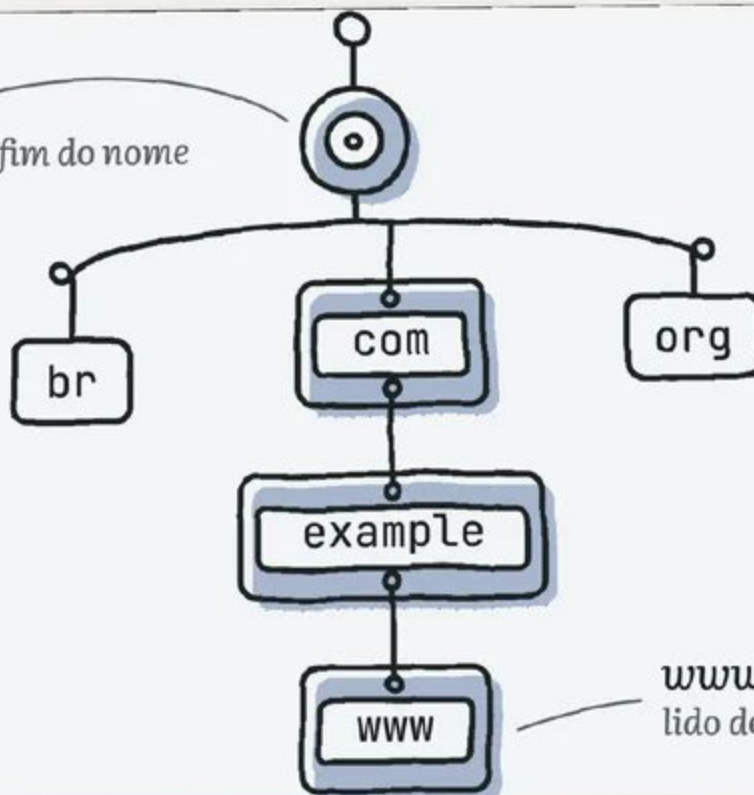


Registros de DNS

A, AAAA, CNAME, MX, TXT e o domínio no GitHub Pages

a raiz
o ponto no fim do nome



www.example.com.
lido de baixo para cima

O DNS traduz um nome no endereço IP

O DNS (Domain Name System) traduz um nome como `www.example.com` no endereço IP de que o navegador precisa para abrir a conexão.

Toda configuração de domínio é feita nele, na forma de **registros**: o site, o e-mail, a prova de que o domínio é seu e quem pode emitir certificado para ele.

Quando um site novo não abre ou o e-mail do domínio cai no spam, a causa costuma estar num desses registros.

porta 53

as consultas usam UDP e TCP; o suporte aos dois é obrigatório em toda implementação de uso geral

antes do DNS

um arquivo só

O `HOSTS.TXT`, mantido por uma central e copiado por FTP por todas as máquinas da rede. A cada máquina nova, o arquivo crescia, e todo mundo precisava baixar de novo.

o DNS

base de dados distribuída

Nenhum servidor conhece todos os nomes. Cada parte da base é mantida por quem é dono dela, e um servidor sabe apontar para o servidor que conhece a parte seguinte.

O nome é um caminho até a raiz da árvore

Cada nó tem um rótulo de até 63 bytes, e o nome completo tem até 255. A raiz tem o rótulo vazio: por isso um nome escrito por inteiro termina em ponto.

