

Fundamentos de Programação

Diagrama de blocos

Prof. M.Sc.: João Paulo Q. dos Santos

E-mail: joao.queiroz@ifrn.edu.br

Página: <http://docente.ifrn.edu.br/joaoqueiroz/>

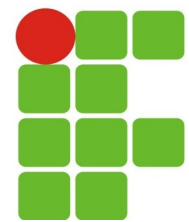
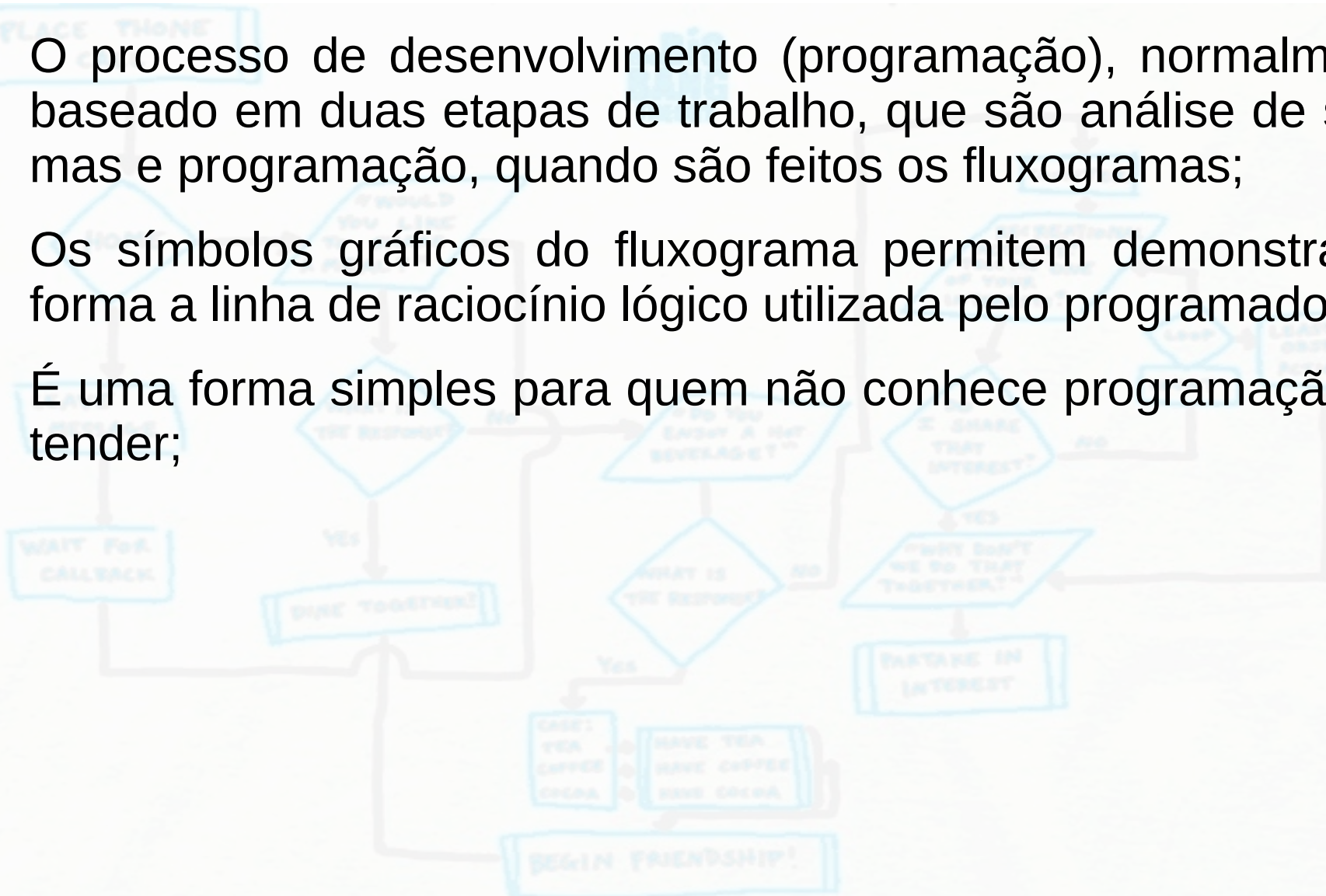


