

Sistemas distribuídos na web

A world wide web (WWW), em seu estágio inicial, era um sistema global para acessar documentos pela internet, sendo composta de inúmeros clientes e servidores. Basicamente, os servidores mantinham os documentos, e os clientes proporcionavam aos usuários do sistema uma interface simples para acessar e mostrar esses documentos. Desde então, tem se aprimorado a capacidade da web com padronização de protocolos e interoperabilidade, o que a tornou mais do que apenas um sistema simples baseado em documentos, fazendo surgir a utilização de serviços, dos quais qualquer usuário pode usufruir.

Os serviços web são uma coleção de protocolos e padrões abertos, tais como TCP/IP, HTTP, Java, HTML e XML, utilizados para trocar dados entre aplicativos ou sistemas pela internet independentemente da sua plataforma ou do seu fabricante.

Para que possamos compreender o funcionamento dos sistemas distribuídos na web, o conteúdo deste capítulo foi dividido em dois grandes itens: modelos de sistemas distribuídos na web e implementação de serviços web.

1 Modelos de sistemas distribuídos na web

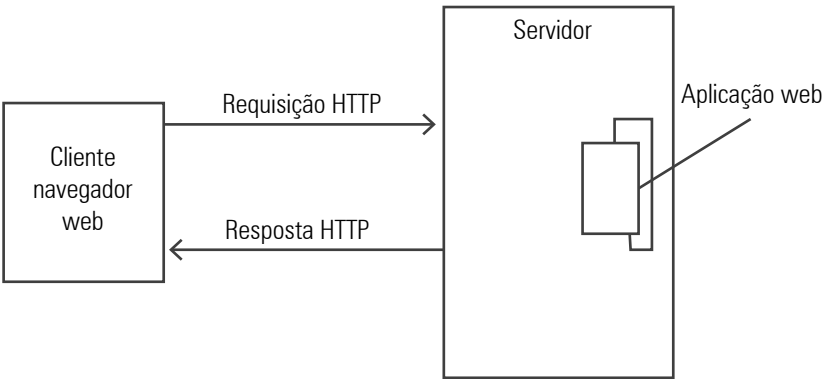
Os sistemas distribuídos baseados na web tornaram as aplicações de rede populares entre usuários finais através da utilização de documentos. Podemos entender um documento web como uma forma de trocar informações, o que se aproxima do modo como as pessoas costumam se comunicar. Em sua aplicabilidade mais simples, os usuários finais enxergam como uma arquitetura cliente-servidor, na qual um documento é buscado na internet. No seu uso mais complexo, com arquiteturas multicamadas, os componentes finais são responsáveis por gerar páginas web com respostas ao cliente, o que traz à tona o conceito de serviços web (TANENBAUM; VAN STEEN, 2008).

Um serviço web (web service) fornece uma interface de serviço que permite aos clientes interagirem com servidores de uma maneira mais geral do que acontece com os navegadores web. Os clientes acessam as operações na interface de um serviço web por meio de requisições e respostas formatadas em XML e, normalmente, transmitidas por HTTP. Os serviços web podem ser acessados de uma maneira mais *ad hoc* do que os serviços baseados em CORBA, permitindo que eles sejam mais facilmente usados em aplicações de internet. (COULOURIS; DOLLIMORE; KINDBERG, 2007, p. 672)

Os sistemas distribuídos baseados na web tradicional utilizam o protocolo HTTP (hypertext transfer protocol) como padrão para troca de mensagens. Já o SOAP (simple object access protocol) é tido como o protocolo padrão para troca de mensagens quando se consideram serviços web, sendo implementado por meio do HTTP. O protocolo HTTP foi

projetado para atender às demandas da arquitetura cliente-servidor, sendo orientado à transferência em ambas as direções. A figura 1 exemplifica o processo de comunicação HTTP; o cliente requisita ao servidor que uma operação seja executada, e este retorna com a resposta ao cliente.

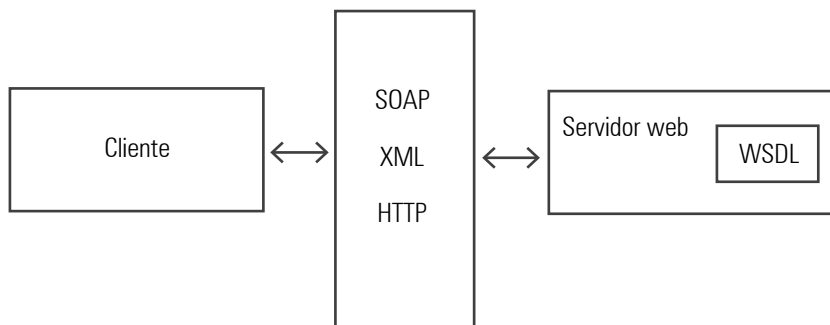
Figura 1 – Comunicação HTTP



Já o protocolo SOAP tem como objetivo fornecer um meio simples de comunicação, permitindo que diferentes partes, que se conheçam muito pouco, consigam trocar mensagens. As mensagens SOAP são baseadas em XML (extensible markup language). O XML é uma linguagem de metamarcação que inclui a definição dos elementos usados para descrever o documento. O WSDL (web service description language) é uma linguagem de definição baseada em XML, sendo usada para descrever a funcionalidade de um serviço web baseado em SOAP, ou seja, além do descritivo do serviço, ela contém a forma de acesso e as operações disponíveis para utilização (BASHAM; SIERRA; BATES, 2005).

