

# Lição 10



## Redes

# Tópicos

Ao final desta lição, o estudante será capaz de:

- Explicar os conceitos básicos de redes
  - Endereço IP
  - Protocolos
  - Portas
  - Paradigma Cliente/Servidor
  - Sockets
- Criar aplicações usando o pacote de Java para redes
  - As classes *ServerSocket* e *Socket*
  - As classes *MulticastSocket* e *DatagramPacket*



# Conceitos básicos de redes: Internet

- Uma rede mundial de computadores interligados
- Permanece funcional apesar da diversidade de hardware e software interconectados
- Possível mediante padrões de comunicação definidos e respeitados
- Assegura compatibilidade e confiabilidade de comunicações

# Conceitos básicos de redes: Endereço IP

- Logicamente similares ao tradicional endereçamento de correio
- Endereço que identifica individualmente um determinado objeto
- Um número de 32 bits usado para identificar unicamente cada computador conectado à Internet

# Conceitos básicos de redes: Protocolo

- Por que protocolos?
- Diferentes tipos de comunicações acontecem na Internet
- Conjunto de regras e padrões que definem um determinado tipo de comunicações pela Internet
- Descreve as seguintes informações:
  - Formato de dados enviados pela Internet
  - Como é enviado
  - Quando é enviado



# Conceitos básicos de redes: Protocolo

- O conceito de protocolo não é inteiramente novo para nós. Considere este tipo de conversa:

"Alô"

"Alô. Eu poderia falar com Joana?"

"Aguarde um momento, por favor"

"Obrigado"

- Protocolo social usado em uma conversa telefônica
- Nos proporciona confiança e familiaridade sobre como nos comportarmos



# Conceitos básicos de redes: Protocolo

- Hypertext Transfer Protocol (HTTP) – Protocolo de Transferência de hipertexto
- File Transfer Protocol (FTP) – Protocolo de Transferência de Arquivo
- Ambos os protocolos têm seu próprio conjunto de regras e padrões sobre como os dados são transferidos



# Conceitos básicos de redes:

## Portas

- Protocolos devem ser utilizados no contexto de um serviço
- HTTP é usado para disponibilizar conteúdo de Web por meio de um serviço HTTP
- Cada computador na Internet pode disponibilizar uma diversidade de serviços
- O tipo de serviço precisa ser conhecido antes que as informações possam ser transferidas



# Conceitos básicos de redes:

## Portas

- Um número de 16 bits que identifica cada serviço oferecido por um servidor de rede
- Empregando um determinado serviço para estabelecer uma linha de comunicação por meio de um protocolo específico
- É necessário conectar-se à porta apropriada

# Conceitos básicos de redes:

## Portas

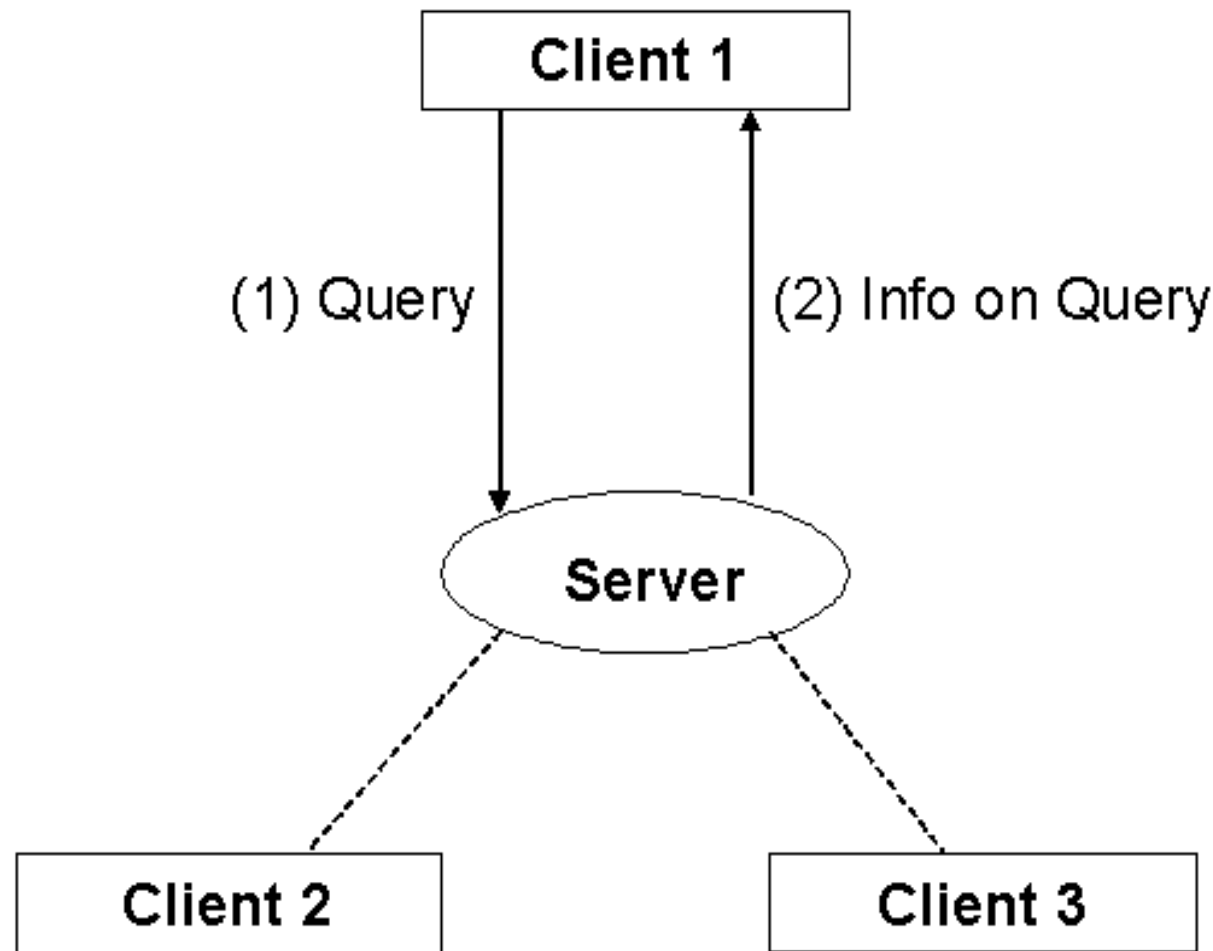
- Porta padrão
- Todos os serviços alocados de dados encontram-se nas porta com valores abaixo de 1024
- Se já estiver em uso por alguma comunicação personalizada, será necessário buscar outros valores



