

# DNS

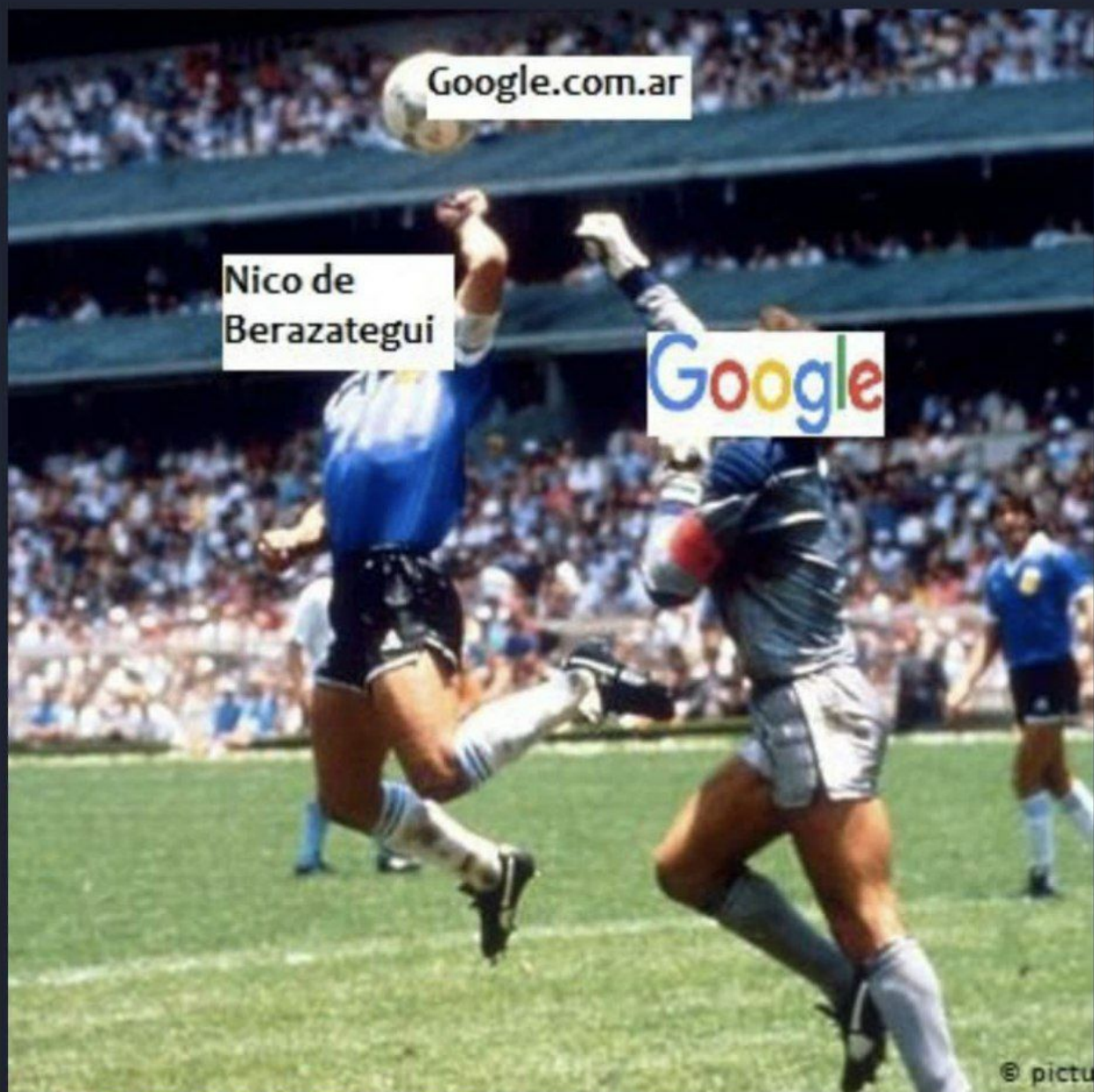
## **Introducción a los Sistemas Distribuidos (75.43)**

Universidad de Buenos Aires, Facultad de Ingeniería

Google.com.ar

Nico de  
Berazategui

Google



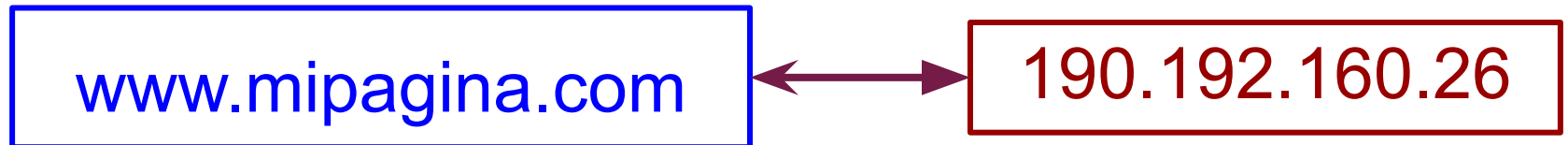
© pictu

# Agenda de hoy

- Repaso DNS
  - Objetivo
  - Servicios
  - Estructura (infra + mensajes)
- Ejemplo
- eDNS + ECS
- Herramientas → dig

# Repaso DNS

- DNS -> **Domain Name System**
- Directorio dónde se almacenan las direcciones IP para cada hostname



# Repaso DNS - Objetivo

Poder identificar con (o “traducir a”) con nombres “fáciles”, a distinta información asociada a componentes de Internet.

Ejemplo:

192.6.85.23 → pepito.com

# Repaso DNS - Características

- A nivel infraestructura
  - Base de datos distribuida y jerárquica
  - Está contenida en varios servidores alrededor del mundo
- En el stack TCP/IP
  - Protocolo de capa de aplicación
  - Función: proveer IP para hostnames
  - Típicamente usa el puerto UDP : 53

# Repaso DNS - Estructura RR

- La base de datos distribuida de DNS guarda resource records (RR)
- Los RR son tuplas de 4 elementos:
  - **Name**: Nombre del dominio (ej: gmail.com)
  - **Type** → A, AAAA, CNAME, MX, ...
  - **Value/Data** → Ej: 190.192.8.64 (type A)
  - **TTL**

⇒ En Firefox, para ver cache table, en la barra:

***about:networking#dns***

# Repaso DNS - Estructura RR

## Tipos de registros comunes:

- **A** → dirección IP
- **AAAA** → dirección IPv6
- **TXT** → registro de texto
- **MX** → intercambio de emails
- **NS** → servidor de nombres
- **CNAME** → registro de nombre canónico (alias)

⇒ En este [link](#) se encuentra una lista completa de los tipos de registro.



# Repaso DNS - Servicios

DNS provee varios servicios:

- Host aliasing
- Mail aliasing
- Load balancing

# Repaso DNS - Servicios

DNS provee varios servicios:

- Host aliasing
- Mail aliasing
- Load balancing



# Repaso DNS - Servicios

El mismo servicio puede tener varios host names.

⇒ Se resuelve con el registro **CNAME** (canonical name).

**CNAME** ~ puntero

name	class	type	data	TTL
www.reddit.com	IN	CNAME	reddit.map.fastly.net	30
reddit.map.fastly.net	IN	A	199.232.9.140	30

# Repaso DNS - Servicios

El mismo servicio puede tener varios host names.

⇒ Se www.reddit.com es una forma más simple de reddit.map.fastly.net

name	class	type	data	TTL
www.reddit.com	IN	CNAME	reddit.map.fastly.net	30
reddit.map.fastly.net	IN	A	199.232.9.140	30

# Repaso DNS - Servicios

DNS provee varios servicios:

- Host aliasing
- Mail aliasing
- Load balancing



# Repaso DNS - Servicios

Los usuarios quieren mails fáciles de recordar (*@gmail.com*) pero los servidores tienen IPs difíciles.

