

Programação Orientada a Serviços

Prof. Diego Cirilo

Aula 03: Evolução

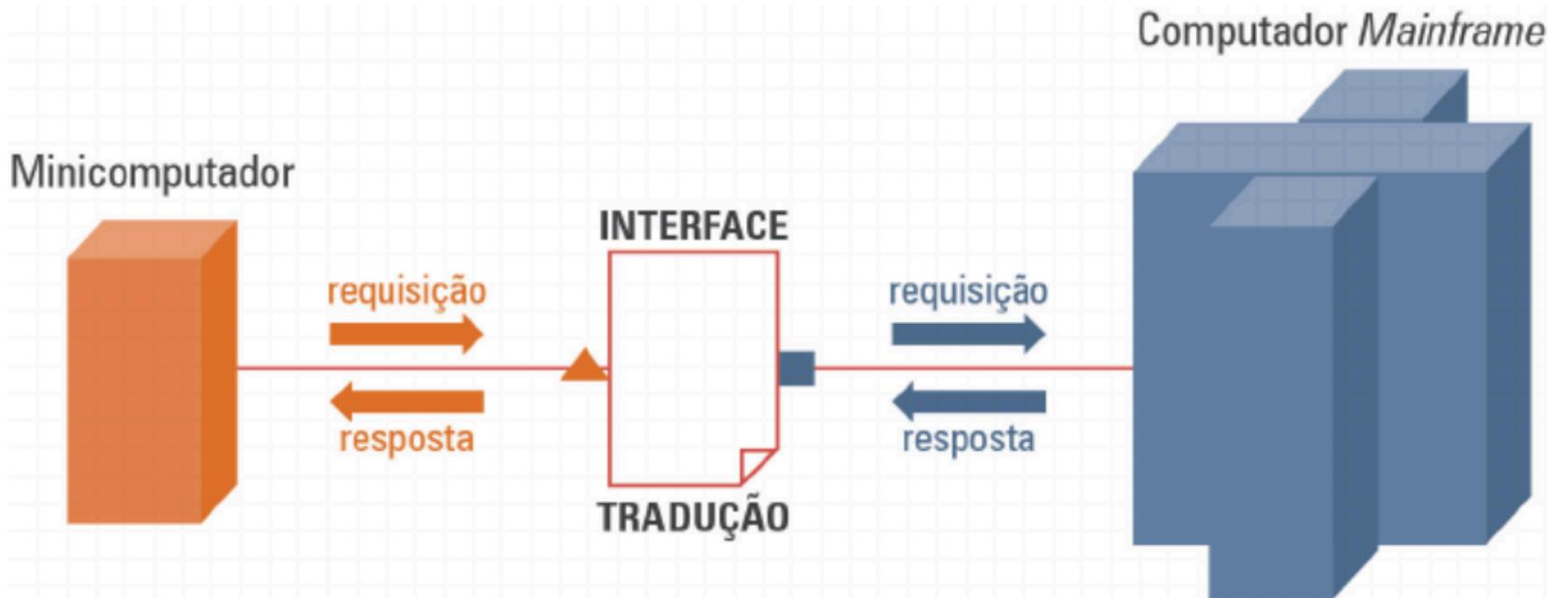
Arquiteturas Monolíticas

- Metodologia básica de desenvolvimento
- Sem problemas para sistemas simples
- Sistemas complexos...
- Reflexo do mercado na época.



Computação Distribuída

- Sistemas grandes: dividir para conquistar
- Melhor alocação de recursos
- Confiabilidade
- Manutenção, etc.

