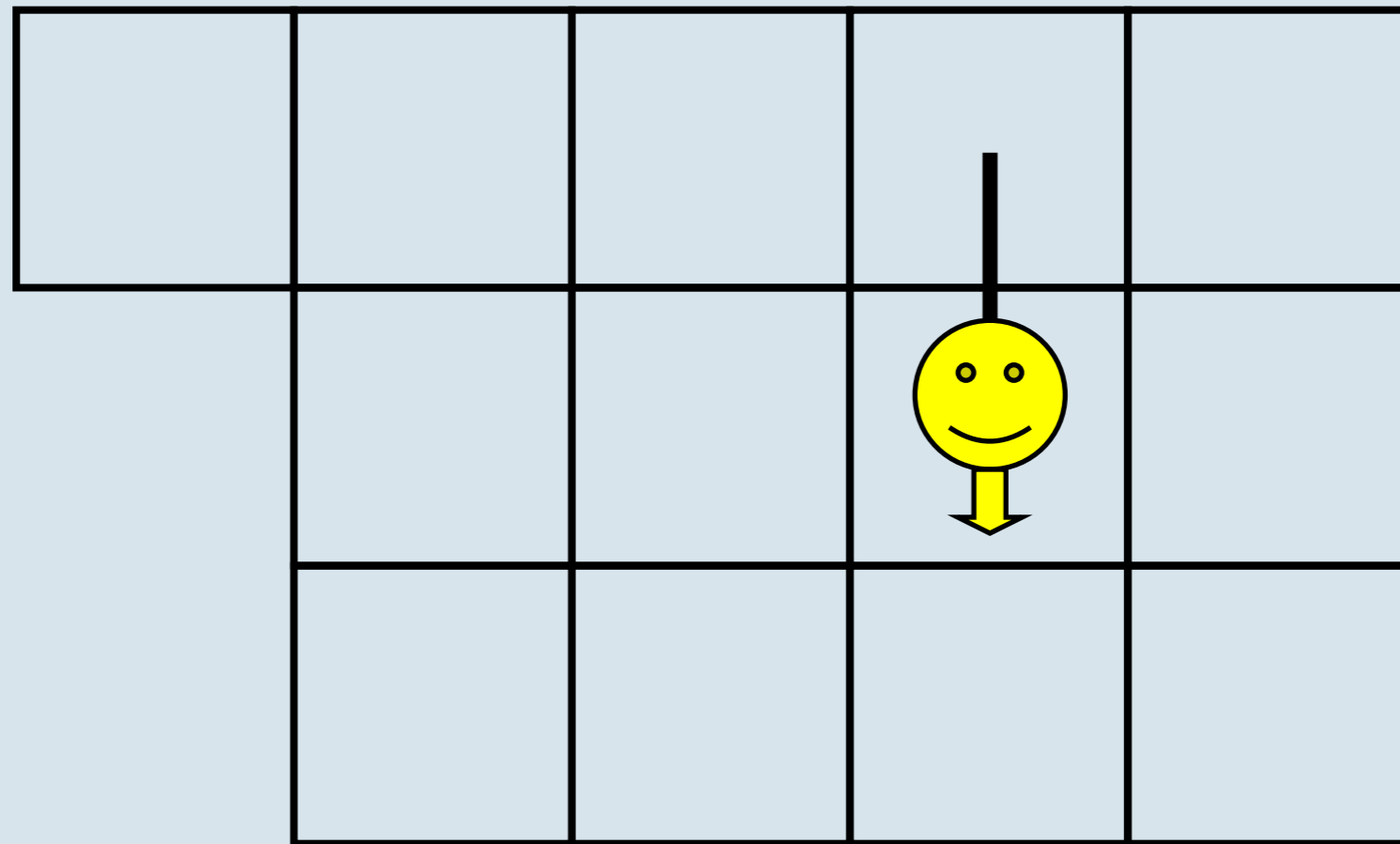
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- HELLO



Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)

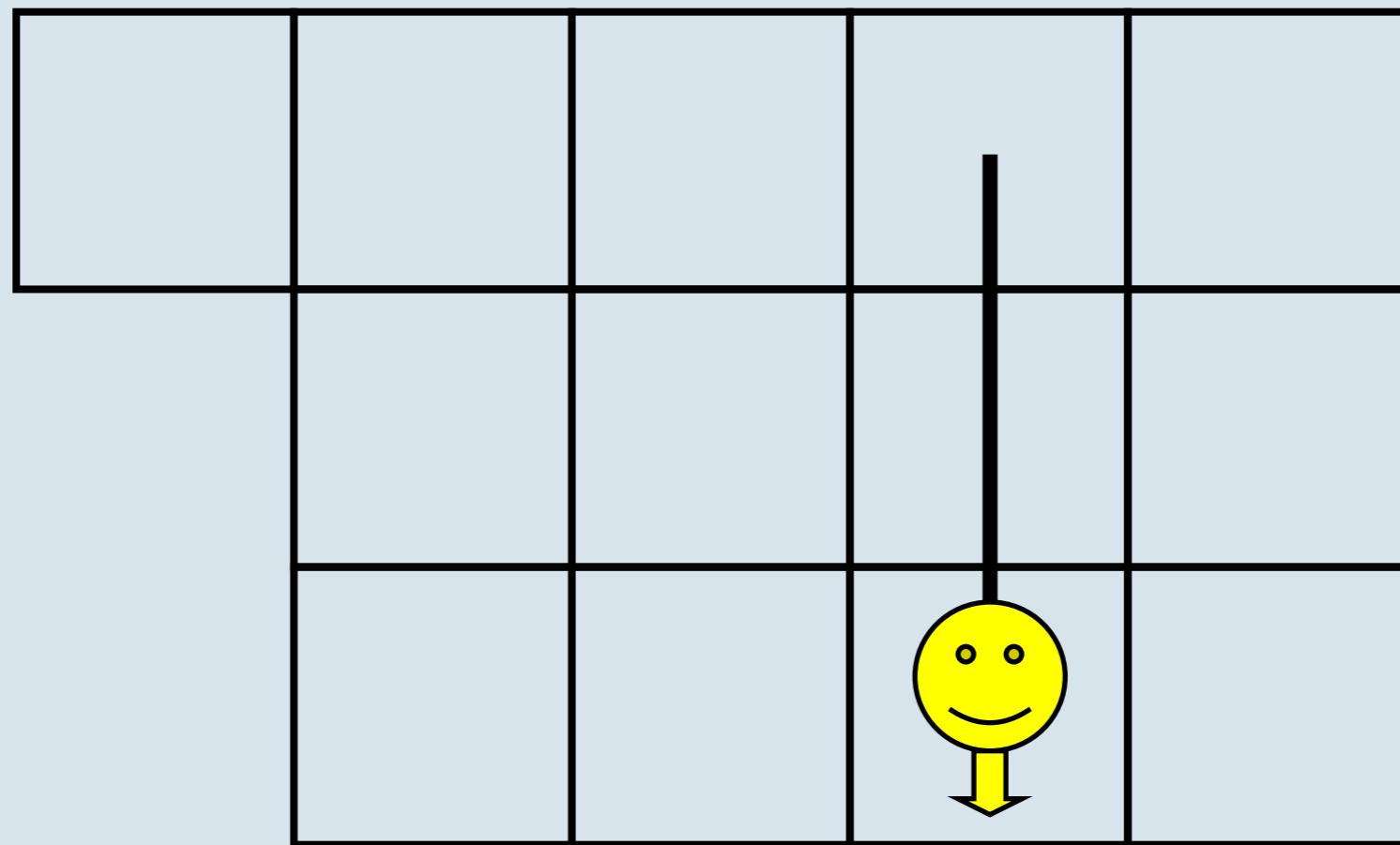
- 
- FORWARD
 - FORWARD
 - RIGHT
 - FORWARD
 - FORWARD
 - RIGHT
 - FORWARD
 - FORWARD
 - RIGHT
 - FORWARD
 - FORWARD
 - HELLO



Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)

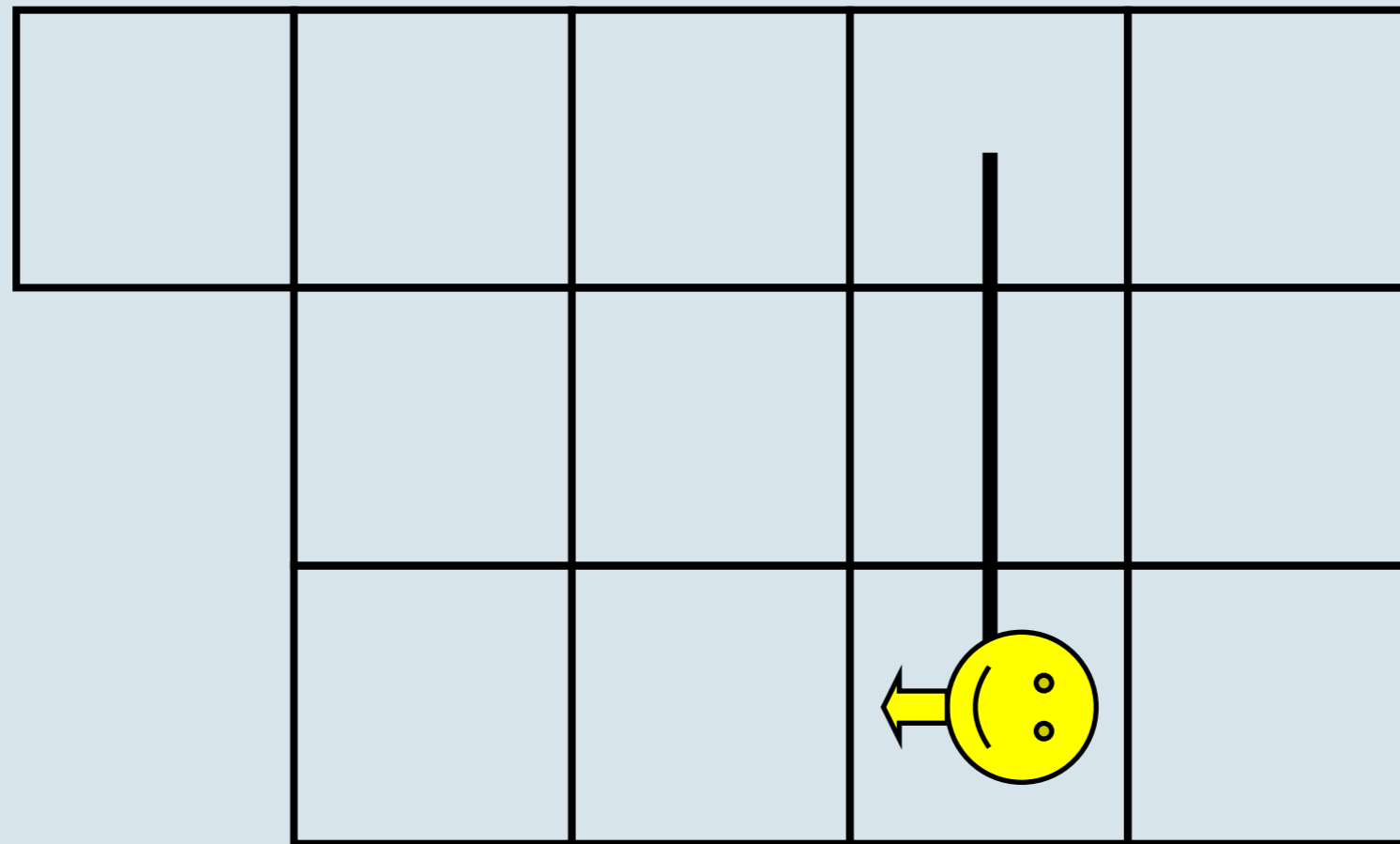
- FORWARD
- – FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- HELLO



Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)

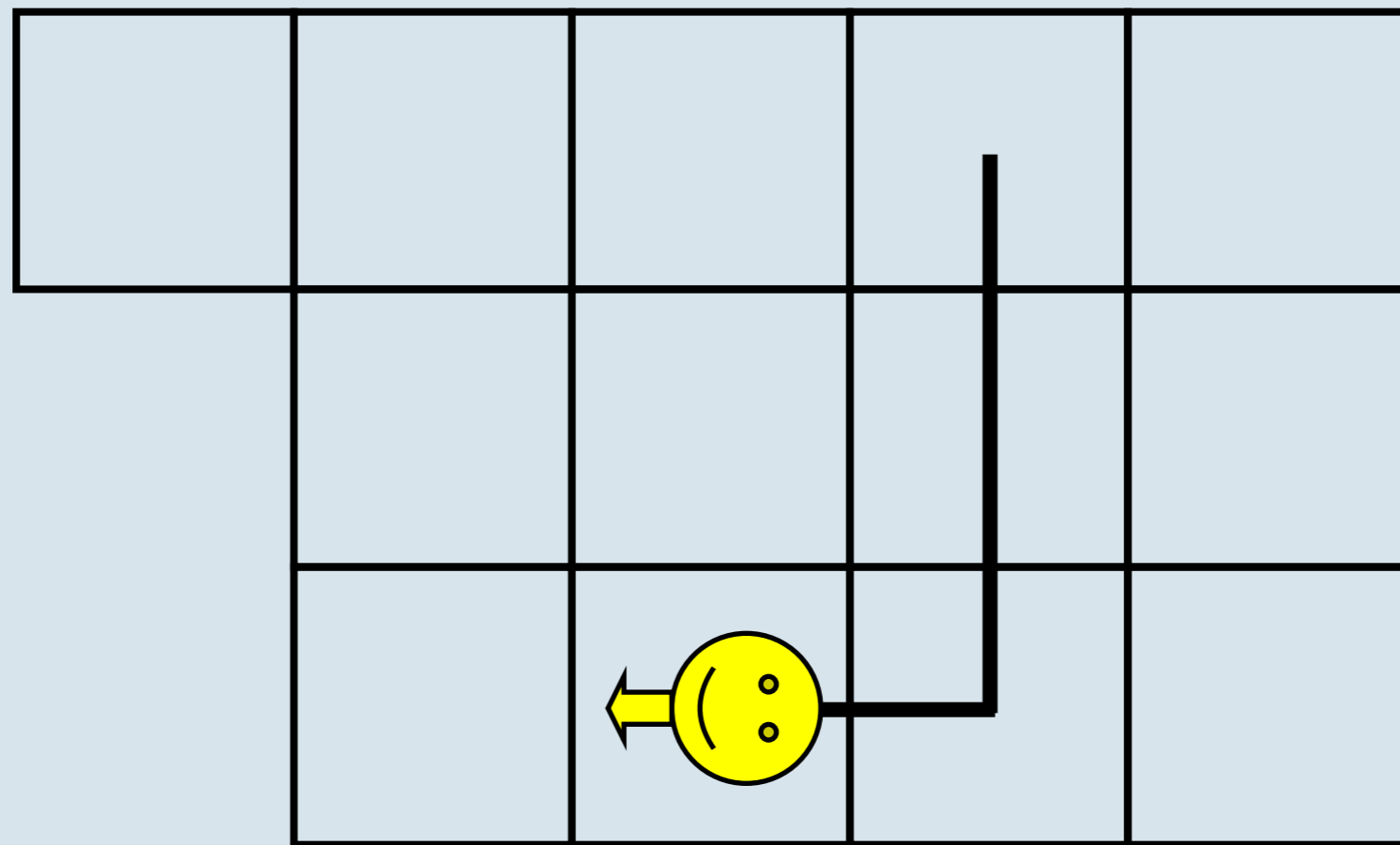
- FORWARD
- FORWARD
- – RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- HELLO



Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)

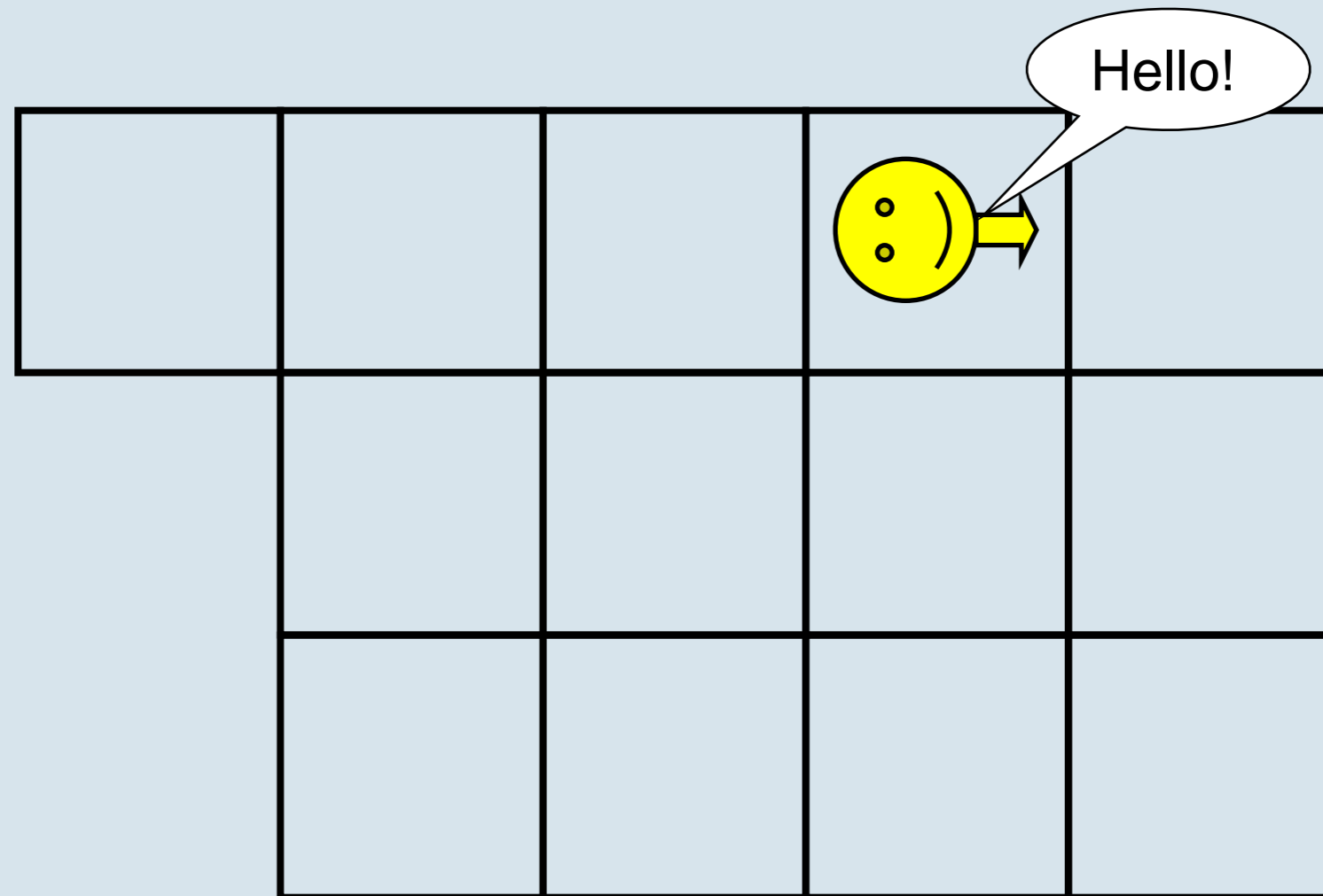
- FORWARD
- FORWARD
- RIGHT
- – FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- HELLO



Um Programa para o Robot

- Vamos considerar que, neste caso, um programa é uma lista de instruções, reconhecidas pelo Robot, que serão executadas por este em sequência (uma após a outra)

- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- RIGHT
- FORWARD
- FORWARD
- – HELLO



Exemplo: Operador Programável

Operações executáveis pelo Operador:

INPUT

- Apaga o quadro A.
- Ouve um número e escreve-o no quadro A.

COPY

- Apaga o quadro B.
- Copia o número no quadro A para o quadro B.

SUM

- Adiciona o número no quadro A com o número no quadro B.
- Substitui o número no quadro B pelo resultado obtido.

DOWN

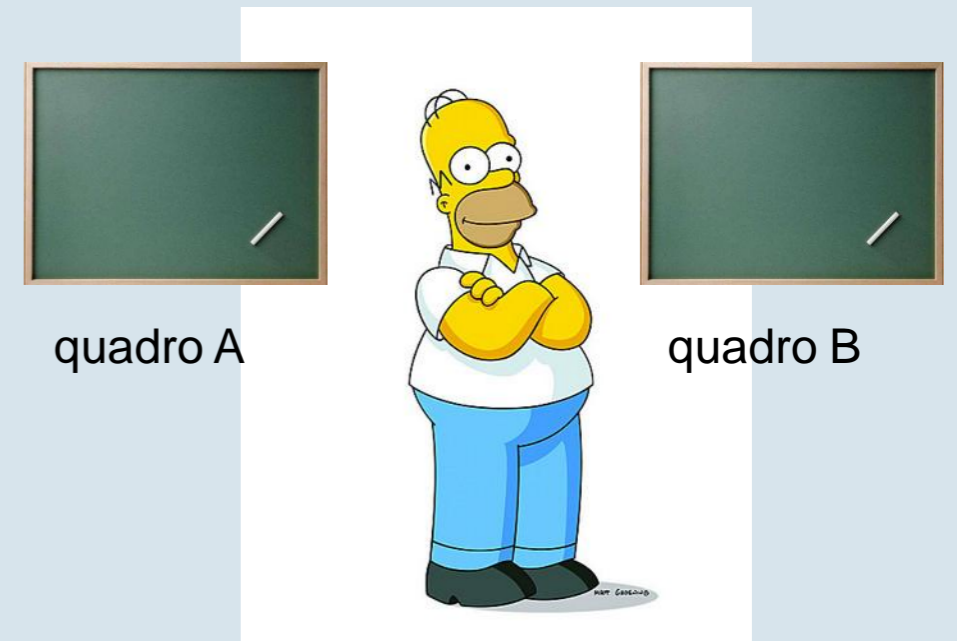
- Subtrai 1 ao número no quadro A.

LOOP x

- Se o número no quadro A é diferente de zero continua pela instrução na linha x.