Diagrama de bloco

- O processo de desenvolvimento (programação), normalmente, baseado em duas etapas de trabalho, que são análise de sistemas e programação, quando são feitos os fluxogramas;
- Os símbolos gráficos do fluxograma permitem demonstrar de forma a linha de raciocínio lógico utilizada pelo programador;
- É uma forma simples para quem não conhece programação entender;



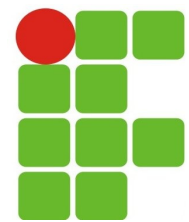
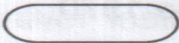

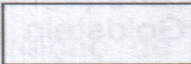

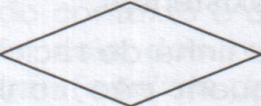






Diagrama de bloco

Símbolo	Significado	Descrição
	Terminal <i>Terminator</i>	O símbolo representa a definição de início e fim do fluxo lógico de um programa. Também é utilizado na definição de sub-rotinas de procedimento ou de função.
	Entrada manual <i>Manual input</i>	Representa a entrada manual de dados, normalmente efetuada em um teclado conectado diretamente ao console do computador.
	Processamento <i>Process</i>	Representa a execução de uma operação ou grupo de operações que estabelecem o resultado de uma operação lógica ou matemática.
	Exibição <i>Display</i>	Representa a execução da operação de saída visual de dados em um monitor de vídeo conectado ao console do computador.
	Decisão <i>Decision</i>	O símbolo representa o uso de desvios condicionais para outros pontos do programa de acordo com situações variáveis.
	Preparação <i>Preparation</i>	Representa a modificação de instruções ou grupo de instruções existentes em relação à ação de sua atividade subsequencial.
	Processo predefinido <i>Predefined process</i>	Definição de um grupo de operações estabelecidas como uma sub-rotina de processamento anexa ao diagrama de blocos.
	Conector <i>Connector</i>	Representa a entrada ou a saída em outra parte do diagrama de blocos. Pode ser usado na definição de quebras de linha e na continuação da execução de decisões.
	Linha <i>Line</i>	O símbolo representa a ação de vínculo existente entre os vários símbolos de um diagrama de blocos. Possui a ponta de uma seta indicando a direção do fluxo de ação.

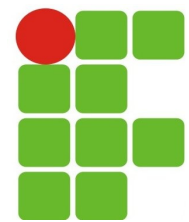


Diagrama de bloco

- Após a exibição dos símbolos que podem ser usados e de suas descrições é pertinente estabelecer algumas regras para sua devida utilização:
 - Os símbolos de identificação gráfica representam uma operação ou conjunto de operações similares, podendo ser identificados por um rótulo (escrever qual ação) relacionado à própria ação do símbolo em uso, somente quando necessário;
 - Os símbolos devem ser conectados uns aos outros por linhas e setas que mostrem explicitamente a direção do fluxo a ser executado pelo programa;
 - A estrutura visual do diagrama deve, a princípio, estar orientada no sentido de cima para baixo, da direita para a esquerda e ser desenhada no centro.

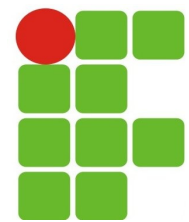


Diagrama de bloco

- A definição e de inicialização e finalização de um diagrama ocorre com o uso do símbolo “terminal” devidamente identificado com um dos rótulos: inicio, fim, retorno ou definição de um nome particular quando for necessário;
- As operações de entrada de dados serão genericamente definidas como “entrada manual” e o de saída como “exibição”;
- A definição de variáveis nas operações de entrada e saída será feita nos símbolos apropriados, quando houver mais de uma variável, serão separadas por vírgulas;
- As operações de processamento matemático e lógico estarão definidas com o símbolo “processamento”. Quando houver mais de uma operação a ser definida em um mesmo símbolo, devem ser separadas por linhas de ação sem o uso de virgulas ou serem escritas em símbolos distintos.

