Capitulo 1

Esse curso focará nesse protocolo tão presente no dia a dia do usuário e do desenvolvedor.

Iremos abordar tópicos como o HTTPS, que é a web segura, entendendo que o HTTP trafega texto puro, já o HTTPS trafega o texto criptografado, e como isso tudo funciona por baixo dos panos.

Veremos também sobre endereços, incluindo domínios, recursos e portas.

Além disso, estudaremos sobre sessão, cookie e o modelo de requisição e resposta do HTTP, mais ainda os parâmetros que são enviados na requisição, seja no seu corpo ou na URL.

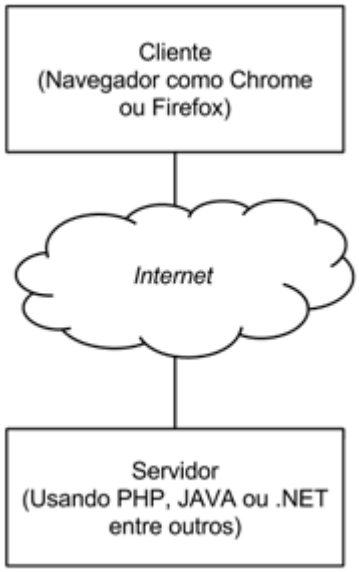
Comentaremos também sobre os serviços REST, já que o HTTP não roda somente no browser, ele também roda no seu aplicativo mobile. Veremos como implementar essa comunicação, que tipo de verbo o HTTP utiliza.

Por último, veremos sobre a nova versão do HTTP, o HTTP2, e o que ele adicionou de melhorias e otimizações que ele realiza para nós.

Nesse treinamento focaremos nos fundamentos da web. Isto é importante pois a grande maioria das aplicações hoje em dia a utilizam de alguma forma ou funcionam dentro dela. Não focaremos em nenhuma plataforma específica de desenvolvimento como Java ou PHP. Focaremos nas regras de comunicação da web.

Quando se fala em HTTP, o primeiro pensamento que vem a nossa mente é sobre a utilização da internet, é o cenário onde vemos realmente na prática a utilização do HTTP. Nós acessamos sites em que seus endereços iniciam com http:// e por isso precisamos conhecer o que realmente está acontecendo ao fazer isso.

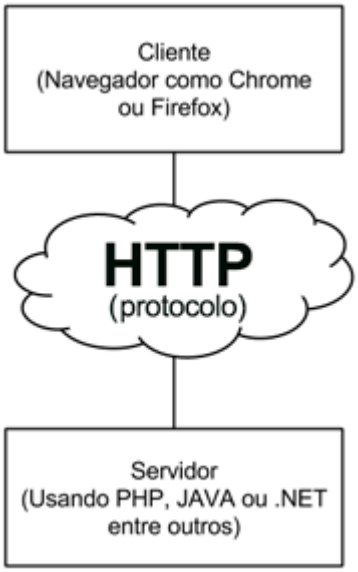
No momento em que acessou este curso, esta aula, entre o navegador e a Alura aconteceu uma comunicação, e esta comunicação tem duas partes bem conhecidas que chamamos de Client-Server ou em português Cliente-Servidor. Este é um modelo arquitetural, ou seja, a internet inteira é baseada nesta arquitetura onde há um cliente que solicita e um servidor que responde.



Em qualquer comunicação é preciso existir algumas regras para que as duas partes consigam se entender com sucesso. Pensando na comunicação do seu navegador entre a Alura ou algum outro site esse conjunto de regras é basicamente um protocolo, onde neste cenário é o HTTP.

Os protocolos são definidos, especificados e disponibilizados para implementação em ambas as partes, para consultar a especificação do HTTP, você pode utilizar o seguinte endereço: https://tools.ietf.org/html/rfc2616.

Resumindo: O HTTP é um protocolo que define as regras de comunicação entre cliente e servidor na internet. Vamos focar nos próximos vídeos e entender melhor esse protocolo tão importante. Mãos à obra!