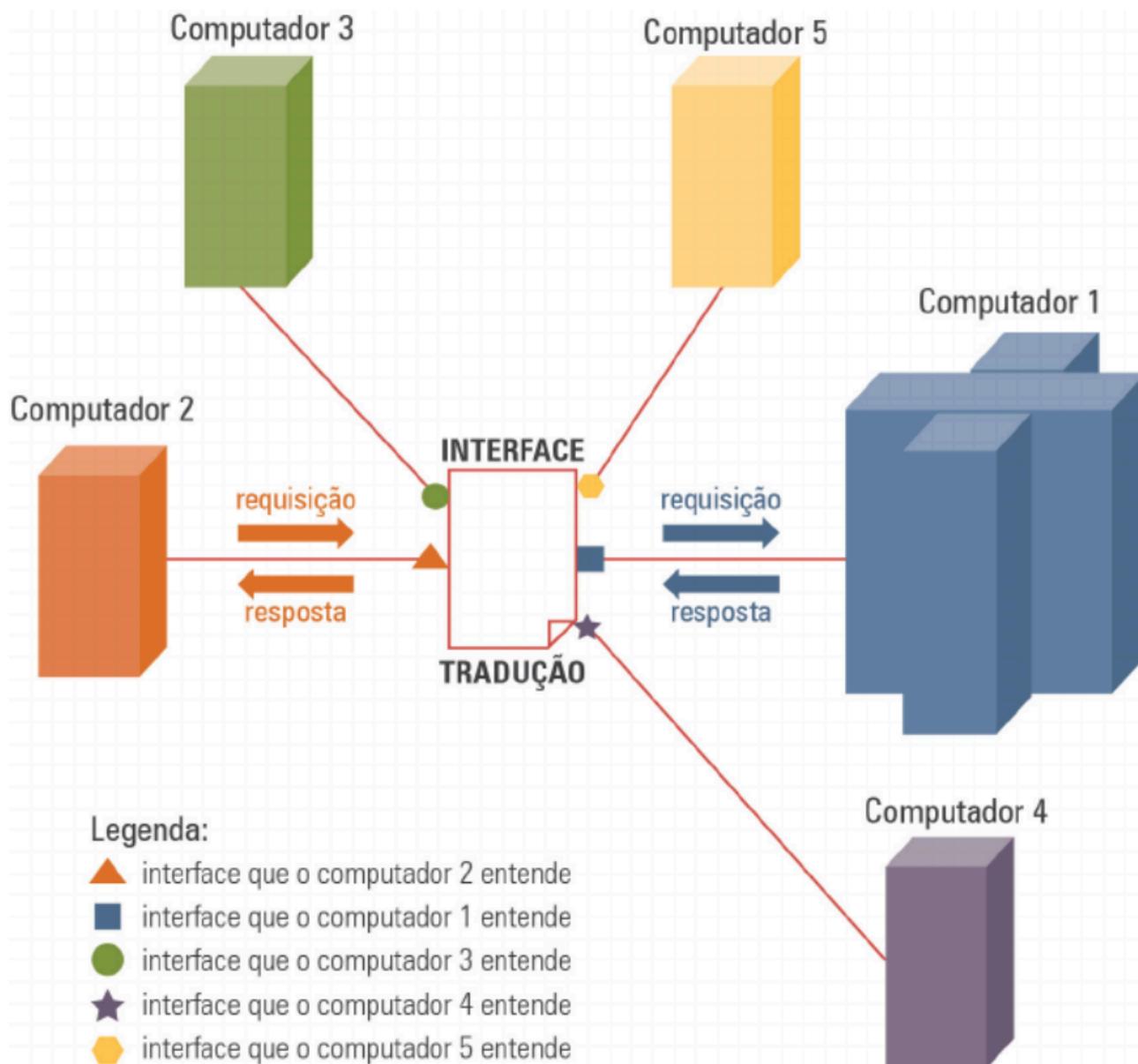


Legenda:

- ▲ interface que o mini-computador entende
- interface que o computador mainframe entende

Interfaces

- Problemas de interoperabilidade
- Padrões proprietários
- Desenvolvimento *on-demand*
- Baixo reuso



