

Introdução a Fundamentos da Web

A Internet



O Projeto Arpa

- Conectar redes físicas separadas sem que as ligações aumentem os recursos de rede para links constantes?
 - troca de pacotes e envolve requisições de dados sendo divididos em pequenos pedaços
 - processados rapidamente sem bloquear a comunicação de outras partes
 - **Internet!!!**



Depois da Arpanet...

- Surgimento de várias outras redes usando a mesma técnica de troca de pacotes – por exemplo, X.25
 - as bases da primeira rede universitária do Reino Unido JANET (Conjunto de rede acadêmica),
 - rede pública americana CompuServe
 - um empreendimento comercial permitindo pequenas empresas e indivíduos a acessarem recursos computacionais por um tempo compartilhado, e depois acesso à Internet
- Estas redes apesar de terem muitas conexões, foram mais redes privadas que a Internet de hoje.

Padronização

- A proliferação de diferentes protocolos de rede logo se tornou um problema
 - Dificuldade de comunicação entre redes
- Esforço na arquitetura de redes mais aberta para substituir o protocolo atual usado na ARPANET.
 - Criação de um sistema que mascara a diferença entre os protocolos de rede usando um novo padrão.
 - 1982: Conexões da ARPANET para fora dos EUA foram convertidas para usar o novo protocolo “TCP/IP”

NÃO
CONFUNDA
INTERNET
COM WEB!!!

“A Internet é um **conglomerado de redes em escala mundial** de milhões de computadores interligados pelo **TCP/IP** que permite o acesso a informações e todo tipo de transferência de dados. Ela carrega uma **ampla variedade de recursos e serviços**”

(Wikipedia)



E a Web nessa história toda???

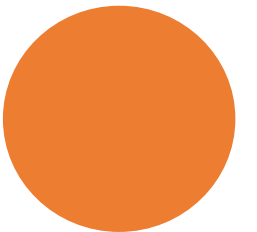
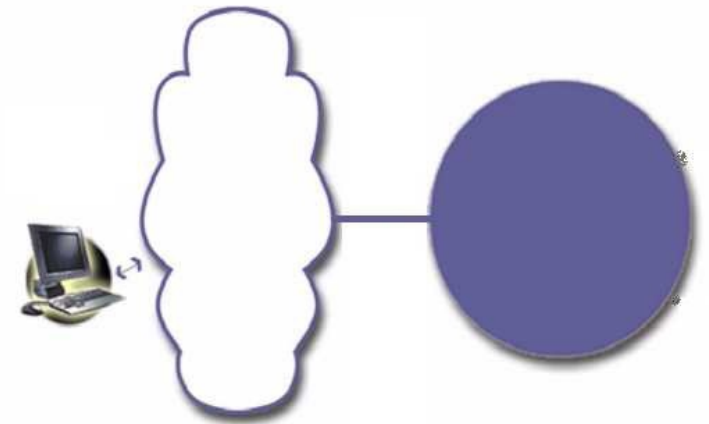
Web é um serviço de documentos interligados por meio de **hipertextos**."

Criação da Web

- Universidade de Minnesota:
 - O **Gopher** foi um sistema de recuperação de informação usado no início dos anos 90, oferecendo um método de entrega de menus de links para arquivos
- Tim Berners-Lee (CERN) e a criação de hipertextos:
 - o texto poderia conter links e referências para outros trabalhos, permitindo o leitor a pular rapidamente de um documento para outro.

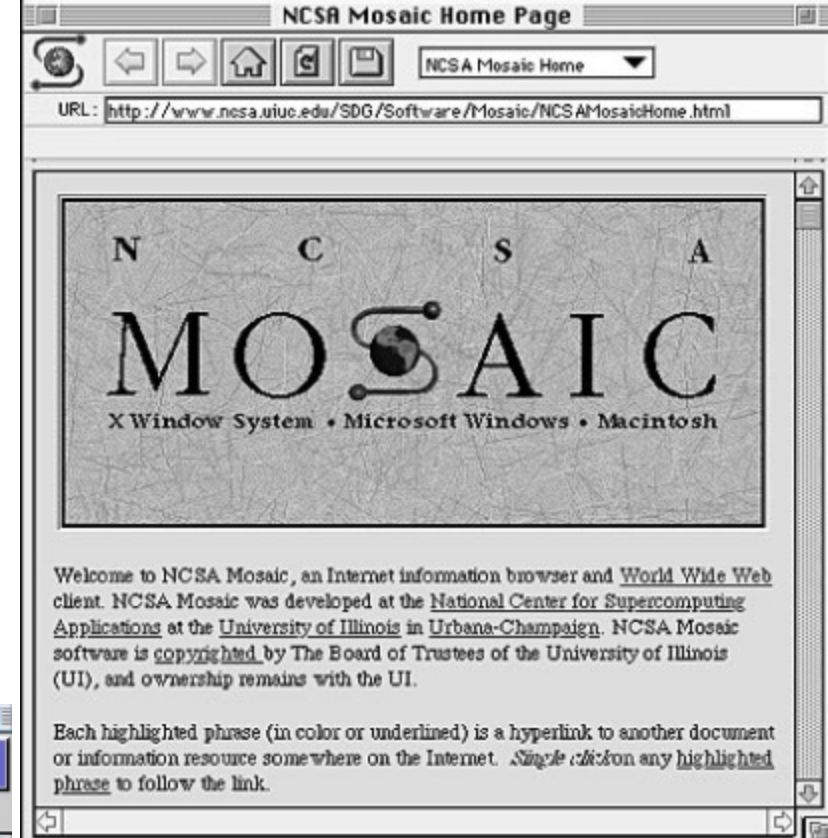
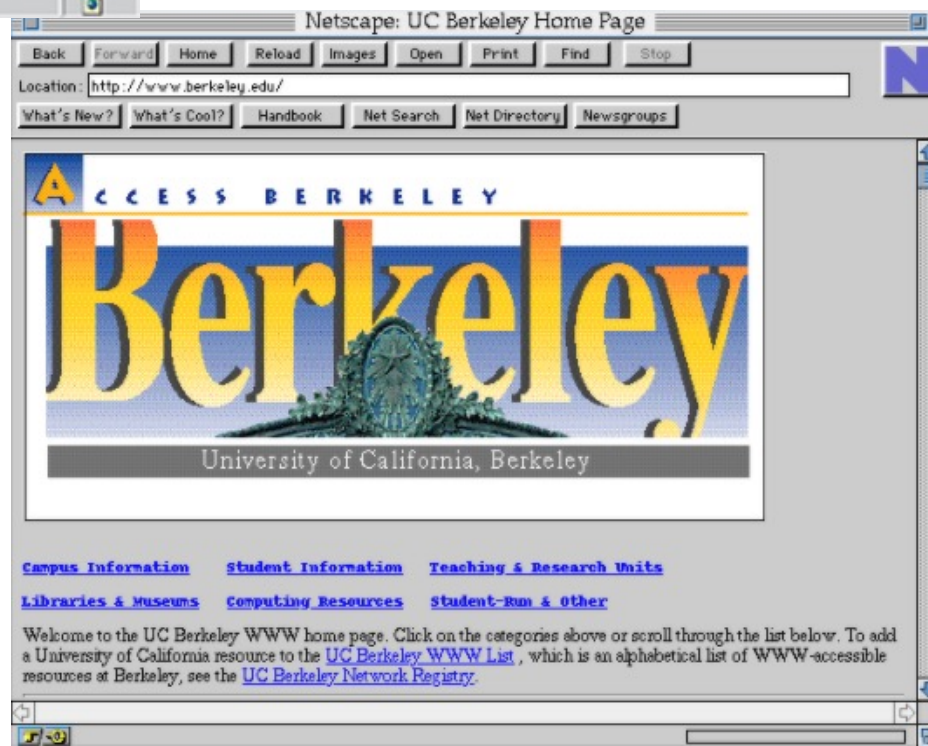
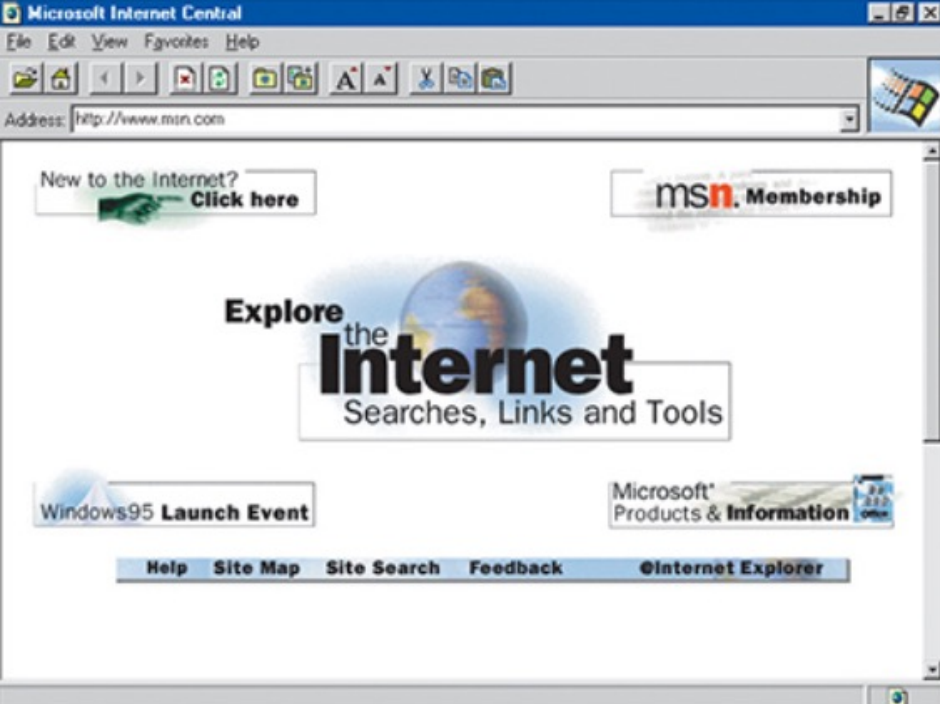
Navegadores (Browsers)