o ponto é a raiz

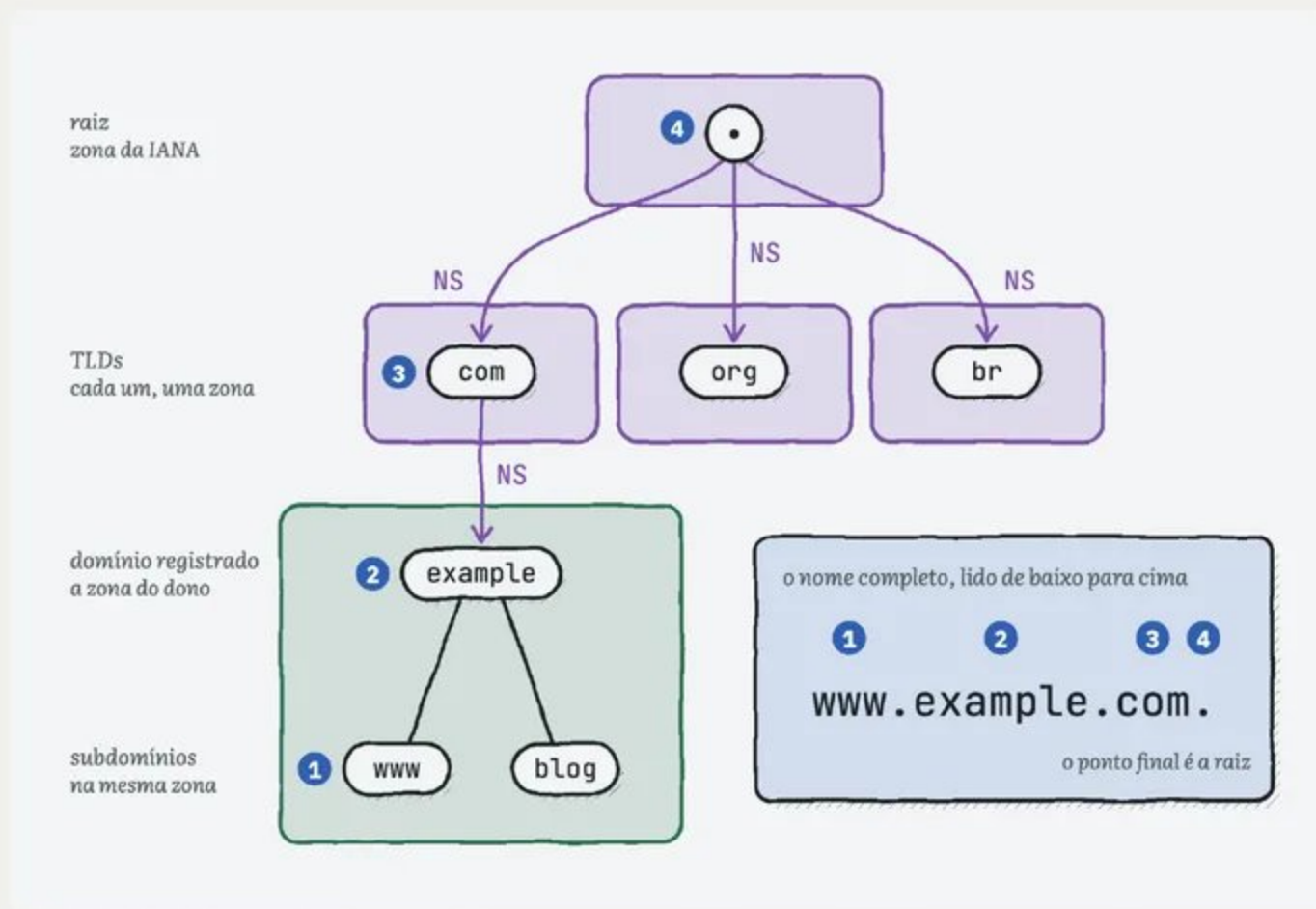
`www.example.com.`: do nó `www` até a raiz, lido da direita para a esquerda.

A raiz (.): a zona dela lista os domínios de primeiro nível.

Os TLDs `com`, `org`, `br`, `io`: 1.437 na lista da IANA em 04/10/2026.

O domínio registrado `example.com`, `exemplo.com.br`.

Os subdomínios `www`, `blog`: criados pelo dono do domínio, sem pedir nada a ninguém.



Cada contorno é uma zona: em roxo, a raiz e os TLDs, que só delegam; em verde, o domínio registrado. As setas são as delegações por NS.

Domínio e zona são coisas diferentes.

Um **domínio** é um galho da árvore, com tudo o que está embaixo dele.

Uma **zona** é o pedaço da árvore que um mesmo conjunto de servidores responde, com autoridade.

A zona com não guarda os endereços de `example.com`: guarda só um recado, quem responde por ele. Esse recado é a **delegação**, feita com registros NS.

Ao comprar um domínio, o registro do TLD grava na zona dele os NS que você indicou.

No `.br`, é o Registro.br; a IANA delega o `.br` ao Comitê Gestor da Internet no Brasil.

O resolvedor sempre começa pela raiz

Ao abrir `www.example.com`, o sistema operacional não sai perguntando para a árvore. Ele pergunta a um **resolvedor recursivo**: o do provedor de internet, o da empresa ou um público, como o do Google ou o da Cloudflare.

