ETEC PROFESSORA MARIA CRISTINA MEDEIROS

Ensino Médio Integrado ao Técnico em Informática para Internet

Igor Neves Silva

**Protocolo HTTP e Repostas**

Ribeirão Pires

2024

Igor Neves Silva

**Protocolo HTTP e Repostas**

Trabalho bimestral avaliativo da matéria de

SW II apresentado ao Ensino

Médio Integrado com Habilitação em Técnico

em Informática, orientado pelo Professor Anderson Silva.

Ribeirão Pires

2024.

# Sumário

1. **Introdução..............................................................................................**
2. **Principais Métodos de Requisição.............................................................**
3. **Códigos de Status de resposta..................................................................**
4. **Conclusão..............................................................................................**
5. **Bibliografia.............................................................................................**
6. **INTRODUÇÃO**

HTTP é um protocolo de transferência que possibilita que as pessoas que inserem a URL do seu site na Web possam ver os conteúdos e dados que nele existem. A sigla vem do inglês Hypertext Transfer Protocol. O termo HTTP foi cunhado por Ted Nelson em 1965 no Projeto Xanadu, que por sua vez foi inspirado numa visão dos anos 30 tida por Vannevar Bush referente ao sistema “memex” de recuperação e gestão de informação baseado em microfilmes descrito em seu ensaio de 1945 “[As We May Think](https://worrydream.com/refs/Bush%20-%20As%20We%20May%20Think%20(Life%20Magazine%209-10-1945).pdf)“.Cada URL que começa com HTTP usa um tipo básico de “protocolo de transferência de hipertexto”. Esse padrão de protocolo de rede é o que permite que navegadores e servidores se comuniquem através da troca de dados.Ou seja, por se tratar de um protocolo da camada de aplicação, o HTTP foca em apresentar a informação, se preocupando menos com a maneira como essa informação é transmitida de um ponto para outro. Isso quer dizer que é possível transferir informações de forma confiável e padronizada, permitindo o acesso a uma infinidade de recursos online assim, o HTTP continua a ser um pilar essencial para a conectividade e o funcionamento da web como a conhecemos hoje.

1. **PRINCIPAIS MÉTODOS DE REQUISIÇÃO**

O protocolo HTTP contém vários métodos de requisição sendo eles:

GET: O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

HEAD: O método HEAD solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta.

POST: O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

PUT: O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

DELETE: O método DELETE remove um recurso específico.

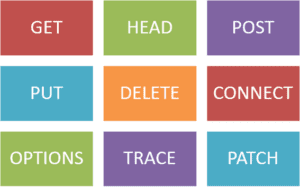
CONNECT: O método CONNECT estabelece um túnel para o servidor identificado pelo recurso de destino.

PATCH: O método PATCH é utilizado para aplicar modificações parciais em um recurso.

OPTIONS: O método OPTIONS é usado para descrever as opções de comunicação com o recurso de destino..

TRACE: O método TRACE executa um teste de chamada loop-back junto com o caminho para o recurso de destino.

**Figura 1**



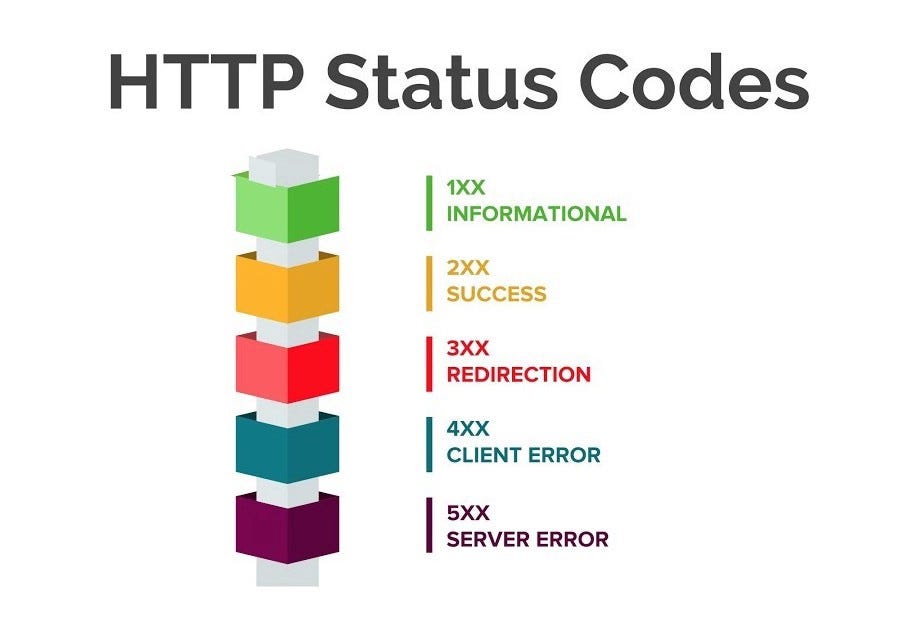
**Fonte:** [**GRAN**](https://blog.grancursosonline.com.br/arquitetura-rest-metodos-de-requisicoes-http/)