⇒ Se resuelve con el registro **MX**

name	class	type	data	TTL
gmail.com	IN	MX	30 alt3.gmail-smtp-in.l.google.com	13
gmail.com	IN	MX	40 alt4.gmail-smtp-in.l.google.com	13

# Repaso DNS - Servicios

Los usuarios quieren mails fáciles de recordar  
(@gmail.com)  
difícil

**Gmail apunta a varios -> MX  
permite brindar correo con  
multiples servers**

⇒ Se resuelve con el registro **MX**

name	class	type	data	TTL
gmail.com	IN	MX	30 alt3.gmail-smtp-in.l.google.com	13
gmail.com	IN	MX	40 alt4.gmail-smtp-in.l.google.com	13

# Repaso DNS - Servicios

DNS provee varios servicios:

- Host aliasing
- Mail aliasing
- Load balancing





# Repaso DNS - Servicios

**Load balancing:** distribuir la carga tráfico de red de un servicio en más de un servidor

- Permite generar redundancia
  - Si tengo 3 servers y se cae uno, la app sigue funcionando
- Permite distribuir la carga
  - Si mucha gente usa el servicio, hay varios servers para dar mayor poder de procesamiento

# Repaso DNS - Servicios

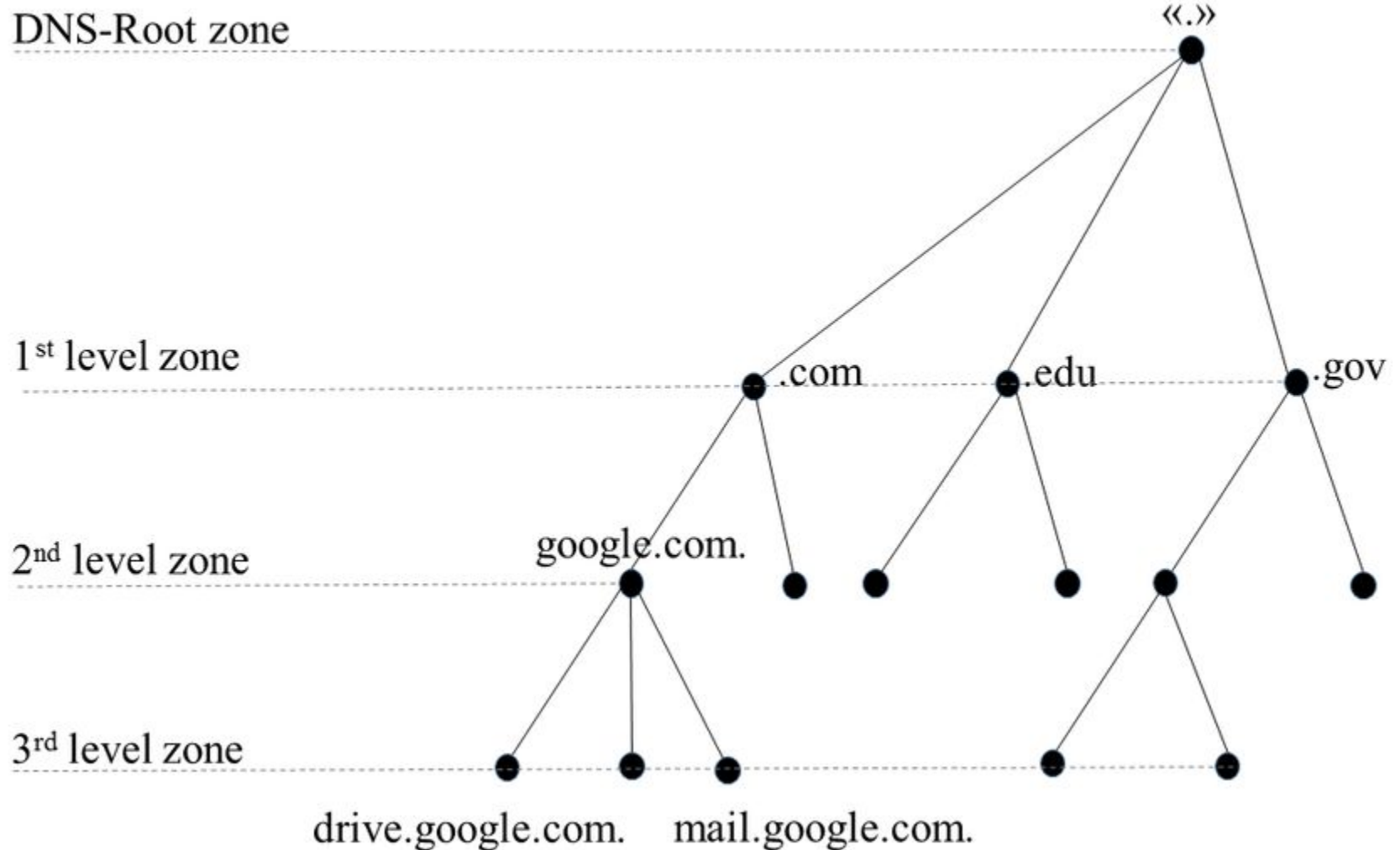
## **¿Cómo es que DNS nos ofrece esto?**

- DNS devuelve todas las IPs existentes para un dado dominio
- Si consultamos varias veces, nos las vuelve a devolver pero con el orden cambiado

# Repaso DNS - Infraestructura

- Descentralizada → ¿Por qué?
  - Single point of failure
  - Latencia a todo el mundo
  - Hardware costoso
- Jerárquica → Responsabilidad y DB distribuida
  - Root servers
  - Top level domain servers (.com, .ar)
  - Servidores autoritativos

# Repaso DNS - Infraestructura



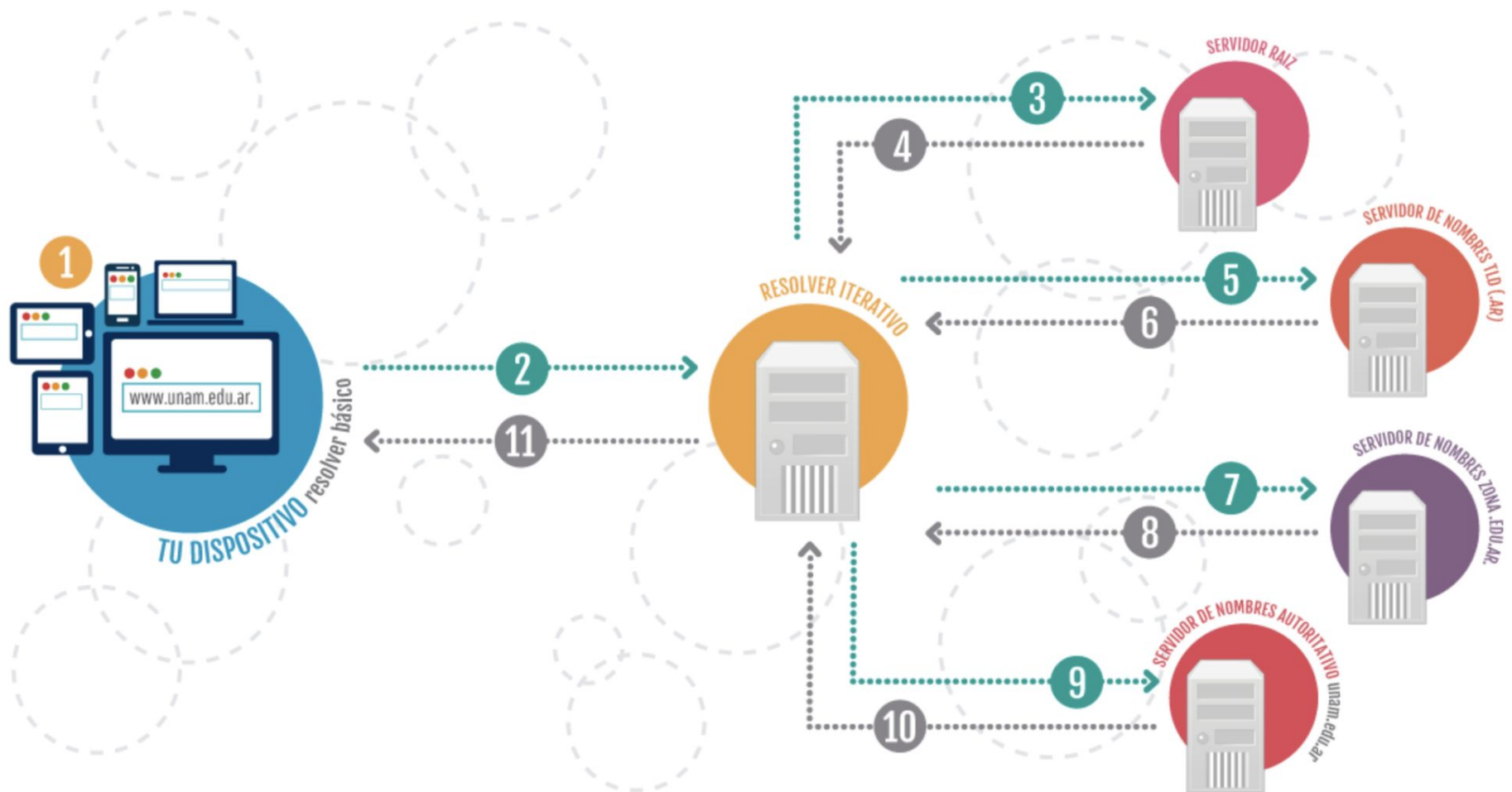
Cons #1	Se envía la query (¿Cuál es la IP de <a href="http://www.youtube.com">www.youtube.com</a> ?) al servidor raíz
RTA #1	El servidor raíz responde con la dirección IP del servidor TLD
Cons #2	Se envía la query (¿Cuál es la IP de <a href="http://www.youtube.com">www.youtube.com</a> ?) al servidor TLD
RTA #2	El servidor TLD responde con la dirección IP del servidor autoritativo
Cons #3	Se envía la query (¿Cuál es la IP de <a href="http://www.youtube.com">www.youtube.com</a> ?) al SA
RTA #3	El SA finalmente responde la dirección IP de <a href="http://www.youtube.com">www.youtube.com</a>