STOP

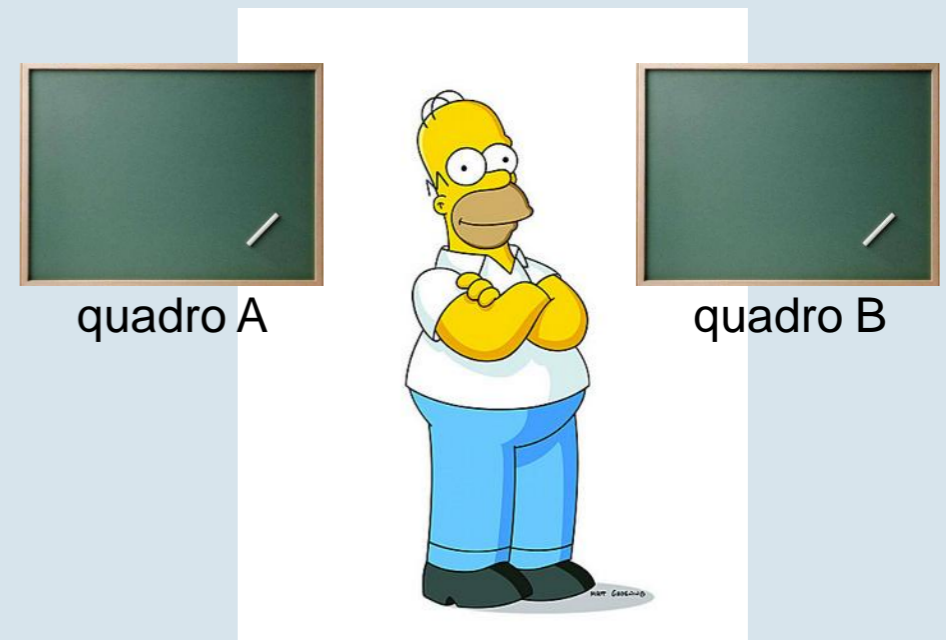
- Termina o programa



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

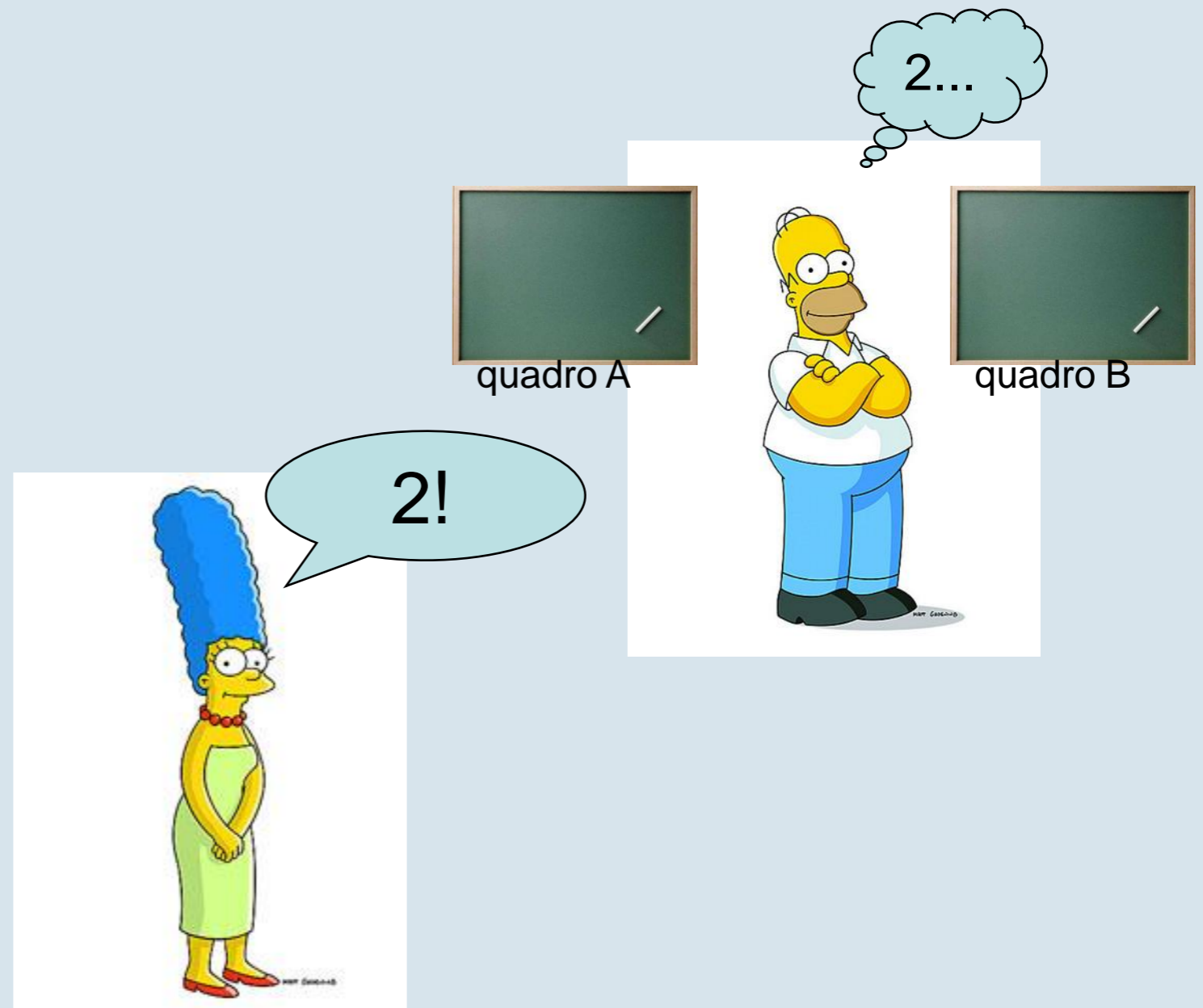
- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



Um programa para o Operador


- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

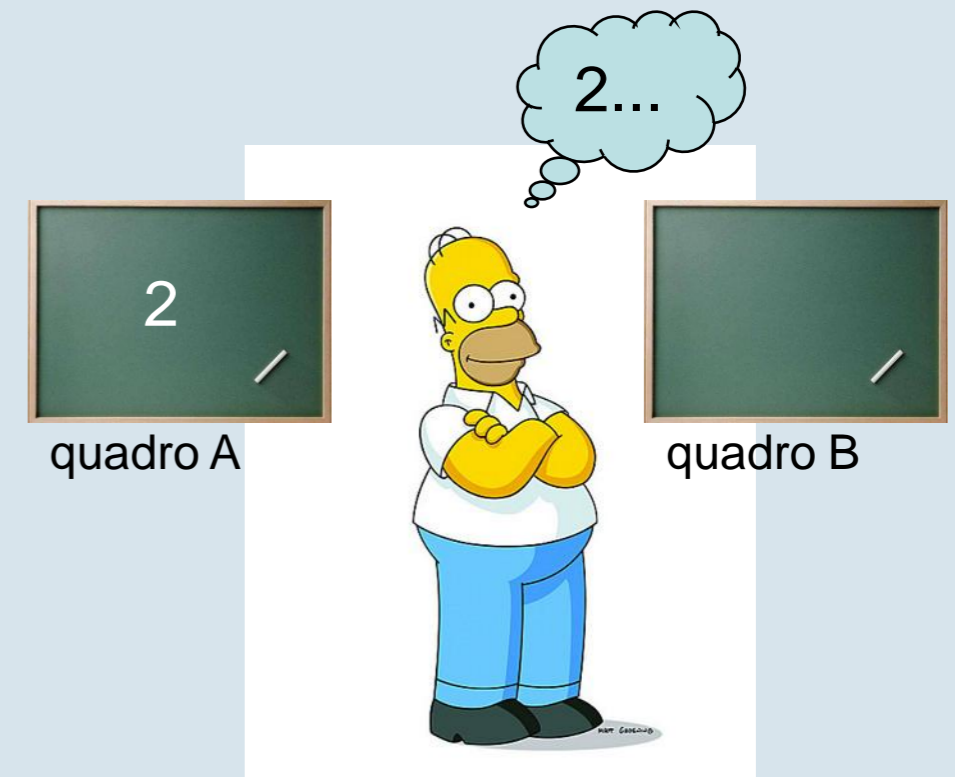
→ 1: INPUT
2: COPY
3: SUM
4: DOWN
5: LOOP 3
6: STOP



