

# Logica de Programacao



Jorge Augusto F. de Souza Edward Boszczowski

# Lógica de Programação

## Roteiro:

O que é lógica de programação

A importância do papel e caneta

O que é algoritmo

Pseudocódigo

Diagrama de Blocos

Diagrama de Chapin



# Lógica de Programação

## Roteiro:

UML

Lógica nas diversas áreas relacionadas

- Geometria computacional
- Animação por computador
- Teoria dos Grafos



# Lógica de Programação

A lógica de Programação

## O que é lógica de Programação?

Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo. (Paulo Sergio de Moraes)



# Lógica de Programação

## A importância do papel e caneta

- Importância do papel e caneta para o programador novato
- Teste de mesa e código fonte aberto



# Lógica de Programação

## A lógica de Programação

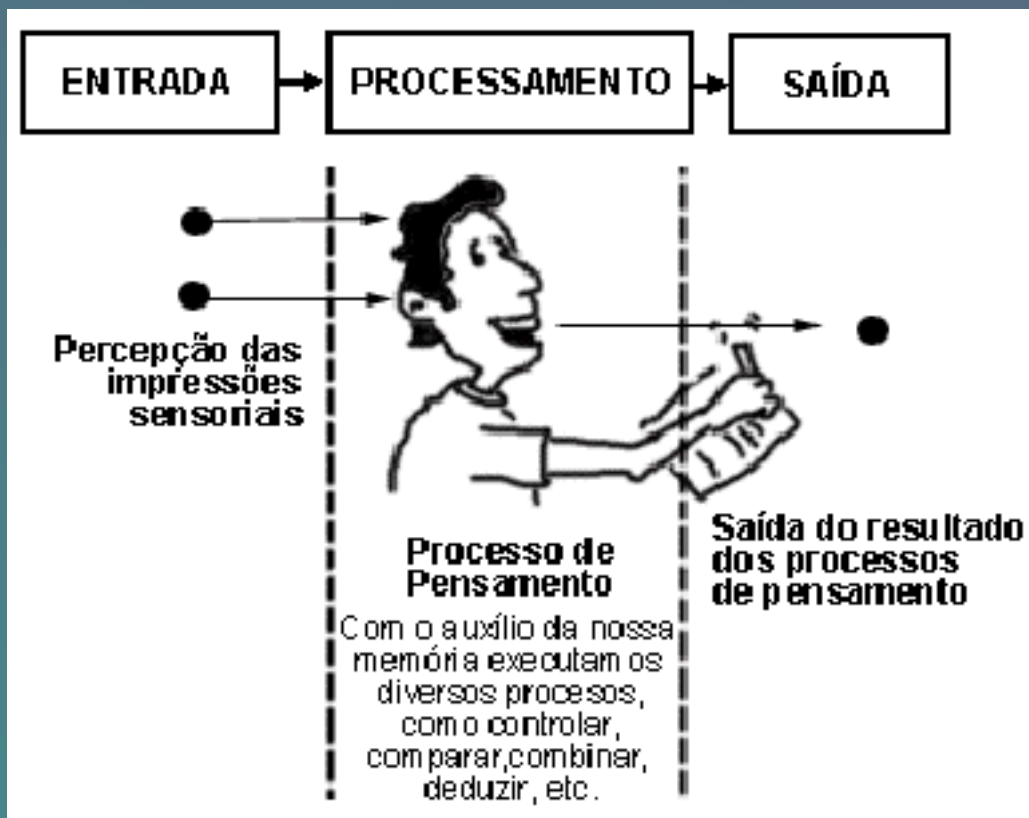
### O que é Algoritmo?

Um algoritmo é formalmente uma seqüência finita de passos que levam a execução de uma tarefa. Podemos pensar em algoritmo como uma receita, uma seqüência de instruções que dão cabo de uma meta específica.

(Paulo Sergio de Moraes)

# Lógica de Programação

Como é composto um programa



# Lógica de Programação

## A lógica de Programação - Pseudocódigo

### O pseudocódigo

Os algoritmos são descritos em uma linguagem chamada pseudocódigo. Este nome é uma alusão à posterior implementação em uma linguagem de programação. Os algoritmos são independentes das linguagens de programação. Ao contrário de uma linguagem de programação não existe um formalismo rígido de como deve ser escrito o algoritmo.