- Navegador Web é um programa que se conecta a um servidor Web ou HTTP e recebe uma página com conteúdo definido com a linguagem de marcação HTML.
- O navegador interpreta a página ou imagem recebida e exibe-a para o usuário.



A “guerra dos navegadores”

- A popularização da web trouxe interesses comerciais.
 - Surgimento de diferentes programas interpretadores de documentos de hipertextos: Netscape Navigator, Mosaic, Internet Explorer, ...
- Netscape e a Microsoft tentaram cada qual obter uma margem competitiva em termos de recursos suportados, a fim de atrair desenvolvedores.
 - “Guerra dos navegadores”.



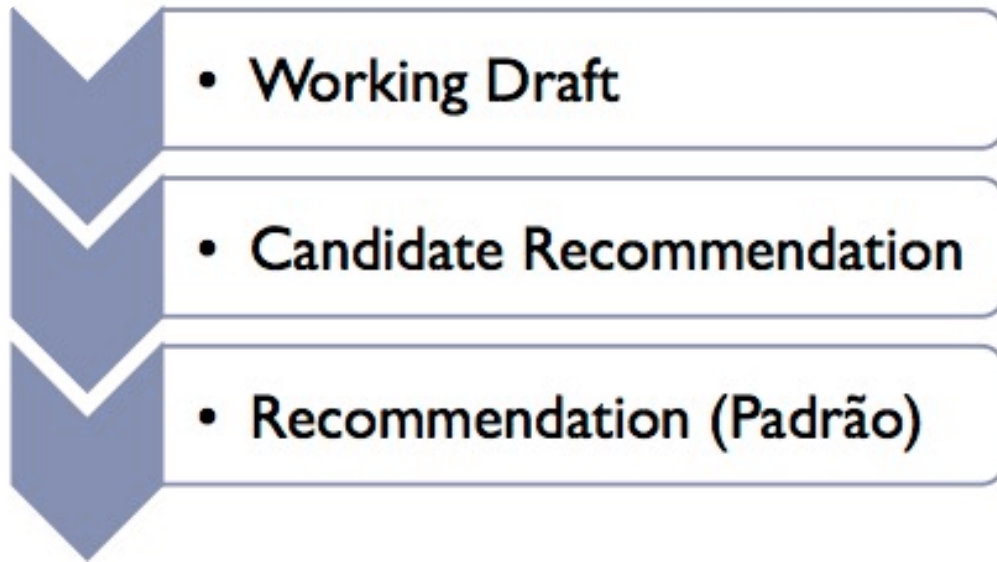
A chegada dos padrões web

- Microsoft e Netscape estiveram focadas em implementar novas funções em vez de consertar os problemas com as funções já suportadas,
 - adicionaram funções proprietárias e
 - criaram funções que competiam diretamente com funções existentes no outro navegador, mas implementadas de uma **forma incompatível**.

A formação da W3C

- Tim Berners-Lee fundou o World Wide Web Consortium (W3C)
 - padronizar os protocolos e tecnologias usados para criar a web
 - conteúdo acessado largamente pela população mundial tanto quanto o possível.
- W3C publicou várias especificações (chamadas “recomendações”) incluindo o HTML, o formato de imagens PNG, e as Folhas de Estilo em Cascata versões 1, 2 e 3.