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

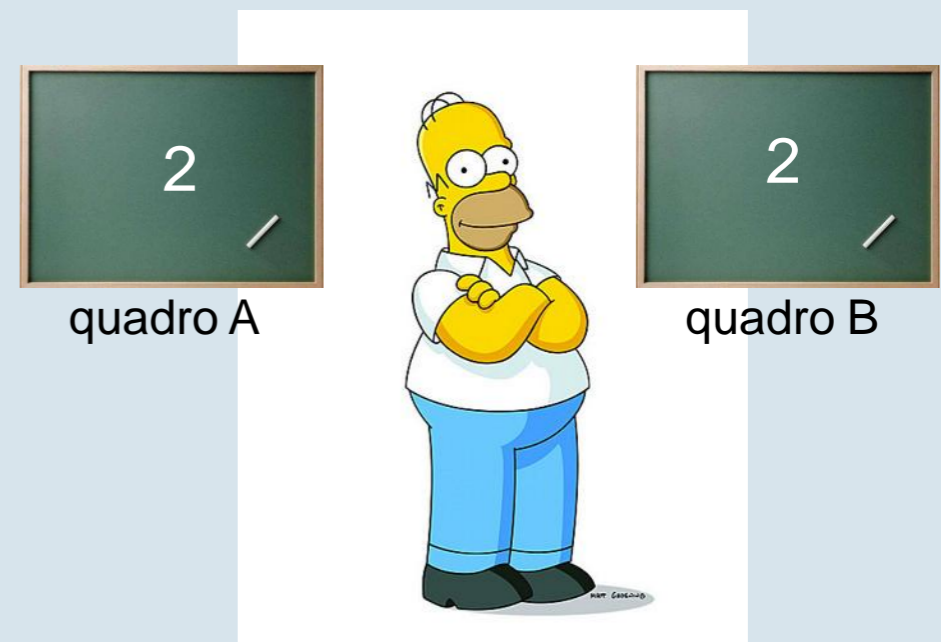
- 
- 1: INPUT
 - 2: COPY
 - 3: SUM
 - 4: DOWN
 - 5: LOOP 3
 - 6: STOP



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



quadro A



quadro B



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



quadro A



quadro B



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



quadro A



quadro B



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



quadro A



quadro B

Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

1: INPUT
2: COPY
3: SUM
4: DOWN
5: LOOP 3
6: STOP



quadro A



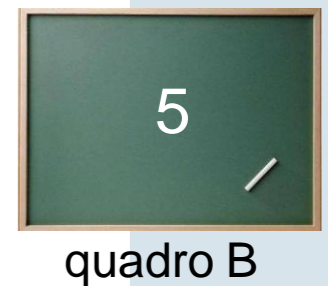
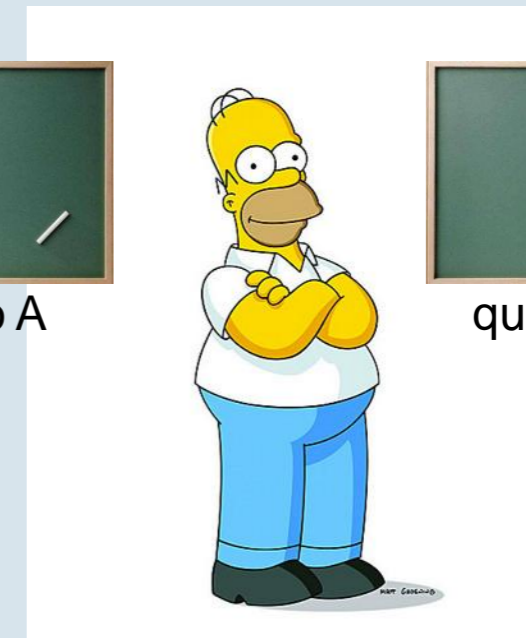
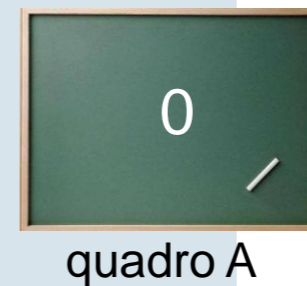
quadro B



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

- 1: INPUT
- 2: COPY
- 3: SUM
- 4: DOWN
- 5: LOOP 3
- 6: STOP



quadro A

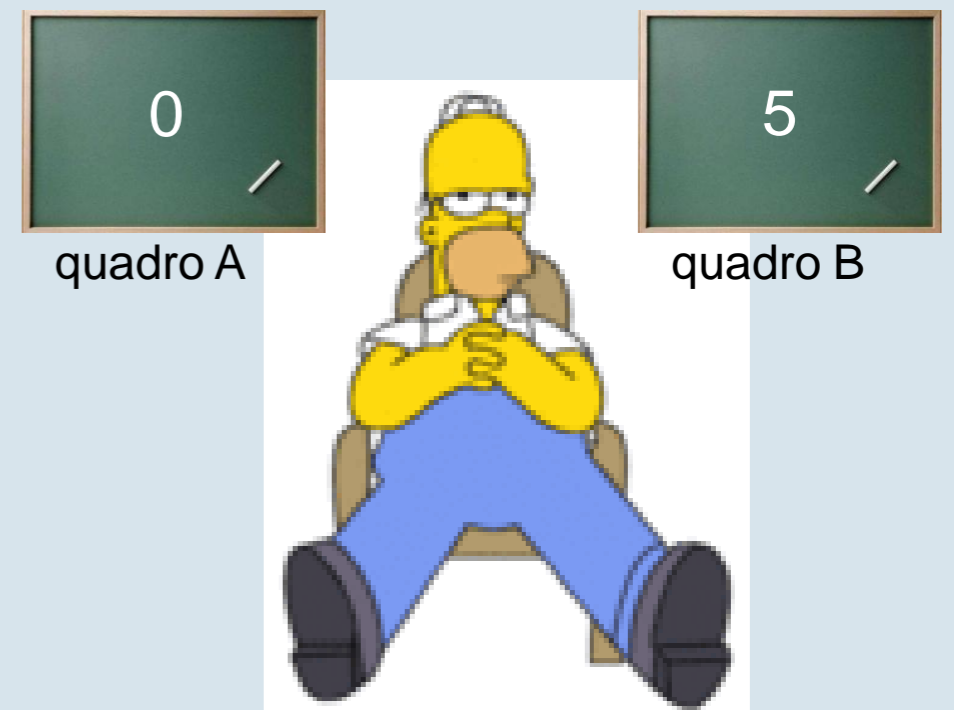


quadro B

Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

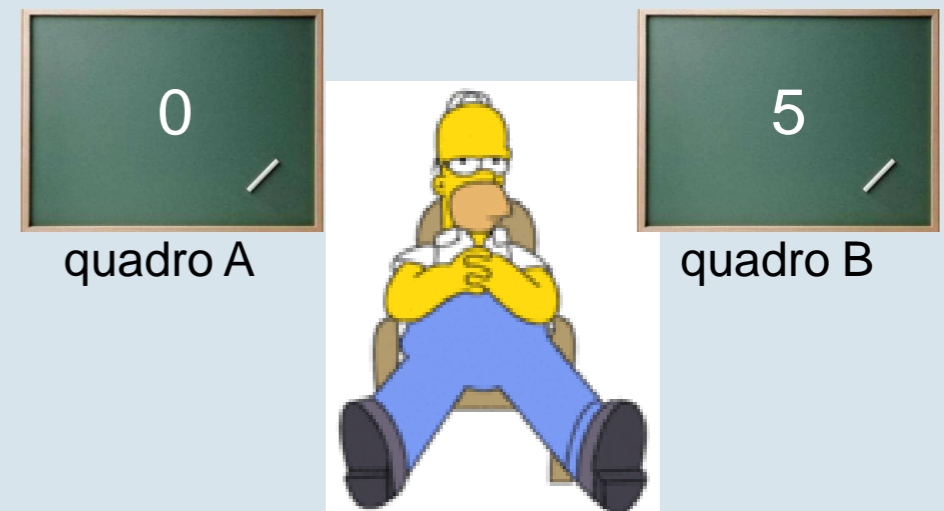
1: INPUT
2: COPY
3: SUM
4: DOWN
5: LOOP 3
→ 6: STOP



Um programa para o Operador

- Neste caso, um programa é uma lista de instruções, que serão executadas pelo Operador em sequência, **excepto** quando uma instrução LOOP for executada.

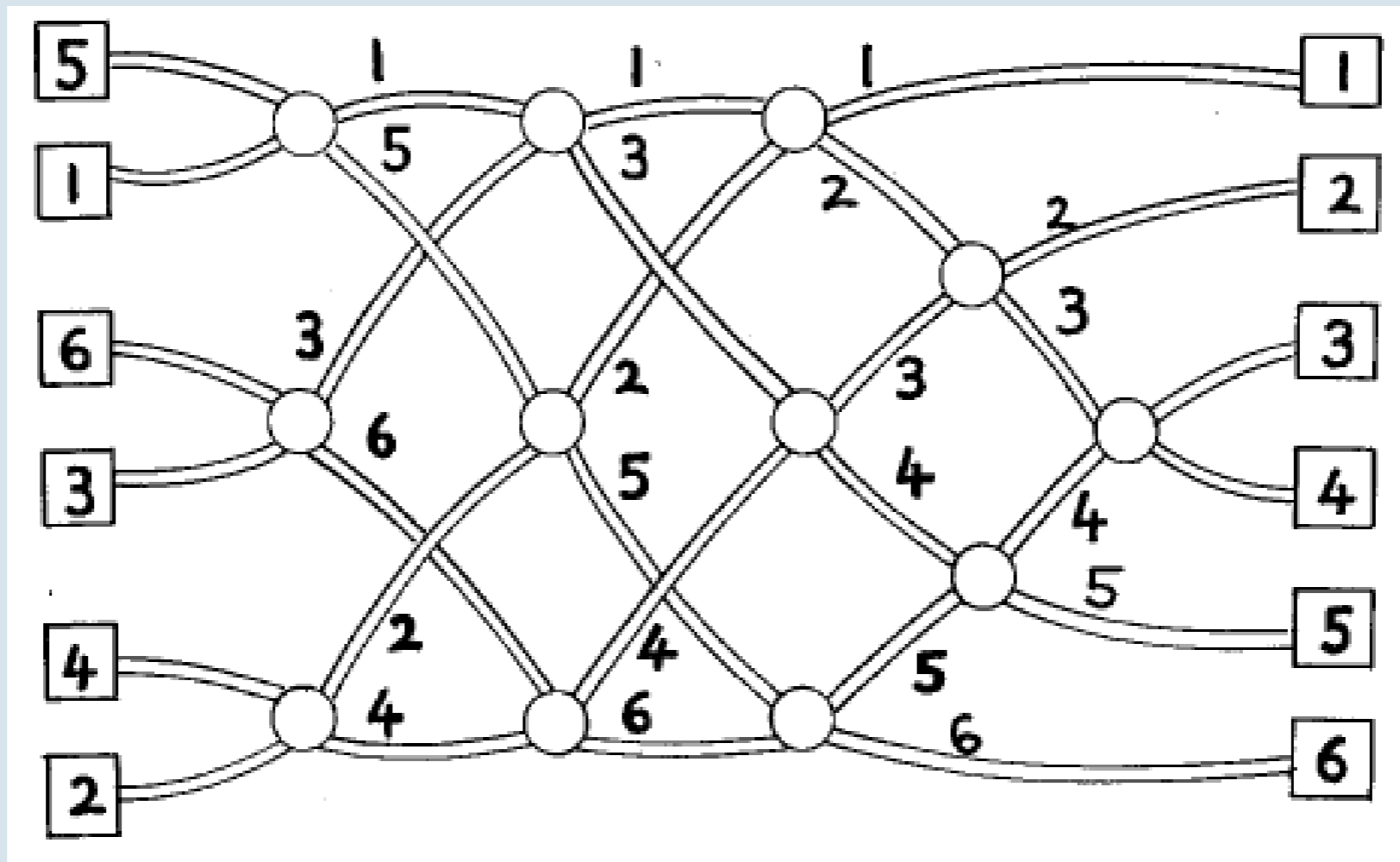
1: INPUT
2: COPY
3: SUM
4: DOWN
5: LOOP 3
6: STOP



- Algumas instruções deste programa são **acções** simples (DOWN), outras envolvem a tomada de **decisões** (LOOP).
- Estas últimas são chamadas operações de controlo, e existem em todas as linguagens de programação.
- O ponto importante a reter é que as decisões a tomar durante a execução estão definidas no programa de forma explícita.

Exemplo: Ordenador Humano

- Num computador vários programas podem executar ao mesmo tempo. Neste exemplo:
- Cada um executa cegamente as suas instruções
- Todos colaboram para um fim comum



Exemplo: Ordenador Humano

Programa executado **cegamente** por cada interveniente:

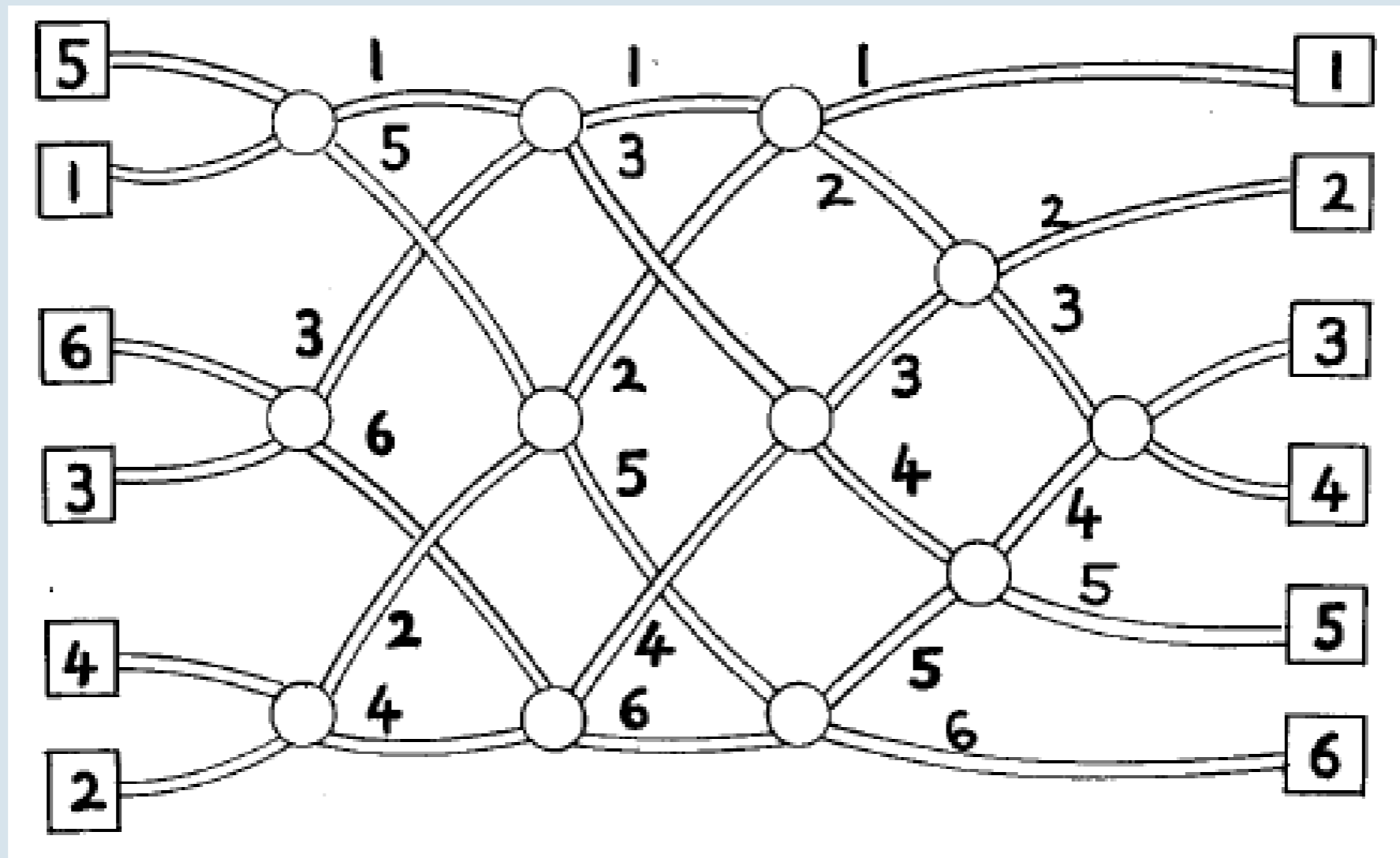
1. Avançar uma casa;
2. Se estiver no fim da rede, então terminar;
3. Esperar por outro colega;
4. Comparar o seu número com o do outro colega;
5. Se o seu número for maior que o do outro colega então dar um passo para a casa da direita, caso contrário dar um passo para a casa da esquerda
6. Continuar na instrução **2**