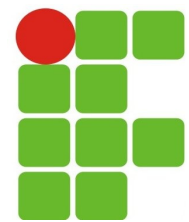


Diagrama de bloco

- As operações de tomada de decisão para condições simples, compostas, sequenciais, encadeadas ou execução de laços interativos (condicionais) serão representadas pelo símbolo de “decisão”, que conterà a condição a ser avaliada logicamente;
- Cada símbolo de decisão pode possuir apenas uma condição lógica;
- É considerada lógica uma condição isolada ou de um conjunto de condições vinculadas com o uso de um operador lógico de conjunção, disjunção ou disjunção exclusiva;
- As operações de laços iterativos e não iterativos (incondicionais) serão representados com o símbolo “preparação” que permite a realização de um grupo de tarefas predefinidas e relacionadas;
- A definição e uso de sub-rotinas são apresentados pelo “processo predefinido”. Esse símbolo deve estar identificado como um rótulo associado a outro diagrama ou diagrama interdependente ao programa e indicado como rótulo do símbolo “terminal” da rotina associada.

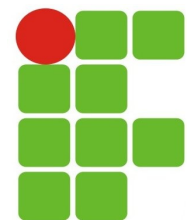


Diagrama de bloco

- As operações que necessitam de conexão utilizarão o símbolo “conector” na finalização de operações de decisões ou na identificação de vínculos entre partes de um programa e, neste caso, devem estar identificados com rótulos alfanuméricos;
- Fica eleito o símbolo “processamento” como curinga, que pode representar qualquer ação definida ou não, desde que a operação seja devidamente identificada por um rótulo descritivo;
- Mas existe a exceção aos símbolos “decisão” e “preparação” que representam operações bem definidas não devem, em hipótese alguma, ser substituídos por qualquer outro símbolo;
- O símbolo “terminal” só pode ser usado duas vezes de cada programa ou programa de sub-rotina. Em nenhuma hipótese pode ser utilizado mais de duas vezes.

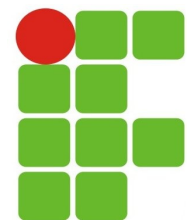


Diagrama de bloco

- A representação gráfica baseada em diagrama de bloco é também referenciada erroneamente como fluxograma;
- Possivelmente o erro de definição do termo ocorre devido a estrutura da palavra original do inglês: flowchart (flow = fluxo, chat = diagrama), portanto diagrama de fluxo;
- Portanto diagrama de fluxo não é o mesmo que fluxograma;
- Mostraremos agora as estruturas lógicas de operação computacional comuns e que serão apresentadas ao longo da disciplina: sequência, decisão simples, decisão composta, laço de repetição condicional pós-teste e pré-teste e laço de repetição incondicional;