# Repaso DNS - Infraestructura

Un componente mas → Local resolver

- Todos los ISPs / organizaciones tienen uno
- Es cercano a los usuarios
- Funcionamiento:
  - El usuario realiza la query al resolver (comportamiento recursivo)
  - Éste inicia el proceso iterativo, comenzando con el servidor raíz
  - Cuando el SA finalmente le contesta la IP, la envía al usuario

# Repaso DNS - Infraestructura



# Repaso DNS - Infraestructura

Muchas queries similares, poco eficiente...

**⇒ DNS Caches**

- Se almacenan en memoria los resultados de las queries a los SA
- Ante nuevas consultas para los mismos hostnames, se devuelve la respuesta cacheada
  - Reducen tiempos
  - Reducen tráfico en la red
- La cache entry tiene un TTL
  - Para bypasssear caídas o Nicos de Berazategui



# Repaso DNS - Infraestructura

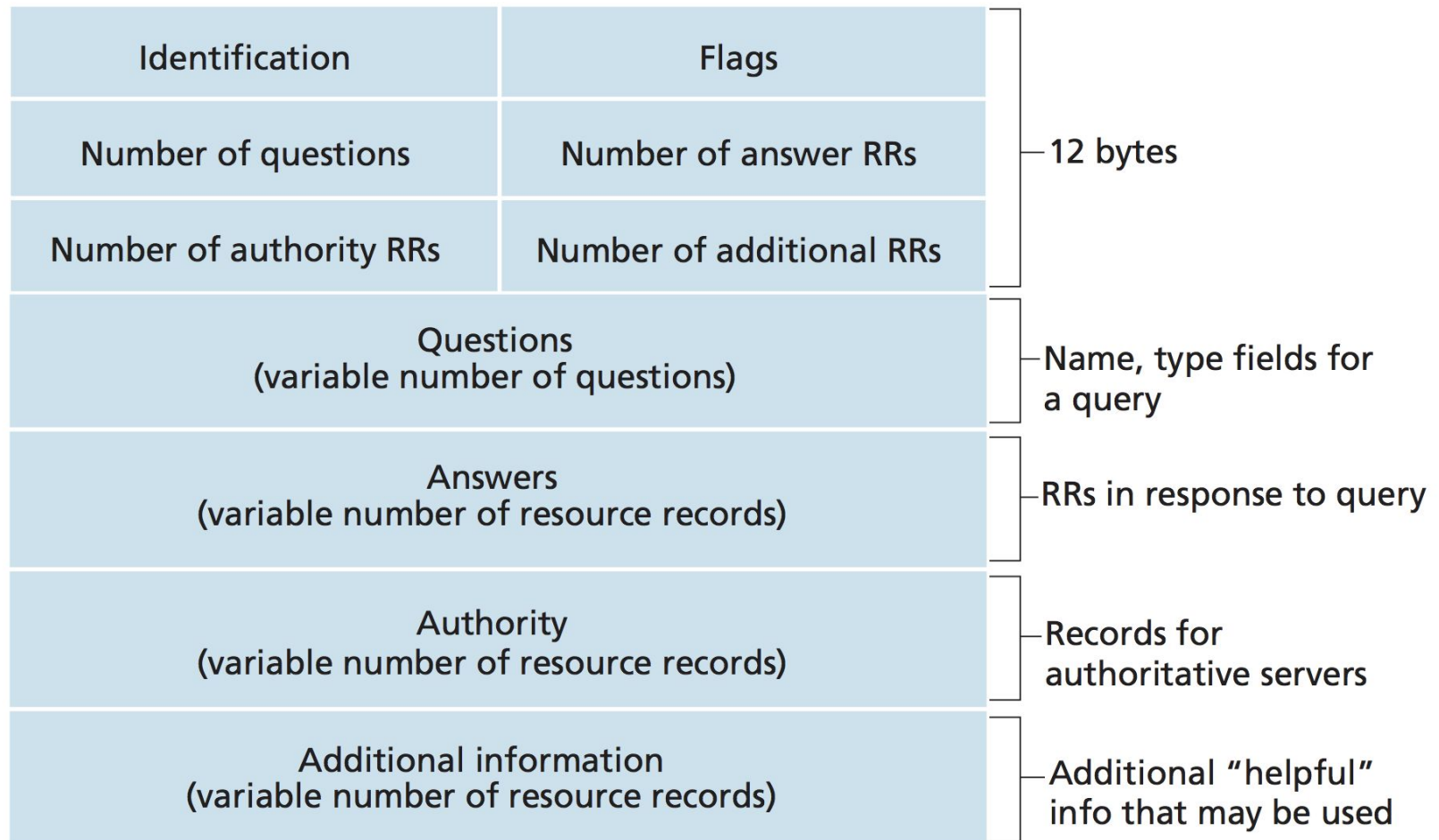
Muchas queries similares, poco eficiente...

⇒ DNS Caches

## *Google Public DNS turns 8.8.8.8 years old*

- Se qu
- An se
  - Evolución del DNS de Google
  - > 1TB de queries por día
  - Problemas de performance y soluciones
    - Muchas muchas caches
    - Anycast → pegarle al más cercano
    - Etc etc... (for more details, <https://careers.google.com/jobs/>)
- La cache entry tiene un TTL → Nico de Berazategui

# Repaso DNS - Estructura msje



**Figure 2.23** ♦ DNS message format

Ejemplo

# Ejemplo

¿Qué pasa cuando  
ingresamos una URL en  
un browser y apretamos  
enter?