# Conceitos básicos de redes: O Paradigma Cliente/Servidor

- Base para aplicativos de redes Java
- Envolve dois elementos importantes:
  - Cliente
  - Servidor
- Cliente conecta-se a um servidor e consulta determinada informação e o Servidor avalia a consulta e retorna informação consultada ao cliente

# Conceitos básicos de redes: O Paradigma Cliente/Servidor



# Conceitos básicos de redes: Sockets

- Abstração, na forma de software, correspondente a um meio de entrada ou saída de comunicação
- Java executa todas as suas comunicações de baixo nível em rede através de sockets
- Canais de comunicação que permitem que você transfira dados através de uma determinada porta
- Um ponto terminal para comunicação entre duas máquinas



# Pacote Java para redes

- Pacote java.net
- Algumas classes existentes no pacote:
  - *ServerSocket*
  - *Socket*
  - *MulticastSocket*
  - *DatagramPacket*



# Classe *ServerSocket*

- Construtores:

```
ServerSocket(int port)
```

```
ServerSocket(int port, int backlog)
```

- Métodos:

```
public Socket accept()
```

```
public void close()
```

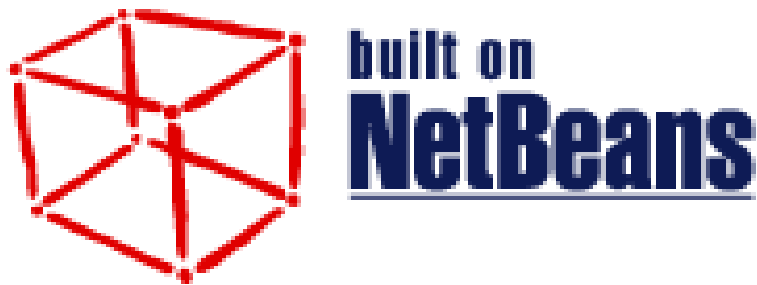
```
public int getLocalPort()
```

```
public boolean isClosed()
```



# Classe *ServerSocket*: Exemplo

- Passaremos agora para o NetBeans





# Classe Socket

- Construtores:

```
Socket(String host, int port)
```

```
Socket(InetAddress address, int port)
```

- Métodos:

```
public void close()
```

```
public InputStream getInputStream()
```

```
public OutputStream getOutputStream()
```

```
public InetAddress getAddress()
```

```
public int getPort()
```

```
public boolean isClosed()
```



# Classe *Socket*: Exemplo

- Passaremos agora para o NetBeans



# Classe *MulticastSocket*

- Construtores:

```
MulticastSocket(int port)
```

- Métodos:

```
public void joinGroup(InetAddress mcastaddr)
```

```
public void leaveGroup(InetAddress mcastaddr)
```

```
public void send(DatagramPacket p)
```



# Classe *DatagramPacket*

- Construtores:

```
DatagramPacket(byte[] buf, int length)
```

```
DatagramPacket(byte[] buf, int length,  
    InetAddress address, int port)
```

- Métodos:

```
public byte[] getData()
```

```
public InetAddress getAddress()
```

```
public int getLength()
```

```
public int getPort()
```



# Sumário

- Conceitos Básicos sobre Redes
  - Endereço IP
  - Protocolo
  - Portas
  - Paradigma Cliente/Servidor
  - Sockets
- Pacote Java para Redes
  - *ServerSocket*
  - *Socket*
  - *MulticastSocket*
  - *DatagramPacket*



# Parceiros

- Os seguintes parceiros tornaram JEDI<sup>TM</sup> possível em Língua Portuguesa:

