ETEC MARIA CRISTINA MEDEIROS

LAURA DE OLIVEIRA SOUSA

**Protocolo HTTP**

Ribeirão Pires

2024

ETEC MARIA CRISTINA MEDEIROS

LAURA DE OLIVEIRA SOUSA

**Protocolo HTTP**

Trabalho apresentado como forma avaliativa bimestral na matéria de Sistemas Web no curso de Informática para Internet

Ribeirão Pires

2024

**SUMÁRIO**

[**1.PROTOCOLO HTTP** 5](#_Toc169545724)

[**1.1** **Funcionamento** 5](#_Toc169545725)

[**1.2** **HTTP X HTTPS** 5](#_Toc169545726)

[**2.MÉTODOS DE REQUISIÇÃO** 6](#_Toc169545727)

[**2.1 Lista** 6](#_Toc169545728)

[**3.STATUS DE RESPOSTA HTTP** 7](#_Toc169545729)

[**3.1 Lista** 7](#_Toc169545730)

[**4.CONCLUSÃO** 7](#_Toc169545731)

# **1.PROTOCOLO HTTP**

O protocolo HTTP é um protocolo de comunicação que permite a transferência de dados entre clientes e servidores na web12. O protocolo funciona como um modelo de requisição e resposta, onde o cliente envia uma mensagem HTTP com um método de solicitação (como GET ou POST) e o servidor responde com outra mensagem HTTP com um código de status (como 200 ou 404)1. O protocolo HTTP também controla aspectos como o cache, a flexibilidade do requisito de origem e a segurança dos dados

## **Funcionamento**

O protocolo HTTP opera em um modelo cliente-servidor1. O funcionamento ocorre por meio de uma troca de mensagens entre o cliente e o servidor, chamadas de requisição (request) e resposta (response)2. Quando um usuário acessa um site digitando a URL no navegador, o navegador envia uma requisição ao servidor solicitando a página correspondente.

### **HTTP X HTTPS**

# A diferença entre HTTP e HTTPS é a segurança. O HTTPS é uma versão mais segura ou uma extensão do HTTP, que utiliza criptografia para proteger os dados durante o tráfego, tornando-os praticamente indecifráveis para qualquer pessoa que esteja interceptando a comunicação. O certificado SSL criptografa o conteúdo transferido e só permite sua leitura a quem tiver uma chave para “abri-lo”1. O HTTP, por sua vez, é um protocolo não seguro, permitindo que terceiros mal-intencionados interceptem e leiam as informações transmitidas através dele

# **2.MÉTODOS DE REQUISIÇÃO**

Entendendo agora o que são esses protocolos, é importante saber o que acontece por trás de toda essa comunicação. Os métodos são comandos que definem o objetivo do REQUEST. Explicando melhor, uma requisição tem 5 partes: método, path, versão do HTTP, cabeçalhos (headers) e corpo (body). Embora esses métodos possam ser descritos como substantivos, eles também são comumente referenciados como HTTP Verbs (Verbos HTTP). Cada um deles implementa uma semântica diferente

## **2.1 Lista**

* GET: Mais comumente usado para passar parâmetros na URL, o método get tem como objetivo recuperar e solicitar informações de um servidor.
* HEAD: Assim como get, ele tem objetivo de recuperar e solicitar dados mas encurtando as respostas. Como por exemplo, ele pode ser usado para pegar um metadado sem carregar o conteúdo complete.
* POST: É um método usado para enviar informações a serem processadas. Como envio de dados do formulário
* PUT: Serve para criar ou atualizar uma informação específica indicada.
* DELETE: Como o próprio nome já diz, ele deleta completamente e permanentemente um dado específico informado
* TRACE: Usado para fins diagnósticos, o trace é eficaz para mostrar o processo que possivelmente deu errado, assim fazendo o debug.

# **3.STATUS DE RESPOSTA HTTP**

O status de resposta ou código de resposta é uma mensagem enviada do servidor para o cliente, identificando o erro/status da sua página. Em outras palavras, após um request para um servidor, ele manda as informações necessárias + o código que explica aquela situação

## **3.1 Lista**

Um código de status HTTP consiste em três dígitos. O primeiro dígito varia de um a cinco, e indica o tipo de status. O segundo e terceiro dígitos referem-se aos códigos de status no intervalo do primeiro dígito.

* 1xx (INFORMATIVO) – o servidor recebeu a solicitação e está processando.
* 2xx (CONFIRMAÇÃO) ‒ o servidor recebeu a solicitação e enviou de volta a resposta esperada.
* 3xx (REDIRECIONAMENTO) ‒ indica que algo mais precisa ser feito ou precisou ser feito para completar a solicitação.
* 4xx (ERRO DO CLIENTE) ‒ indica que a solicitação não pode ser concluída ou contém a sintaxe incorreta.
* 5xx (ERRO NO SERVIDOR) ‒ o servidor falhou ao concluir a solicitação.
* 200 (SUCESSO) -O servidor atingiu com êxito o objetivo. A requisição foi feita perfeitamente.
* 400 (ERRO) – A requisição não pode ser feita devido à erro na sintaxe
* 404 (NÃO ENCONTRADO) – Sua requisição não pode ser concluida pois as informações não foram encontradas.

# **4.CONCLUSÃO**

Em resumo, o protocolo HTTP, com seus códigos de status e métodos, forma a base da web moderna. Entender esses elementos é crucial para desenvolver aplicações web mais eficientes e resilientes, além de ser fundamental para resolver problemas e garantir uma boa experiência ao usuário. As novas versões, como HTTP/2 e HTTP/3, prometem tornar a comunicação na web ainda mais rápida e segura, ressaltando a importância do HTTP no cenário digital presente e futuro.Com as informações apresentadas, é possível entender como um link/URL funciona e o que está por trás dele. Este trabalho mostra como tudo está interconectado e interdependente, criando a internet e a experiência de navegação que conhecemos hoje.

**REFERÊNCIA**

Hypertext Transfer Protocol – Wikipédia, a enciclopédia livre (wikipedia.org). Acesso em: 22 jun. 2024.

https://planejadorweb.com.br/http-entenda-o-que-e-para-que-serve-e-como-funciona/. Acesso em: 22 jun. 2024.

https://www.homehost.com.br/blog/internet/o-que-e-http/. Acesso em: 22 jun. 2024.

https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods. Acesso em: 22 jun. 2024.

MELO, Diego. O que é HTTP?. **Tecnoblog,** 2022. Disponível em: https://tecnoblog.net/responde/o-que-e-http/. Acesso em: 17 jun. 2024.