

<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

# Trabajo práctico grupal N° 2

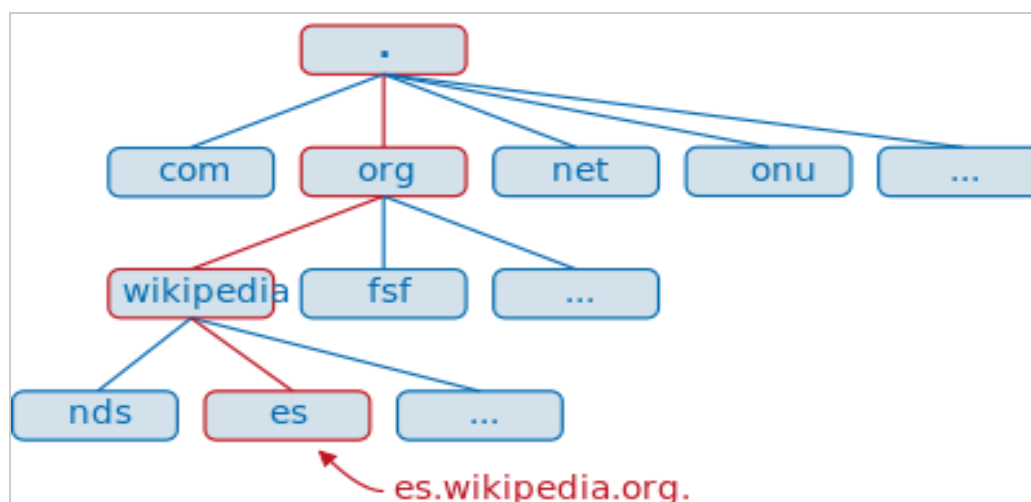
## DNS - Domain Name System

### Introducción

**Domain Name System** o **DNS** (en español «Sistema de Nombres de Dominio») es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a [Internet](#) o a una [red privada](#). Este sistema asocia información variada con [nombres de dominios](#) asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

El servidor DNS utiliza una [base de datos](#) distribuida y [jerárquica](#) que almacena información asociada a [nombres de dominio](#) en redes como [Internet](#). Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a [direcciones IP](#) y la localización de los servidores de [correo electrónico](#) de cada dominio.

### Jerarquía DNS



<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

## Árbol DNS

El espacio de nombres de dominio tiene una estructura arborescente. Las hojas y los nodos del árbol se utilizan como etiquetas de los medios. Un nombre de dominio completo de un objeto consiste en la concatenación de todas las etiquetas de un camino. Las etiquetas son cadenas alfanuméricas (con '-' como único símbolo permitido), deben contar con al menos un carácter y un máximo de 63 caracteres de longitud, y deberá comenzar con una letra (y no con '-') (ver la [RFC 1035](#), sección "2.3.1. Preferencia nombre de la sintaxis "). Las etiquetas individuales están separadas por puntos. Un nombre de dominio termina con un punto (aunque este último punto generalmente se omite, ya que es puramente formal). Un FQDN correcto (también llamado Fully Qualified Domain Name), es por ejemplo este: `www.example.com.` (incluyendo el punto al final).

Un nombre de dominio debe incluir todos los puntos y tiene una longitud máxima de 255 caracteres.

