



**UNIVERSIDADE
JOAQUIM CHISSANO**

**FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO**

Modelos de Linguagens

■ **Modelo imperativo**

- Linguagens expressam sequências de comandos que realizam transformações sobre dados
- base: máquina de von Neumann
- orientadas a procedimentos

■ **Modelo declarativo**

- Linguagens que não possuem o conceito de sequências de comandos
- Linguagens funcionais, baseadas em funções
- Linguagens lógicas, baseada em axiomas lógicos

Modelos de execução

■ **Sequencial**

- Uma computação é realizada após o término da anterior
- Controle de fluxo de execução interno ao programa:
 - ◆ Sequência
 - ◆ Seleção
 - ◆ Iteração
 - ◆ Invocações

■ **Concorrente**

- Múltiplas computações podem ser executadas simultaneamente
- Computações paralelas
 - ◆ Múltiplos processadores compartilham memória
- Computações distribuídas
 - ◆ múltiplos computadores conectados por uma rede de comunicação

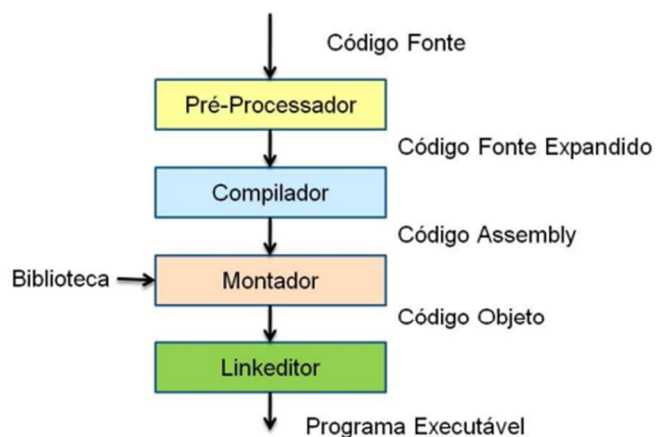


**UNIVERSIDADE
JOAQUIM CHISSANO**

FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO

Modelo de Execução/Construção

■ Compilada



■ Interpretada



Modelo de Nível

■ Baixo Nível

- ◆ Próximo à Máquina

```
MOV AX,0002
MOV BX,0001
ADD AX,BX
```



■ Alto Nível

- ◆ Próximo ao programador

```
int A, B;
int main()
{
    A = 2;
    B = 1;
    A = A + B;
}
```



**UNIVERSIDADE
JOAQUIM CHISSANO**

**FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO**

Modelo de Aplicação

- Científica
 - ◆ Algol, Fortran, Matlab
- Comerciais
 - ◆ Cobol, Dataflex
- Processamento de Listas
 - ◆ Lisp
- Gerais
 - ◆ Java, C, C++, Visual Basic
- Inteligência Artificial
 - ◆ PROLOG
- Web
 - ◆ JSP, PHP, HTML

Critérios para avaliação de LP

- Legibilidade
- Simplicidade
- Expressividade
- Ortogonalidade
- Confiabilidade
- Portabilidade



**UNIVERSIDADE
JOAQUIM CHISSANO**

**FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO**

Legibilidade

- Facilidade de ler e escrever programas
- Legibilidade influi:
 - desenvolvimento e depuração de programas
 - manutenção de programas
 - desempenho de equipes de programação
- Fatores que melhoram a legibilidade:
 - Abstração de dados
 - Comandos de controle
 - Modularização de programas
 - Documentação
 - Convenções léxicas, sintaxe e semântica
 - ◆ Exemplo em Java: nomes de classes iniciam por letra maiúscula
nomes de campos usam letras minúsculas

Simplicidade

- Representação de cada conceito seja simples de aprender e dominar:
 - Simplicidade sintática exige que a representação seja feita de modo preciso, sem ambiguidades
 - ◆ Contraexemplo 1: `A++; A=A+1; A+=1; ++A.`
 - ◆ Contraexemplo 2: `a, b := x+1, y + 3;`
 - Simplicidade semântica exige que a representação possua um significado independente de contexto
 - ◆ Contraexemplo: `private: B b; class Campo: private Campo`
 - Simplicidade não significa concisão
 - ◆ A linguagem pode ser concisa mas usar muitos símbolos especiais: Ex. Linguagens Funcionais



**UNIVERSIDADE
JOAQUIM CHISSANO**

**FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO**

Expressividade

- Representação clara e simples de dados e procedimentos a serem executados pelo programa
 - Exemplo: tipos de dados em Pascal
- Expressividade x concisão
 - Muito concisa: falta expressividade?
 - Muito extensa: falta simplicidade?
- Linguagens mais modernas
 - Incorporam apenas um conjunto básico de representações de tipos de dados e comandos
 - Aumentam o poder de expressividade com bibliotecas de componentes
 - ◆ Ex. BigInteger em Java

Ortogonalidade

- conjunto de construções primitivas pode se combinar em um número grande de maneiras para construir as estruturas de uma linguagem
- Possibilidade de combinar entre si, sem restrições, os componentes básicos da LP
 - Exemplo: permitir combinações de estruturas de dados, como arrays de qualquer tipo.
 - Contraexemplo: não permitir que um array seja usado como parâmetro de uma função
- Componente de primeira ordem: pode ser livremente usado em expressões, atribuições, como argumento e retorno de procedimentos