# Lógica de Programação

## A lógica de Programação - Pseudocódigo

### O pseudocódigo

O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.



# *Exemplo bubblesort*

=> algoritmo do bubblesort em portugol

=> Supoe-se que os indices dos arrays vao de 1 a n

void bubblesort( array de inteiro: a, inteiro: n )

i, j, temp, trocas : inteiro;

para i de n ate 2 faca;

trocas = 0;

para j de 1 ate j-1 faca;

se  $a[j] > a[j + 1]$  entao

=> troca i com j

temp =  $a[i]$ ;

$a[i] = a[j]$ ;

$a[j] = temp$ ;

trocas = trocas + 1;

fim\_para

se trocas = 0 entao pare; => array ordenado

fim\_para

fim\_bubblesort

# Exemplo bubblesort - C++

```
#include "Bubblesort.h"
```

```
void bubblesort(int *a, int n)
```

```
{
```

```
    int j;
```

```
    for(indice j= n-1; j>0; j--)
```

```
        for(i=0; i<j; i++)
```

```
            if(a[i+1] < a[i])
```

```
                troca(a[i+1], a[i]);
```

```
}
```

# Exemplo bubblesort - Java

```
public static void bubbleSort(int[] a) {  
    for (int i = 0; i < a.length-1; i++) {  
        for (int j = 0; j < a.length-i-1; j++) {  
            if (a[j] > a[j+1]) {  
                swap(a, j, j+1);  
            }  
        }  
    }  
}
```

```
private static void swap(int[] a, int i, int j) {  
    int temp = a[i];  
    a[i] = a[j];  
    a[j] = temp;  
}
```

# Exemplo bubblesort - ML

```
fun fix ( f,x) =  
  let val fx = f(x)  
  in  
    if x = fx then x  
    else fix(f,fx)  
end;  
  
fun bubble ([]) = []  
  | bubble([a]) = [a]  
  | bubble(a::b::x) =  
    if a <= b then a::bubble(b::x)  
    else b::bubble(a::x);  
  
fun bubblesort( lista ) = fix (bubble,lista);
```

# Lógica de Programação

## A lógica de Programação – Diagrama de Bloco



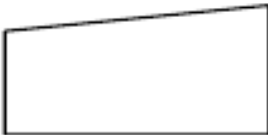

### Diagrama de Bloco

O diagrama de blocos é uma forma padronizada e eficaz para representar os passos lógicos de um determinado processamento.

Com o diagrama podemos definir uma seqüência de símbolos, com significado bem definido, portanto, sua principal função é a de facilitar a visualização dos passos de um processamento.



# Lógica de Programação

Símbolo	Função
 TERMINAL	Indica o INÍCIO ou FIM de um processamento  Exemplo: Início do algoritmo
 PROCESSAMENTO	Processamento em geral  Exemplo: Calculo de dois números
 ENTRADA DE DADO MANUAL	Indica entrada de dados através do Teclado  Exemplo: Digite a nota da prova 1
 EXIBIR	Mostra informações ou resultados  Exemplo: Mostre o resultado do calculo

# Lógica de Programação

A lógica de Programação – Diagrama de Chapin

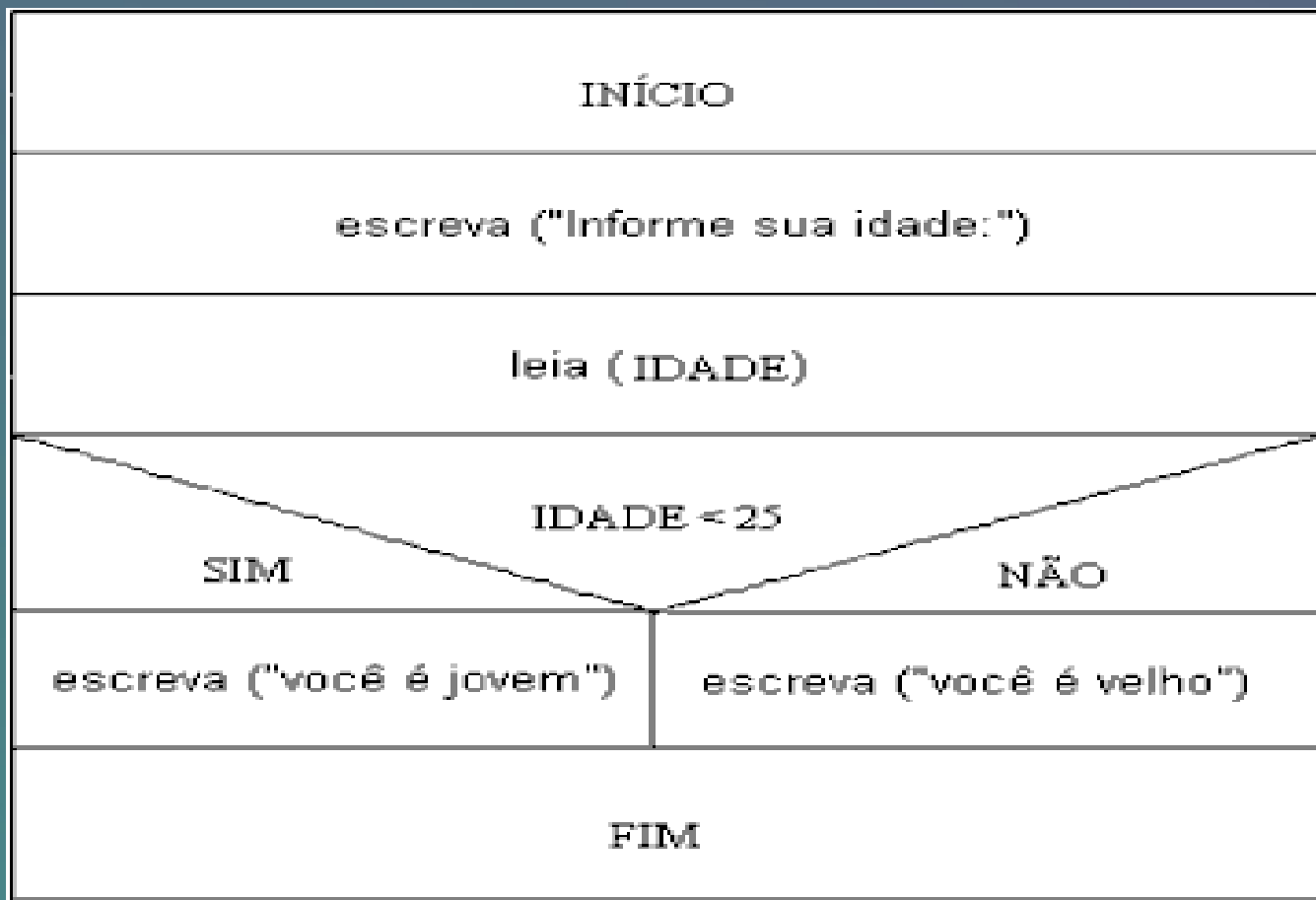
## Diagrama de Chapin

O diagrama foi desenvolvido por Nassi e Shneiderman e ampliado por Ned Chapin. A grande vantagem desse tipo de diagrama é a representação das estruturas que têm um ponto de entrada e um ponto de saída e são compostas pelas estruturas básicas de controle de sequência, seleção e repetição.





# Lógica de Programação



# Lógica de Programação

## A lógica de Programação – UML

### UML

A UML é uma linguagem de modelagem de propósito geral no campo da engenharia de software. Ela se originou devido a necessidade de se usar uma metodologia em um projeto orientado a objetos e é composta de três tipos de diagramas

\* Diagrama de comportamento. Um tipo de diagrama que ilustra características comportamentais de um sistema ou processo de negócio. Isto inclui atividade, máquina de estado, e diagramas de caso de uso bem como os quatro diagramas de interação.