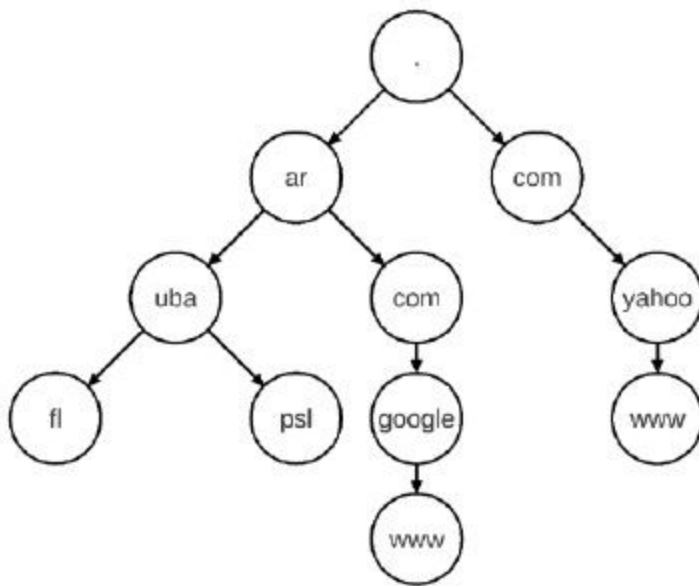
Un nombre de dominio se escribe siempre de derecha a izquierda. El punto en el extremo derecho de un nombre de dominio separa la etiqueta de la raíz de la jerarquía (en inglés, root). Este primer nivel es también conocido como dominio de nivel superior (TLD - Top Level Domain).

Los objetos de un dominio DNS (por ejemplo, el nombre del equipo) se registran en un archivo de zona, ubicado en uno o más servidores de nombres.

<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

## Objetivo

Para poder utilizar las estructuras de la materia definiremos nuestra Jerarquía DNS de la siguiente forma:



Cuando un cliente conectado al servidor A desea enviar un correo a otro, conectado al servidor B, el primero debe averiguar, de alguna manera, la dirección del segundo, para transferir el correo. Este es uno de los objetivos del DNS.

Este sistema parte de organizar jerárquicamente los nombres de dominio de Internet. Asociado a cada nombre de dominio se puede almacenar un dato, como por ejemplo la dirección IP de un servidor web, o la dirección IP de un servidor de correo. Esta organización se muestra en la figura, donde se representan los dominios fi.uba.ar, psi.uba.ar, www.google.com.ar y www.yahoo.com.

Este sistema presenta dos grandes ventajas. Por un lado, podemos delegar cada subárbol a una entidad distinta.

Por ejemplo, el subárbol que comienza en el nodo uba.ar puede ser administrado por la Universidad de Buenos Aires. Esto quiere decir que la UBA podrá agregar, quitar o modificar subdominios, sin alterar el funcionamiento del resto del sistema.

La segunda ventaja es que ninguna entidad necesita almacenar el árbol completo.

<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

Veamos esta segunda ventaja en detalle con un ejemplo. Supongamos que cada nodo del árbol de la figura representa un servidor DNS.

Además supongamos que cada servidor almacena únicamente la dirección de los servidores información asociada a un nombre de dominio determinado, por ejemplo fi.uba.ar?

En primer lugar, el servidor debe conocer de antemano la dirección del servidor raíz, al cual le consultará la dirección del servidor “ar”.

Con esta segunda dirección, el servidor de correos ya puede consultar la dirección del servidor “uba”. Finalmente, al consultar a este tercer servidor por el nombre de dominio “fi”, el mismo responderá con el dato asociado al mismo.

En nuestro caso, este dato es la dirección del servidor de correos de la Facultad de Ingeniería.

En este trabajo práctico se desarrollará un TDA para simular el funcionamiento del sistema de nombres de dominio de Internet.

De esta forma, cuando un servidor inseguro quiera obtener la dirección de otro para pasarle un mensaje, deber realizar la consulta a través del TDA DNS, quien se encargará de recorrer la jerarquía de dominios hasta encontrar el servidor buscado.

## Definición de TDA's

### TDADNS

La estructura del TDA queda a cargo del Grupo.

Primitivas:

**int crearDNS(TDADNS \*dns, int tamDato);**

**Descripción:** Crea una instancia de TDA DNS.

**Precondiciones:** dns tiene sizeof(TDADNS) bytes reservados. dns no creado. tamDato es el tamaño del tipo de dato T que tendrán los elementos que se van a guardar asociados a un dominio.

