Paradigmas de Linguagem de Programação

EDUARDO HABIB BECHELANE MAIA

HABIB@CEFETMG.BR

O que é um Paradigma?

É um modelo imperativo de uma realidade

Permite organizar as ideias com vista:

- Ao atendimento dessa realidade
- À determinação de qual é a melhor forma de atuar sobre essa realidade

O que é Paradigma de Programação?

Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns.

Características de Paradigmas de Programação

Gramática e significado bem definidos:

sintaxe: gramática (forma);

semântica: significado].

Implementável (executável):

com eficiência ‘‘aceitável’’.

Universal:

deve ser possível expressar todo problema computável.

Propriedades Desejáveis

Legibilidade	Confiabilidade	Eficiência	Facilidade de aprendizado
Ortogonalidade	Reusabilidade	Modificabilidade	Portabilidade

Por que tantas Linguagens?

PROPÓSITOS DIFERENTES

AVANÇOS TECNOLÓGICOS

INTERESSES COMERCIAS

CULTURA E BACKGROUND CIENTÍFICO

Paradigma estruturado

O Paradigma Estruturado

Em linguagens puramente imperativas (Assembly) é muito fácil o programador criar códigos de difícil leitura

Ex: Saltos

Programação estruturada

programador tem maior controle sobre sobre o fluxo de execução do programa.

Estruturas de sequência

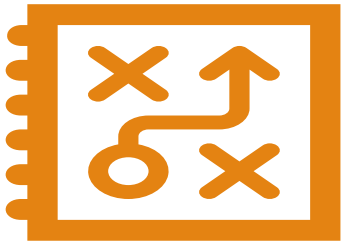
Estruturas de decisão

Estruturas de iteração

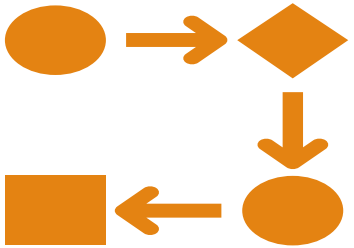
Comandos são organizados em grupos, chamados funções, que podem ser evocados em momentos diferentes.

Elementos

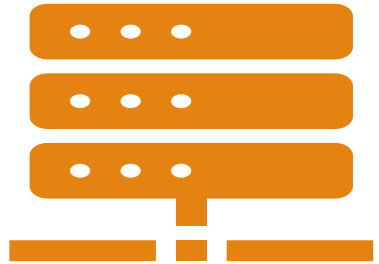
Definição de tipos de dados



Expressões e atribuições



Estruturas de controle de fluxo



Definição de sub-rotinas

Vantagens do modelo estruturado

Eficiência

Paradigma dominante e bem estabelecido

Modelagem natural de aplicações do mundo real

Melhor controle sobre o
fluxo de execução do código

É fácil de se entender.

Desvantagens do paradigma estruturado

Ainda se foca em como a tarefa deve ser feita e não em o que deve ser feito.

Pode gerar códigos confusos

tratamento dos dados são misturados com o comportamento do programa.

Alguns exemplos

Ada

Algol

Basic

C

PHP

Cobol

Fortran

Pascal

Python

Paradigma Orientado a objetos

O Paradigma Orientado a Objetos

Aplicação é estruturada em módulos (classes).

Que agrupam um estado (atributos).

- e operações (métodos) sobre este.

Classes podem ser estendidas e/ou usadas como tipos.

Encapsulamento.

Reuso de código.

Facilidade de manutenção.

Polimorfismo.

Vantagens do Paradigma Orientado a Objetos

Todas as do paradigma estruturado.

Estimula:

Modularidade;
Reusabilidade;
Extensibilidade.

Aceitação comercial crescente.

Alguns exemplos de linguagens do modelo OO

C++	C#	Java	VB.NET
Object Pascal	Objective-C	Python	Smalltalk

Funcional

Paradigma funcional

Alto nível de abstração

Ela enfatiza a aplicação de funções.

Soluções elegantes, concisas e poderosas.

Resultados das funções dependem apenas dos valores de entrada.

Forte fundamentação teórica.

Reutilização

Paradigma funcional

Ausência de estados

Tudo é função

Paradigma funcional

Por que estudar o paradigma funcional?

Visão clara de conceitos fundamentais

- Abstração
- Recursão
- Genericidade, sobrecarga, etc.