W3C – www.w3.org



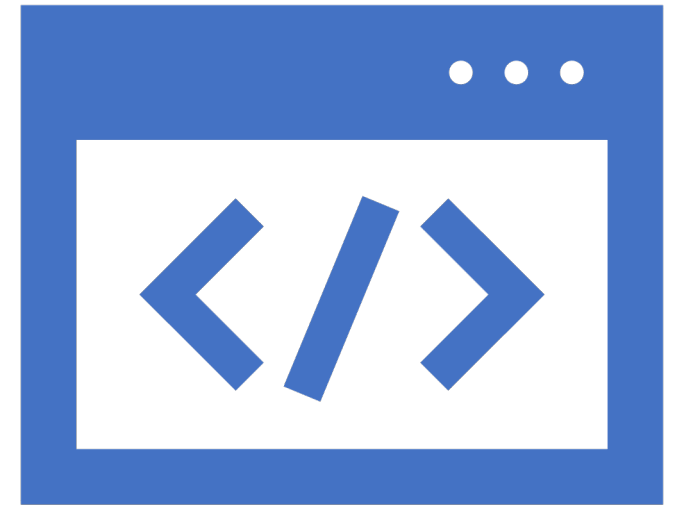
- Órgão responsável pela padronização de iniciativas ligadas à Web
 - Ex.: HTML, XML e iniciativas relacionadas, entre outros
- Especificações dessas iniciativas são classificadas de acordo com seu nível de “maturidade”

O Projeto Padrões Web

- 1998: principais navegadores: Internet Explorer 4 e o Netscape Navigator 4
- Uma versão beta do Internet Explorer 5 foi lançada
 - implementava um novo e proprietário HTML dinâmico.
 - os desenvolvedores web profissionais deveriam saber **cinco maneiras diferentes de escrever JavaScript**.


Padrões Web

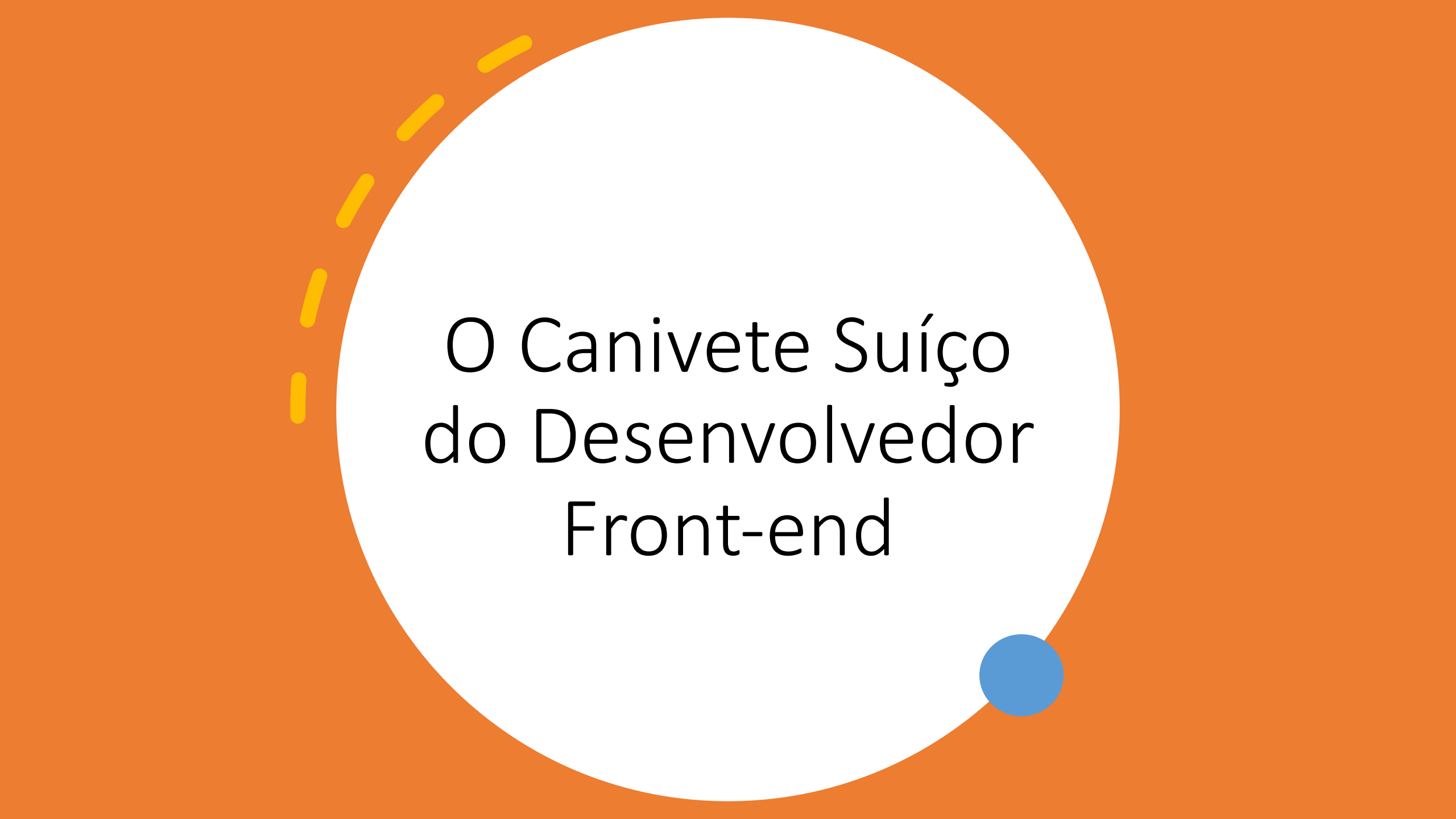
- Um grupo de desenvolvedores web e web designers profissionais se uniu.
 - O grupo “Projeto Padrões Web” (ou “Web Standards Project” - WaSP).
 - Transformando as recomendações da W3C em “padrões”,
 - Microsoft e a Netscape teriam que se adaptar as recomendações da W3C





Deu certo!!! Ou
quase...

- No lançamento de versões seguintes, os navegadores passaram a dar suporte aos padrões estipulados,
 - Salvo algumas poucas exceções...
- 



O Canivete Suíço do Desenvolvedor Front-end

Sugestões de ferramentas

- Editor para códigos
 - Visual Studio Code, Sublime, Atom
- Sincronizador Browser/Editor
 - BrowserSync, LiveServer
- Browser com ferramentas de desenvolvimento
 - Firefox, Chrome
- Git / serviço remoto de armazenamento de repositórios git
 - Se necessário, instale um cliente git no seu computador; use o github para armazenar seus projetos
- Ferramenta de prototipação
 - Whimsical, Figma

Tecnologias Web

HTML, CSS e JavaScript



Dados Estruturados, Dados não estruturados...

- Dados estruturados
 - Obedecem um padrão; possuem um formato predefinido
 - Mais fácil de processar
 - Exemplo: Tabelas, bancos de dados relacionais, ...
- Dados não estruturados
 - Imprevisíveis
 - Difíceis de processar
 - Textos, documentos, imagens, multimídia em geral

A maior parte dos dados de interesse corporativos são não estruturados...