O que aprendemos neste capítulo?

- Na internet sempre tem um cliente e um servidor.
- Entre o cliente e o servidor precisam haver regras de comunicação.
- As regras são definidas dentro de um protocolo.
- HTTP é o protocolo mais importante na internet.

Para saber mais: Peer-To-Peer

Você já usou torrent para baixar algum arquivo na internet? Caso sim, aproveitou um outro modelo de comunicação, o P2P ou Peer-To-Peer!

O modelo Cliente-Servidor não é o único modelo de comunicação na rede, nem sempre o mais adequado. Por exemplo, imagine que precisemos contar as letras de 20 palavras. No caso do modelo Cliente-Servidor, quem fará esse trabalho é o servidor, certo? E se precisar contar as letras de 1 milhão de palavras? Muito trabalhoso para o servidor, não?

O modelo Cliente-Servidor tenta centralizar o trabalho no servidor, mas isso também pode gerar gargalos. Se cada Cliente pudesse ajudar no trabalho, ou seja, assumir um pouco da responsabilidade do servidor, seria muito mais rápido. Essa é a ideia do P2P! Não há mais uma clara divisão entre Cliente-Servidor, cada cliente também é servidor e vice-versa!

Isto é útil quando você precisa distribuir um trabalho ou necessita baixar algo de vários lugares diferentes. Faz sentido?

Usando algum aplicativo de Torrent, o protocolo utilizado não é o HTTP, e sim o protocolo P2P, como BitTorrent ou Gnutella.

Para saber mais: Arquitetura da Alura

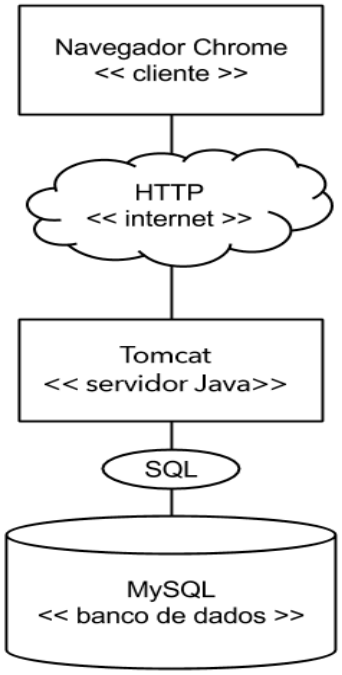
Agora já sabemos que existe um cliente, o navegador, como Chrome e Firefox, e um servidor, a Alura. Para definir as regras de comunicação entre cliente e servidor, existe o protocolo HTTP.

Também já sabemos que o servidor usa alguma plataforma, como PHP, Java, .Net ou outros. Qual plataforma realmente é utilizada? Não é tão fácil de descobrir, pois o HTTP, de propósito, não está focado em alguma plataforma específica e esconde isso de nós. Bom, eu não vou esconder nada e vou contar para vocês que a Alura usa a plataforma Java e o servidor concreto se chama Tomcat.

Também já falamos que o SQL é uma linguagem para consultar o banco de dados. Alura usa SQL para acessar o banco de dados MySQL.

Com essas informações já temos uma breve ideia da arquitetura da Alura!

Cliente <--- HTTP ---> Servidor Java <--- SQL ---> Banco de dadosCOPIAR CÓDIGO



O que você aprendeu nesse capítulo?

- A arquitetura Cliente-Servidor
- Um protocolo é um conjunto de regras
- HTTP é um protocolo que define as regras de comunicação entre cliente e servidor na internet.
- HTTP é o protocolo mais importante da Internet

Atividades

Aluno!

Neste treinamento, vamos falar sobre a "sigla" mais importante da internet: o HTTP. O objetivo é entender o protocolo HTTP detalhadamente. Quanto mais o desenvolvedor souber sobre este protocolo, melhor, pois ele é utilizado em todas aplicações web.