# Ejemplo

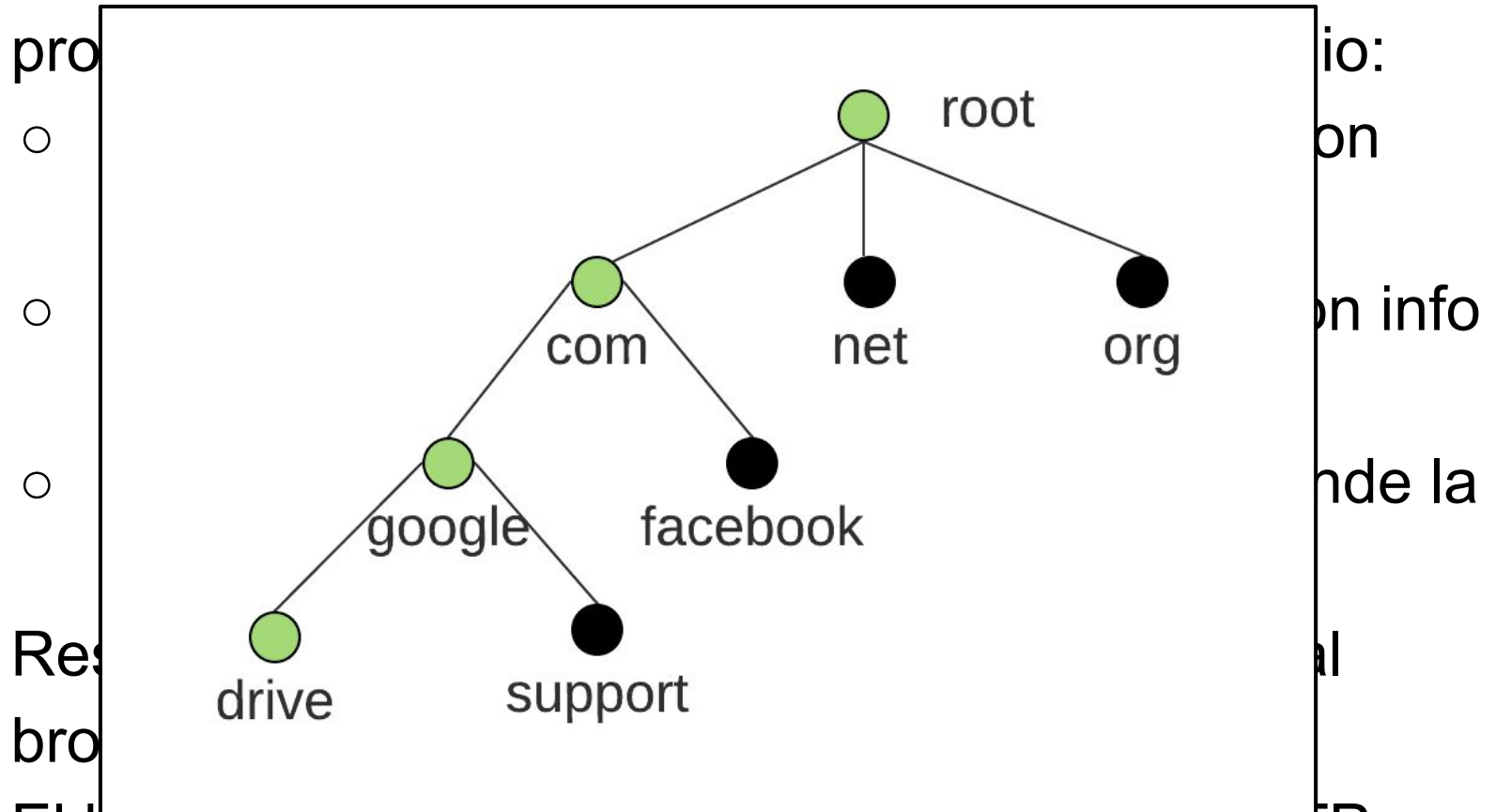
- Ponemos ***drive.google.com*** en nuestro browser
- El browser busca la URL en la(s) cachés de DNS:
  - Primero, checkea su propia caché
  - Si no lo encuentra, system call para chequear caché del SO
- Si no se tiene, se escala al DNS resolver que tengamos configurado:
  - Puede ser por ej, 8.8.8.8 (DNS de google)
  - También puede ser el default gateway (nuestro router local)

# Ejemplo

- Si el resolver no lo tiene cacheado, comienza proceso iterativo para averiguar la IP del dominio:
  - Se lanza la query al root server; responde con info del TLD **.com**
  - Se lanza la query al TLD **.com**; responde con info del SA **google.com**
  - Se lanza la query al SA **google.com**; responde la IP de **drive.google.com**
- Resolver cachea la respuesta, y se la redirige al browser (que tambien la cachea).
- El browser ahora puede realizar la request HTTP

# Ejemplo

- Si el resolver no lo tiene cacheado, comienza



- Res...
- bro...
- El browser ahora puede realizar la request HTTP

eDNS

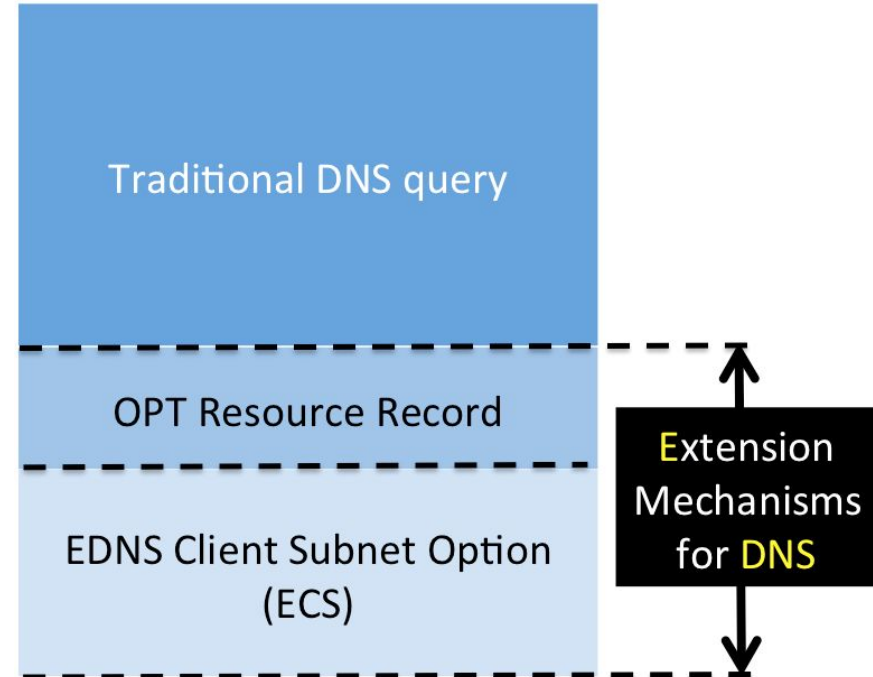


# eDNS

- DNS fue creado a principios de la década de 1980.
- El protocolo básico DNS presentaba ciertas limitaciones:
  - Tamaño de varios campos
  - Códigos de retorno
  - Tipos de etiquetas
  - Mensajes de 512 bytes máximo
- Se vuelve una necesidad poder mandar mas info en los mensajes DNS
  - Difícil cambiar el protocolo existente; hay que conservar compatibilidad

# eDNS

- En 1999 se publica [\[RFC 2671\]](#).
- No hay cambios en el header de DNS ⇒ Compatibilidad con implementaciones anteriores.
- Se crea un nuevo pseudo-RR: **OPT** (solo existen en mensajes)
- Permite agregar nuevos RR.