1. **CÓDIGOS DE STATUS DE RESPOSTA**

Os códigos HTTP são divididos em cinco classes de resposta. Cada classe de código de status representa um tipo de comunicação que acontece entre o servidor e o cliente. São elas:

1. [Respostas Informativas](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status#respostas_informativas) (100 – 199)
2. [Respostas bem-sucedidas](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status#respostas_bem-sucedidas) (200 – 299)
3. [Mensagens de redirecionamento](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status#mensagens_de_redirecionamento) (300 – 399)
4. [Respostas de erro do cliente](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status#respostas_de_erro_do_cliente) (400 – 499)
5. [Respostas de erro do servidor](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status#respostas_de_erro_do_servidor) (500 – 599)

**Figura 2**



**Fonte: MEDIUM**

**100 Continuar**

Essa resposta provisória indica que o cliente deve continuar a solicitação ou ignorar a resposta se a solicitação já estiver concluída.

**101 Mudando Protocolos**

Esse código é enviado em resposta a um cabeçalho de solicitação [Upgrade](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Upgrade) do cliente e indica o protocolo para o qual o servidor está mudando.

**102 Processamento**

Este código indica que o servidor recebeu e está processando a requisição, mas nenhuma resposta está disponível ainda.

**103 Sugestões Iniciais**

Este código de status destina-se principalmente a ser usado com o cabeçalho [Link](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Headers/Link), permitindo que o agente do usuário inicie o [pré-carregamento](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/rel/preload" \o "Esta página está disponível apenas em inglês no momento) recursos enquanto o servidor prepara uma resposta.

**200 OK**

A solicitação foi bem-sucedida. O significado do resultado de "sucesso" depende do método HTTP:

**201 Criado**

A requisição foi bem sucedida e um novo recurso foi criado como resultado. Esta é normalmente a resposta enviada após as solicitações POST ou algumas solicitações PUT.

**202 Aceito**

A solicitação foi recebida, mas ainda não foi atendida. É sem compromisso, pois não há como no HTTP enviar posteriormente uma resposta assíncrona indicando o resultado da solicitação. Destina-se a casos em que outro processo ou servidor manipula a solicitação ou processamento em lote.

**203 Não Autorizado**

Este código de resposta só aparece ao usar um servidor proxy. O servidor proxy modifica o código de status de resposta 200 enviado pelo servidor de origem antes de passá-lo para o cliente.

**204 Nenhum Conteúdo**

Não há conteúdo para enviar para esta solicitação, mas os cabeçalhos podem ser úteis. O agente do usuário pode atualizar seus cabeçalhos em cache para este recurso com os novos.

**205 Resetar Conteúdo**

Diz ao agente do usuário para redefinir o documento que enviou esta solicitação.

**206 Conteúdo Parcial**

Este código de resposta é usado quando o cabeçalho [Range](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Headers/Range) é enviado do cliente para solicitar apenas parte de um recurso.

**300.1 Redirecionamento**

A classe de código de status 300 sinaliza que o agente de usuário precisa realizar ações adicionais para que a solicitação seja atendida, pois o recurso solicitado foi redirecionado para outro lugar. Devido à sua função, essa classe de código de status geralmente aparece no redirecionamento de URL. No entanto, você pode não ver esse código se a segunda solicitação usar o método GET ou HEAD. Nesse caso, o agente pode realizar a ação adicional sem que o usuário interaja.

**300.2 Múltipla Escolha**

Esse código de status de redirecionamento indica que o agente de usuário pode escolher um dos recursos solicitados disponíveis. Ele aparece quando há várias extensões de nome de arquivo ou ambiguidade na solicitação.

**301 Movido Permanentemente**

Esse código de status geralmente aparece com redirecionamentos de página permanentes. Ele comunica a realocação permanente do recurso para outra URL, indicada nos cabeçalhos de resposta Location.