Na figura 2, é demonstrada a forma de comunicação do cliente com o servidor web através do SOAP.

Figura 2 – Comunicação SOAP



Complementarmente, Tanenbaum e Van Steen (2008) enfatizam que, como os serviços web utilizam a internet como meio de comunicação, é necessário se preocupar com o desempenho, a tolerância, a falha e a segurança. Para atingir esse objetivo, é necessário o uso de cache, replicação e técnicas padronizadas já aplicadas em outros sistemas.



PARA SABER MAIS

A W3C (World Wide Web Consortium) é uma organização que padroniza a web e tem como principal objetivo potencializar o seu uso através do desenvolvimento de protocolos e fóruns abertos para a comunidade em geral. Em seu site oficial, é possível visualizar o descritivo sobre o SOAP, sua aplicabilidade e exemplos. Para verificar o conteúdo, busque no site por “simple object access protocol”.

2 Implementação de serviços web

Para compreendermos como funciona o serviço web, nada melhor do que uma demonstração prática. Para alcançar esse objetivo, este tópico é dividido na implementação de dois exemplos. O primeiro exemplo

é uma demonstração simples de acesso cliente-servidor utilizando o protocolo HTTP, e o segundo exemplo é a utilização do SOAP com Python.

No primeiro exemplo, para a comunicação entre um cliente e um servidor, é necessário que exista um servidor web. Para demonstrar a comunicação, será utilizado o servidor web embutido junto ao Python, e o utilizaremos para mostrar uma página web para o usuário final.

Para ativá-lo é bem simples: temos que entrar no diretório em que está a página web e ativá-lo por meio do comando a seguir. Caso não haja um arquivo chamado "index.html", será mostrado o diretório corrente.

```
python -m SimpleHTTPServer
```

O comando informado ativou um servidor web na porta 8000. Ao acessar o IP do servidor, aparecerá o conteúdo conforme mostrado na figura 3, fazendo o uso de requisição HTTP e resposta HTTP, como visto no capítulo anterior.

Figura 3 – Resposta de acesso HTTP



No segundo exemplo, utilizaremos a biblioteca SOAPpy para demonstrar os recursos do SOAP para Python. Essa biblioteca fornece um recurso simples com interação dinâmica entre clientes e servidores. O exemplo escolhido é o programa disponível no site oficial do SOAPpy, que mostra a mensagem "Hello World" na tela do cliente.

Para isso, é necessário criar o script a seguir no lado do servidor.

```
import SOAPpy
def hello():
    return "Hello World"
server = SOAPpy.SOAPServer(("localhost", 8080))
server.registerFunction(hello)
server.serve_forever()
```

Posteriormente, será necessário criar o script a seguir no lado do cliente para que seja possível a comunicação com o servidor, bem como a interação entre ambos.

```
import SOAPpy
server = SOAPpy.SOAPProxy("http://localhost:8080/")
print server.hello()
```

Ao invocar o método, a mensagem "Hello World" será apresentada na tela do usuário com a comunicação através do SOAP no lado do servidor utilizando a biblioteca SOAPpy, conforme mostra a figura 4.

Figura 4 – Resposta de acesso SOAP

```
estudante@estudante:~/soap$ python Client_SOAP.py
Hello World
Estudante$estudante:~/soap$
```



PARA SABER MAIS

A comunidade Python brasileira mantém uma wiki para ajudar os desenvolvedores com as principais dúvidas e divulgar a linguagem de programação Python de maneira geral. O atrativo dessa wiki são os tutoriais com exemplos didáticos e práticos dos principais assuntos de desenvolvimento Python. Ao acessar a wiki da Python Brasil, procure pelo termo “web service” para ter acesso a um exemplo do uso de SOAP na criação de um serviço web.

Considerações finais

Neste capítulo, aprendemos a importância da internet para os sistemas distribuídos e acompanhamos a sua evolução ao longo dos anos. Foi possível constatar a necessidade da padronização de protocolos e de interoperabilidade, independentemente da plataforma ou do fabricante, associados à utilização de serviços web. Tivemos a oportunidade de conhecer também os modelos de sistemas distribuídos para a web e de visualizar como funciona uma simples requisição HTTP e um serviço web baseado em SOAP.

Para demonstrar os conceitos e sua aplicabilidade, fechamos o capítulo com a implementação de um serviço web em Python.

Referências

BASHAM, Bryan; SIERRA, Kathy; BATES, Bert. **Use a cabeça!**: servlets & JSP. Rio de Janeiro: Alta Books, 2005.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas distribuídos**: conceitos e projeto. 4. ed. Porto Alegre: Bookman, 2007.

TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Sistemas distribuídos**: princípios e paradigmas. São Paulo: Prentice Hall, 2008.