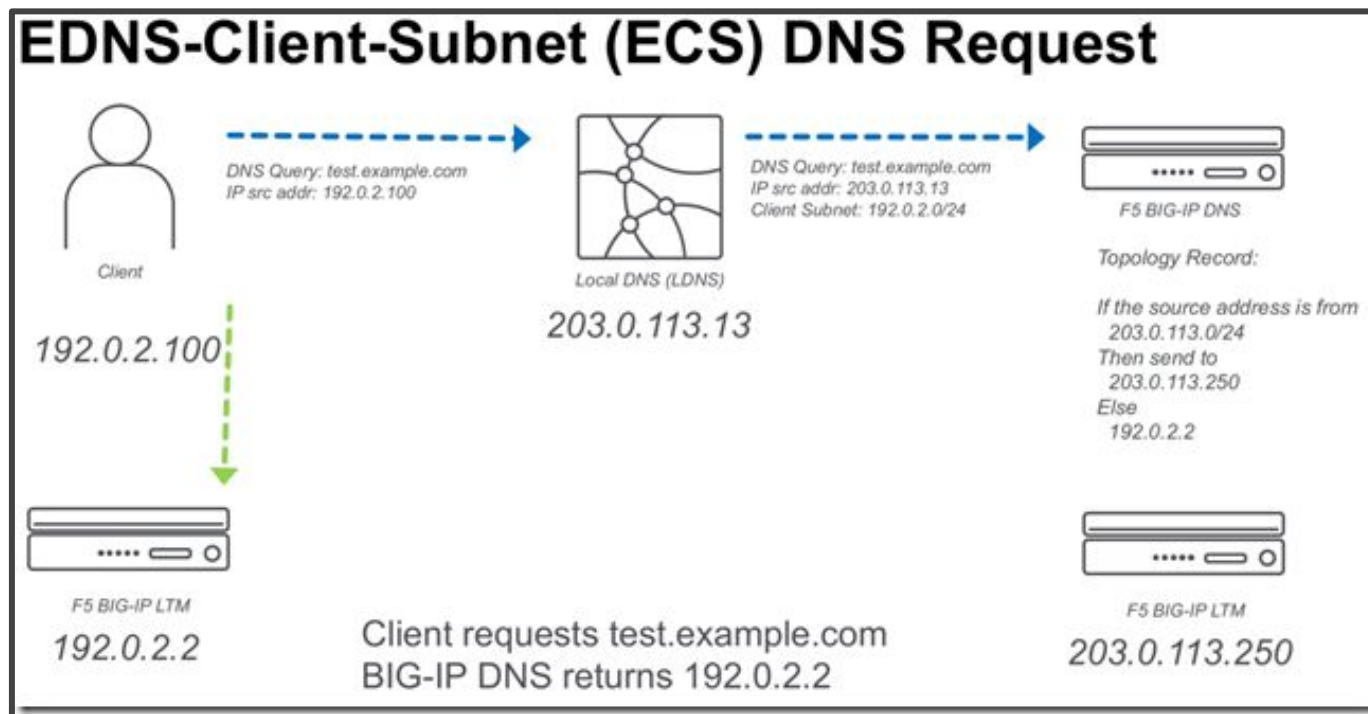


# eDNS

- Aplicación:
  - **ECS** → Compartir información de la subnet del cliente para acelerar tiempos de respuesta (CDNs)
  - **DNSSEC** → Extensiones de seguridad (autenticación criptográfica)
- Problemas:
  - Facilita ataques DNS Amplification → Consumir del bandwidth de un servicio o red, a través de engaño de DNS resolvers

# eDNS - ECS [\[RFC 7871\]](#)

- Utiliza eDNS para enviar la información del generador de la query hasta el servidor autoritario.
- Los servidores DNS autoritativos pueden devolver distintos registros en base a esta información.

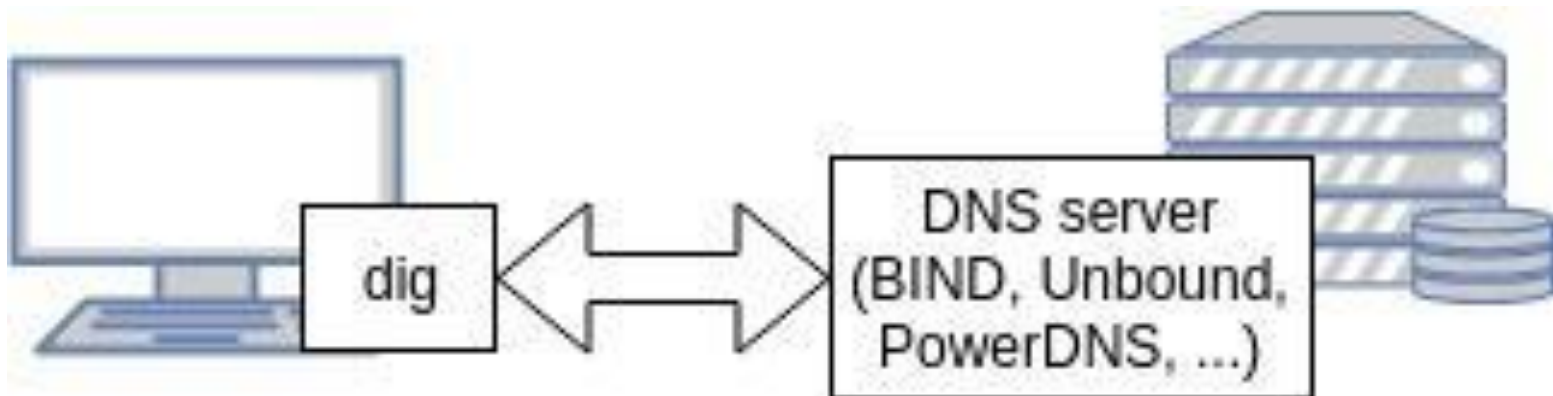


# Herramientas

> dig <

# dig

- Dig (Domain Information Groper) CLI tool
- Herramienta para consultar servidores DNS



# dig - Uso básico

```
➔ ~ dig @server domain type
```

Donde:

- **@server** es el nombre o dirección IP del servidor DNS al que se le realiza la consulta. Si no se especifica un servidor usará los servidores que encuentre en /etc/resolv.conf.
- **domain** es el dominio sobre el que se solicita información
- **type** es el tipo de consulta (por defecto busca registros tipo A)

⇒ El uso básico y el resto de las opciones se pueden encontrar con  
`man dig`.

# dig - Ejemplo

➔ ~ dig fi.uba.ar

```
; <<>> DiG 9.10.6 <<>> fi.uba.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44328
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;fi.uba.ar.                IN      A

;; ANSWER SECTION:
fi.uba.ar.                 3600 IN    A      157.92.49.38

;; Query time: 11 msec
;; SERVER: 192.168.1.102#53(192.168.1.102)
;; WHEN: Tue Mar 19 14:24:10 -03 2019
;; MSG SIZE rcvd: 125
```



# dig - Ejemplo

→ ~ dig fi.uba.ar

; <<>> DiG 9.10.6 <<>> fi.uba.ar

;; global options: +cmd

;; Command line options:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44328

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 2

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;fi.uba.ar. IN A

;; ANSWER SECTION:

fi.uba.ar. 3600 IN

;; Query time: 11 msec

;; SERVER: 192.168.1.102#53

;; WHEN: Tue Mar 19 14:24:52

;; MSG SIZE rcvd: 125

**Application info section** tiene información sobre la versión de dig, el dominio consultado y las opciones globales usadas.

# dig - Ejemplo

→ ~ dig fi.uba.ar

; <<>> DiG 9.10.6 <<>> fi.uba.ar

;; global options: +cmd

;; Command line options:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44328

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 2

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;fi.uba.ar. IN A

;; ANSWER SECTION:  
fi.uba.ar.

;; Query time: 11 m

;; SERVER: 192.16

;; WHEN: Tue Mar

;; MSG SIZE rcvd

**Header section** respuesta del servidor dns. Los mensajes de DNS son únicos.

Flags:

- qr indica que es una respuesta

- rd y ra indican la forma en que se resolvió el pedido.

# dig - Ejemplo

➔ ~ dig fi.uba.ar

```
; <<>> DiG 9.10.6 <<>> fi.uba.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44328
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;fi.uba.ar.                IN      A

;; ANSWER SECTION:
fi.uba.ar.                3600 IN    A      157.92.49.38

;; Query time: 11 msec
;; SERVER: 192.168.1.102#53(192.168.1.102)
;; WHEN: Tue Mar 19 14:24:10 -03 2019
;; MSG SIZE rcvd: 125
```

# dig - Ejemplo

➔ ~ dig fi.uba.ar

```
;; <<>> DiG 9.10.6 <<>> fi.uba.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44328
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 4096
;; QUESTION SECTION:
;fi.uba.ar.                IN      A

;; ANSWER SECTION:
fi.uba.ar.                3600 IN    A      157.92.49.38

;; Query time: 11 msec
;; SERVER: 192.168.1.102#53(192.168.1.102)
;; WHEN: Tue Mar 19 14:24:10 -03 2019
;; MSG SIZE rcvd: 125
```

# dig - Ejemplo

➔ ~ dig fi.uba.ar

```
;; <<>> DiG 9.10.6 <<>> fi.uba.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;; udp: 4096
;; QUESTION SECTION:
;fi.uba.ar.                IN      A

;; ANSWER SECTION:
fi.uba.ar.                 3600 IN      A      157.92.49.38

;; Query time: 11 msec
;; SERVER: 192.168.1.102#53(192.168.1.102)
;; WHEN: Tue Mar 19 14:24:10 -03 2019
;; MSG SIZE rcvd: 125
```

## Answer section

Respuesta de la consulta.