Não tem um meio termo???

- SIM!
- Dados semi estruturados
 - Dar alguma ordem para a bagunça
 - Define regras flexíveis para estruturação de dados
 - Facilita o processamento
 - Exemplos:
 - Notas fiscais eletrônicas, formulários, páginas web...

Linguagens de Marcação

Linguagens de marcação
são usadas para
estruturar documentos

XML, HTML

XML

XML é uma Recomendação W3C;

XML apenas descreve os dados e o que eles significam

XML foi desenvolvido para estruturar, armazenar e enviar os dados

“XML is a cross-platform, software and hardware independent tool for transmitting information” – W3Schools

Com XML a troca de dados entre sistemas incompatíveis é possível

As *tags* do XML não são pré-definidas, você deve criar as suas próprias *tags*, obedecendo a um pequeno conjunto de regras de sintaxe.

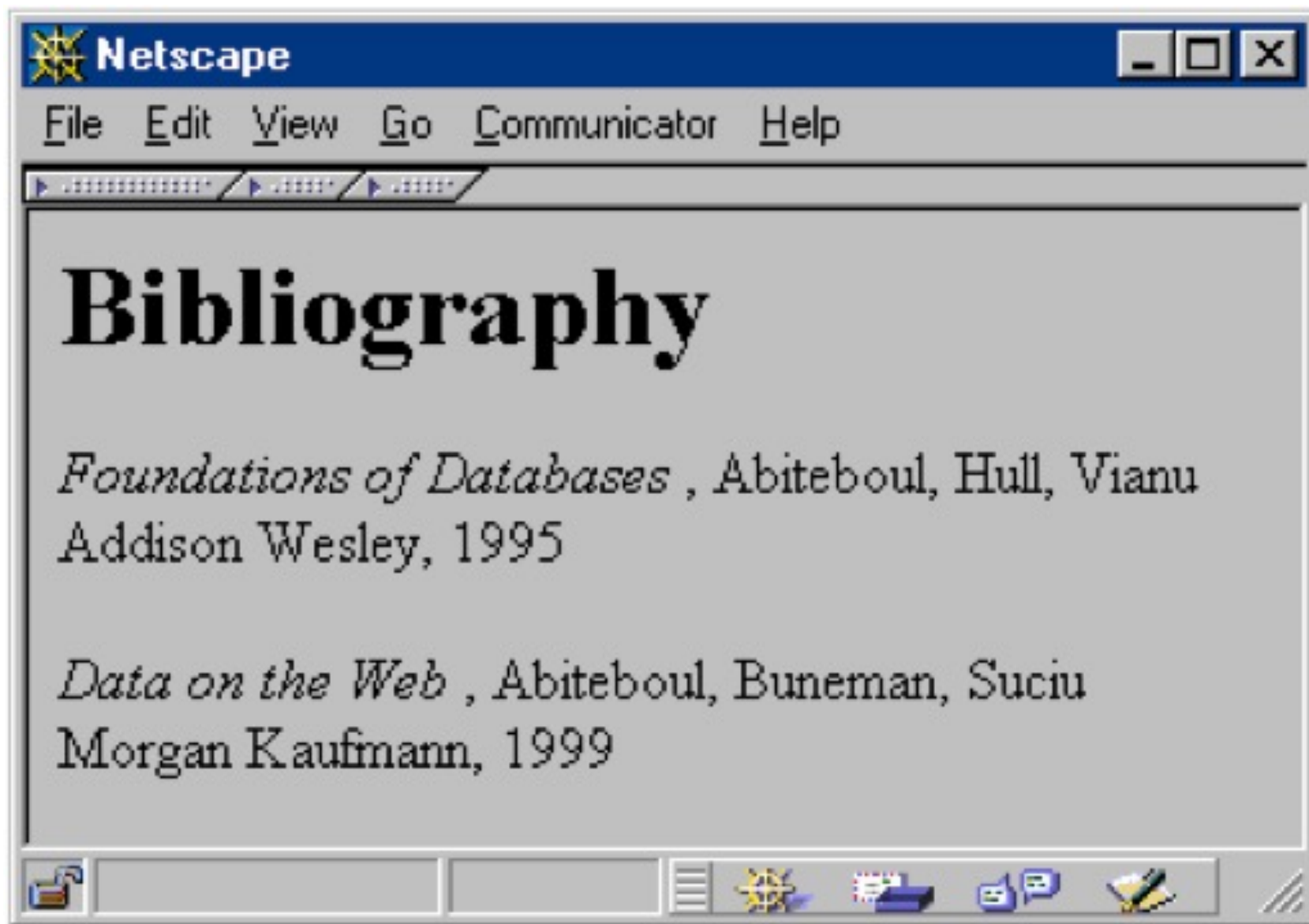
HTML vs XML

HTML – estrutura
formato de
documentos web

- HTML tem um conjunto fixo de tags e não descreve conteúdo obrigatoriamente

XML – descreve o
conteúdo do
documento

- Usuário define suas próprias tags para criar uma estrutura
- Um documento XML não tem nenhuma instrução para apresentação



Fonte HTML

```
<!doctype html>
<html>
<head>
  <title>A bibliography on Databases</title>
  <meta charset="utf-8">
</head>
<body>
  <h1> Bibliography </h1>
  <p> <em> Foundations of Databases </em> Abiteboul, Hull, Vianu <br>
  Addison Wesley, 1995
</p>
  <p> <em> Data on the Web </em> Abiteoul, Buneman, Suciu <br>
  Morgan Kaufmann, 1999
</p>
</body>
</html>
```

HTML: Conjunto pré-definido de elementos (tags) para especificação da estrutura e apresentação de um documento

XML: Elementos (tags) definidos pelo usuário da linguagem e servindo para descrever o conteúdo e a estrutura.

Fonte XML

```
<?xml version="1.0" encoding="utf-8"?>
<bibliography>
  <book>
    <title>Foundations of Databases</title>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <publisher>Addison Wesley</publisher>
    <year>1995</year>
  </book>
  <book>
    <title>Data on the Web</title>
    <author>Abiteboul</author>
    <author>Buneman</author>
    <author>Suciu</author>
    <publisher>Morgan Kaufmann</publisher>
    <year>1999</year>
  </book>
</bibliography>
```

Sintaxe XML

- Documento XML
 - Sequência de elementos que englobam texto ou outros elementos
 - Elementos podem conter atributos
 - Todos os documentos XML devem conter a “declaração XML”
 - Define a versão do XML e a codificação de caracteres usada no documento
 - Parâmetros
 - version indica a versão da linguagem (1.0 ou 1.1) - obrigatório
 - encoding indica a codificação de caracteres utilizada no documento – opcional
- <?xml version="1.0" encoding="uft-8"?>*

Sintaxe XML

- **Todos os documentos XML devem conter um, e apenas um, elemento raiz**
`<raiz> ... </raiz>`
- Entre as *tags* do elemento raiz que estarão todas as outras *tags* do seu documento XML
- **Todos os elementos devem ter uma tag de fechamento**
- Existem duas construções válidas:
 - `<teste>Teste 1</teste>` o elemento teste tem conteúdo
 - `<teste />` o elemento teste é vazio

An abstract geometric pattern on the left side of the slide, featuring various colored squares (black, blue, purple, orange, green, pink) and lines of different thicknesses and colors (grey, black, white) arranged in a complex, interconnected manner.

XML

- Elemento
 - Delimitado por marcas (*tags*)
 - Possuem uma marca inicial e uma marca final
 - Tudo o que estiver delimitado por essas marcas faz parte do conteúdo do elemento
 - Ex:<empregado>João</empregado>
- Atributo
 - Podem aparecer dentro da marca **inicial** de um elemento
 - Ex:<empregado **cod="E01"**>João</empregado>

XML

Todos os elementos devem ter uma *tag* de fechamento

- Existem duas construções válidas:
 - `<teste>Teste 1</teste>` o elemento teste tem conteúdo
 - `<teste />` o elemento teste é vazio

XML é *case sensitive*

- `<teste>` é diferente de `<Teste>`

Os elementos XML devem estar corretamente aninhados

- Incorreto: `<i>texto da tag</i>`
- Correto: `<i>texto da tag</i>`

Tags podem ter Atributos, caso tenham devem estar entre aspas.

- Correto: `<aluno ID="2212"></aluno>`
- Incorreto: `<aluno=2212></aluno>`

XML

Os nomes dos elementos devem seguir as regras

- Nomes das tags podem possuir letras, números e outros caracteres.
- Nomes não podem começar com número ou caractere de pontuação.
- Nomes não podem começar com as letras XML e suas variações
- Nomes não podem conter espaços

Sintaxe dos comentários no documento:

`<!--Comentário-->`

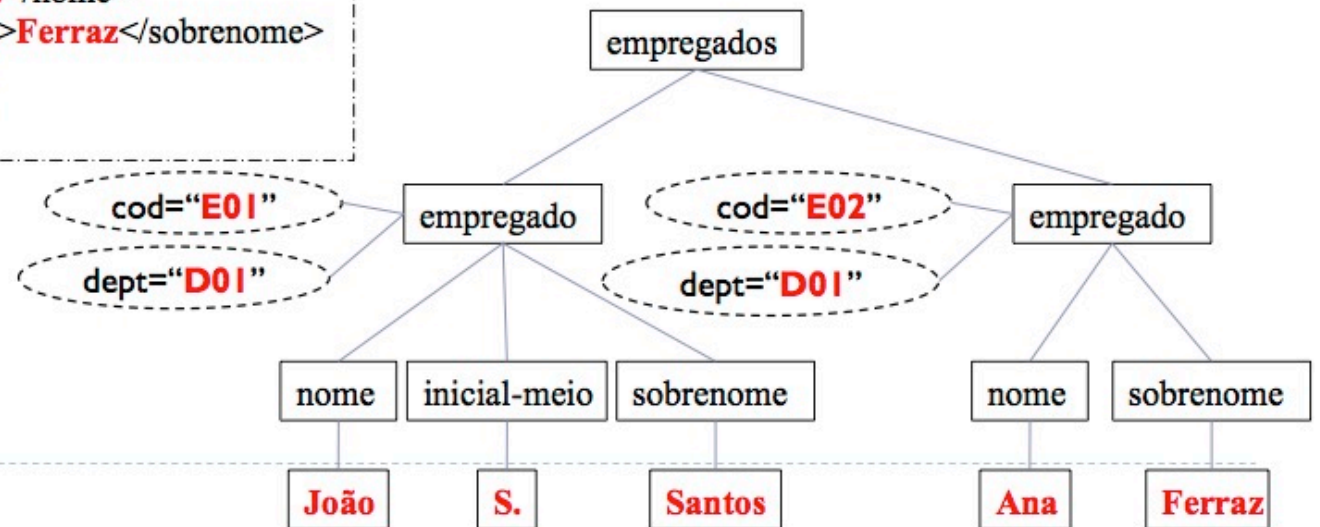
Exemplo de Documento XML

```
<? xml version="1.0" ?>  
<empregados>  
  <empregado cod="E01" dept="D01">  
    <nome>João</nome>  
    <inicial-meio>S.</inicial-meio>  
    <sobrenome>Santos</sobrenome>  
  </empregado>  
  <empregado cod="E02" dept="D01">  
    <nome>Ana</nome>  
    <sobrenome>Ferraz</sobrenome>  
  </empregado>  
</empregados>
```

Documentos XML como árvores

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

- ▶ Elementos, atributos e **texto** → nodos
- ▶ Relações pai/filho → arestas



Documentos XML Bem-Formados

- Por representar uma estrutura de árvore, algumas restrições se aplicam a documentos XML
 - Raiz única (elemento documento ou elemento raiz)
 - Todas as marcas são fechadas
 - Elementos são bem aninhados (marcas fecham na ordem inversa à em que foram abertas)
 - Exemplo de elemento não bem-aninhado
<empregado><nome>João</empregado></nome>
 - Atributos não se repetem no mesmo elemento
 - Nomes de elementos são sensíveis a maiúsculas e minúsculas

