Páginas dinâmicas

Eduardo Ferreira dos Santos

Ciência da Computação
Centro Universitário de Brasília – UniCEUB

Agosto, 2016

Sumário

Enviando parâmetros

- Para que o conteúdo seja dinâmico os sistemas Web precisam suportar algum tipo de interação;
- Para o usuário interagir é necessário permitir o envio de parâmetros para o servidor Web.

```
http://localhost/hello.php?name=Manuel&title=Dr
```

- O servidor Web envia os parâmetros para o interpretador da linguagem, que tem o papel de armazenar e processar os dados.
- Veja o exemplo para a linguagem PHP:

$HTTP_Request GET /hello.php?name=Manuel&title=Dr HTTP/1.1
$QUERY_STRING name=Manuel&title=Dr
$_GET["name"] Manuel
$_GET["title"] Dr

Sistemas de templates

- Gerando HTML:
  - Nenhuma da soluções utilizadas até agora permitem a separação entre HTML e código;
  - A lógica da aplicação está misturada ao HTML.
- A mistura entre código HTML e aplicação é tão comum que, para mitigar o problema, foram desenvolvidos os sistemas de template (*template systems*).
  - Template: arquivo HTML que contém *tags* (ou marcações) indicando o conteúdo que deve ser substituído;
  - Normalmente existe um *gateway CGI* que substitui as *tags* pelo conteúdo:
    - Todas as definições de *layout* e estilo estão no template;
    - Todo o conteúdo e processamento está no CGI;
    - Separação clara entre conteúdo e lógica.
- Diversas implementações em diferentes linguagens.

Engenharia de templates

- Algumas vezes, a lógica da aplicação está embutida no template;
- Questões relativas à Engenharia de Software:
  - O que é separação de contexto (*separation of concerns*)?
  - O que é decisão de projeto (*design choice*)?
  - O que são dependências?

Template Engines

As *template engines* tentam resolver o problema das dependências tornando o processamento da aplicação independente do layout.

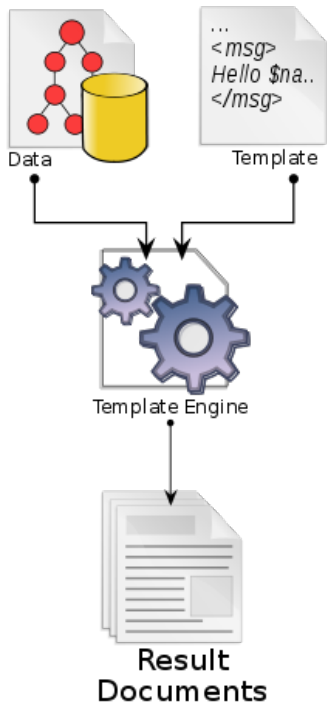

Figura 1.1: Diagrama para template Engines [1]

[1] Fonte: https://pt.wikipedia.org/wiki/Ficheiro:TempEngGen015.svg

Exemplo PHP

Listing 1: Exemplo de engine em PHP [Pérez-Quiñones, 2011]

```
function generate_page($file, $variables)
{
  // read the file
  // search for %symbol% and replace with $variables['symbol']
}
```

Listing 2: Arquivo de template "chamando" as variáveis [Pérez-Quiñones, 2011]

```
<html>
<body>
  Hello %NAME%
</body>
</html>
```

Listing 3: Chamada da engine [Pérez-Quiñones, 2011]

```
$names = array("NAME"=>"Joe", "LAST"=>"Smith");
generate_page("temp.html", $names);
```

UniCEUB
Centro Universitário de Brasília

Limitações

- Qual o contexto em que nossa engine de template falharia?
  - Dados estruturados;
  - Arrays;
  - Formatação condicional.

Desafios [Andersson et al., 2006]

- Alguns desafios inerentes às aplicações para a Web:
  - Sessões compartilhadas e portáveis;
  - Compartilhamento de músicas;
  - Compartilhamento de fotos;
  - Compartilhamento de documentos.
- O que normalmente não é possível compartilhar é a sua experiência utilizando a Internet;
- Utilização de dispositivos móveis;
- Computação ubíqua.

O computador pode me ajudar a ser tudo o que eu preciso?

