

# **Desenvolvimento baseado em Webservices**

**Prof. Emílio Dias**  
emiliodias@gmail.com  
<http://www.github.com/emiliodias>

# Conteúdo da disciplina

- Introdução a comunicação inter-processos
- Discussões gerais sobre a Web
- Introdução a Webservices
- Webservices “RESTful”
- Segurança de Webservices com OAuth2
- Webservices SOAP
- Utilização do framework Spring

# Avaliação

- Presença e participação em sala de aula.
  - **Aula 1:** Apresentação da disciplina, e introduções, os alunos não serão avaliados.
  - **Aula 2 à 5:** 20 pontos por presença + participação

**Participação** = Interesse no conteúdo apresentado, questionamentos, execução de exercícios quando propostos.

# **Comunicação inter-processos**

**Motivação para o estudo,  
entendimento de comunicação  
inter-processos e relação com  
Webservices.**

# Comunicação inter-processos

- Passagem de Mensagem pode ser suportada por duas operações de comunicação (send e receive);
- A comunicação se dá pelo envio da mensagem (seqüência de bytes) do emissor para o receptor;
- Essa troca de mensagem pode envolver a sincronização dos processos envolvidos.

# **Comunicação inter-processos**

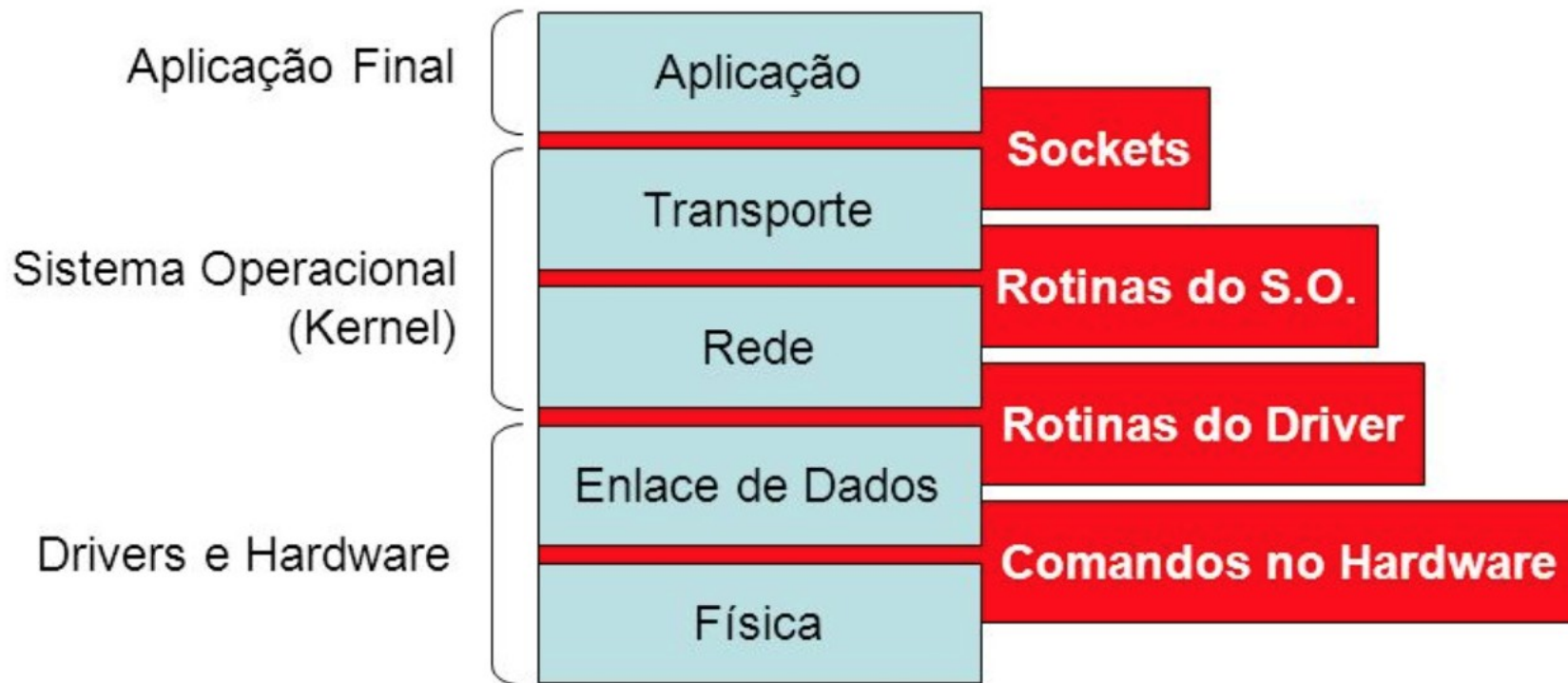
## **Comunicação síncrona**

Ambos, emissor e receptor sincronizam-se a cada mensagem. Assim tanto o send como o receive são operações bloqueantes.