Como a realocação é permanente, os mecanismos de pesquisa redirecionarão todo o tráfego da página antiga para a nova quando os rastreadores web encontrarem esse código de status. Por esse motivo, quem tem um site costuma usar esse código de status ao transferir ou alterar seu domínio, ou mesclar dois sites.

**302 Encontrado**

Assim como no código de status 301, o código 302 redireciona um usuário para outro URL. A diferença está na realocação, a qual é apenas temporária. Ela solicita que o navegador processe solicitações futuras usando a URL antiga.

**303 Consulte Outro**

O servidor enviou esta resposta para direcionar o cliente a obter o recurso solicitado em outro URI com uma solicitação GET.

**304 Não Modificado**

O servidor está informando ao navegador que o recurso não mudou desde que foi armazenado em cache. Portanto, o navegador pode acessar seu cache local para o recurso solicitado, reduzindo o tempo de carregamento da página.

**305 Use Proxy**

Definido em uma versão anterior da especificação HTTP para indicar que uma resposta solicitada deve ser acessada por um proxy. Foi descontinuado devido a questões de segurança em relação à configuração em banda de um proxy

.**307 Redirecionamento Temporário**

O servidor envia esta resposta para direcionar o cliente a obter o recurso solicitado em outra URI com o mesmo método usado na solicitação anterior. Tem a mesma semântica do código de resposta HTTP 302 Found, com a exceção de que o agente do usuário não deve alterar o método HTTP usado: se um POST foi usado na primeira solicitação, um POST deve ser usado no segundo pedido.

**308 Redirecionamento Permanente**

O Redirecionamento Permanente funciona de forma semelhante ao código 301 Movido Permanentemente. No entanto, apresenta as mesmas características do código 307, pois não altera o método HTTP e nem o corpo da mensagem original.

**400 Solicitação Inválida**

O servidor não pode ou não irá processar a solicitação devido a algo que é percebido como um erro do cliente (por exemplo, sintaxe de solicitação malformada, enquadramento de mensagem de solicitação inválida ou roteamento de solicitação enganosa)

**401 Não Autorizado**

Embora o padrão HTTP especifique "unauthorized", semanticamente, essa resposta significa "unauthenticated". Ou seja, o cliente deve se autenticar para obter a resposta solicitada.

**402 Pagamento Necessário**

Este código de resposta está reservado para uso futuro. O objetivo inicial da criação deste código era usá-lo para sistemas digitais de pagamento, no entanto, este código de status é usado raramente e não existe nenhuma convenção padrão.

**403 Proibido**

erro 403 indicando que uma solicitação não foi autorizada

O código 403 tem um propósito semelhante à resposta 401 Não Autorizado. Ele indica uma recusa em autorizar a solicitação mesmo com credenciais de login válidas. Portanto, o usuário não deve repetir a solicitação.

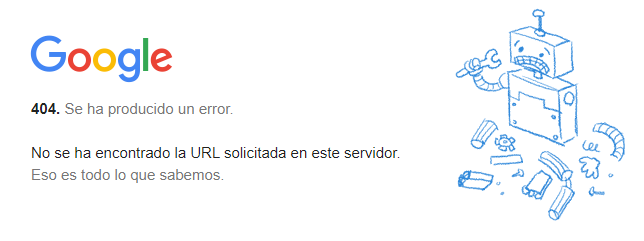
As causas mais comuns são as permissões limitadas por exemplo, um usuário com função de Escritor tentando acessar páginas apenas para Editores.

**404 Não Encontrado**

Erro 404 da página informando falha ao localizar recurso, esse código de status informa ao usuário que o servidor falhou ao localizar o recurso solicitado. No entanto, ele não consegue identificar se o recurso está ausente de forma temporária ou permanentemente.

A mensagem de erro varia dependendo do navegador. Muitos sites por exemplo, personalizam a mensagem incluindo uma breve explicação sobre o erro. Algumas das causas mais comuns são URL digitados incorretamente, problemas de armazenamento em cache e propagação de domínio incompleta.

**Figura 3**



**Fonte:** Suport.google.com

**408 Tempo de Solicitação Esgotado**

O servidor de origem pode enviar este código de resposta se um usuário não apresentar um pedido dentro do tempo que o servidor estava preparado para esperar. Como esse erro é causado principalmente por conexões lentas de internet ou URLs incorretos, o usuário pode repetir a solicitação em outro momento sem alterar o conteúdo.