Alto acoplamento

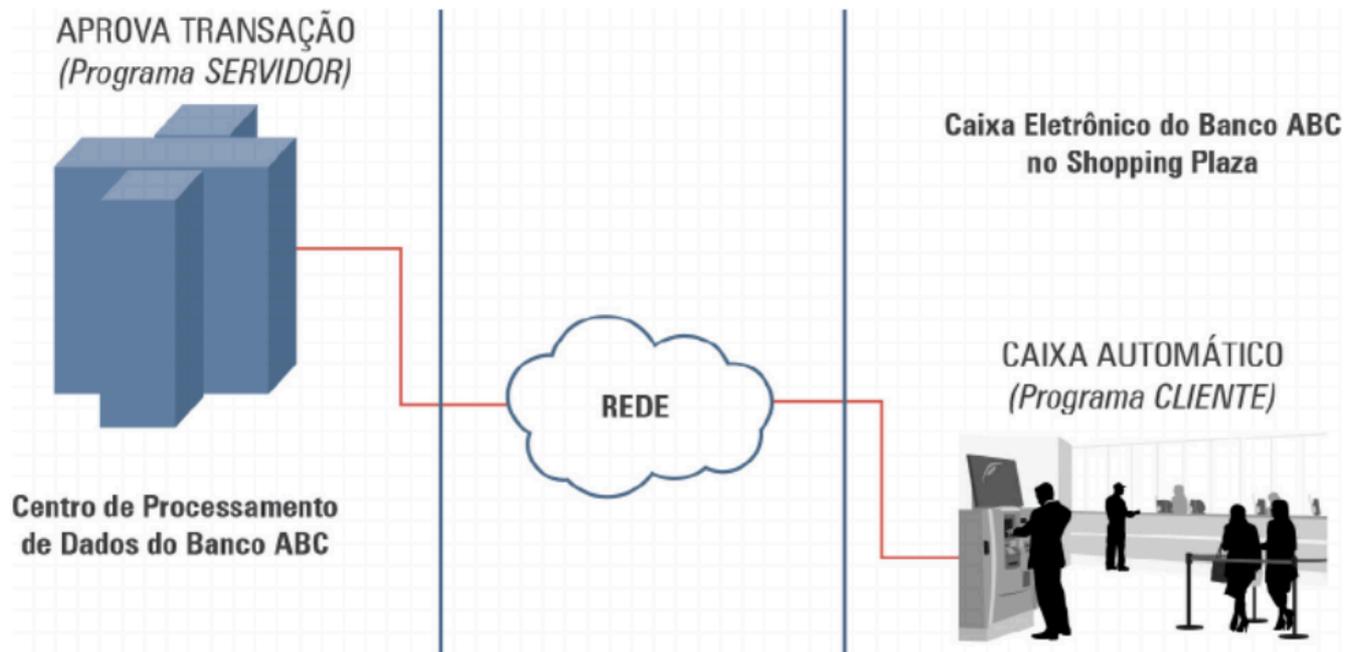
- Sistemas desenvolvidos em conjunto
- O funcionamento de cada elemento depende dos outros
- Protecionismo?
- Alto custo.

Orientação a objetos

- Reuso
- Padrões
- Melhor organização do código
- Eficiência/Baixo custo
- Problema: o reuso dependia da linguagem

Arquitetura Cliente-Servidor

- Advento da internet
- Possibilidade de "reuso" em diferentes linguagens
- Só funciona com padrões abertos...



Padrões

- Permitem a interoperabilidade
- Grupos industriais colaborativos
- *W3C - World Wide Web Consortium*
- Tim Berners-Lee
- HTTP/HTML
- Funciona em outras áreas: tomadas, USB-C, Bluetooth, P2, etc.

Exemplos

- HTTP: *Hyper Text Transfer Protocol*
- XML: *eXtensible Markup Language*
- SOAP: *Simple Object Access Protocol*
- WSDL: *Web Services Description Language*
- UDDI: *Universal Description, Discovery and Integration*
- REST: *REpresentational State Transfer*

```
graph LR; A[Silos: aplicações monolíticas] --> B[Computação distribuída e mensageria]; B --> C[Orientação a objeto e Cliente-Servidor]; C --> D[Padrões Uniformes]; D --> E[Arquitetura Orientada e Serviço];
```

Silos:
aplicações
monolíticas

Computação
distribuída e
mensageria

Orientação a objeto
e Cliente-Servidor

Padrões
Uniformes

Arquitetura
Orientada e
Serviço

Dúvidas?