Exemplo de desafio

E se pudéssemos construir uma espécie de nutricionista pessoal online?
Vamos chamá-lo de Doutor [Andersson et al., 2006]:

09h00 Você liga para o Doutor do seu celular:

Doutor Qual foi o seu café?

Você Um copo de suco, duas fatias de pão e uma xícara de café com leite.

Doutor Qual o tamanho do copo de suco?

Você Médio

10h45 Você come um bolo trazido por um companheiro de trabalho a avisa o Doutor. Ele informa o que você já comeu e quando de caloria ainda pode consumir.

Exemplo de desafio (cont.)

13h30 Você come um sanduíche e uma Coca Zero. Você acessa o Doutor e reporta o consumo.

16h00 Você faz uma pausa para o lanche e come salgadinho e refrigerante. Você decide acessar o celular e informar o Doutor do seu consumo.

19h00 Você malha por 45 minutos e informa o Doutor pelo celular.

20h30 Você chega em casa, janta e o Doutor fornece o relatório diário de quanto você consumiu e se está perto ou longe da meta. Você está ficando gordo.

Modelagem do Doutor

- Alguns componentes identificados no exemplo:
  - Um modelo adaptativo de usuário;
  - Banco de dados de calorias para diferentes comidas;
  - Algum conhecimento sobre nutrição;
  - Total de calorias consumidas x Peso estimado por idade.
  - Interface Web para navegadores;
  - Interface Web para dispositivos móveis;
  - Um sistema de reconhecimento de voz?
- Objetivo principal da Web: conectar-se com outras pessoas;
- Qual a importância da conexão com outras pessoas percebida na Web atual?
- Como você arrumou seu último emprego/estágio?

Limitações do HTTP

- Protocolo HTTP Não armazena estado (stateless) e anônimo;
- Quando a conexão é finalizada, toda a comunicação é finalizada;
- Como criar uma experiência contínua em cima dessa limitação do HTTP?

Limitações do HTTP

- Protocolo HTTP Não armazena estado (stateless) e anônimo;
- Quando a conexão é finalizada, toda a comunicação é finalizada;
- Como criar uma experiência contínua em cima dessa limitação do HTTP?
- Como armazenar o estado entre diferentes requisições?

Limitações do HTTP

- Protocolo HTTP Não armazena estado (stateless) e anônimo;
- Quando a conexão é finalizada, toda a comunicação é finalizada;
- Como criar uma experiência contínua em cima dessa limitação do HTTP?
- Como armazenar o estado entre diferentes requisições?
  - Arquivo de log no servidor?

Limitações do HTTP

- Protocolo HTTP Não armazena estado (stateless) e anônimo;
- Quando a conexão é finalizada, toda a comunicação é finalizada;
- Como criar uma experiência contínua em cima dessa limitação do HTTP?
- Como armazenar o estado entre diferentes requisições?
  - Arquivo de log no servidor?
  - HTTP não sabe quem está conectando.
  - No mesmo endereço IP podem estar potecialmente milhares de usuários.

Enviando parâmetros

- Para que o conteúdo seja dinâmico os sistemas Web precisam suportar algum tipo de interação;
- Para o usuário interagir é necessário permitir o envio de parâmetros para o servidor Web.

```
http://localhost/hello.php?name=Manuel&title=Dr
```

- O servidor Web envia os parâmetros para o interpretador da linguagem, que tem o papel de armazenar e processar os dados.
- Veja o exemplo para a linguagem PHP:

$HTTP_Request GET /hello.php?name=Manuel&title=Dr HTTP/1.1
$QUERY_STRING name=Manuel&title=Dr
$_GET["name"] Manuel
$_GET["title"] Dr

Utilização de URI's

- Suponha que o usuário acesse o seguinte endereço HTTP:

```
http://www.amazon.com/exec/obidos/ASIN/1588750019/
```

- O número 1588750019 representa o *International Standard Book Number* – ISBN – que identifica unicamente o livro;
- O servidor da Amazon altera a URL para adicionar o identificador da sessão:

```
http://www.amazon.com/exec/obidos/ASIN/1588750019/
                    103-9609966-7089404
```

- Ao continuar acessando diferentes livros, é possível perceber que todos possuirão o mesmo identificador de sessão.