Ajuda na programação em outros paradigmas

Possibilita

Alta produtividade

Programas mais concisos

Menos erros

Exemplo

Quick sort – Haskell

```
qs []    = []
qs (x:xs) = qs [y | y <- xs, y < x]
            ++ [x]
            ++ qs [y | y <- xs, y >= x]
```

```
int particao(int vec[], int inicio, int
   fim) {
  int i, j;
  i = inicio;
  for (j = inicio + 1; j <= fim; ++j) {
    if (vec[j] < vec[inicio]) {
      ++i;
      troca(&vec[i], &vec[j]);
    }
  }
  troca(&vec[inicio], &vec[i]);
  return i;
}
```

Vantagens:

Alto nível de abstração

- Programas pequenos, claros e rápidos.

Desvantagens

Podem vir a ser menos eficientes.

Exemplos:

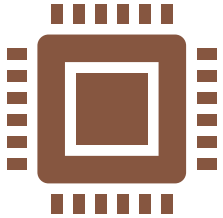
Lisp, ML, Miranda e Haskell

Paradigma funcional

Paradigma Paralelo e Distribuído

Introdução

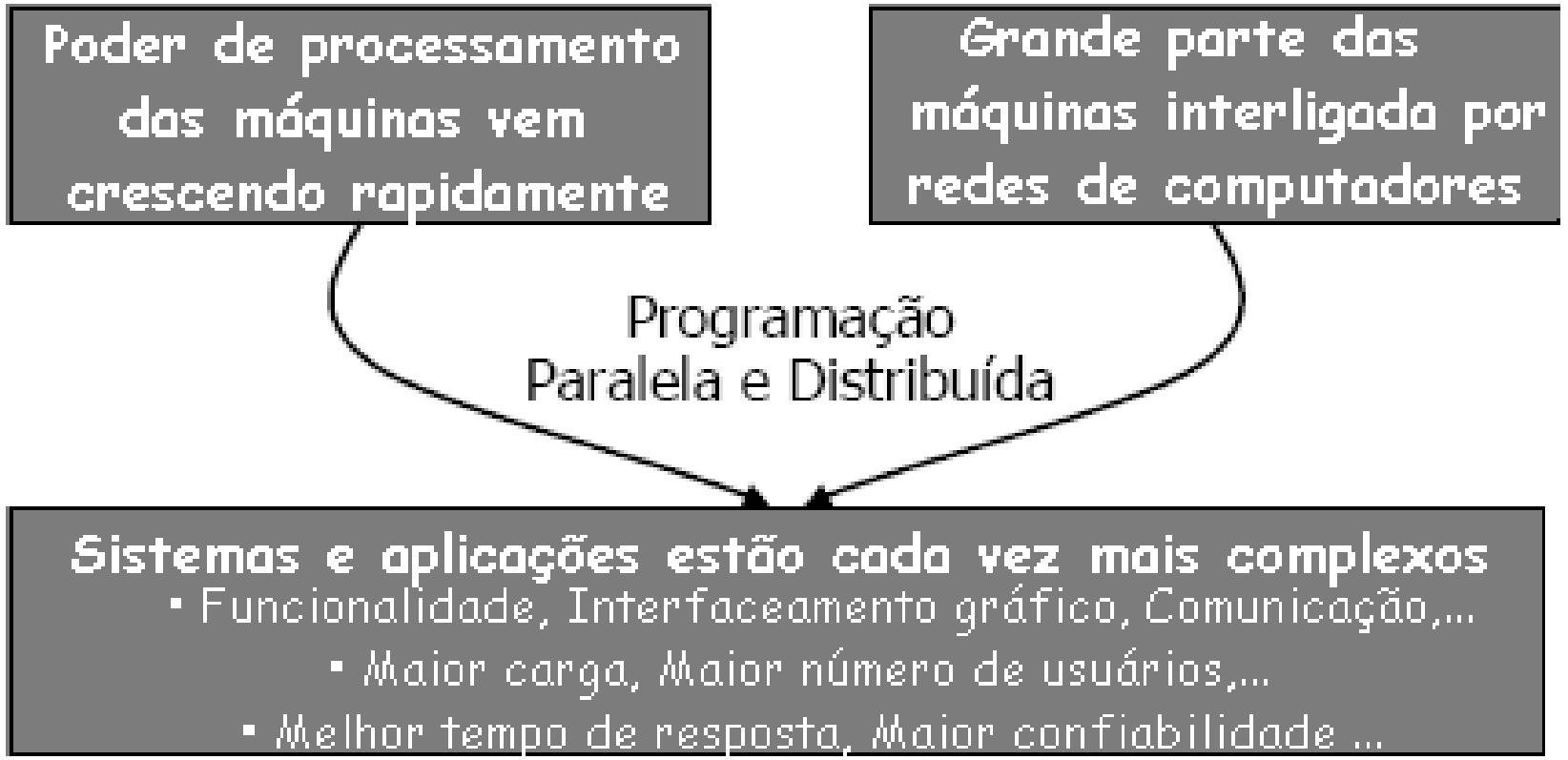
Poder de processamento das máquinas está crescendo rapidamente.