No entanto, não focaremos em como essas aplicações são criadas e funcionam internamente. Para isso, existem várias plataformas, como PHP, .NET ou Java (entre muitas outras) que não abordaremos. Temos treinamentos dedicados para conhecer estas plataformas.

Resumindo, nosso foco será o protocolo HTTP!

1. Falamos tanto sobre essa sigla, mas você sabe qual é o significado do HTTP?
    A. Alternativa correta.
    B. High Text Transmission Protocol.
    C. Alternativa correta.
    D. Heavy Transmission Text Protocol.
    E. Help Text Transfer Protocol.
    F. **Hypertext Transfer Protocol:**
    Alternativa correta: No mundo de TI, temos muitas siglas e abreviações! O que menos importa é decorar esses nomes, mas é preciso entender o que há por trás. Nesse treinamento vamos focar nos principais conceitos do protocolo HTTP, aquilo que realmente importa para o desenvolvedor.

2. O protocolo HTTP segue o modelo Client-Server. O que o navegador (como Chrome ou Firefox) representa nesse modelo? O cliente ou o servidor?

A. **Cliente: Alternativa correta:**
Alternativa correta: Exato, nós que estamos utilizando o navegador somos o cliente da Alura, que nos fornece o conteúdo, logo ela é o servidor.
B. Servidor.
C. Nem um, nem outro.

Nesse modelo, o navegador representa o cliente. É importante saber que nem só navegadores dominam o protocolo HTTP. Ainda veremos mais sobre isso neste curso.

3. O cliente inicia a comunicação e o servidor responde. No entanto, qual é o papel do HTTP entre o cliente e o servidor?
    A. Definir uma estrutura de dados
    B. Definir o melhor algoritmo de pesquisa
    C. **Estabelecer regras de comunicação:** Alternativa correta exatamente, o HTTP foi feito para estabelecer regras de comunicação entre o modelo Cliente-Servidor que funciona na Web.
    D. Comprimir os dados

Se você compreende este texto, é porque você sabe português! Para que alguém consiga se comunicar com você, esse alguém deverá usar o português (supondo que você desconheça outro idioma, é claro). Isso significa que, sua regra (protocolo) de comunicação com o mundo é a língua portuguesa, que define a forma com que as informações devem chegar até você (através do vocabulário, regras de gramática e etc. uma outra pessoa que conheça português irá usar do mesmo formato, já que vocês possuem um idioma em comum.

Na internet, como já vimos, o idioma mais comum é o HTTP. Ele é responsável por definir a forma de como os dados são trafegados na rede através de várias regras. Portanto, todo mundo que conhece o idioma HTTP poderá receber e enviar dados e participar dessa conversa!

4. O HTTP não é o único protocolo de comunicação que existe. Aliás, existem milhares de protocolos no mundo de TI, no entanto o HTTP é de longe o mais popular.

Na lista abaixo, há um item que não representa um protocolo para internet.

Qual é exatamente? Pesquise se for necessário.

A. FTP.
B. SQL.
Alternativa correta: SQL (Structured Query Language) não é um protocolo para internet, e sim uma linguagem de consulta para banco de dados.
C. BitTorrent.