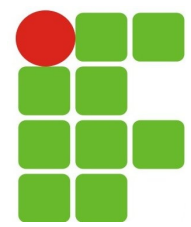
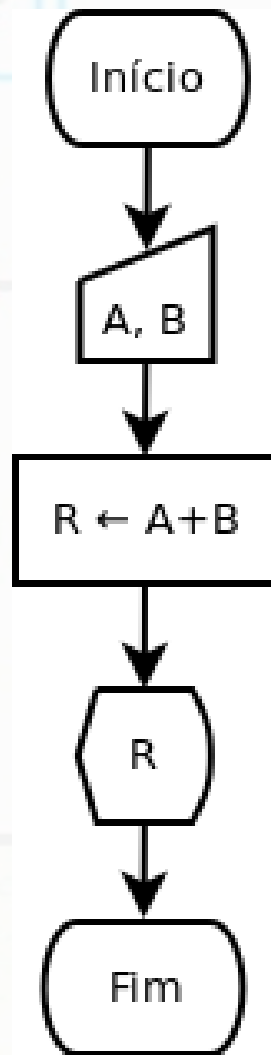
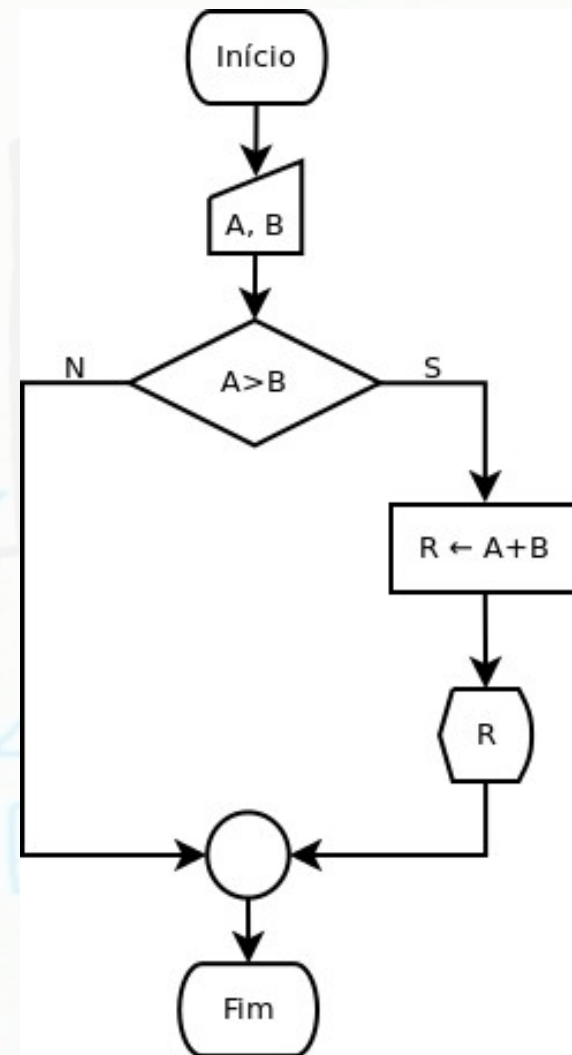