Grande parte das máquinas são interligadas por redes

Sistemas e aplicações estão cada vez mais complexos.

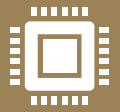
Introdução


Poder de processamento das máquinas vem crescendo rapidamente
Grande parte das máquinas interligada por redes de computadores
Programação Paralela e Distribuída
Sistemas e aplicações estão cada vez mais complexos
• Funcionalidade, Interfaceamento gráfico, Comunicação,...
• Maior carga, Maior número de usuários,...
• Melhor tempo de resposta, Maior confiabilidade ...

Paradigma paralelo

Consiste em executar simultaneamente várias partes de uma mesma aplicação.

Tornou-se possível a partir do desenvolvimento de sistemas operacionais multi-tarefa, multi-thread e paralelos.

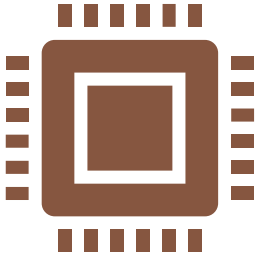


As aplicações são executadas paralelamente:

- Em um mesmo processador.
- Em uma máquina multiprocessada.
- Em um grupo de máquinas interligadas que se comporta como uma só máquina.

Paradigma Distribuído

Consiste em executar aplicações cooperantes em máquinas diferentes.



Tornou-se possível a partir da popularização das redes de computadores.

Intranets

Internet

Outras redes públicas ou privadas

Diferenças entre sistemas paralelos e distribuídos

Paralelo	Distribuído
Fortemente acoplados	Fracamente acoplados
Sistemas paralelos são mais previsíveis	Mais imprevisíveis
Troca de mensagens são praticamente em tempo real	Troca de mensagens influenciada pela rede
Maior controle sobre os recursos computacionais	Menor controle sobre os recursos computacionais

Vantagens

Ambos

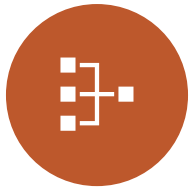
- Maior poder computacional
- Aumenta a vazão
- Compartilhamento de recursos
- Mais confiáveis
- Reutilização

Para os sistemas distribuídos

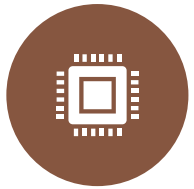
- Separação física possibilita independência
- Escalável

Para os sistemas paralelos

- Diminui a latência

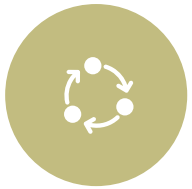


HETEROGENEIDADE DO SISTEMA

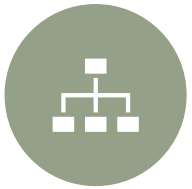


SINCRONIZAÇÃO DAS CÓPIAS OU CACHE (SE HOUVER);

AUMENTO DA VAZÃO IMPLICA EM AUMENTO DO USO DA REDE;



TRANSPARÊNCIA NO ACESSO PARALELO NÃO É FÁCIL DE SE IMPLEMENTAR.



GERENCIAMENTO DO SISTEMA

CONTROLE DE ACESSO AOS RECURSOS COMPARTILHADOS

FALHAS DA REDE PODEM COMPROMETER O SISTEMA.

SEGURANÇA E SIGILO.

Desvantagens Sistemas Distribuídos

Tendências

Integração de paradigmas

Aumentar o domínio da aplicação

Cautela

Não viole princípios básicos