- nombre del dominio
- TTL (segundos en cache)
- Clase
- Tipo del registro
- Dirección IP resuelta

# dig - Ejemplo


➔ ~ dig fi.uba.ar

```
;; <<>> DiG 9.10.6 <<>> fi.uba.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;; udp: 4096
;; QUESTION SECTION:
;fi.uba.ar.                IN      A

;; ANSWER SECTION:
fi.uba.ar.                 3600 IN    A      157.92.49.38

;; Query time: 11 msec
;; SERVER: 192.168.1.102#53(192.168.1.102)
;; WHEN: Tue Mar 19 14:24:10 -03 2019
;; MSG SIZE rcvd: 125
```



## Statistics section

### Estadísticas sobre la consulta

- tiempo que tomó
- name server que lo resolvió
- puerto que se usó para la consulta
- momento en que se hizo
- tamaño del paquete recibido en bytes (sin tener en cuenta las cabeceras de las capas inferiores).

# dig - Ejemplo 2

```
➔ ~ dig @ns1.fi.uba.ar fi.uba.ar
```

```
; <<>> DiG 9.10.6 <<>> @ns1.fi.uba.ar fi.uba.ar  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status:  
NOERROR, id: 34472  
;; flags: qr aa rd; QUERY: 1, ANSWER: 1,  
AUTHORITY: 5, ADDITIONAL: 4  
;; WARNING: recursion requested but not available
```

```
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags;; udp: 4096  
;; QUESTION SECTION:  
;fi.uba.ar. IN A
```

```
;; ANSWER SECTION:  
fi.uba.ar. 3600 IN A 157.92.49.38
```

```
;; AUTHORITY SECTION:
```

```
fi.uba.ar. 3600 IN NS ns3.fi.uba.ar.  
fi.uba.ar. 3600 IN NS ns1.uba.ar.  
fi.uba.ar. 3600 IN NS ns2.fi.uba.ar.  
fi.uba.ar. 3600 IN NS ns2.uba.ar.  
fi.uba.ar. 3600 IN NS ns1.fi.uba.ar.
```

```
;; ADDITIONAL SECTION:
```

```
ns1.fi.uba.ar. 3600 IN A 157.92.49.2  
ns2.fi.uba.ar. 3600 IN A 157.92.49.3  
ns3.fi.uba.ar. 3600 IN A 157.92.56.2
```

```
;; Query time: 24 msec  
;; SERVER: 157.92.49.2#53(157.92.49.2)  
;; WHEN: Tue Mar 19 11:30:20 -03 2019  
;; MSG SIZE rcvd: 192
```

# dig - Ejemplo 2

```
➔ ~ dig @ns1.fi.uba.ar fi.uba.ar
```

```
; <<>> DiG 9.10.6 <<>> @ns1.fi.uba.ar fi.uba.ar  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status:  
NOERROR, id: 34472  
;; flags: qr aa rd: QUERY: 1 ANSWER: 1.
```

**Authority section**

Muestra el nombre de los  
name servers disponibles.  
Aparece si se consulta un  
servidor autoritativo.

## ;; AUTHORITY SECTION:

fi.uba.ar.	3600	IN	NS	ns3.fi.uba.ar.
fi.uba.ar.	3600	IN	NS	ns1.uba.ar.
fi.uba.ar.	3600	IN	NS	ns2.fi.uba.ar.
fi.uba.ar.	3600	IN	NS	ns2.uba.ar.
fi.uba.ar.	3600	IN	NS	ns1.fi.uba.ar.

## ;; ADDITIONAL SECTION:

ns1.fi.uba.ar.	3600	IN	A	157.92.49.2
ns2.fi.uba.ar.	3600	IN	A	157.92.49.3
ns3.fi.uba.ar.	3600	IN	A	157.92.56.2

```
;; Query time: 24 msec  
;; SERVER: 157.92.49.2#53(157.92.49.2)  
;; WHEN: Tue Mar 19 11:30:20 -03 2019  
;; MSG SIZE rcvd: 192
```



# dig - Ejemplo 2

```
➔ ~ dig @ns1.fi.uba.ar fi.uba.ar
```

```
; <<>> DiG 9.10.6 <<>> @ns1.fi.uba.ar fi.uba.ar  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY status:
```

## Additional section

Muestra las direcciones ip de los name servers. Aparece junto con la authority section.

```
;; ANSWER SECTION:  
fi.uba.ar.      3600 IN      A      157.92.49.38
```

## ;; AUTHORITY SECTION:

fi.uba.ar.	3600	IN	NS	ns3.fi.uba.ar.
fi.uba.ar.	3600	IN	NS	ns1.uba.ar.
fi.uba.ar.	3600	IN	NS	ns2.fi.uba.ar.
fi.uba.ar.	3600	IN	NS	ns2.uba.ar.
fi.uba.ar.	3600	IN	NS	ns1.fi.uba.ar.

## ;; ADDITIONAL SECTION:

ns1.fi.uba.ar.	3600	IN	A	157.92.49.2
ns2.fi.uba.ar.	3600	IN	A	157.92.49.3
ns3.fi.uba.ar.	3600	IN	A	157.92.56.2

```
;; Query time: 24 msec  
;; SERVER: 157.92.49.2#53(157.92.49.2)  
;; WHEN: Tue Mar 19 11:30:20 -03 2019  
;; MSG SIZE rcvd: 192
```

# dig - Servidores diferentes

➔ ~ dig @8.8.8.8 google.com

```
; <<>> DiG 9.10.6 <<>> @8.8.8.8 google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 54528
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY:
0, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 512
;; QUESTION SECTION:
;google.com.                IN      A
```

```
;; ANSWER SECTION:
google.com.                295     IN      A
172.217.30.142
```

```
;; Query time: 64 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Mar 19 11:37:59 -03 2019
;; MSG SIZE rcvd: 55
```

**8.8.8.8** es un servidor DNS  
publico de google.

➔ ~ dig @208.67.222.222 google.com

```
; <<>> DiG 9.10.6 <<>> @208.67.222.222 google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 25637
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY:
0, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 4096
;; QUESTION SECTION:
;google.com.                IN      A
```

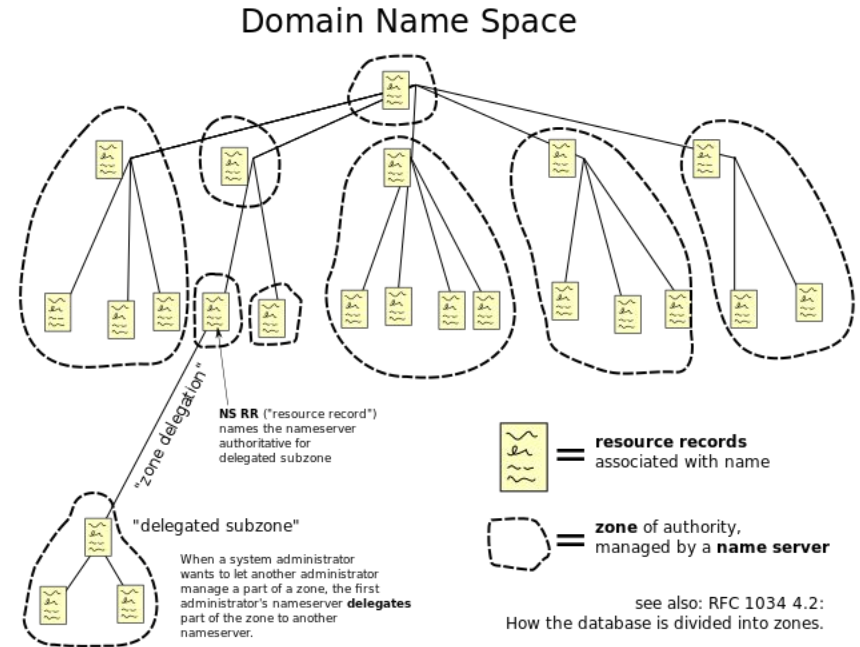
```
;; ANSWER SECTION:
google.com.                300     IN      A
172.217.30.238
```

```
;; Query time: 162 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Tue Mar 19 11:40:40 -03 2019
;; MSG SIZE rcvd: 55
```

**208.67.222.222** es un servidor  
DNS publico de OpenDNS.

# Delegación de zonas

Para ver la delegación de zonas desde los servidores de nombre raíces con dig se puede usar la opción +trace, esto fuerza a ignorar los caches y resolver de forma iterativa.