Tipos de Elemento

- Composto

- Contém outros (sub)-elementos

<empregado>

<nome>Ana</nome>

<sobrenome>Ferraz</sobrenome>

</empregado>

- Textual

- Contém somente texto

<nome>Ana</nome>

- Misto

- Contém texto e sub-elementos

<endereco> Rua das Flores, 75

<cidade>Rio de Janeiro**</cidade>**

</endereco>

- Vazio

- Elemento sem conteúdo

<engenheiro></engenheiro>

<engenheiro/>

Outras considerações importantes

- Elementos são ordenados
- Atributos **não são** ordenados

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
</empregados>
```



Documentos Diferentes!

Outras considerações importantes

- Elementos são ordenados
- Atributos **não são** ordenados

<pre><? xml version="1.0" ?> <empregados> <empregado dept="D01" cod="E01" > <nome>João</nome> <inicial-meio>S.</inicial-meio> <sobrenome>Santos</sobrenome> </empregado> <empregado cod="E02" dept="D01"> <nome>Ana</nome> <sobrenome>Ferraz</sobrenome> </empregado> </empregados></pre>	<pre><? xml version="1.0" ?> <empregados> <empregado cod="E01" dept="D01"> <nome>João</nome> <inicial-meio>S.</inicial-meio> <sobrenome>Santos</sobrenome> </empregado> <empregado cod="E02" dept="D01"> <nome>Ana</nome> <sobrenome>Ferraz</sobrenome> </empregado> </empregados></pre>
---	--

Documentos Iguais

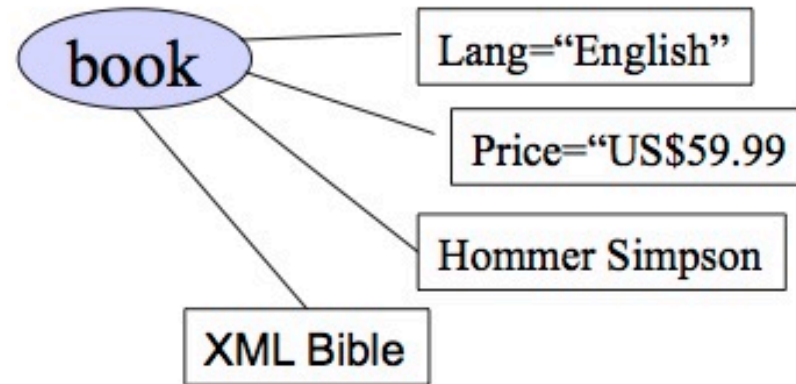
Elementos x Atributos

- Não há regras
- Atributos apresentam algumas restrições
 - Não são extensíveis
 - Não permitem múltiplos valores
 - Não descrevem estruturas
- Recomendação: em geral, **preferir elementos, e usar atributos para informações secundárias**
- Metadados (dados sobre os dados) devem ser representados como atributos

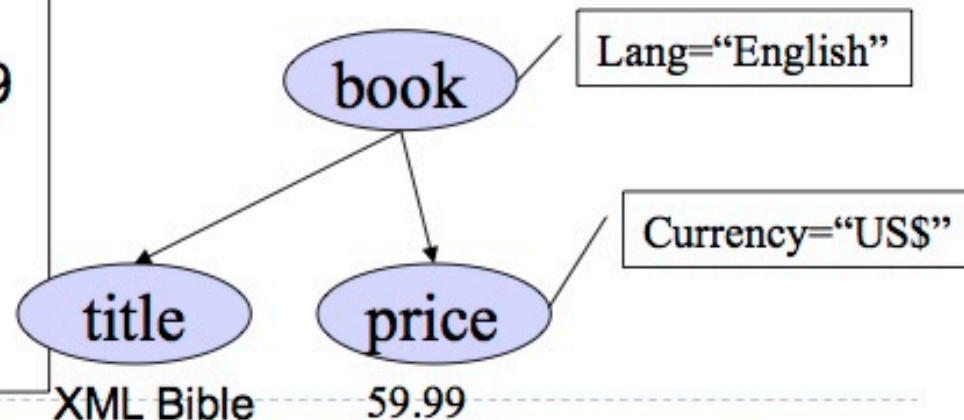
Ex: <price currency="US">59.99</price>

Elementos e Atributos