D. SMTP

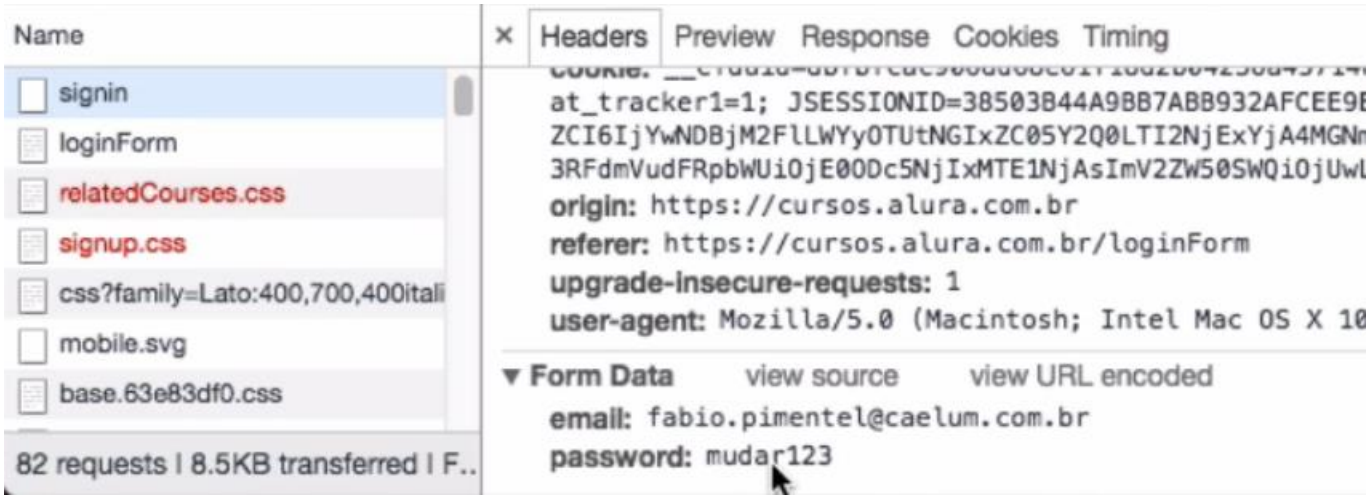
Um banco de dados cuida dos dados de uma aplicação, é parecido com uma planilha de Excel. O SQL ajuda muito a acessar esses dados.

Um banco de dados não se preocupa em como os dados serão visualizados, ele só administra os dados! Aqui na Alura, o banco de dados guarda informações sobre os usuários, cursos, perguntas, respostas, etc.

Capitulo 2

Sabendo que o HTTP é o protocolo que define as regras de comunicação na web, precisamos observar algumas coisas. Quando usamos o HTTP, todos os dados enviados entre cliente e servidor são transmitidos em texto puro, inclusive dados sensíveis, como login e senha!

Quando acessamos a Alura por exemplo, precisamos fornecer informações de autenticação, essas informações são nosso e-mail e senha, que são enviadas e validadas pela plataforma para que assim consigamos assistir as aulas. Estas informações são enviadas em texto limpo e é possível visualizá-las pelas ferramentas do desenvolvedor do navegador. A aba network nos possibilita isso.



Mas por que é importante sabermos isso? Quando o navegador pede informações da Alura, nessa comunicação há vários intermediários. Por exemplo, usando uma conexão Wi-Fi, os dados do navegador passam primeiro para o roteador Wi-Fi, e do roteador passam para o modem do provedor, do modem para algum servidor do provedor de internet, como Oi ou NET.

É muito provável que existam outros servidores intermediários no provedor antes que os dados realmente cheguem no servidor da Alura. Com a resposta é a mesma coisa, ela volta passando por esses servidores no meio antes de chegar até nosso navegador. O problema é, quando usamos HTTP, qualquer servidor no meio pode espionar os dados enviados, algo totalmente inseguro! Imagine se essas informações fossem relativas a contas bancárias. Não seria nada seguro!

Para estes outros cenários, existe o HTTPS, que basicamente é o HTTP comum, porém com uma camada adicional de segurança/criptografia que antes era SSL, mas posteriormente passou a ser também TLS. É muito comum que estas duas siglas sejam encontradas juntas como SSL/TLS por se tratarem da mesma questão de segurança. Sendo assim, temos dois termos:

1. HTTP: HyperText Transfer Protocol.
2. SSL/TLS: Secure Sockets Layer / Transport Layer Security.

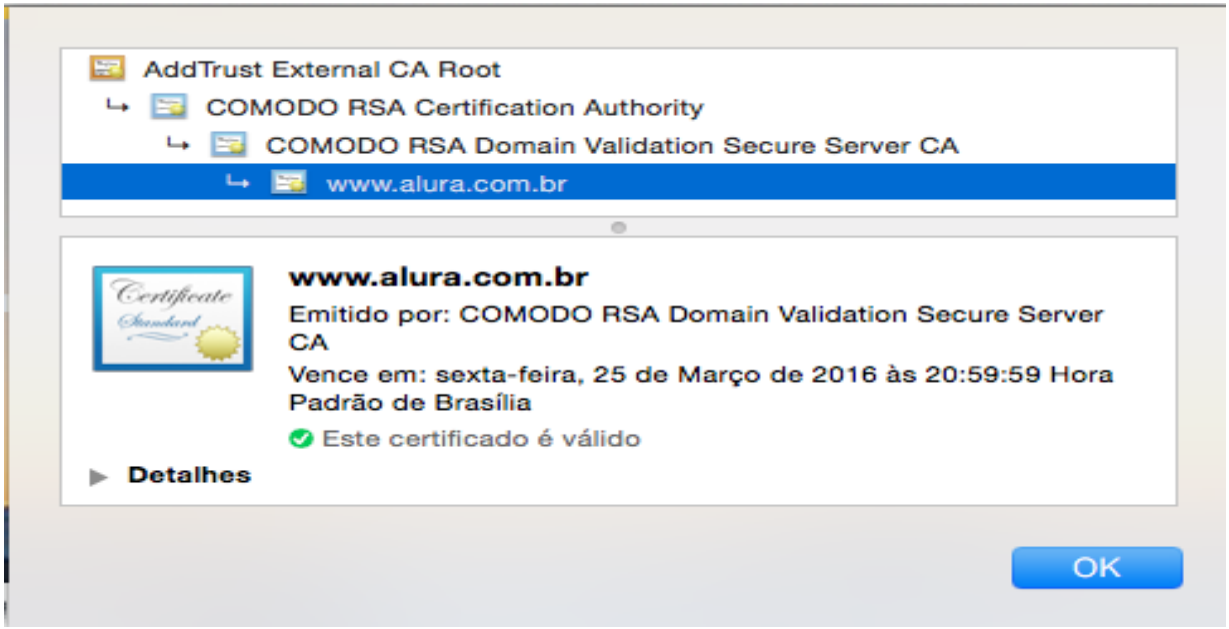
Ao acessarmos o site da Alura pelo navegador podemos perceber que ele já usa o protocolo **HTTPS:**

Reparem que no navegador, ao lado do https, aparece um cadeado e que ao clicarmos no cadeado podemos ver mais informações sobre HTTPS. Uma dessas informações indica que a Alura tem uma identidade confirmada. O que isso quer dizer?

O HTTPS para garantir segurança usa criptografia baseada em chaves públicas e privadas e para gerar essas chaves públicas e privadas é preciso garantir a identidade de quem possui essas chaves e isso é feito a partir

de um certificado digital, ou seja, um certificado digital é utilizado para identificar determinada entidade e ainda é utilizada para geração das chaves de criptografia.

Apesar disso, ainda é necessário que uma autoridade certificadora, que nada mais é que um órgão ou entidade confiável, garanta não apenas a identidade do site, mas também a validade do certificado. No caso da Alura a autoridade certificadora é a COMODO RSA Domain Validation, mas existem outras.



Dito isso, como tudo funciona? Os navegadores em posse da chave pública criptografam as informações e as enviam para o servidor que as descriptografa com a chave privada. É importante notar que apenas a chave privada descriptografa as informações criptografadas com a pública, e também que deve-se manter a chave privada segura.

O que aprendemos nesse capítulo?

- Só com HTTPS a web é segura.
- HTTPS significa usar um certificado digital no servidor.
- O certificado prova a identidade e tem validade
- O certificado possui uma chave pública.

A chave é utilizada pelo navegador para criptografar os dados.