# Lógica de Programação

## A lógica de Programação – UML

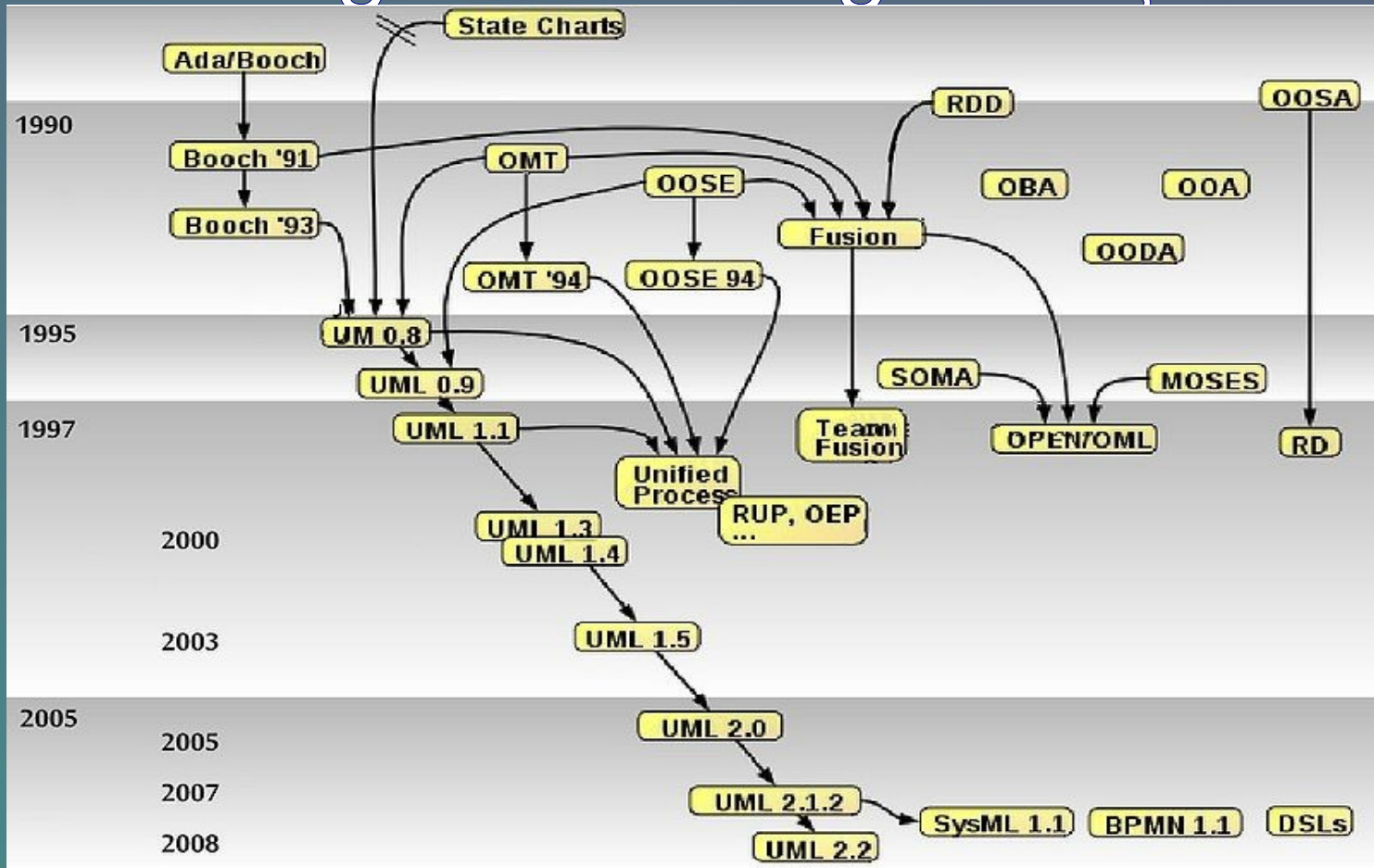
### UML

\* Diagramas de interação. Um subconjunto de diagramas de comportamento que enfatiza interações entre os objetos. Isto inclui comunicação, panorama de interação, sequência e diagramas de temporização.

