Http – protocolo de transferência de hipertexto – protocolo de comunicação para transferir dados na web

Protocolo: padrão(conjunto de regras) que controla e possibilita conexão, comunicação e, transferência de dados entre sistemas computacionais – rede de computadores

Em uma estrutura http é necessário dois sides -> cliente e servidor (entidades de comunicação)

Aplicações: são softwares que realizam tarefas especificas – tem diversos conceitos

Interface gráfica do usuário, Diagrama, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

FORNECE SERVICO A UMA REDE DE COMPUTADORES

Servidores de aplicação: disponibiliza e gerencia recursos e um ambiente para execução de aplicações. O servidor de aplicações responde a algumas questões comuns a todas as aplicações, como [segurança](https://pt.wikipedia.org/wiki/Seguran%C3%A7a), garantia de [disponibilidade](https://pt.wikipedia.org/wiki/Disponibilidade), [balanceamento de carga](https://pt.wikipedia.org/wiki/Balanceamento_de_carga) e [tratamento de exceções](https://pt.wikipedia.org/wiki/Tratamento_de_exce%C3%A7%C3%B5es).

* [**Servidor web**](https://pt.wikipedia.org/wiki/Servidor_web): Servidor responsável pelo armazenamento de páginas de um determinado [site](https://pt.wikipedia.org/wiki/Site), requisitados pelos clientes através de [*browsers*](https://pt.wikipedia.org/wiki/Browser).
* **Servidor de banco de dados**: Servidor que possui e manipula informações contidas em um banco de dados
* **Servidor**[**DNS**](https://pt.wikipedia.org/wiki/DNS): Servidores responsáveis pela conversão de endereços de sites em [endereços IP](https://pt.wikipedia.org/wiki/Endere%C3%A7o_IP) e vice-versa.
* [**Servidor proxy**](https://pt.wikipedia.org/wiki/Proxy): Servidor que atua como um [cache](https://pt.wikipedia.org/wiki/Cache), armazenando páginas da [internet](https://pt.wikipedia.org/wiki/Internet) recém-visitadas, aumentando a velocidade de carregamento destas páginas ao chamá-las novamente.[[3]](https://pt.wikipedia.org/wiki/Servidor#cite_note-3)
* [**Servidor de sistema operacional**](https://pt.wikipedia.org/w/index.php?title=Servidor_de_sistema_operacional&action=edit&redlink=1): permite compartilhar o sistema operacional de uma máquina com outras, interligadas na mesma rede, sem que essas precisem ter um sistema operacional instalado, nem mesmo um HD próprio.
* **As camadas da internet**
* Para responder a essa pergunta, começaremos entendendo melhor quais são as camadas da internet e onde o HTTP se encaixa.
* As duas primeiras camadas da internet são a **Física** e a de **Enlace**. Nelas, estão as partes físicas que possibilitam a nossa navegação via internet. Podemos citar como exemplos a conexão com cabos de fio, o Wi-fi e até o 5G.
* A próxima é a camada de **Rede**. Um exemplo dessa camada é a conexão IP. Você já deve ter ouvido falar dos endereços IP, certo? Eles se encontram na camada de Rede.
* Em seguida, temos a camada de **Transporte**, que engloba os protocolos TCP e UDP. Essa camada garante que as informações cheguem de um ponto A até um ponto B.
* Por fim, temos a camada de **Aplicação**, onde está o HTTP. Essa camada é composta pelo browser usado no computador e no smartphone e pelos aplicativos instalados no celular. Toda vez que usarmos o HTTP, estamos usando a camada de Aplicação.

HTTP utiliza a camada de transporte do tcp para fazer a transferencia de dados

Diagrama

O conteúdo gerado por IA pode estar incorreto.SQL é uma linguagem para consultar o banco de dados.

Cliente <--- HTTP ---> Servidor Java(plataforma do servidor)

<--- SQL ---> Banco de dadosCopiar có

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Acabamos de aprender que URLs são utilizadas para definir, de forma padronizada, como localizar um recurso (como um website ou um arquivo) na Internet.

O recurso é aquilo que vem depois do nome do servidor.

 Em outras palavras, uma URL representa uma identificação de um recurso (URI) através do endereço(localização)

Uma imagem contendo Diagrama

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

TCP É USADO PELO HTTP PARA TRANSPORTAR AS MENSGAGENS

Uma imagem contendo Diagrama

O conteúdo gerado por IA pode estar incorreto.Diagrama

O conteúdo gerado por IA pode estar incorreto.

GET / HTTP/1.1

Tabela

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Tela de celular com publicação numa rede social

O conteúdo gerado por IA pode estar incorreto.

Quando falamos de **Cookies**, na verdade queremos dizer **Cookies HTTP** ou **Cookie web**. Um cookie é um pequeno arquivo de texto, normalmente criado pela aplicação web, para guardar algumas informações sobre o usuário no navegador. Quais são essas informações depende um pouco da aplicação. Pode ser que fique gravado alguma preferência do usuário. Ou algumas informações sobre as compras na loja virtual ou, como vimos no vídeo, a identificação do usuário. Isso depende da utilidade para a aplicação web.