Certificado digital

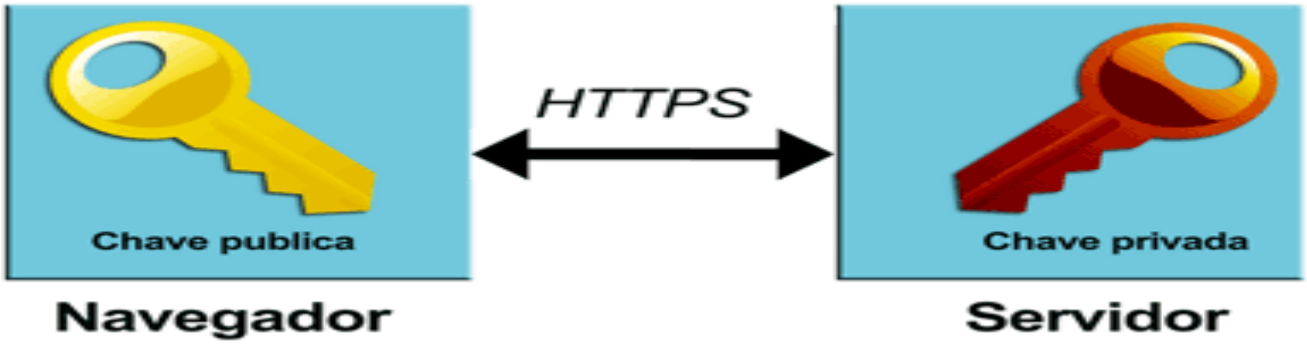
Quando precisamos informar nossos dados a algum servidor, queremos ter certeza que este servidor realmente representa a entidade em questão. Queremos confiar em quem estamos fornecendo nossos dados!

Um certificado digital prova uma identidade para um site, onde temos informações sobre o seu domínio e a data de expiração desse certificado.

Além disso, o certificado ainda guarda a chave pública que é utilizada para criptografar (cifrar) os dados que são trafegados entre cliente e servidor.

Para Saber Mais: As chaves do HTTPS

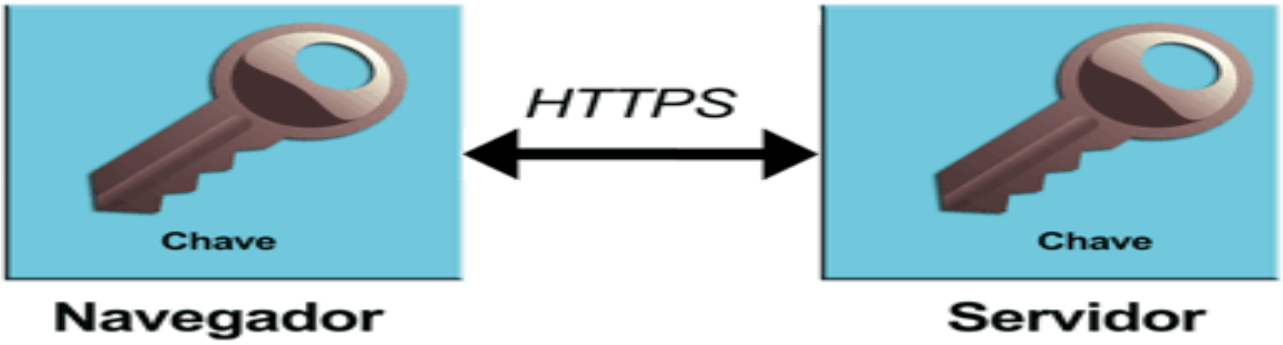
Aprendemos no vídeo que o HTTPS usa uma chave pública e uma chave privada. As chaves estão ligadas matematicamente, o que foi cifrado pela chave pública só pode ser decifrado pela chave privada. Isso garante que os dados cifrados pelo navegador (chave pública) só podem ser lidos pelo servidor (chave privada). Como temos duas chaves diferentes envolvidas, esse método de criptografia é chamado de criptografia assimétrica. No entanto, a criptografia assimétrica tem um problema, ela é lenta.