Exemplo: Ordenador Humano



Exemplo: Ordenador Humano

- Temos aqui 6 programas a correr em simultâneo
- Cada um executa cegamente as suas instruções
- Todos colaboram para um fim comum



Exemplo: Ordenador Humano

Programa executado **cegamente** por cada interveniente:

1. Avançar uma casa;
2. Se estiver no fim da rede, então terminar;
3. Esperar por outro colega;
4. Comparar o seu número com o do outro colega;
5. Se o seu número for maior que o do outro colega então dar um passo para a casa da direita, caso contrário dar um passo para a casa da esquerda
6. Continuar na instrução 2

Neste exemplo, as operações possíveis são **acções, observações, decisões.**

Exemplo: Ordenador Humano

Programa executado **cegamente** por cada “agente”

1. **Avançar uma casa;**
2. Se estiver no fim da rede, então **terminar;**
3. Esperar por outro colega;
4. Comparar o seu número com o do outro colega;
5. Se o seu número for maior que o do outro colega então **dar um passo para a casa da direita**, caso contrário **dar um passo para a casa da esquerda**
6. **Continuar na instrução 2**

Neste exemplo, as operações possíveis são **acções**, **observações**, **decisões**.

Exemplo: Ordenador Humano

Programa executado **cegamente** por cada “agente”

1. Avançar uma casa;
2. Se estiver no fim da rede, então terminar;
3. **Esperar por outro colega;**
4. Comparar o seu número com o do outro colega;
5. Se o seu **número for maior que o do outro** colega então dar um passo para a casa da direita, caso contrário dar um passo para a casa da esquerda
6. Continuar na instrução 2

Neste exemplo, as operações possíveis são **acções, observações, decisões.**

Exemplo: Ordenador Humano

Programa executado **cegamente** por cada “agente”

1. Avançar uma casa;
2. **Se** estiver no fim da rede, **então** terminar;
3. Esperar por outro colega;
4. Comparar o seu número com o do outro colega;
5. **Se** o seu número for maior que o do outro colega **então** dar um passo para a casa da direita, **caso contrário** dar um passo para a casa da esquerda
6. Continuar na instrução 2

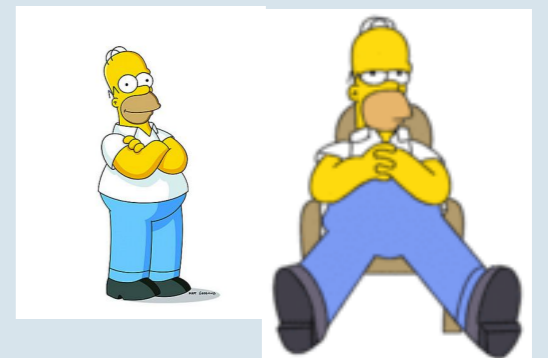
Neste exemplo, as operações possíveis são **acções**, **observações**, **decisões**.

Aspectos importantes a reter

- A programação, em sentido lato, é a **actividade central** na Engenharia Informática
- A programação é uma actividade **criativa**, por isso **difícil**
- Um programa descreve como se executa uma tarefa
 - ordenar uma lista de números
 - visualizar uma fotografia
 - extrair informação de uma base de dados
 - imprimir um documento
 - ...
- Um programa envolve em geral um encadeamento de
 - accões
 - observações
 - decisões

Três cenários semelhantes

- Programação do ROBOT
 - Máquina: **Robot**
 - Instruções específicas (acções): FORWARD, LEFT, ...
 - Programa: sequência de acções
- Programação do Operador
 - Máquina: **Operador**
 - Instruções (acções, decisões) : COPY, LOOP x, ...
 - Programa: sequência de acções, decisões
- Programação da Equipa
 - Máquina: **Vários Agentes** (multiprocessador)
 - Operações: observações, acções, decisões
 - Programa: Vários, em execução simultânea



Aspectos importantes a reter

- Um programa descreve um plano de acção, descrito de **forma precisa**, que pode ser executado por um computador sem requerer criatividade nenhuma da parte deste
- Todas as funcionalidades desempenhadas por computadores são definidas por programas
- Um computador moderno executa várias aplicações e programas em simultâneo
- Cada programa é escrito numa certa linguagem de programação.
- Actualmente, são usadas muitas linguagens
 - C
 - Assembler
 - Java (que vamos usar em “Introdução à Programação”)
 - ...

Aspectos importantes a reter

- Os computadores são irresponsáveis !
- Tudo o que um programa faz, incluindo os eventuais erros que possa cometer, é da exclusiva responsabilidade dos programadores que o escreveram
- Em muitas situações, a correcção do software é muito crítica (sistema financeiros, centrais nucleares)
- Assim, a programação é uma actividade:
 - muito crítica
 - muito dispendiosa
 - requer muita concentração e disciplina
 - ... mas é também muito divertida

Construção de um Programa

Texto do Programa
(source code)

```
str = new Scanner(System.in);  
N = str.nextInt();  
str.nextLine();  
while(N!=-1) {  
    q = new int[8];  
    p = new int[N][8];  
    ...
```

criado com um editor de texto

Compilador

o compilador é uma aplicação que traduz o texto do nosso programa em código máquina

computador executa o código máquina

instruções máquina
(object code)