Diagrama de bloco



Sequência



Decisão simples

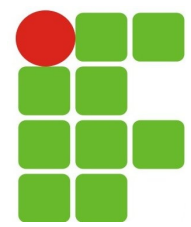
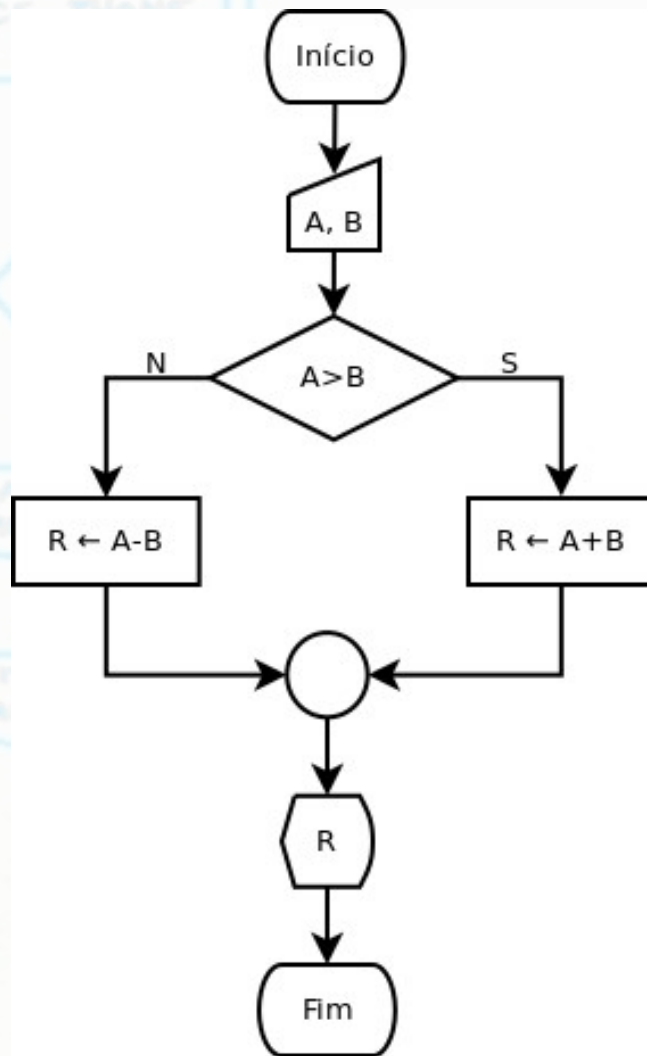
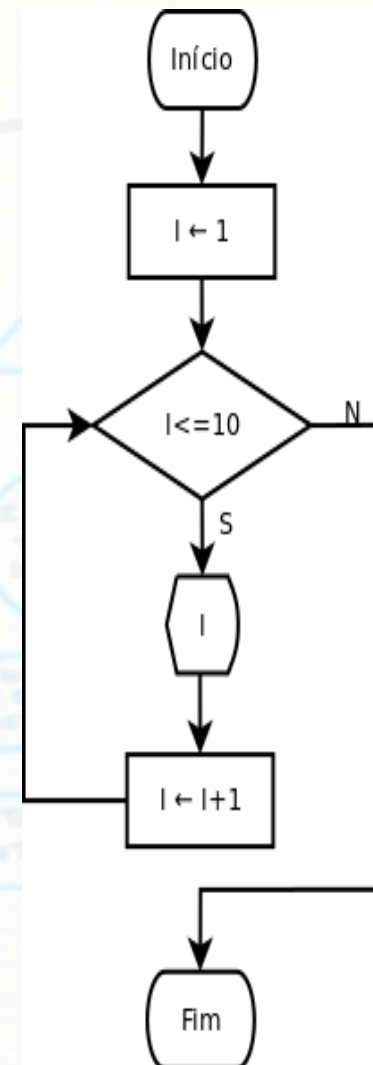