## **Comunicação assíncrona**

Após o send o processo pode continuar executando. A operação receive pode ser bloqueante ou não.

# Comunicação inter-processos



**Arquitetura em camadas TCP/IP**



# Comunicação inter-processos

## Sockets

- Abstração que disponibiliza um ponto final para comunicação.
- A comunicação inter-processos se dá através da transmissão de mensagem entre um socket em um processo e outro socket em outro processo.
- Para o processo receptor o socket deve estar ligado ao endereço internet local e a uma porta local, no computador onde ele executa.
- Cada socket está associado com um protocolo particular (TCP e UDP)

# Comunicação inter-processos

## Modelos de comunicação

- Request/Response
- RMI
- **RPC → Modelo utilizado pelo protocolo SOAP**
- **REST → Modelo utilizado na criação de Webservices RESTful**
- Outros...

# **Discussões gerais sobre a Web**

**O que é a Web?**

# Discussões gerais sobre a Web

## O que é a Web?

- Algo que tem uma URL...
- Plataforma Web...
- Tudo que funciona no Browser...
- Quem sabe???

**Fonte:** [https://www.mnot.net/blog/2014/12/04/what\\_is\\_the\\_web](https://www.mnot.net/blog/2014/12/04/what_is_the_web)

**Tradução:** <https://www.infoq.com/br/news/2015/03/o-que-e-web>



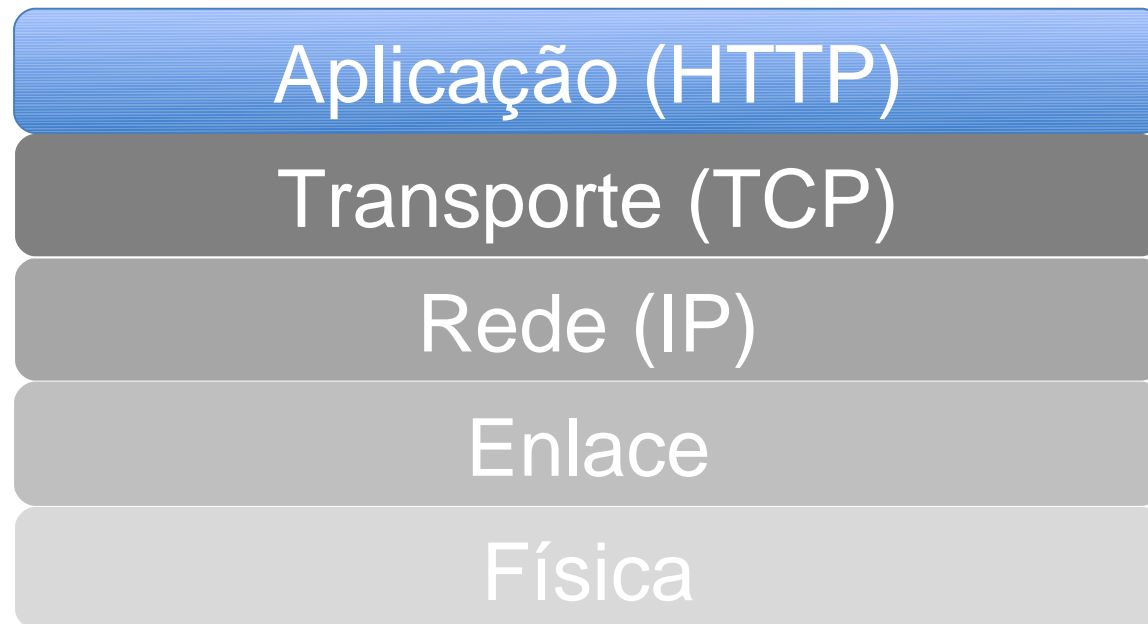
Hi, I'm Mark Nottingham. I currently co-chair the IETF [HTTP](#) and [QUIC](#) Working Groups, and am a member of the [Internet Architecture Board](#). I usually write here about the Web, protocol design, HTTP, and caching.

