

LISTA DE EXERCÍCIOS – CAMADA DE APLICAÇÃO

- 1) Para uma sessão de comunicação entre um par de processos, qual processo é o cliente e qual é o servidor?
- 2) Que informação é usada por um processo rodando em um *host* para identificar um processo rodando em outro *host*?
- 3) Conceitualmente, qual a diferença entre arquitetura de rede e arquitetura de aplicação de rede? Descreva três arquiteturas de aplicação de rede distintas, dando exemplos de aplicações que se enquadram em cada uma delas.
- 4) Aplicações diferentes têm necessidades de serviço diferentes, e estas necessidades costumam ser classificadas em três dimensões distintas. Que dimensões são essas? Forneça exemplos de aplicações de rede que você conhece e identifique as necessidades de serviço das mesmas.
- 5) Quais os protocolos de transporte da Internet e quais os serviços oferecidos por estes protocolos para a camada de aplicação?
- 6) Qual a função dos *cookies*? Dê um exemplo de uma aplicação de rede que requer o uso de *cookies*, explicando como a aplicação tira proveito deste recurso.
- 7) Descreva os diferentes tipos de serviço fornecidos pelo DNS.
- 8) Explique o processo de resolução do endereço <http://www.fib.br> via DNS. Na sua resposta, comente sobre os papéis do servidores local, raiz, de domínio de alto nível e oficial.
- 9) Um cliente TCP pode executar antes que um servidor TCP? E se for um cliente e um servidor UDP? Explique a sua resposta.
- 10) Um cliente TCP Java que estabelece uma conexão com um servidor TCP informa o IP e o número da porta no momento em que cria o seu objeto `Socket`:

```
Socket socketEnvio = new Socket("www.unifacs.br", 80);
```

Já um cliente UDP que envia um datagrama para um servidor UDP informa o IP e o número da porta quando cria o objeto `DatagramPacket`:

```
DatagramSocket ds = new DatagramSocket();  
...  
InetAddress ip = InetAddress.getByName("dns.unifacs.br");  
DatagramPacket dp = new DatagramPacket(msg, msg.length, ip, 53);  
ds.send(dp);
```

Porque esta diferença? Na sua resposta, relacione estes modelos de programação distintos com o tipo de serviço oferecido por cada um dos protocolos.