500 Internal Server Error, que significa que o servidor teve algum problema interno na hora de tratar a requisição.

O código 500 acontece com frequência quando estamos desenvolvendo uma aplicação web e, ao testar, percebemos que erramos algo na lógica que gerou um problema no servidor.

De forma geral, erros da classe 5xx significam que houve algum problema no servidor.

Nesta aula vamos utilizar o comando openssl, que permite, entre outras coisas, gerar chaves privadas e certificados digitais.

git clone https://github.com/alura-cursos/curso-react-alurabooks.git

# entra na pasta do frontend

cd curso-react-alurabooks

# seleciona a versão correta

git checkout aula-5

# instala as dependências

npm install

# compila o frontend e o disponibiliza através de um servidor no endereço http://localhost:3000

npm start

# baixa nosso backend

git clone https://github.com/alura-cursos/api-alurabooks.git

# entra na pasta do backend

cd api-alurabooks

# instala as dependências que estão listadas no arquivo package.json

npm install

# executa o backend e o disponibiliza através de um servidor no endereço http://localhost:8000

npm run start-auth

Tela de celular com fundo preto

O conteúdo gerado por IA pode estar incorreto.

INSERÇÃO DE CERTIFICADO DIGITAL PARA SEGURANÇA ->> TRATAMENTO DE AUTORIZAÇÕES WIRESHARK

Para simularmos uma situação em que um hacker entra na nossa rede e intercepta os requests HTTP, usaremos um programa chamado **Wireshark**

om isso, percebemos que o Wireshark conseguiu capturar o protocolo de texto e ler o seu conteúdo. Desse modo, temos uma falha de segurança no projeto. Para resolvê-la, introduziremos a versão segura do HTTP: o **HTTPS**.

Para realizar essa modificação, o primeiro passo é gerar uma entidade e uma chave de criptografia para o nosso site. Faremos isso digitando o seguinte comando no Terminal, na pasta "api-alurabooks", e pressionando a tecla "Enter" em seguida:

openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt

Usaremos o OpenSSL, um programa que trabalha com criptografia. Ele será usado para gerar tanto a chave privada do servidor quanto o certificado do servidor, ou seja, a sua identidade.

CRIAR OS ARQUIVOS PARA ENTIDADE E CHAVE DE CRIPTOGRAFIA E DEPOIS IMPLEMENTAR NO CODIGO

Em seguida, verificaremos se o Wireshark conseguiu capturar as informações transmitidas via HTTPS. Com o filtro usado, estamos vendo apenas as atividades com HTTP. O Wireshark não nos permite filtrar por HTTPS porque ele usa um protocolo chamado TLS, que corresponde à segurança da camada de transporte sobre o TCP.

Por esse motivo, atualizaremos a barra de texto no topo do Wireshark para "tcp.port == 8000 && tls". Ao fazer isso, ele exibirá as mensagens enviadas de maneira criptografada, com o HTTPS que habilitamos.

HTTPS CONFIGURADO NO BACKEND

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.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Nslookup

Texto

O conteúdo gerado por IA pode estar incorreto.

No caso do certificado, surge uma sequência de letras e números ininteligíveis. Existe outro comando baseado no programa OpenSSL capaz de decodificar o conteúdo do certificado. Assim, digitaremos "openssl x509 -in server.crt -text" no Terminal e pressionaremos a tecla "Enter".

Para inspecionar a chave, usaremos o comando "openssl rsa -in server.key -text -noout" e pressionaremos a tecla "Enter".

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

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.

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.

Publica -> privada -> encaminha pro servidor simétrico somente para aquele momento para ser utilizada no restante da comnicação

Tela de celular

O conteúdo gerado por IA pode estar incorreto.

Passando parâmetros para buscar

Diagrama

O conteúdo gerado por IA pode estar incorreto.

/livros?categoria=3&autor=1

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Em alguns momentos se faz necessário avisar na própria requisição um formato específico esperado para a resposta.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

TCP É UM TUBO Q PASSA REQUISÃO NO HTTP1 TINHA QUE SER UMA ATRAS DA OTURA

OU SEJA ERA ENVIADO A REQUISICAO E TINHA QUE ESPERAR A RESPONSE PARA DEPOIS MANDAR OUTRA

HTTP2 PERMITE REQUISICOES EM PARALELO

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Tela preta com letras brancas

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

A principal mudança do HTTP/3 é o uso do QUIC para enviar mensagens HTTP/2. Já o QUIC, por sua vez, é baseado no UDP, e é mais eficiente do que usar o TCP. Para conseguir isso, o QUIC introduz o mínimo de funcionalidade necessária para obter confiabilidade nas mensagens (como ocorre no TCP), mas sem sacrificar a performance.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

ROOT SERVER

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Tabela

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Tela preta com letras brancas

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.