O resolvedor faz o trabalho de descer a árvore, guarda o que aprendeu e sempre sabe por onde começar: os servidores raiz.

não 13 máquinas

13 identidades

de `a.root-servers.net` a
`m.root-servers.net`

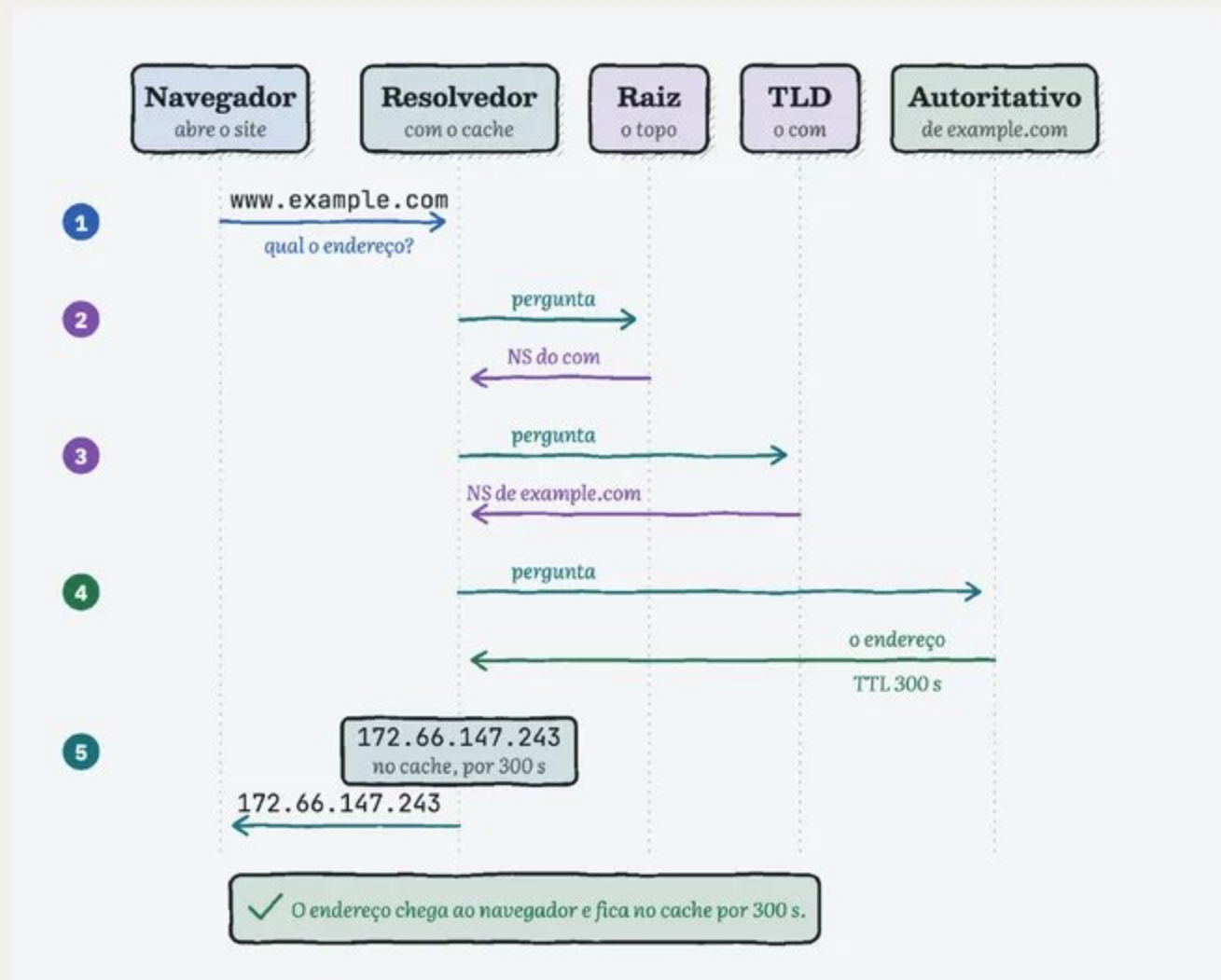
12 organizações

independentes operam os servidores
raiz

mais de 2.000

instâncias, no `root-servers.org` em
04/10/2026: cada identidade é servida
por muitas máquinas

Da raiz ao autoritativo, em cinco passos



- 1 O navegador pergunta ao resolvedor o endereço de `www.example.com`.
- 2 A raiz não sabe o endereço, mas indica os servidores do `com`.
- 3 O servidor do `com` indica os servidores de `example.com`.
- 4 O servidor autoritativo de `example.com` responde com o endereço.
- 5 O resolvedor guarda a resposta pelo TTL e entrega ao navegador.