# Delegación de zonas

→ ~ dig google.com +trace

; <<>> DiG 9.10.6 <<>> google.com +trace

;; global options: +cmd

.	518398	IN	NS	l.root-servers.net.
.	518398	IN	NS	g.root-servers.net.
.	518398	IN	NS	f.root-servers.net.
.	518398	IN	NS	k.root-servers.net.
.	518398	IN	NS	m.root-servers.net.
.	518398	IN	NS	c.root-servers.net.
.	518398	IN	NS	j.root-servers.net.
.	518398	IN	NS	i.root-servers.net.
.	518398	IN	NS	b.root-servers.net.
.	518398	IN	NS	d.root-servers.net.
.	518398	IN	NS	h.root-servers.net.
.	518398	IN	NS	e.root-servers.net.
.	518398	IN	NS	a.root-servers.net.

;; Received 733 bytes from 192.168.1.100#53(192.168.1.100) in 6 ms

# Delegación de zonas

```
com.      172800   IN   NS   a.gtld-servers.net.
com.      172800   IN   NS   b.gtld-servers.net.
com.      172800   IN   NS   c.gtld-servers.net.
com.      172800   IN   NS   d.gtld-servers.net.
com.      172800   IN   NS   e.gtld-servers.net.
com.      172800   IN   NS   f.gtld-servers.net.
com.      172800   IN   NS   g.gtld-servers.net.
com.      172800   IN   NS   h.gtld-servers.net.
com.      172800   IN   NS   i.gtld-servers.net.
com.      172800   IN   NS   j.gtld-servers.net.
com.      172800   IN   NS   k.gtld-servers.net.
com.      172800   IN   NS   l.gtld-servers.net.
com.      172800   IN   NS   m.gtld-servers.net.
```

;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

```
google.com. 172800   IN   NS   ns2.google.com.
google.com. 172800   IN   NS   ns1.google.com.
google.com. 172800   IN   NS   ns3.google.com.
google.com. 172800   IN   NS   ns4.google.com.
```

;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

```
google.com. 300    IN   A    172.217.30.142
```

;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms

# Delegación de zonas

→ ~ dig google.com +trace

; <<>> DiG 9.10.6 <<>> google.com +trace

;; global options: +cmd

.	518398	IN	NS	l.root-servers.net.
.	518398	IN	NS	g.root-servers.net.
.	518398	IN	NS	f.root-servers.net.
.	518398	IN	NS	k.root-servers.net.
.	518398	IN	NS	m.root-servers.net.
.	518398	IN	NS	c.root-servers.net.
.	518398	IN	NS	j.root-servers.net.
.	518398	IN	NS	i.root-servers.net.
.	518398	IN	NS	b.root-servers.net.
.	518398	IN	NS	d.root-servers.net.
.	518398	IN	NS	h.root-servers.net.
.	518398	IN	NS	e.root-servers.net.
.	518398	IN	NS	a.root-servers.net.

;; Received 733 bytes from 192.168.1.100#53(192.168.1.100) in 6 ms

# Delegación de zonas

→ ~ dig google.com +trace

; <<>> DiG 9.10.6 <<>> google.com +trace

;; global options: +cmd

.	518398	IN	NS	l.root-servers.net.
.	518398	IN	NS	g.root-servers.net.
.	518398	IN	NS	f.root-servers.net.
.	518398	IN	NS	k.root-servers.net.
.	518398	IN	NS	m.root-servers.net.
.	518398	IN	NS	c.root-servers.net.
.	518398	IN	NS	j.root-servers.net.
.	518398	IN	NS	i.root-servers.net.
.	518398	IN	NS	d.root-servers.net.
.	518398	IN	NS	e.root-servers.net.
.	518398	IN	NS	a.root-servers.net.

;; Received 733 bytes from 192.168.1.100#53(192.168.1.100) in 6 ms

# Delegación de zonas

```
com.      172800   IN   NS   a.gtld-servers.net.
com.      172800   IN   NS   b.gtld-servers.net.
com.      172800   IN   NS   c.gtld-servers.net.
com.      172800   IN   NS   d.gtld-servers.net.
com.      172800   IN   NS   e.gtld-servers.net.
com.      172800   IN   NS   f.gtld-servers.net.
com.      172800   IN   NS   g.gtld-servers.net.
com.      172800   IN   NS   h.gtld-servers.net.
com.      172800   IN   NS   i.gtld-servers.net.
com.      172800   IN   NS   j.gtld-servers.net.
com.      172800   IN   NS   k.gtld-servers.net.
com.      172800   IN   NS   l.gtld-servers.net.
com.      172800   IN   NS   m.gtld-servers.net.
```

;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

```
google.com. 172800   IN   NS   ns2.google.com.
google.com. 172800   IN   NS   ns1.google.com.
google.com. 172800   IN   NS   ns3.google.com.
google.com. 172800   IN   NS   ns4.google.com.
```

;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

```
google.com. 300    IN   A    172.217.30.142
```

;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms



# Delegación de zonas

```
com.      172800  IN   NS   a.gtld-servers.net.
com.      172800  IN   NS   b.gtld-servers.net.
com.      172800  IN   NS   c.gtld-servers.net.
com.      172800  IN   NS   d.gtld-servers.net.
com.      172800  IN   NS   e.gtld-servers.net.
com.      172800  IN   NS   f.gtld-servers.net.
com.      172800  IN   NS   g.gtld-servers.net.
com.      172800  IN   NS   h.gtld-servers.net.
com.      172800  IN   NS   i.gtld-servers.net.
com.      172800  IN   NS   j.gtld-servers.net.
com.      172800  IN   NS   k.gtld-servers.net.
com.      172800  IN   NS   l.gtld-servers.net.
com.      172800  IN   NS   m.gtld-servers.net.
;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

google.com. 172800  IN   NS   ns2.google.com.
google.com. 172800  IN   NS   ns1.google.com.
google.com. 172800  IN   NS   ns3.google.com.
google.com. 172800  IN   NS   ns4.google.com.
;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

google.com. 300   IN   A     172.217.30.142
;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms
```

# Delegación de zonas

```
com.      172800   IN      NS      a.gtld-servers.net.
com.      172800   IN      NS      b.gtld-servers.net.
com.      172800   IN      NS      c.gtld-servers.net.
com.      172800   IN      NS      d.gtld-servers.net.
com.      172800   IN      NS      e.gtld-servers.net.
com.      172800   IN      NS      f.gtld-servers.net.
com.      172800   IN      NS      g.gtld-servers.net.
com.      172800   IN      NS      h.gtld-servers.net.
com.      172800   IN      NS      i.gtld-servers.net.
com.      172800   IN      NS      j.gtld-servers.net.
com.      172800   IN      NS      k.gtld-servers.net.
com.      172800   IN      NS      l.gtld-servers.net.
com.      172800   IN      NS      m.gtld-servers.net.
;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

google.com. 172800   IN      NS      ns2.google.com.
google.com. 172800   IN      NS      ns1.google.com.
google.com. 172800   IN      NS      ns3.google.com.
google.com. 172800   IN      NS      ns4.google.com.
;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

google.com. 300     IN      A       172.217.30.142
;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms
```

# Delegación de zonas

```
com.      172800   IN      NS      a.gtld-servers.net.
com.      172800   IN      NS      b.gtld-servers.net.
com.      172800   IN      NS      c.gtld-servers.net.
com.      172800   IN      NS      d.gtld-servers.net.
com.      172800   IN      NS      e.gtld-servers.net.
com.      172800   IN      NS      f.gtld-servers.net.
com.      172800   IN      NS      g.gtld-servers.net.
com.      172800   IN      NS      h.gtld-servers.net.
com.      172800   IN      NS      i.gtld-servers.net.
com.      172800   IN      NS      j.gtld-servers.net.
com.      172800   IN      NS      k.gtld-servers.net.
com.      172800   IN      NS      l.gtld-servers.net.
com.      172800   IN      NS      m.gtld-servers.net.
;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

google.com. 172800   IN      NS      ns2.google.com.
google.com. 172800   IN      NS      ns1.google.com.
google.com. 172800   IN      NS      ns3.google.com.
google.com. 172800   IN      NS      ns4.google.com.
;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

google.com. 300     IN      A       172.217.30.142
;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms
```

