**Camada de Aplicação**

* Ignora o transporte e tenta fazer uma conexão entre as duas pontas.
* **Socket:** aplicação rodando num hospedeiro que envia e recebe mensagens da rede.
* Cada aplicação possui uma porta definida ou a ser definida para que o S.O possa saber para qual aplicação enviar os pacotes.
* **Definições da camada de aplicação:**

1. Tipo da mensagem
2. Sintaxe da mensagem
3. Regras de quando e como processos enviam e respondem mensagens.

* **Protocolo HTTP (Porta 80)**

**-**Permite a transferência de dados entre redes de computadores, principalmente na internet.

**-URL: protocolo://domínio/caminho**

* **Modelo cliente/servidor**

**-**Cliente requisita algo e o servidor responde.

**-O HTPP trabalha em cima do protocolo TPC**

**Por que?** Porque garante mais segurança.

**-O HTTP é um protocolo sem estado**, pois o servidor não guarda informações sobre requisições passadas do cliente.

* **HTTP não persistente**

**-**No máximo um objeto é enviado por uma conexão TPC.

**-**Utiliza http 1.0

**-Processo:** verifica se o servidor está disponível para a requisição do cliente e responde se sim ou não, se sim, envia um objeto e encerra a conexão TPC e assim sucessivamente até serem enviados todos os objetos requisitados.

**RTT:** tempo para um pacote trafegar do cliente ao servidor e retornar.

**Ttotal = 2RTT + tempo de transmissão**

**Ttotal = n objetos (2RTT+ tempo de transmissão)**

* **HTTP persistente**

**-**Múltiplos objetos podem ser enviados por uma única conexão TPC entre o cliente e o servidor.

**-**Utiliza http 1.1

* **Formato das mensagens HTTP**

**-Requisição:**

**-**Linha de requisição

**-**Linhas do cabeçalho

**-Resposta:**

**-**Linha de status

**-**Linhas do cabeçalho

**-**Dados requisitados

* **Método GET**

Solicita um recurso por meio do http.

* **Método HEAD**

Parecido com o GET mas sem que o corpo da mensagem seja retornado.

* **Método POST**

Envia dados para serem processados.

* **Cookies**

**-**É um pedaço de texto que um servidor Web armazena no disco rígido do usuário

**-**Salva o estado da sessão do usuário

**Ex:** carrinho de um site de compras.

**-**Problemas com privacidade, pois coleta dados do usuário.

* **Caches Web**

**-**Serve para otimizar o tempo de resposta entre o cliente e o servidor.

**-Processo:** o servidor intermediário (entre o cliente e o servidor final) guarda uma cópia dos objetos requisitados, assim quando for requisitado os mesmos objetos novamente, o servidor intermediário envia a cópia.

**-**Funilamento pode ser resolvido com cache

**-**É necessário um GET condicional para caso o objeto tenha uma versão atualizada o cache não seja enviado para o cliente. Isto é feito com base na data de modificação na requisição http.