Azul é o navegador, petróleo o resolvedor, roxo os servidores que só indicam o caminho e verde o autoritativo, que tem a resposta.

O dig +trace mostra cada degrau

Cada servidor responde só o que sabe. A raiz e o servidor do com não têm o endereço e respondem com uma **indicação** (referral): os NS do nível de baixo.

Só o **autoritativo** de example.com tem a resposta final, com o **TTL**, em segundos: o tempo que o resolvedor guarda a resposta.

A próxima pergunta, de qualquer pessoa que use o mesmo resolvedor, sai do cache sem passar pela árvore.

```
dig +trace www.example.com

.                600      IN    NS    a.root-servers.net.
.                600      IN    NS    m.root-servers.net.
;; Received 811 bytes from fe80::1%14#53(fe80::1%14)

com.             172800   IN    NS    a.gtld-servers.net.
com.             172800   IN    NS    m.gtld-servers.net.
;; Received 840 bytes from 192.112.36.4#53(g.root-servers.net)

example.com.     172800   IN    NS    hera.ns.cloudflare.com.
example.com.     172800   IN    NS    elliottns.cloudflare.com.
;; Received 363 bytes from 192.12.94.30#53(e.gtld-servers.net)

www.example.com. 300       IN    A     172.66.147.243
www.example.com. 300       IN    A     104.20.23.154
;; Received 76 bytes from 162.159.44.228#53(elliottns.cloudflare.com)
```



A saída real de 04/10/2026, resumida: o resolvedor local deu a lista da raiz, a raiz indicou o com, o com indicou a Cloudflare, e a Cloudflare deu o endereço.

Uma zona lista os registros do domínio

example.com.zone

```
$ORIGIN example.com.  
$TTL 3600
```

; a zona: quem responde por ela e os tempos

```
@      IN  SOA  ns1.example.net. hostmaster.example.com. (  
                2026100401 ; serial  
                7200      ; refresh  
                900       ; retry  
                1209600   ; expire  
                300      ) ; minimum (cache negativo)
```

```
@      IN  NS  ns1.example.net.
```

```
@      IN  NS  ns2.example.net.
```

; o site no GitHub Pages: o domínio principal e o www

```
@      IN  A    185.199.108.153  
@      IN  A    185.199.109.153  
@      IN  A    185.199.110.153  
@      IN  A    185.199.111.153  
@      IN  AAAA  2606:50c0:8000::153  
@      IN  AAAA  2606:50c0:8001::153  
@      IN  AAAA  2606:50c0:8002::153  
@      IN  AAAA  2606:50c0:8003::153
```

```
| www      IN  CNAME  usuario.github.io. ← com ponto: nome absoluto  
_github-pages-challenge-usuario IN TXT "a1b2c3d4e5f6"
```

Cada registro tem cinco partes:

nome	www, ou @ para o próprio domínio
TTL	em segundos; sem ele, o \$TTL
classe	na prática, sempre IN, de internet
tipo	A, CNAME, MX, TXT...
dado	o endereço, o nome ou o texto

O @ é o próprio domínio (o *apex*), e um nome sem ponto final, como `www`, ganha o domínio no fim: vira `www.example.com`.

Em 04/10/2026, essa zona foi carregada no Unbound 1.24.2, e cada registro do post foi consultado com `dig`.

zona testada no Unbound 1.24.2

Dez tipos, e cada um guarda uma coisa

Tipo	O que guarda	Uso típico
A	um endereço IPv4	o site, a API
AAAA	um endereço IPv6	o mesmo, em IPv6
CNAME	outro nome, do qual este é apelido	www apontando para um serviço de hospedagem
MX	o servidor que recebe e-mail, com prioridade	o e-mail do domínio
TXT	texto livre	SPF, DKIM, DMARC, verificação de domínio
NS	os servidores que respondem pela zona	a delegação
SOA	os dados da zona: dono, serial e tempos	um por zona, no apex
CAA	quem pode emitir certificado	proteger o HTTPS
SRV	servidor e porta de um serviço	protocolos como SIP e IMAP
PTR	o nome de um endereço	DNS reverso

A e AAAA dão o endereço; CNAME, o apelido

A e AAAA

o endereço

O A guarda um endereço IPv4, de 32 bits, e o AAAA, um IPv6, de 128 bits. Um nome pode ter vários de cada: o cliente recebe todos e escolhe um, o que já reparte a carga e dá alternativa se um endereço falhar.