# Delegación de zonas

```
com.      172800  IN      NS      a.gtld-servers.net.
com.      172800  IN      NS      b.gtld-servers.net.
com.      172800  IN      NS      c.gtld-servers.net.
com.      172800  IN      NS      d.gtld-servers.net.
com.      172800  IN      NS      e.gtld-servers.net.
com.      172800  IN      NS      f.gtld-servers.net.
com.      172800  IN      NS      g.gtld-servers.net.
com.      172800  IN      NS      h.gtld-servers.net.
com.      172800  IN      NS      i.gtld-servers.net.
com.      172800  IN      NS      j.gtld-servers.net.
com.      172800  IN      NS      k.gtld-servers.net.
com.      172800  IN      NS      l.gtld-servers.net.
com.      172800  IN      NS      m.gtld-servers.net.
;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

google.com. 172800  IN      NS      ns1.google.com.
google.com. 172800  IN      NS      ns2.google.com.
google.com. 172800  IN      NS      ns3.google.com.
google.com. 172800  IN      NS      ns4.google.com.
;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

google.com. 300    IN      A       172.217.30.142
;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms
```

# Delegación de zonas

```
com.      172800  IN      NS      a.gtld-servers.net.
com.      172800  IN      NS      b.gtld-servers.net.
com.      172800  IN      NS      c.gtld-servers.net.
com.      172800  IN      NS      d.gtld-servers.net.
com.      172800  IN      NS      e.gtld-servers.net.
com.      172800  IN      NS      f.gtld-servers.net.
com.      172800  IN      NS      g.gtld-servers.net.
com.      172800  IN      NS      h.gtld-servers.net.
com.      172800  IN      NS      i.gtld-servers.net.
com.      172800  IN      NS      j.gtld-servers.net.
com.      172800  IN      NS      k.gtld-servers.net.
com.      172800  IN      NS      l.gtld-servers.net.
com.      172800  IN      NS      m.gtld-servers.net.
;; Received 1170 bytes from 199.7.91.13#53(d.root-servers.net) in 7 ms

google.com. 172800  IN      NS      ns1.google.com.
google.com. 172800  IN      NS      ns2.google.com.
google.com. 172800  IN      NS      ns3.google.com.
google.com. 172800  IN      NS      ns4.google.com.
;; Received 772 bytes from 192.42.93.30#53(g.gtld-servers.net) in 167 ms

google.com. 300  IN      A       172.217.142.142
;; Received 55 bytes from 216.239.32.10#53(ns1.google.com) in 26 ms
```

# DNS - Confidencialidad

¿Qué significa que los mensajes  
DNS no son confidenciales?

# DNS - Confidencialidad

- ¿Quién puede ver los mensajes DNS?

# DNS - Confidencialidad

- ¿Quién puede ver los mensajes DNS?
  - Man-in-the-middle
  - ISPs
  - Cualquiera router en el camino de los mensajes.
  - El gobierno



# DOH

## DNS over HTTPS

# DOH

- Descripto en [\[RFC8484\]](#)
- Enviar mensajes DNS sobre HTTPS
- Resuelve el problema de confidencialidad de los mensajes DNS.

# DOH - Estandar

- GET o POST **/dns-query**
- Parámetros:
  - **dns**: DNS request codificada en base64.
- Response MIME Type:
  - application/dns-message

# DOH - Implementaciones

- Google DoH Resolver:
  - <https://dns.google/dns-query>
- Cloudflare DoH Resolver:
  - <https://cloudflare-dns.com/dns-query>

# DOH - Query

```
curl \  
-H 'accept: application/dns-message '  
'https://cloudflare-dns.com/dns-query  
?dns=tTcBIAABAAAAAAAAABAmZpA3ViYQJhcgAA  
AQABAAApEAAAAAAAAAAAAA '
```

# DOH - Response

- Mensaje DNS binario
- Representación en hexadecimal

```
00000000: b537 8180 0001 0001 0000 0001 0266 6903  .7.....fi.
00000010: 7562 6102 6172 0000 0100 0102 6669 0375  uba.ar.....fi.u
00000020: 6261 0261 7200 0001 0001 0000 0d1b 0004  ba.ar.....
00000030: 9d5c 3126 0000 2905 ac00 0000 0000 00    .\1&..).....
```

# DOH - Response

- Mensaje DNS binario
- Representación en hexadecimal

```
00000000: b537 8180 0001 0001 0000 0001 0266 6903  .7.....fi.
00000010: 7562 6102 6172 0000 0100 0102 6669 0375  uba.ar.....fi.u
00000020: 6261 0261 7200 0001 0001 0000 0d1b 0004  ba.ar.....
00000030: 9d5c 3126 0000 2905 ac00 0000 0000 00    .\1&..).....
```

## Clarísimo, ¿no?

# DOH - Cloudflare

- Cloudflare fue mas alla del estandar.
- Parámetros:
  - **name**: Campo nombre del RR
  - **type**: Tipo del RR
- Response MIME Type:
  - application/dns-json



# DOH - Query

```
curl \  
-H 'accept:application/dns-json'  
'https://cloudflare-dns.com/dns-query  
?name=fi.uba.ar&type=A'
```

# DOH - Response

```
{  
  "Status": 0, "TC": false, "RD": true,  
  "RA": true, "AD": false, "CD": false,  
  "Question": [{  
    "name": "fi.uba.ar.",  
    "type": 1  
  }],  
  "Answer": [{  
    "name": "fi.uba.ar.",  
    "type": 1,  
    "TTL": 3600,  
    "data": "157.92.49.38"  
  }]  
}
```

# DOH - Response

Field	Description
Status	The Response Code of the DNS Query. These are defined here: <a href="https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-6">https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-6</a>
TC	If true, it means the truncated bit was set. This happens when the DNS answer is larger than a single UDP or TCP packet. TC will almost always be false with Cloudflare DNS over HTTPS because Cloudflare supports the maximum response size.
RD	If true, it means the Recursive Desired bit was set. This is always set to true for Cloudflare DNS over HTTPS.
RA	If true, it means the Recursion Available bit was set. This is always set to true for Cloudflare DNS over HTTPS.
AD	If true, it means that every record in the answer was verified with DNSSEC.
CD	If true, the client asked to disable DNSSEC validation. In this case, Cloudflare will still fetch the DNSSEC-related records, but it will not attempt to validate the records.
Question: name	The record name requested.
Question: type	The type of DNS record requested. These are defined here: <a href="https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4">https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4</a>
Answer: name	The record owner.
Answer: type	The type of DNS record. These are defined here: <a href="https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4">https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4</a>
Answer: TTL	The number of seconds the answer can be stored in cache before it is considered stale.
Answer: data	The value of the DNS record for the given name and type. The data will be in text for standardized record types and in hex for unknown types.

Documentación: <https://developers.cloudflare.com/1.1.1.1/dns-over-https/json-format/>

# Referencias

- DoS:  
<https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/>
- DNS Amplification:  
<https://umbrella.cisco.com/blog/2014/03/17/dns-amplification-attacks/>
- Distintos ataques DoS:  
<https://www.imperva.com/learn/application-security/?orderby=views&category=ddos>
- Botnets:  
<https://thehackernews.com/2017/12/hacker-ddos-mirai-botnet.html>
- Mirai:  
<https://thehackernews.com/2017/12/hacker-ddos-mirai-botnet.html>
- Cloudflare DoH:  
<https://developers.cloudflare.com/1.1.1.1/dns-over-https/>

# Referencias

- <https://linux.die.net/man/1/dig>
- <https://tools.ietf.org/html/rfc1034>
- <https://tools.ietf.org/html/rfc1035>
- <https://tools.ietf.org/html/rfc2929>
- [https://en.wikipedia.org/wiki/List\\_of\\_DNS\\_record\\_types](https://en.wikipedia.org/wiki/List_of_DNS_record_types)
- [http://www.tcpipguide.com/free/t\\_DNSMessageHeaderandQuestionSectionFormat.htm](http://www.tcpipguide.com/free/t_DNSMessageHeaderandQuestionSectionFormat.htm)