UniCEUB
Centro Universitário de Brasília

Cookies

- Ao invés de brincar com hyperlinks, utilizados armazenamento no browser;
- Uma das formas de armazenar dados é utilizar informações conhecidas como cookies[2];
- Exemplo: ao enviar a primeira requisição, o servidor escreve:

```
1
Set-Cookie: cart_contents=1588750019; path=/
```

- Enquanto o navegador estiver aberto, a seguinte informação é adicionada no header:

```
2
Cookie: cart_contents=1588750019
```

- Quais os problemas dessa abordagem?

[2] *Persistent Client State HTTP Cookies* – Especificação preliminar
`http://wp.netscape.com/newsref/std/cookie_spec.html`

Armazenamento no servidor

- Qual a melhor maneira de armazenar a informação?
  - Planilhas eletrônicas?
  - Arquivos de texto?
- Problema da concorrência;
- Utilização de bancos de dados.

RDBMS

- Vantagens do banco de dados relacional:
  - Resolve o problema da concorrência;
  - Utiliza linguagem declartiva SQL;
  - Isola partes importantes de erros do programador:
    - Chaves primárias;
    - Não permite alterações não estruturadas;
    - Três operações: INSERT, UPDATE, SELECT
  - Performance.
- Desvantagens:
  - Difícil manutenção;
  - Trocar de banco pode ser um grande desafio;
  - O problema das consultas.