CNAME

um nome que aponta para outro

Ao perguntar por `www.example.com`, o resolvedor recebe “`www.example.com` é `usuario.github.io`” e repete a pergunta para o nome novo. Quem cuida do nome canônico troca os endereços sem você mexer na sua zona.

A regra que mais pega, da RFC 1034 e da RFC 2181: **um nome com CNAME não pode ter nenhum outro registro.**

Como o apex sempre tem SOA e NS, **o domínio principal não pode ser um CNAME**. Por isso, no GitHub Pages, o `www` usa CNAME, e o `example.com` usa A e AAAA.

MX e TXT servem a outros; NS e SOA, à zona

MX

quem recebe o e-mail

Cada servidor tem uma preferência: **o menor número é tentado primeiro**. Com 10 `mx1` e 20 `mx2`, o `mx2` só entra se o `mx1` não responder. O alvo precisa ter endereço próprio, nunca um CNAME.

TXT

texto que outros sistemas leem

Guarda o SPF, o DKIM, o DMARC e as **verificações de domínio**: o GitHub, o Google e os serviços de e-mail pedem um código num nome combinado, para provar que o domínio é seu. Só o dono da zona consegue criar o registro.

NS

a delegação

Lista os servidores que respondem por uma zona. Aparece na zona de cima (o com dizendo quem responde por `example.com`) e na própria zona, que repete a lista.

SOA

start of authority

Abre a zona, e há um só: o servidor principal, o e-mail do responsável, o serial e quatro tempos. O minimum é o TTL do **cache negativo**:
`nao-existe.example.com` voltou com TTL de 300.

o menor entre 300 e 3600

Os menos comuns, e dois fora do padrão

CAA quem emite certificado

Com 0 issue "letsencrypt.org", só a Let's Encrypt pode; `issuewild` vale para certificados curinga. Sem CAA, qualquer uma pode.

SRV servidor e porta

Num nome `_servico._protocolo.dominio`, com prioridade e peso. É como clientes de SIP, XMPP e e-mail descobrem onde se conectar.

PTR do endereço para o nome

192.0.2.10 vira `10.2.0.192.in-addr.arpa`, com os números invertidos. Quem cria é o dono do bloco de IPs (o provedor), não o dono do domínio.