Diagrama de bloco



Decisão composta



Laco de repetição condicional pre-teste

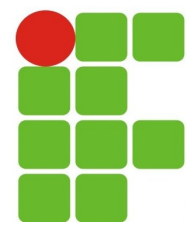
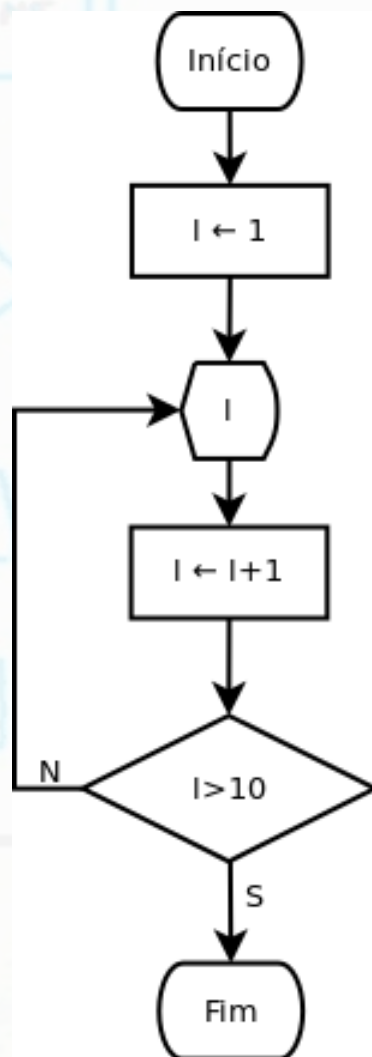
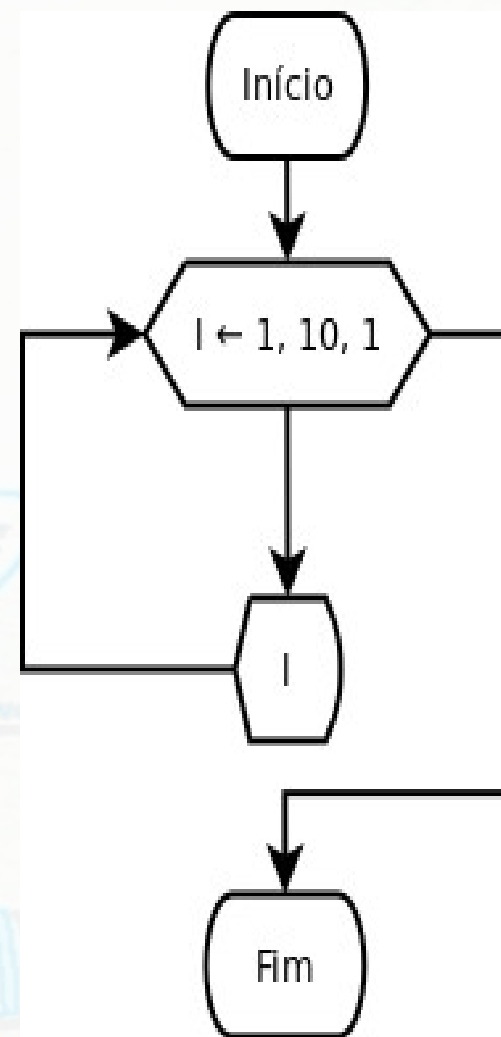


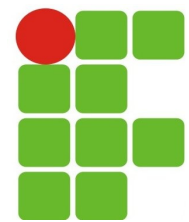
Diagrama de bloco



Laco de repetição condicional pós-teste

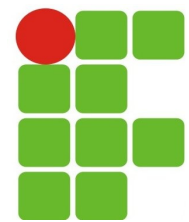


Laco de repetição incondicional



Compiladores, Interpretadores e Tradutores

- Quando o projeto de um programa fica pronto, é necessário transformá-lo em software;
- Para tanto, faz-se necessário a conversão do projeto para uma linguagem de programação;
- A ferramenta que possa editar o texto do programa, pode ser um editor de texto ex.: bloco de notas, gedit, VI;
- Ou uma IDE (*Integrated Development Environment*) Ambiente de Desenvolvimento Integrado ex.: Netbeans, Eclipse, CodeBlocks;
- As ferramentas de tradução são programas que permitem fazer a tradução de um programa escrito em linguagem formal para outra:
 - Ex.: escrevo um programa em uma linguagem C++ e quero reescrever na linguagem Java, **c2j** é um exemplo de tradutor de programas entre linguagens;



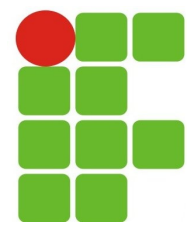
Compiladores, Interpretadores e Tradutores

- Ferramentas de interpretação são programas que executam um código-fonte escrito em uma linguagem na própria memória principal do computador;
- Sem que seja executado diretamente no processador central da máquina;
- Esse tipo de ferramenta permite rápida execução dos programas, no entanto não assegura acesso ao código fonte:
 - Ex.: BASIC, PERL, PYTHON, JavaScript, LOGO.



Compiladores, Interpretadores e Tradutores

- As ferramentas de compilação são programas que traduzem para uma linguagem de baixo nível (linguagem de máquina) um código fonte escrito em linguagem de alto nível;
- Ao fazer essa tradução, o código-fonte se torna um programa-objeto (escrito em linguagem de alto nível compatível com o processador em uso);
- Depois faz a ligação do programa-objeto com as rotinas de execução de programas do sistema operacional, tornando um código executável;



the Big RANG THEORY

Perguntas?