Esquema relacional

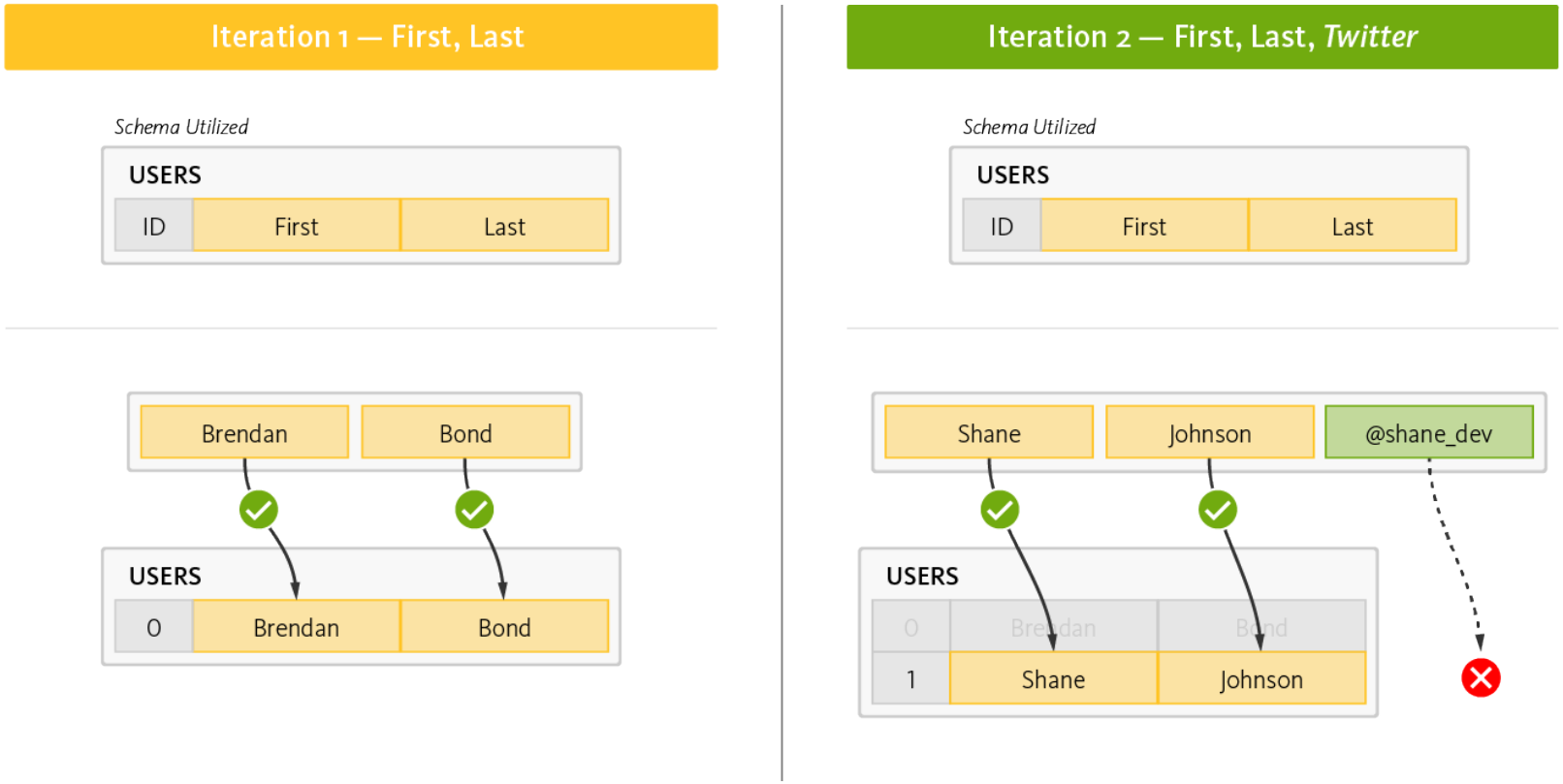

Figura 2.1: No esquema relacional, não é fácil adicionar um novo atributo quando necessário [4]

[3] Fonte: `http://www.couchbase.com/binaries/content/gallery/website/`

Flexibilidade

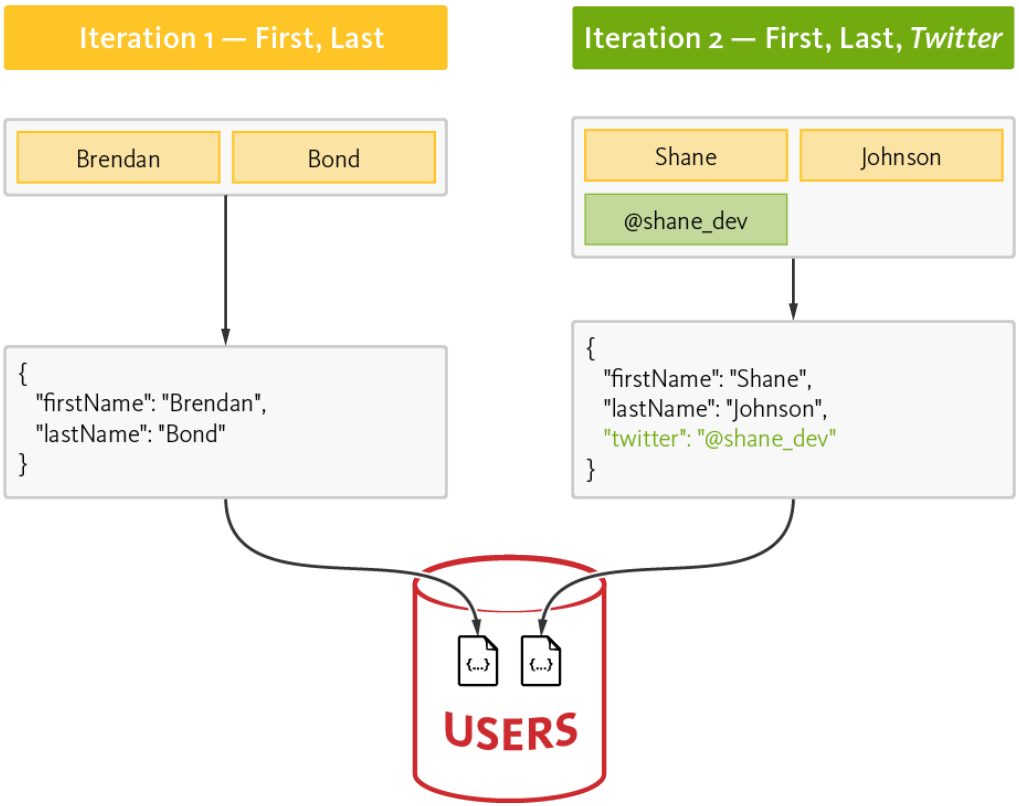

Figura 2.2: No esquema não relacional a alteração do esquema é dinâmica [5]