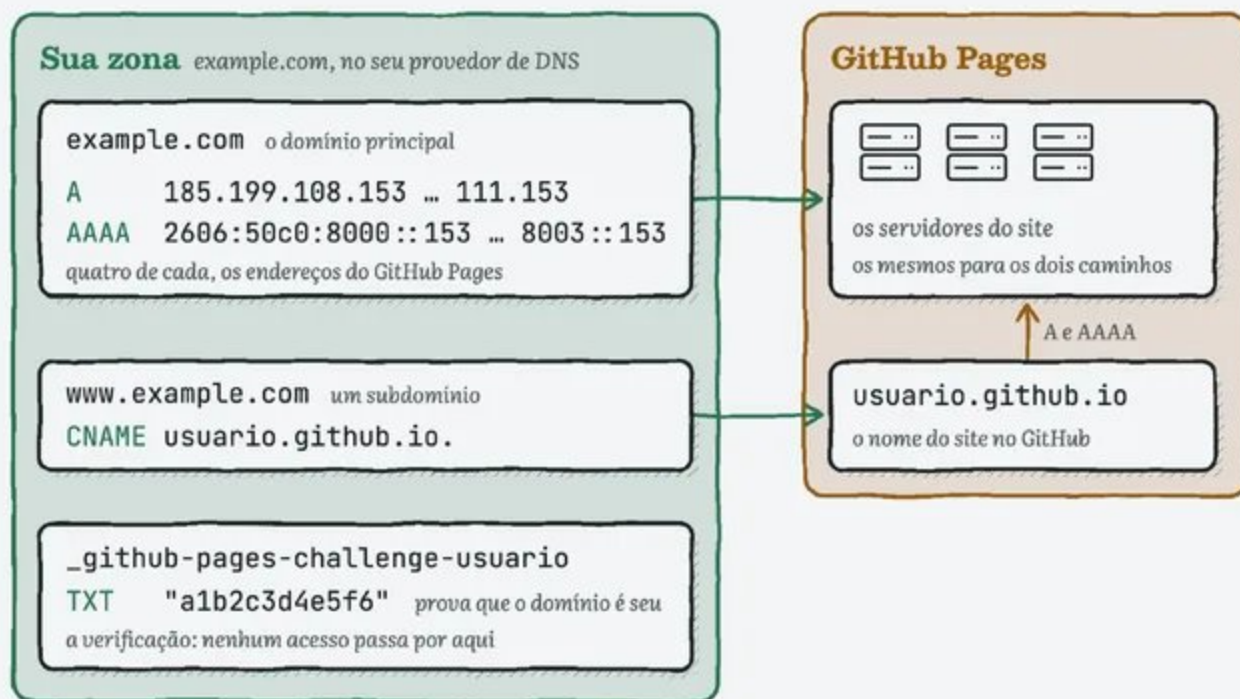
HTTPS e SVCB a conexão na consulta

Entregam ao navegador, na mesma consulta, os parâmetros da conexão, como o suporte a HTTP/3, e aceitam apelido no apex, o que o CNAME não faz.

ALIAS e ANAME recurso do provedor

Os dois não são tipos do padrão: o rascunho do ANAME expirou em 2021. O provedor resolve o destino e publica os endereços no apex; o nome e o comportamento mudam de um para outro.

O apex usa A e AAAA; o www, um CNAME



O domínio principal: quatro A e quatro AAAA, com os endereços do GitHub Pages. Com ALIAS ou ANAME no provedor, ele pode apontar para usuario.github.io.

Um subdomínio: um CNAME para usuario.github.io, sem o nome do repositório, mesmo num repositório de projeto.

Com os dois configurados, o GitHub redireciona um para o outro. O endereço vai em **Settings** → **Pages** → **Custom domain**.

Por branch, o domínio vai num arquivo CNAME na raiz dela; por GitHub Actions, um que exista é ignorado. *

** o caso deste blog (Actions)*

Verde é a sua zona e âmbar o GitHub; o TXT de verificação fica embaixo, sem seta, e não leva tráfego a lugar nenhum.

Verifique o domínio e ligue o HTTPS

verificar o domínio

um TXT

Em **Settings** → **Pages** da conta (não do repositório), o GitHub dá um código para publicar num TXT em
`_github-pages-challenge-usuario.example.com`.

Com o domínio verificado, outra conta não consegue publicar um site nele.

ligar o HTTPS

Let's Encrypt

O GitHub pede o certificado à Let's Encrypt sozinho, e o **Enforce HTTPS** faz todo acesso por HTTP ir para o HTTPS.

A ou AAAA a mais no apex, apontando para outro lugar, atrapalham a emissão, e um CAA precisa incluir letsencrypt.org



Cuidado

Não use um registro curinga (*.example.com) apontando para o GitHub Pages. A documentação avisa que ele abre espaço para alguém tomar subdomínios seus, mesmo com o domínio verificado.

Receber pede o MX; enviar bem, três TXT

example.com.zone

```
; o e-mail
@      IN  MX    10 mx1.example.net.
@      IN  MX    20 mx2.example.net.
@      IN  TXT    "v=spf1 include:_spf.example.net -all"
s1._domainkey IN TXT "v=DKIM1; k=rsa; p=MIIBIjANBgkqh...IDAQAB"
_dmarc  IN  TXT    "v=DMARC1; p=quarantine; rua=mailto:dmarc@example.com"
```

10 consultas

ao DNS, no máximo, na avaliação do SPF, e cada **include** conta

fácil de estourar

SPF

quem pode enviar

Um TXT no apex, começando por v=spf1: o include aceita os servidores do provedor, e o -all diz que o resto falha (o ~all só marca como suspeito).

DKIM

a chave pública

Confere a assinatura das mensagens, num TXT em seletor._domainkey.example.com. O seletor (s1) permite uma chave por serviço que envia.