Por outro lado, temos a criptografia simétrica, que usa a mesma chave para cifrar e decifrar os dados, como na vida real, onde usamos a mesma chave para abrir e fechar a porta. A criptografia simétrica é muito mais rápida, mas infelizmente não tão segura. Como existe apenas uma chave, ela ficará espalhada pelos clientes (navegadores) e qualquer um, que tem a posse dessa chave, pode decifrar a comunicação.

Agora, o interessante é que o HTTPS usa ambos os métodos de criptografia, assimétrica e simétrica. Como assim? Muita calma, tudo o que aprendemos é verdade! Só faltou o grande final :)

No certificado, vem a chave pública para o cliente utilizar, certo? E o servidor continua na posse da chave privada, ok? Isso é seguro, mas lento e por isso o cliente gera uma chave simétrica ao vivo. Uma chave só para ele e o servidor com o qual está se comunicando naquele momento! Essa chave exclusiva (e simétrica) é então enviada para o servidor utilizando a criptografia assimétrica (chave privada e pública) e então é utilizada para o restante da comunicação.



Então, HTTPS começa com criptografia assimétrica para depois mudar para criptografia simétrica. Essa chave simétrica será gerada no início da comunicação e será reaproveitada nas requisições seguintes. Bem-vindo ao mundo fantástico do HTTPS :)

Atividades

Enviando dados com HTTP

1. O que acontece com nossos dados quando usamos HTTP, ou seja, sem a letra S ao final?
   - **A. Os dados são transportados em texto puro para o servidor, visível para qualquer um.**
     Alternativa correta: Exato, nossos dados são enviados em texto puro, ficando visível para qualquer um que consiga interceptar nossa conexão!
   - B. Os dados são criptografados, para impedir a visualização por intermediários.
   - C. Usamos automaticamente um certificado digital para provar a identidade de um site.

Quando usamos HTTP, os dados são enviados em texto puro. O que pode ser perigoso, já que assim deixamos os dados abertos para intermediários.

Características do HTTPS

1. Sobre as características do HTTPS, selecione todas as opções abaixo que estejam corretas:

   - **A. A chave privada fica apenas no lado do servidor.**
     Alternativa correta: Exato, a chave privada é utilizada para descriptografa os dados que foram criptografados com a chave pública, por isso ela é importante e deve ficar apenas em posse do servidor.
   - B. HTTP significa usar um certificado digital do servidor.
   - **C. O certificado prova a identidade e tem validade.**
     Alternativa correta: Correto, todo certificado tem uma data validade e serve para provar a identidade entre o cliente e o servidor.
   - **D. O certificado guarda a chave pública.**
     Alternativa correta: Perfeito, é no certificado digital que encontramos a chave pública utilizada para criptografar os nossos dados.

Lembrando o HTTP não utiliza criptografia nenhuma e é inseguro! Para deixar a web segura devemos usar o HTTPS sempre.

Autoridade certificadora

1. Qual é a finalidade das autoridades certificadoras?
   - **A. Garantir que podemos confiar naquele certificado (identidade).**
     Alternativa correta: Exato, a principal função de uma entidade certificadora é garantir que os certificados que estão sendo utilizados podem ser confiados.
   - B. Importar/Exportar chaves públicas do servidor.
   - C. Usada para registrarmos nomes de domínio (DNS).
   - D. Realizar a criptografia dos dados da requisição.

Essa garantia é feita através de uma assinatura digital. A autoridade certificadora (CA) assina digitalmente o certificado! Como na vida real, existem também no mundo digital: assinaturas!

Uma autoridade certificadora (CA - Certificate Authority) é um órgão que garante ao navegador e ao usuário que a identidade de um servidor (por exemplo o servidor da Alura) é realmente válida. Portanto, podemos trocar informações com este sem riscos!