# Introdução a Webservices

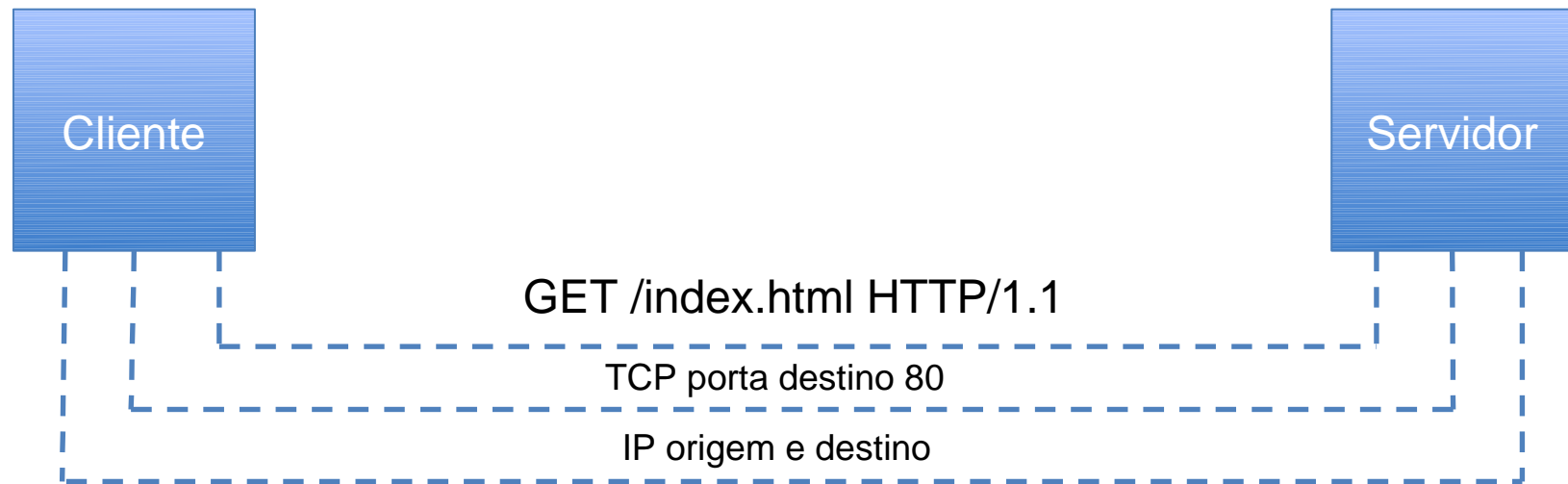
## O que são?

Tecnologia de comunicação inter-processos que utilizam como mecanismo básico de comunicação as tecnologias presentes na Web, sendo a principal o protocolo HTTP.

# O protocolo HTTP



# O protocolo HTTP



# O protocolo HTTP: URI

URI – Universal Resource Identifier

URI

<http://www.unitri.edu.br/posjava>

URL URN



# O protocolo HTTP: Métodos

- **GET**

- Utilizado quando existe a necessidade de se obter um recurso.

- **POST**

- Utilizamos o método POST quando desejamos criar um recurso

- **PUT**

- Semelhante ao método POST, esse método nos permite a atualização de um recurso já existente

- **DELETE**

- Como o próprio nome sugere, o DELETE é utilizado quando existe a necessidade de remoção de um recurso

# O protocolo HTTP: Respostas

- **1xx**
  - Informações gerais
- **2xx**
  - Sucesso
- **3xx**
  - Redirecionamento
- **4xx**
  - Erro no cliente
- **5xx**
  - Erro no servidor

# O protocolo HTTP

**Outras importantes características do protocolo HTTP**

- **Media-type (Content-type)**
- **Caching**

# REST

UNIVERSITY OF CALIFORNIA,  
IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Roy Thomas Fielding

2000

Roy Fielding

American computer scientist



Roy Thomas Fielding is an American computer scientist, one of the principal authors of the HTTP specification and the originator of the Representational State Transfer architectural style. [Wikipedia](#)

**Born:** 1965 (age 53 years), [Laguna Beach, California, United States](#)

**Education:** [University of California, Irvine](#) (2000)

**Field:** [Computer Science](#)

**Books:** [Dean and the Dark Angels](#), [Dyslexia](#)

**Organization founded:** [Apache Software Foundation](#)

# REST

## Constraints

- Cliente/Servidor
- Stateless
- Cache
- Interface Uniforme
- Sistema em camadas
- Código sob demanda

# Webservices RESTful

O que são?

Webservices que utilizam como base para sua arquitetura as constraints definidas pelo modelo arquitetural REST.

**Obrigado!**