DMARC

o que fazer na falha

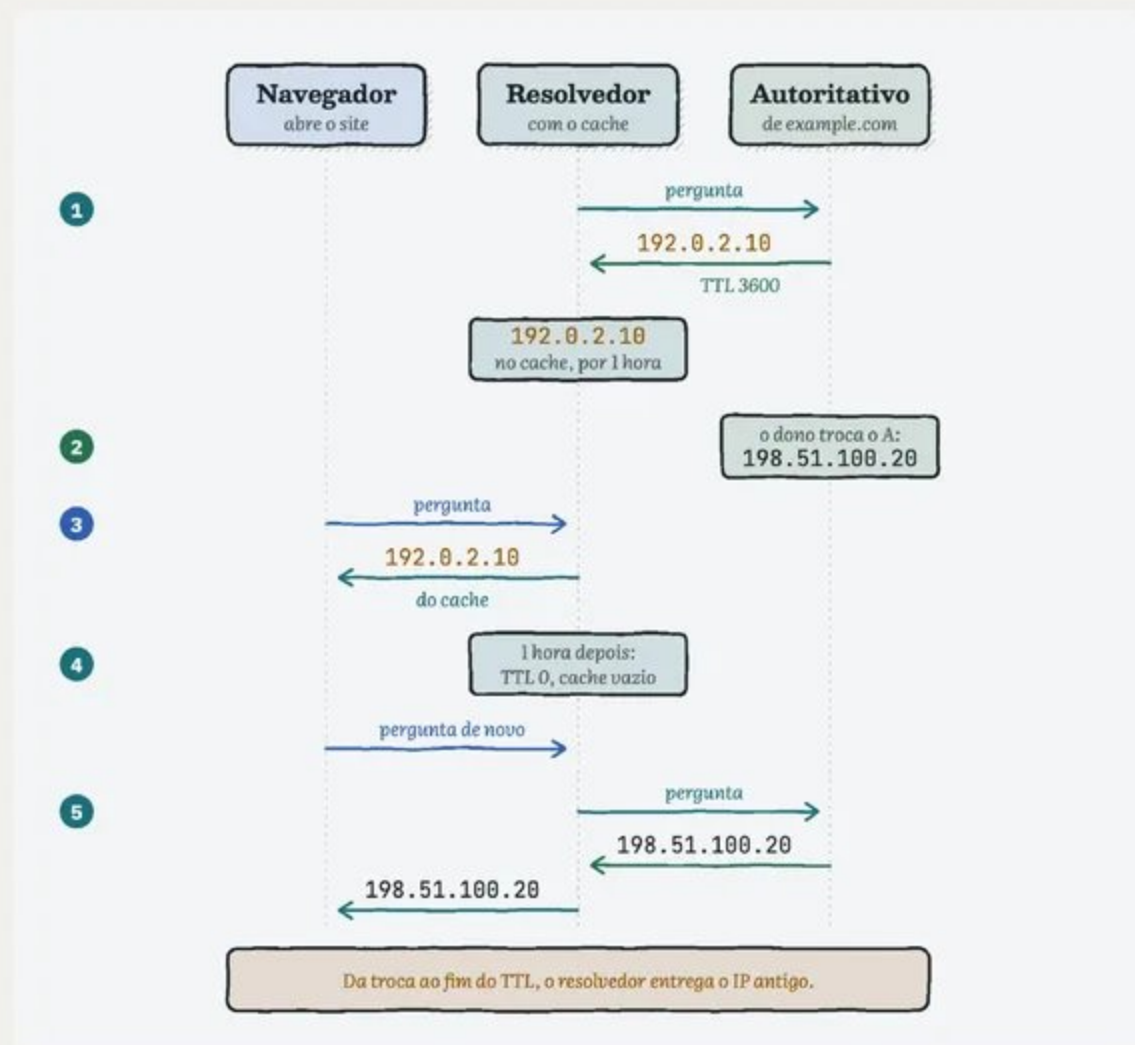
p=none, p=quarantine ou p=reject, e os relatórios em rua, num TXT em _dmarc.example.com. Padrão da IETF desde maio de 2026: a RFC 9989, que substituiu a ~~RFC 7489~~.

O DMARC observa antes de recusar

- ✓ O caminho comum é começar o DMARC com `p=none`, ler os relatórios por algumas semanas para descobrir quem mais envia em nome do domínio (o sistema de cobrança, a ferramenta de newsletter) e só então subir para `quarantine` e `reject`.