```
<book lang="English" price="US  
$59.99"  
title="XML Bible"  
author="Hommer Simpson">  
...  
</book>
```



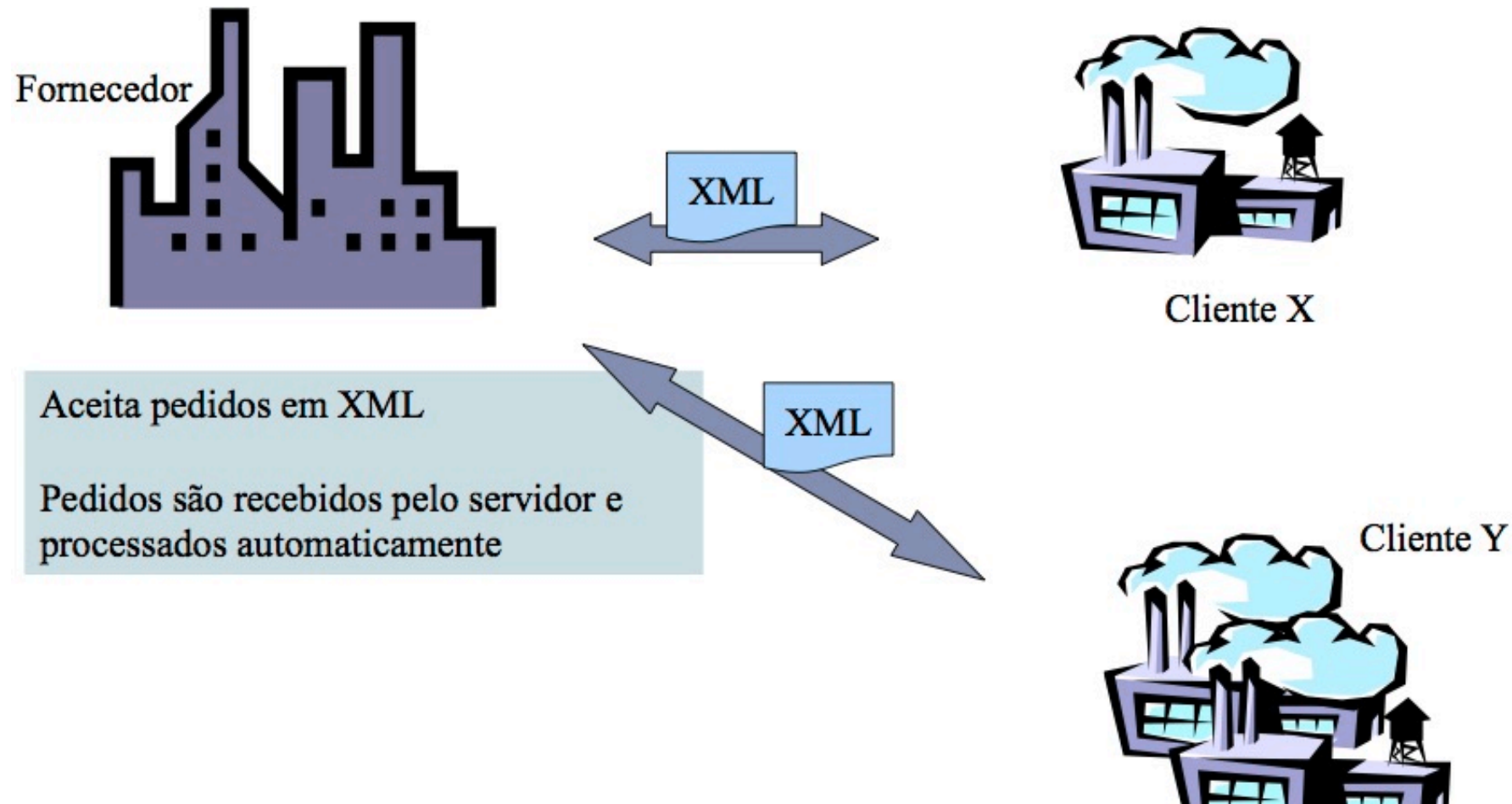
```
<book lang="English">  
  <price currency="US$"> 59.99  
  </price>  
  <title>XML Bible </title>  
  ...  
</book>
```



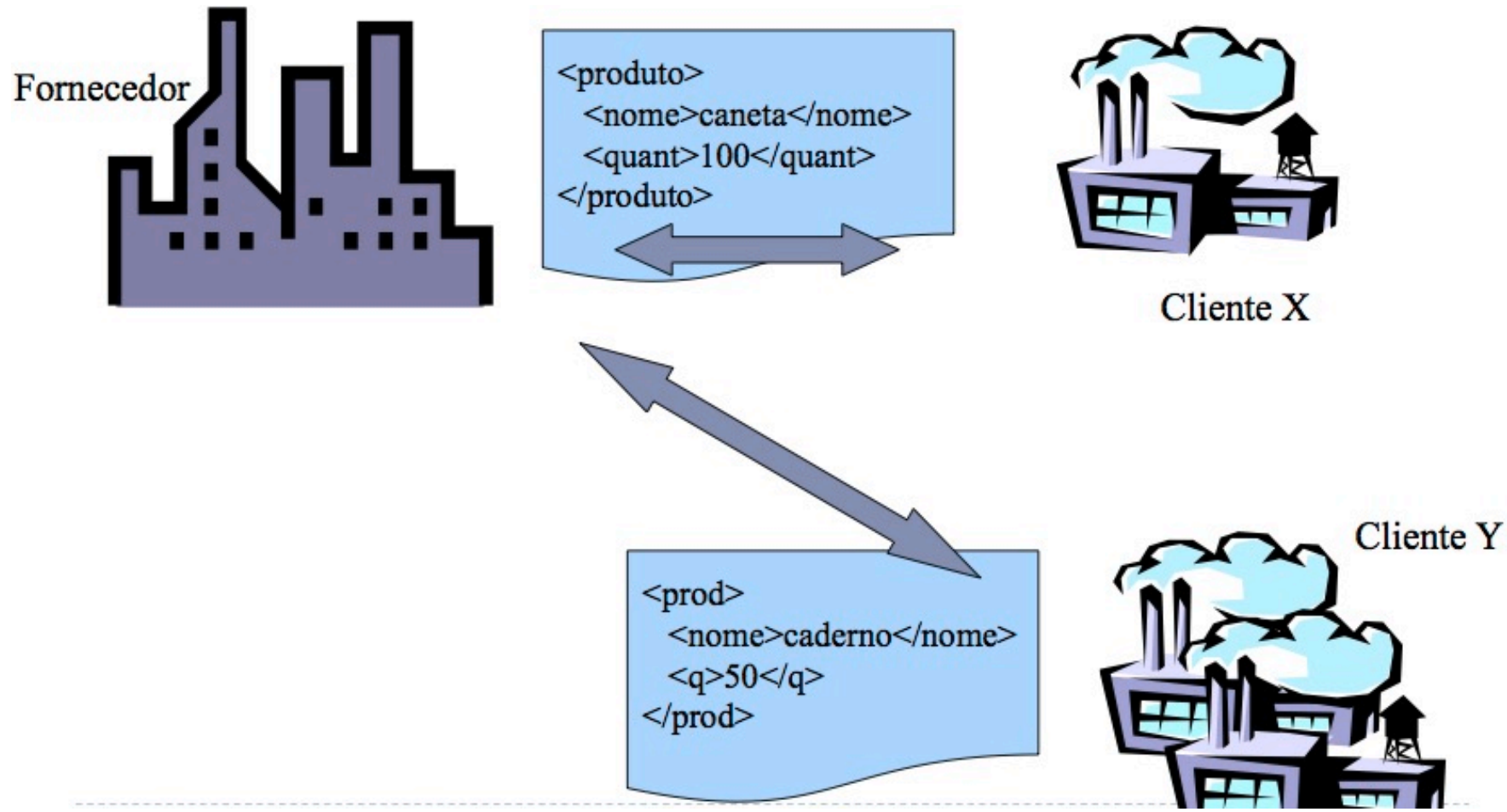
Interoperabilidade

- Chave do sucesso de XML:
 - Usuários podem definir suas próprias marcas
 - Durante muito tempo, foi a solução para interoperabilidade. Progressivamente, tem sido substituída por tecnologias menos prolixas, como JSON.
- E a interoperabilidade, como fica
 - Vamos considerar um exemplo prático

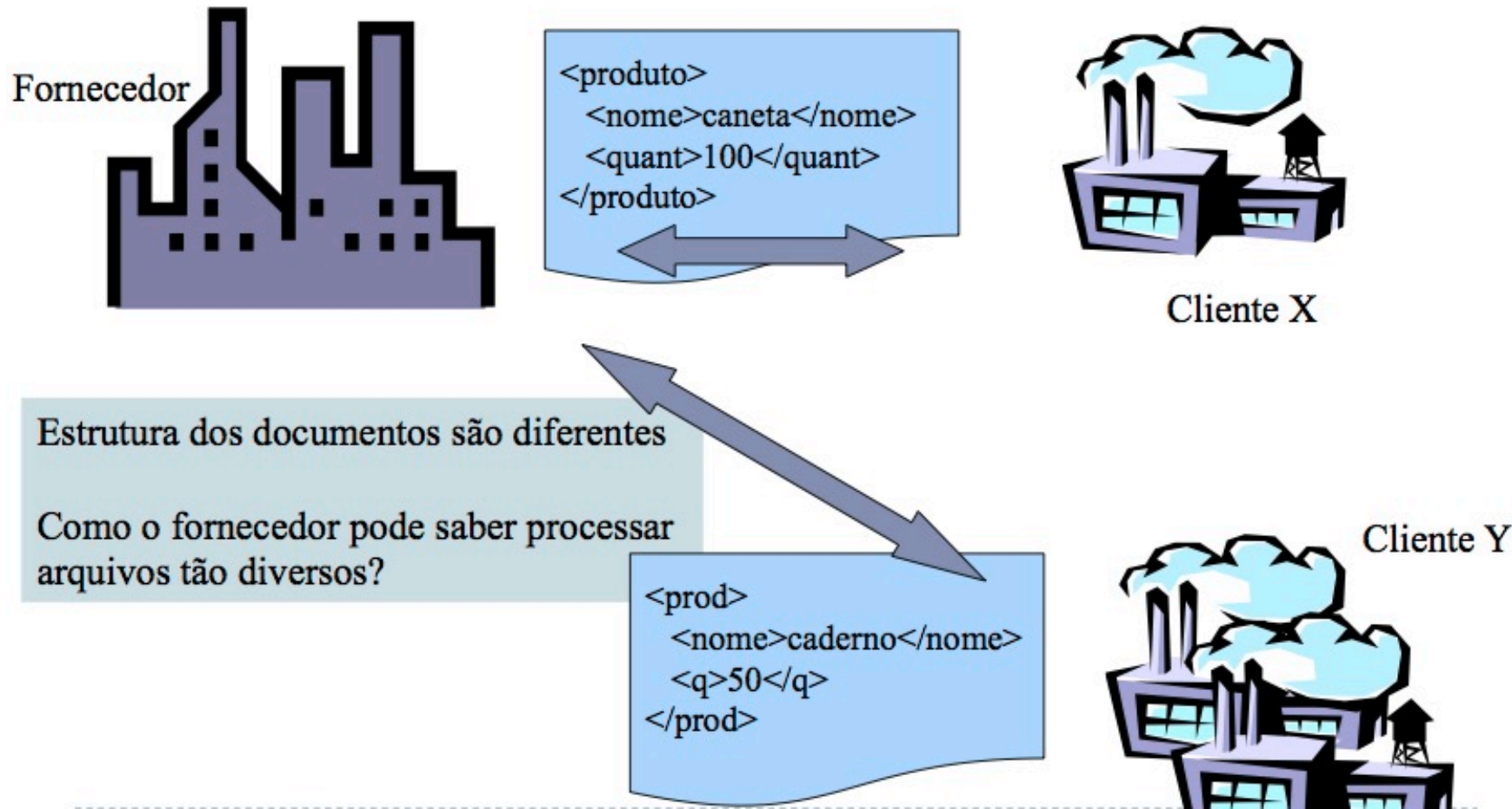
Exemplo prático: interoperabilidade



Exemplo prático: interoperabilidade



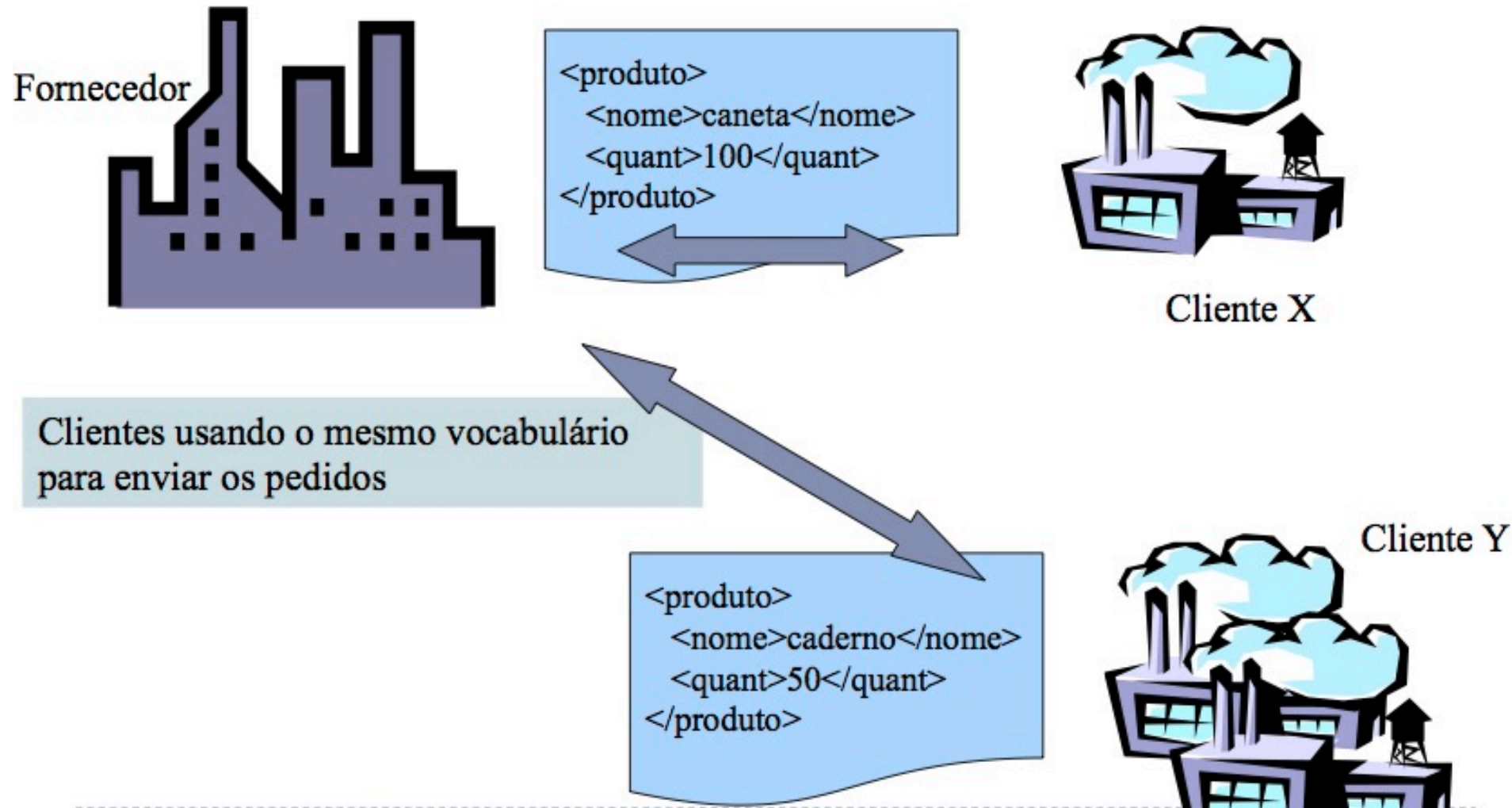
Exemplo prático: interoperabilidade



Solução

- Pode-se definir um vocabulário usando uma linguagem de esquemas para XML (DTD ou XML Schema)
 - Fornecedor define o vocabulário (estrutura, nomes das marcas, tipos de dados)
 - Cliente usa o vocabulário para enviar os pedidos

Interoperabilidade



Exercícios

- Escrever um documento XML para representar uma receituário médico
 - Lembre-se: é importante pensar em como estes documentos serão **estruturados**, e não em como serão **apresentados**



Ana Maria Marina , 7 anos

Uso interno


Xarope SemTosse

1 colher 3x ao dia

Uso Externo

Gyellow

aplicar no braço 1x ao dia ao deitar


Dr. Juca

20/10/2001

Exercícios

- Faça o seu currículo em XML
- Informações obrigatórias:
 - Dados pessoais
 - Formação
 - Idiomas
 - Cursos adicionais

Leituras complementares

Moro, M., Braganholo, V., “Desmistificando XML: da Pesquisa à Prática Industrial”, disponível em:
<http://www2.ic.uff.br/~vanessa/papers/moro2009-jai.pdf>

“Extensible Markup Language (XML)”, disponível em:
<http://www.w3.org/XML/>

“XML Tutorial” disponível em:
<http://www.w3schools.com/xml/>