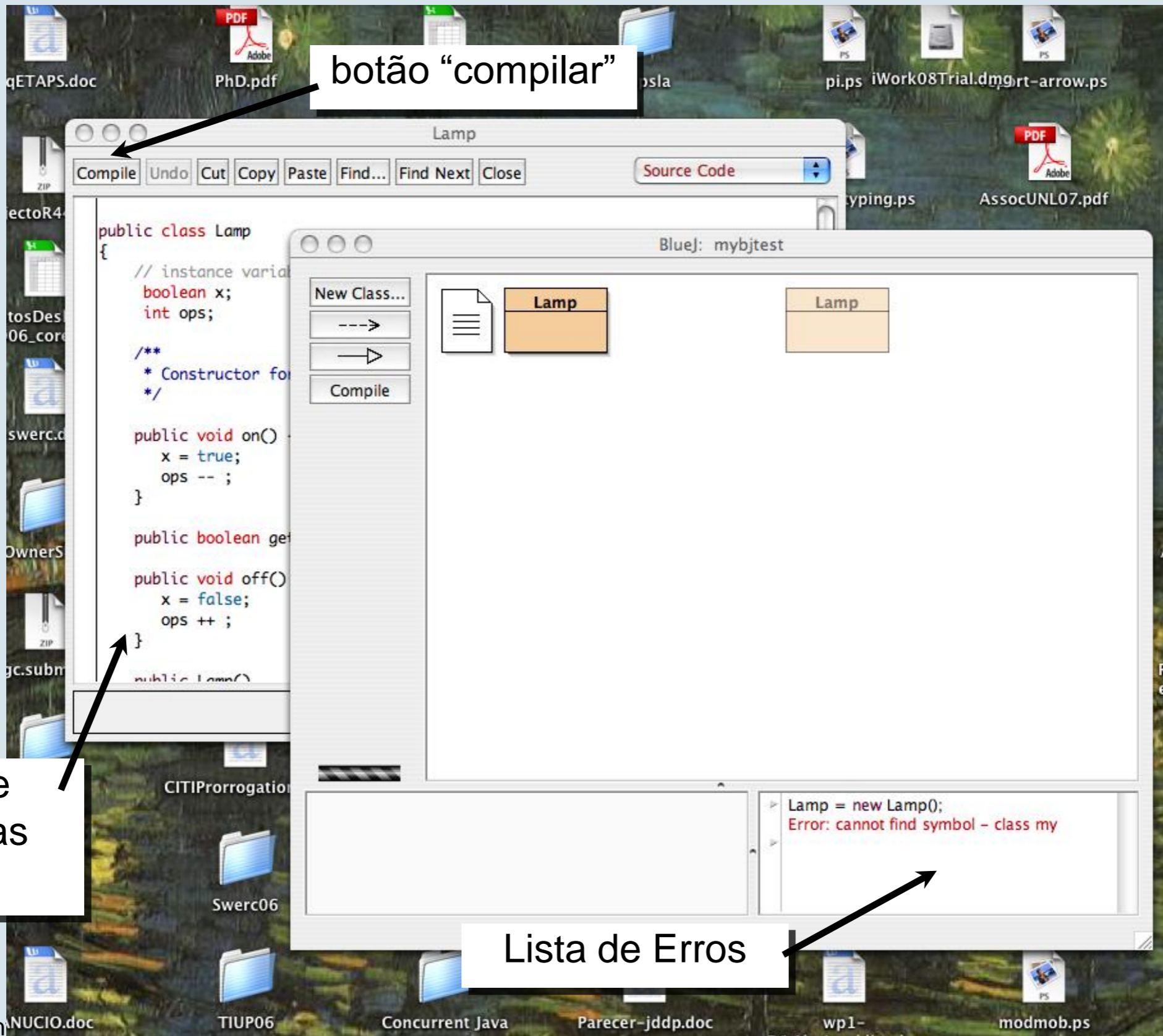
```
100100100100111010010011101  
001001110100100111010010011  
101001001110100100111010010  
011101001001110100100111010  
010011101001001110100100111  
010010011101001001110100100  
111010010011101001001110100  
100111010010011101001001110  
100100111010010011101001001  
111001001010010001000010000  
10
```



IDE (Integrated Development Environment)

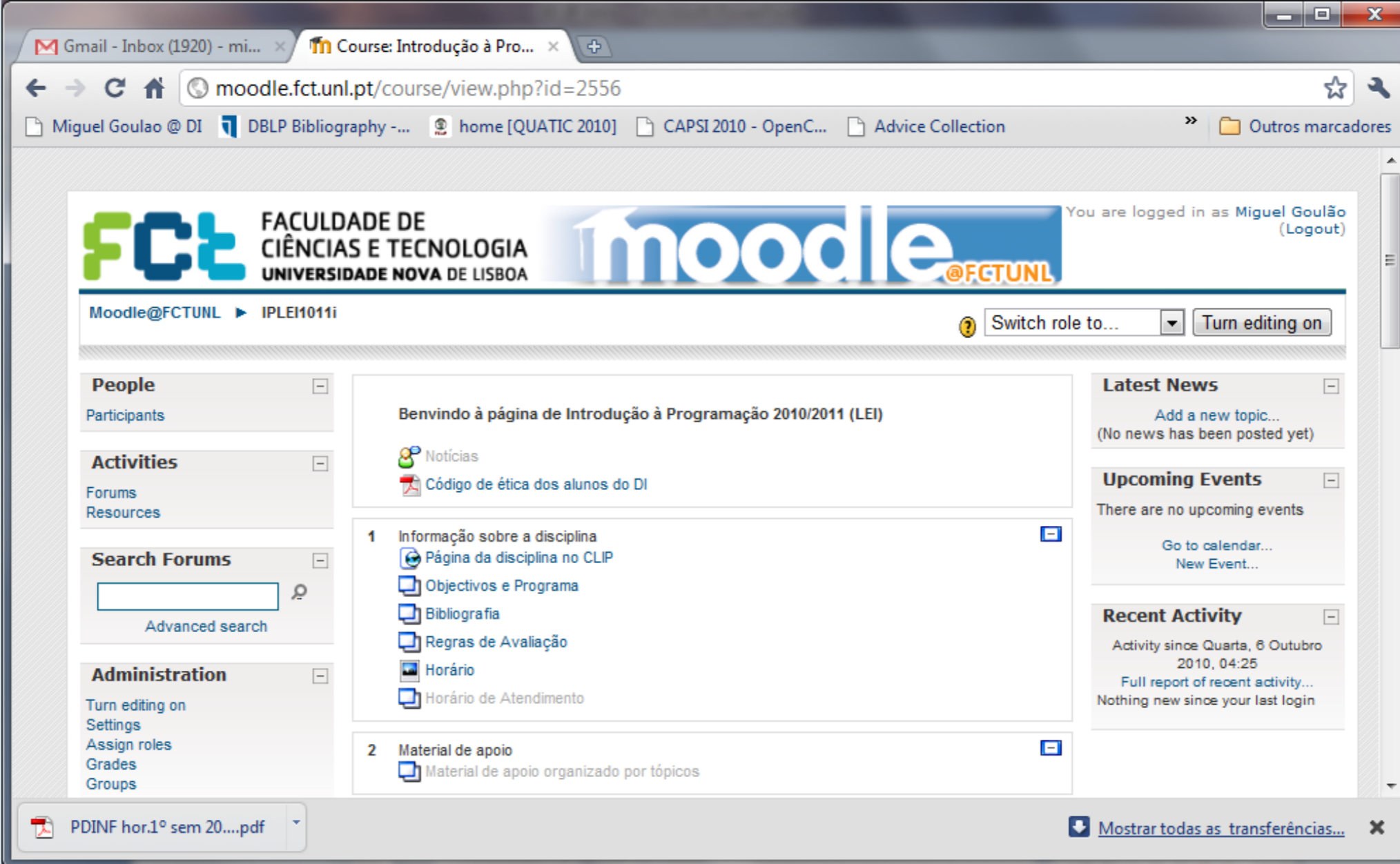
- IDE: Integrated Development Environment
- IDE: Ambiente de (suporte à) Programação
- Sistema de software, com uma interface gráfica sofisticada, que integra os vários componentes necessários à produção de programas:
 - editor de texto (para escrever os programas fonte)
 - compilador (para transformar os programas em código executável)
 - depurador (para ajudar a corrigir erros)
 - etc ...
- Em IP, vamos para já usar o IDE BlueJ

O IDE BlueJ



Objectivos, Programa, Bibliografia...

- Poderá encontrar tudo sobre a disciplina de Introdução à Programação no **moodle** (password de inscrição: **IP1011**)



The screenshot shows a web browser window displaying a Moodle course page. The browser's address bar shows the URL moodle.fct.unl.pt/course/view.php?id=2556. The page header includes the FCT logo, 'FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA', and the Moodle logo. The user is logged in as Miguel Goulão. The course title is 'Moodle@FCTUNL ► IPLEI1011i'. The main content area is titled 'Bemvindo à página de Introdução à Programação 2010/2011 (LEI)' and contains a list of links: 'Notícias', 'Código de ética dos alunos do DI', '1 Informação sobre a disciplina' (with sub-links for 'Página da disciplina no CLIP', 'Objectivos e Programa', 'Bibliografia', 'Regras de Avaliação', 'Horário', and 'Horário de Atendimento'), and '2 Material de apoio' (with a link for 'Material de apoio organizado por tópicos'). The right sidebar features 'Latest News', 'Upcoming Events', and 'Recent Activity' sections. The left sidebar contains 'People', 'Activities', 'Search Forums', and 'Administration' sections. A PDF file 'PDINF hor.1º sem 20....pdf' is visible in the bottom left corner.

<http://moodle.fct.unl.pt/course/view.php?id=2556>