A propagação é o cache expirando



A documentação do GitHub fala em até 24 horas. Mas nada é empurrado para os servidores do mundo: o autoritativo responde o valor novo na hora, e cada resolvidor continua entregando o valor antigo, do cache, até o TTL dele acabar.

Três hábitos

Baixe o TTL antes da troca.

Um dia antes, para 300 segundos: os caches seguram o valor antigo por no máximo cinco minutos.

Limpe o cache de um resolvidor público.

O Google Public DNS tem uma página para limpar o cache de um nome.

Lembre do cache negativo.

Consultar um nome antes de criá-lo guarda o “não existe” pelo minimum do SOA.

O dig com +short mostra só a resposta

Terminal

```
dig example.com A +short
dig www.example.com CNAME +short
dig example.com MX +short
dig _dmarc.example.com TXT +short
dig @8.8.8.8 example.com A +short
```

Terminal: as respostas, na zona de exemplo

```
$ dig example.com MX +short
10 mx1.example.net.
20 mx2.example.net.

$ dig www.example.com CNAME +short
usuario.github.io.

$ dig _dmarc.example.com TXT +short
"v=DMARC1; p=quarantine; rua=mailto:dmarc@example.com"
```

A última pergunta direto ao resolvedor do Google, sem passar pelo da sua rede.

Comparar a resposta do autoritativo (dig @ns1.example.net ...) com a de um resolvedor mostra se a diferença é cache ou erro na zona.

Quando o site não abre, dig +trace mostra em que degrau a resposta muda: se a delegação do TLD aponta para os servidores certos, e se o autoritativo responde o que você espera.

Cinco erros que derrubam o site ou o e-mail

o ponto final

1/5

Esquecer o ponto final.

Na zona, `www IN CNAME usuario.github.io`, sem o ponto, vira `usuario.github.io.example.com.`, que não existe. Confira a resposta com `dig`.

CNAME e outro registro

2/5

CNAME junto de outro registro.

O servidor nem sempre recusa: o Unbound 1.24.2 carregou, sem aviso, CNAME e TXT no mesmo nome.

dois SPF

3/5

Dois registros SPF.

Dois TXT com `v=spf1` no mesmo nome dão erro permanente, e a verificação falha para todo e-mail.

MX para CNAME

4/5

MX apontando para CNAME.

O padrão proíbe, e alguns servidores de e-mail recusam a entrega.

esquecidos

5/5

Registros esquecidos.

Um A antigo no apex, ao lado dos quatro do GitHub, desvia visitas e trava o certificado. Um subdomínio apontando para um serviço desligado pode ser tomado.

Para ir além

- ✓ Quando um domínio não abre, dig `+trace` é o primeiro comando: mostra a consulta inteira, da raiz até a resposta.

[bind9.readthedocs.io](#) > [manpages](#) > `dig`

[docs.github.com](#) > [Pages](#) > [managing a custom domain](#)

As fontes do post

RFC 1034 e RFC 1035: o espaço de nomes, as zonas e os registros

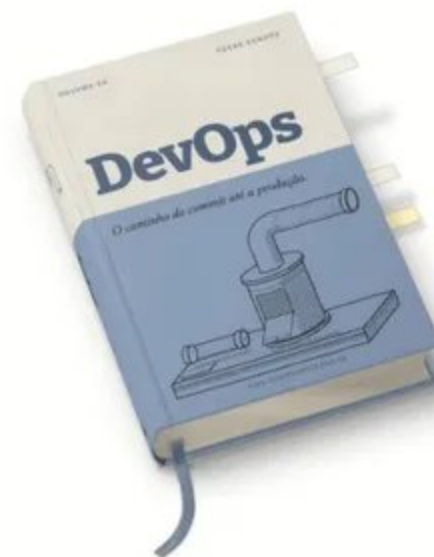
RFC 2181 e RFC 2308: o TTL, o CNAME exclusivo e o cache negativo

RFC 7208, RFC 6376 e RFC 9989: SPF, DKIM e DMARC

IANA, root-servers.org e Google Public DNS

GitHub Docs: domínio personalizado, verificação e HTTPS

Do livro



Volume 04 • 3 artigos

DevOps

[Ver o livro](#) →

[blog.cesarschutz.com.br](#)

cs Cesar Schutz *blog*