**UNIVERSIDADE
JOAQUIM CHISSANO**

**FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO**

Portabilidade

- **Multiplataforma:** capacidade de um software rodar em diferentes plataformas sem a necessidade de maiores adaptações
 - Sem exigências especiais de hardware/software
 - Exemplo: aplicação compatível com Linux e Windows
- **Longevidade:**
 - ciclo de vida útil do software e o do hardware não precisam ser síncronos; ou seja, é possível usar o mesmo software após uma mudança de hardware

Confiabilidade

- **Mecanismos que facilitem a produção de programas que atendam às suas especificações**
 - **Tipagem forte:** o processador da linguagem deve:
 - ◆ assegurar que a utilização dos diferentes tipos de dados seja compatível com a sua definição
 - ◆ evitar que operações perigosas, tal como aritmética de ponteiros, seja permitida
 - **Tratamento de exceções:** sistemas de tratamento de exceções permitem construir programas que:
 - ◆ possuam definições de como proceder em caso de comportamento não usual
 - ◆ possibilitem tanto o diagnóstico quanto o tratamento de erros em tempo de execução



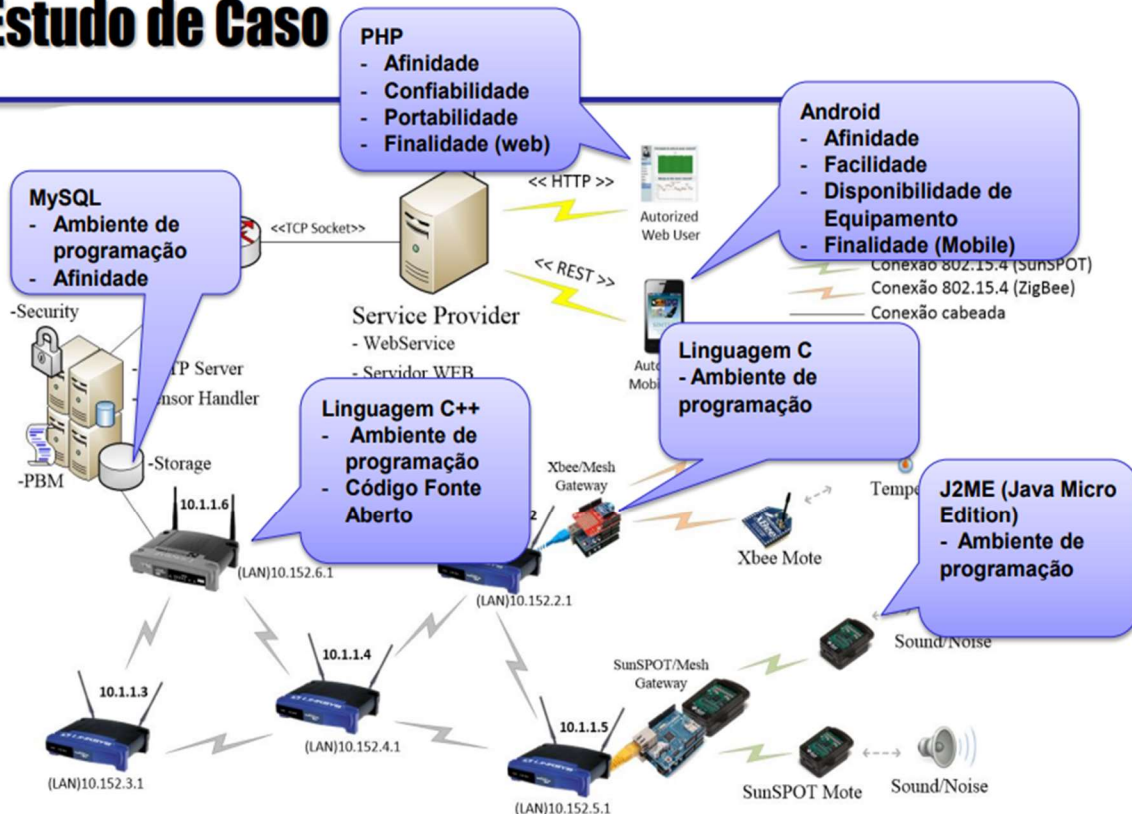
**UNIVERSIDADE
JOAQUIM CHISSANO**

**FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO**

Critérios para escolha de uma LP

- **Implementação**
 - Disponibilidade quanto a plataforma;
 - Eficiência;
- **Competência na LP**
 - Experiência do programador;
 - Competência do grupo envolvido;
- **Portabilidade**
 - Execução em várias plataformas
- **Sintaxe**
 - Certos tipos de problemas acomodam-se melhor a certos modelos;
- **Semântica**
 - Aplicação X facilidade;
- **Ambiente de Programação**
 - Ferramentas diminuem o esforço;
 - Uso de bibliotecas
- **Modelo de computação**

Estudo de Caso



FIM...CONTINUA NA PRÓXIMA AULA. POR: COMPILADO NELSON ALI COM RECURSO A FONTES ABERTAS NA INTERNET.