**409 Conflito**

O servidor não pode concluir a solicitação devido a um conflito com o recurso de destino. Esse erro pode ser causado quando vários usuários atualizam um arquivo simultaneamente ou o arquivo recém-carregado ser mais antigo que o armazenado no servidor, criando conflitos de versão.

**410 Perdido**

Este código de status é semelhante ao 404 Não Encontrado. A diferença é que o 410 indica que o recurso está ausente no servidor de origem permanentemente e que não há redirecionamento.

**411 Duração Necessária**

Um servidor pode enviar essa resposta quando a solicitação está sem seu cabeçalho Content-Length.

**415 Tipo de Mídia Não Suportado**

O servidor não suporta o tipo de mídia incluído na entidade de solicitação do cliente.

**417 Falha na Expectativa**

O servidor não pode atender aos requisitos declarados no campo Expect request-header.

**419 Tempo Limite de Autenticação**

Esse código de status não faz parte do padrão HTTP e ele ocorre quando uma autenticação válida expira. Ele serve como uma alternativa ao código de status 401 Não Autorizado.

**422 Entidade Não Processável**

A solicitação foi bem formada, mas não pôde ser atendida devido a erros semânticos.

**423 Bloqueado**

O recurso de origem ou destino de um método está bloqueado. Este código de erro faz parte da especificação WebDAV (link em inglês), seu mecanismo de bloqueio.

**424 Falha de Dependência**

A solicitação falhou devido à falha de uma solicitação anterior.

**426 Atualização Necessária**

O cliente deve atualizar para um protocolo diferente. O servidor deve enviar um campo de cabeçalho de atualização contendo o protocolo solicitado.

**440 Tempo de Login Excedido**

O cliente deve fazer login novamente porque sua sessão expirou.

**444 Sem Resposta**

Esse código de status não padrão aparece apenas nos arquivos do NGINX, instruindo o servidor a se desconectar sem enviar uma resposta ao cliente. É usado para negar solicitações maliciosas.

**449 Tente Novamente**

O servidor não pode processar a solicitação porque o cliente não forneceu as informações necessárias.

**451 Indisponível por Motivos Legais**

O servidor não pode acessar o recurso devido a restrições legais do governo.

**500 Erro no Servidor**

O servidor encontrou uma situação com a qual não sabe lidar.

**501 Não Implementado**

O método da requisição não é suportado pelo servidor e não pode ser manipulado. Os únicos métodos que servidores devem suportar (e portanto não devem retornar este código) são GET e HEAD.

**503 Serviço Indisponível**

O servidor não pode lidar com a solicitação porque ficou sem recursos ou está em manutenção.

Os servidores geralmente enviam esse código de resposta devido a problemas temporários. A resposta vem com o campo de cabeçalho de solicitação Retry-After e o tempo de recuperação estimado.

**505 Versão HTTP Não Suportada**

A versão HTTP usada na requisição não é suportada pelo servidor.

**507 Espaço Insuficiente de Armazenamento**

O método não pôde ser executado no recurso porque o servidor não pode armazenar a representação necessária para concluir a solicitação com êxito.

**510 Não Estendido**

Extensões adicionais à solicitação são necessárias para que o servidor a atenda.

**511 Autenticação de Rede Necessária**

Indica que o cliente precisa se autenticar para obter acesso à rede.

**521 Servidor Inativo**

Este código de resposta da Cloudflare aparece quando o servidor de origem não consegue se conectar ao proxy reverso. Pode ser porque o servidor de origem está bloqueando conexões de endereços IP específicos da Cloudflare.

1. **CONCLUSÃO**

O protocolo HTTP (Hypertext Transfer Protocol) desempenha um papel fundamental na comunicação entre clientes e servidores na World Wide Web. Ao longo das décadas, o HTTP evoluiu para atender às demandas crescentes da internet, tornando-se mais eficiente, seguro e versátil. Através do HTTP, é possível transferir informações de forma confiável e padronizada, permitindo o acesso a uma infinidade de recursos online. Sua importância transcende as fronteiras da tecnologia, impactando diretamente a maneira como interagimos, compartilhamos dados e acessamos conteúdos. Assim, o HTTP continua a ser um pilar essencial para a conectividade e o funcionamento da web como a conhecemos hoje.

1. **BIBLIOGRAFIA**

<https://rockcontent.com/br/blog/http/> Escrito por: [Ivan de Souza](https://rockcontent.com/br/blog/author/ivan/)

<https://developer.mozilla.org/ptBR/docs/Web/HTTP/Methods>

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status>