[5] Fonte: http://www.couchbase.com/binaries/content/gallery/website/common/why-nosql/figure2_flexible_schema.png

JOIN relacional

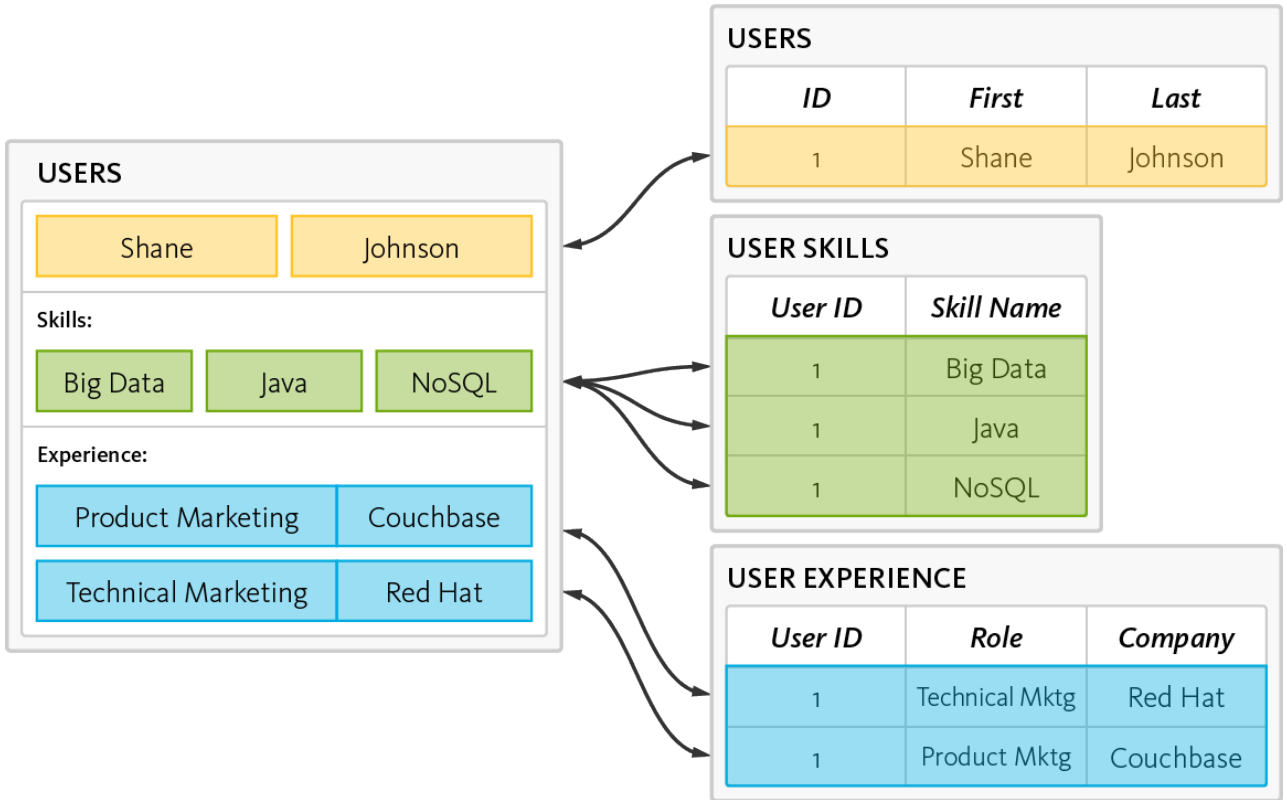

Figura 2.3: O objeto é "dividido" em diferentes tabelas [6]

[6] Fonte: http://www.couchbase.com/binaries/content/gallery/website/common/why-nosql/figure3_object_to_relational.png