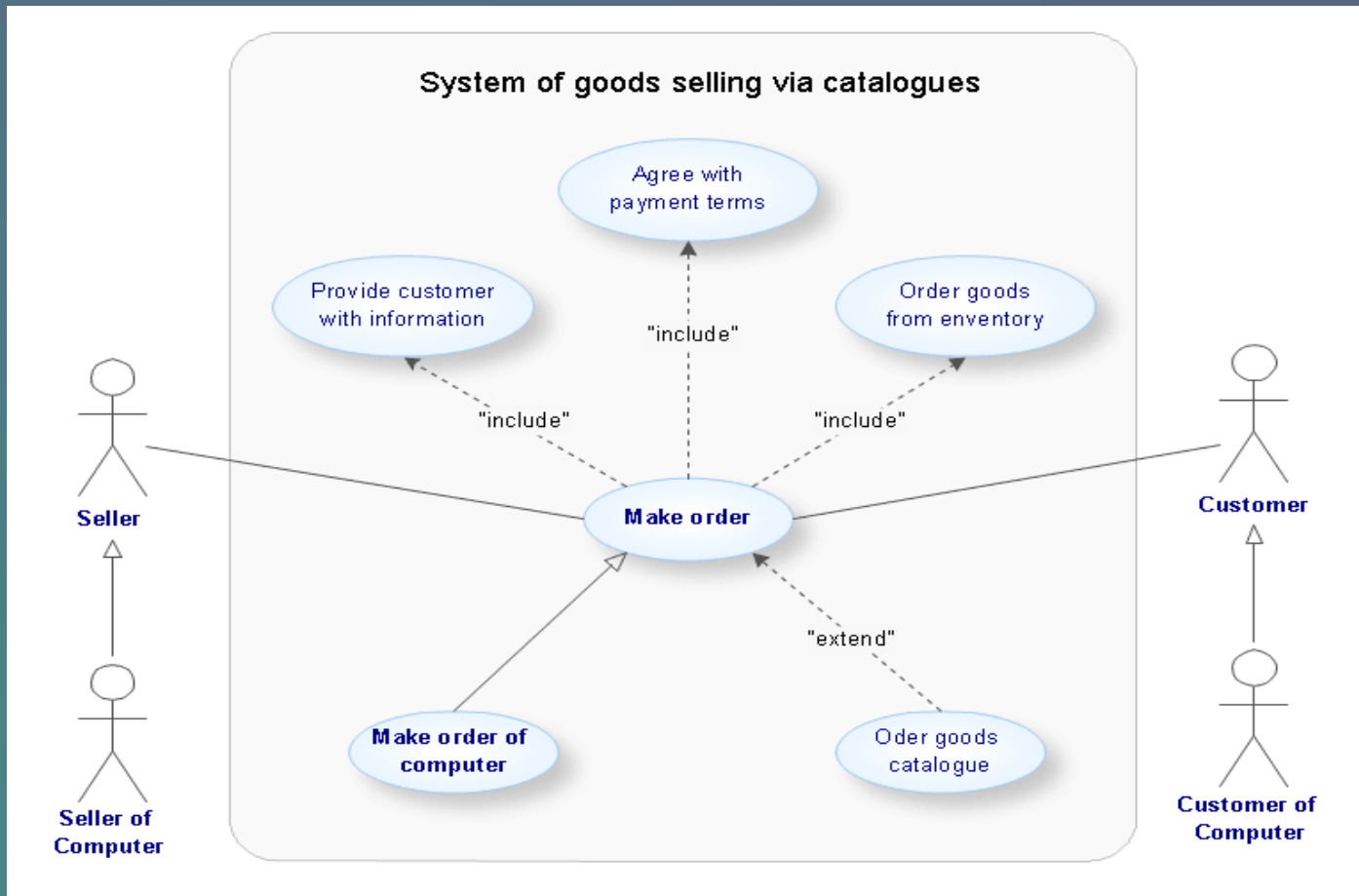
\* Diagramas de estrutura. Um tipo de diagrama que ilustra os elementos de uma especificação que não são respectivos ao tempo. Isto inclui diagramas de pacote, objeto, distribuição, componente, estrutura composta e classe.