Objeto íntegro

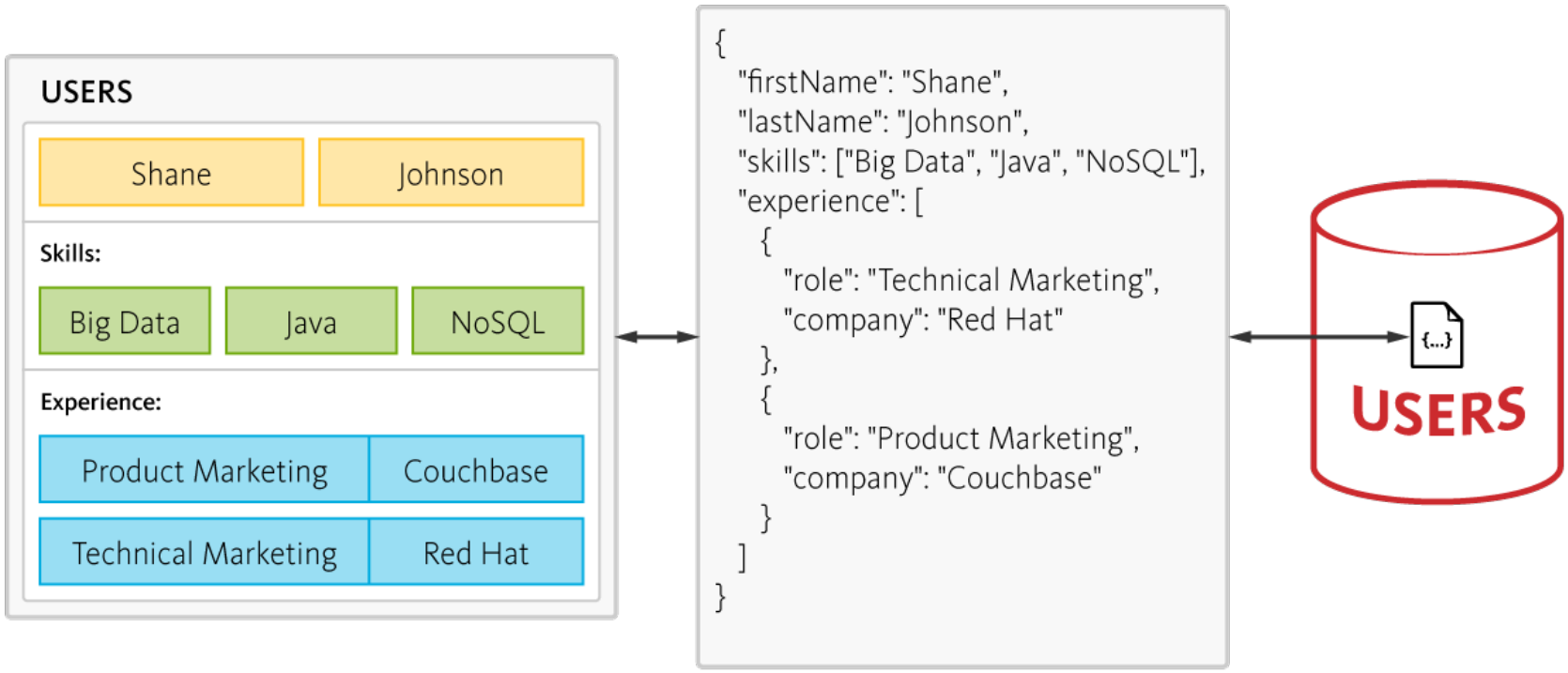

Figura 2.4: Sempre o objeto inteiro é carregado [7]

[7] Fonte: http://www.couchbase.com/binaries/content/gallery/website/common/why-nosql/figure5_object_to_document.png

Desafios do desenvolvimento

- Há pouco tempo, o desenvolvimento era feito por diferentes atores:
  - Programador;
  - Designer;
  - DBA.
- Nova figura: full stack developer.
- Você é capaz de administrador quantos bancos de dados?
- Na nova Web, tudo é serviço!
- Importância da utilização de padrões e frameworks.

OBRIGADO!!!
PERGUNTAS???


Andersson, E. A., Greenspun, P., Grumet, A., Eve Andersson, P. G., et al. (2006).
Software engineering for Internet applications.
Number Sirsi) i9780262511919.


Pérez-Quiñones, M. A. (2011).
Internet software.
Disponível em `http://dopey.cs.vt.edu/courses/cs4244-S11/`
Acessado em 03/08/2016.