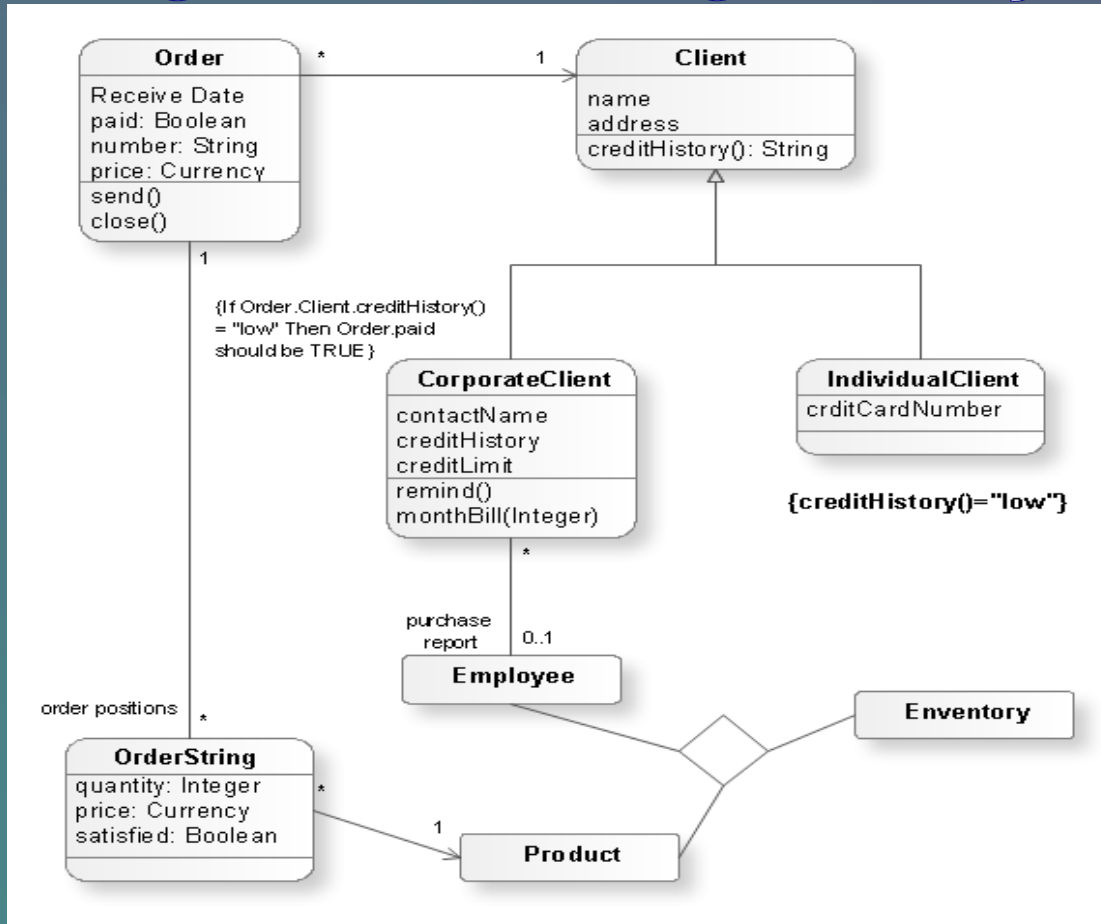
# Lógica de Programação



# Lógica de Programação



# Lógica de Programação



# Lógica de Programação

Logica nas diversas areas relacionadas – Geometria  
Computacional

## Geometria Computacional

A lógica por trás da Geometria Computacional se resume a conhecer alguns fundamentos geométricos que quando juntados formam o algoritmo desejado para se levar a cabo uma dada tarefa.



# Lógica de Programação

Logica nas diversas areas relacionadas – Geometria Computacional

Dessa forma há algumas noções básicas que são utilizadas para uma infinidade de algoritmos e se faz necessário conhecê-las, pois a grande maioria dos algoritmos desta categoria as usa.

Algumas dessas noções são: vetores, matrizes, operações em vetores e matrizes, distância euclidiana.



# Lógica de Programação

Logica nas diversas areas relacionadas – Animacao por computador

## Animação por computador

Para se poder produzir uma animação é necessário que uma dada seqüência de imagens seja passada num tempo baixo, dando a impressão de movimento. Este tempo é pouco maior que 1/20 segundos.

Tal lógica também vale para a animação produzida pela TV e pelo cinema.

# Lógica de Programação

Logica nas diversas areas relacionadas – Teoria dos Grafos

## Teoria dos Grafos

Um grafo é uma estrutura formada por nós e arestas. Os nós são pontos, locais, cidades, enfim, objetos distintos que podem ser enumerados. As arestas expressam as ligacoes entre vertices, ou seja, uma relação entre estes.

Os grafos são um conceito muito importante, estando presente em diversos problemas do mundo real.



# Lógica de Programação

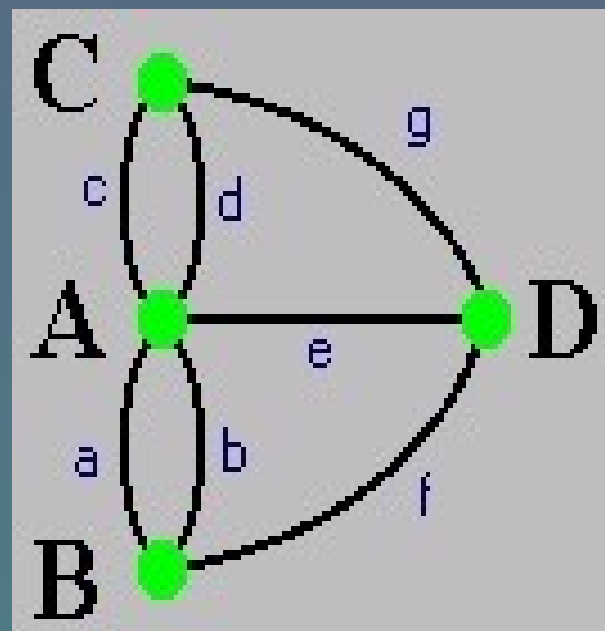
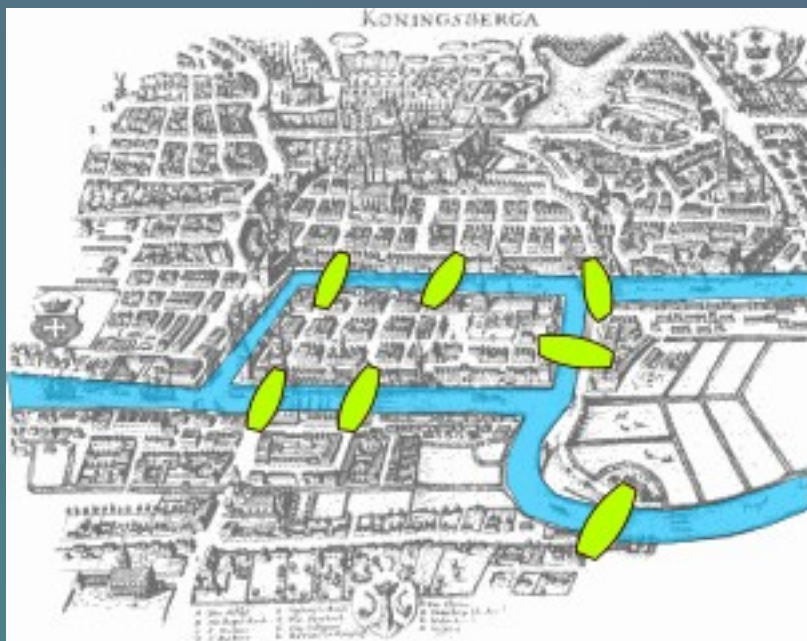
Logica nas diversas areas relacionadas – Teoria dos Grafos

Para resolver um problema com teoria de grafos, devemos primeiro formulá-lo, identificando os vértices e arestas, e, em seguida formular um algoritmo para encontrar a solução de nosso problema.

Tal algoritmo geralmente envolve uma solução com uma busca no grafo, percorrendo-o.



# Lógica de Programação



Problema das pontes de Königsberg

# Lógica de Programação

## Referências Bibliográficas

1. [http://pt.wikipedia.org/wiki/Bubble\\_sort](http://pt.wikipedia.org/wiki/Bubble_sort)
2. Algoritmos - Logica para desenvolvimento de programacao de computadores Jose Augusto N.G. Manzano, Jayr Figueiredo de Oliveira, Ed. Erica, Sao Paulo - SP, 2000, 10 ed.
3. [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)
4. <http://www.agilemodeling.com/essays/umlDiagrams.htm>
5. [http://en.wikipedia.org/wiki/Computer\\_animation](http://en.wikipedia.org/wiki/Computer_animation)
6. Paulo Sergio de Moraes, Lógica de Programação, Disponível em:  
<ftp.unicamp.br/pub/apoio/treinamentos/logica/logica.pdf>  
[http://blogs.claritycon.com/blogs/peter\\_miller/archive/2007/10/10/3307.aspx](http://blogs.claritycon.com/blogs/peter_miller/archive/2007/10/10/3307.aspx)