**Postcondiciones:** Se inicializa la estructura apuntada por dns.

<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

**void destruirDNS(TDADNS \*dns);**

**Descripción:** Destruye una instancia de TDA DNS.

**Precondiciones:** dns creado y vacío.

**Postcondiciones:** Se destruyó la instancia dns.

**int agregarDominio(TDADNS \*dns, char\* dominio, void\* dato);**

**Descripción:** Agrega un dominio con un dato asociado.

**Precondiciones:** dns creado. dominio es un nombre de dominio válido y no existente en dns. dato apunta a una variable de tipo T.

**Postcondiciones:** Se agregó el dominio dominio, y se asoció el valor dato.

**void obtenerValor(TDADNS dns, char\* dominio, void\* dato);**

**Descripción:** Obtiene el valor asociado a un dominio.

**Precondiciones:** dns creado. dominio es un nombre de dominio válido existente en dns. dato apunta a una variable de tipo T.

**Postcondiciones:** Se copió a dato el valor asociado al dominio dominio.

**int existeDominio(TDADNS dns, char\* dominio);**

**Descripción:** Devuelve TRUE si dominio existe en dns. En cualquier otro caso devuelve FALSE.

**Precondiciones:** dns creado. dominio es un nombre de dominio válido.

**Postcondiciones:** No realiza modificaciones.

**void eliminarDominio(TDADNS \*dns, char\* dominio);**

**Descripción:** Elimina un dominio.

**Precondiciones:** dns creado. dominio es un nombre de dominio válido y existente en dns.

**Postcondiciones:** Se eliminó dominio, y su valor asociado, de dns

<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

## Archivo de configuración

El archivo será un txt dns.txt con todos los dominios del DNS con la siguiente estructura. (url e IP asociada). (Crear un archivo con 50 direcciones/IP). Esto inicializará el árbol del DNS.

```
www.fi.uba.ar 157.92.49.38
www.google.com xxx.xxx.xxx.xxx
www.google.com.ar xxx.xxx.xxx.xxx
www.yahoo.com.ar xxx.xxx.xxx.xxx
www.yahoo.com xxx.xxx.xxx.xxx
www.hotmail.com xxx.xxx.xxx.xxx
```

## Archivos de salida esperados

La ejecución del TP, con los argumentos adecuados, debe dar como salida por pantalla el resultado de la consulta.

## Ejecución

Siempre se indica la urlOrigen que debe estar en el DNS para poder ejecutar el comando. De no ser así se debe retornar un mensaje de Error indicando que la url no se encuentra en el DNS.

```
tp2g -dnsSend [urlOrigen] [urlDestino] [mensaje] [archivo.log]
Muestra en pantalla el mensaje encriptado
```

```
tp2g -dnsGetIP [urlOrigen] [urlDestino] [archivo.log]
Muestra en pantalla los datos de la IP Origen y Destino
```

```
tp2g -dnsAddDomain [url] [IP] [archivo.log]
Muestra en pantalla el dominio e IP agregados.
```

```
tp2g -dnsDeleteDomain [url] [archivo.log]
Muestra en pantalla el dominio e IP eliminados.
```

En el archivo.log deben guardarse los datos de la operacion y el horario en que fue realizada la misma.

```
[fecha]-[hora]-[comando][urlOrigen][IPOrigen][urlDestino][IPDestino][mensaje][mensajeEncriptado]
```

<b>Algoritmos y Programación II [75.41]</b>	<b>Trabajo Práctico Grupal N° 2</b>
Cátedra Lic. Gustavo Carolo	DNS
2° Cuatrimestre 2016	Fecha de entrega: <b>20/11/2016</b>

## TDAEncriptador

La implementación del TDA Encriptador quedará a Cargo del Grupo. Básicamente realizaremos la encriptación por transposición, en base a un diccionario que será la clave para desencriptar el mensaje original.

Cada dominio tendrá su diccionario propio, por lo que deberemos realizar una estructura con la dirección ip de los dominios que estén en el dns y su diccionario de encriptación.

Para simplificar tomaremos para el mensaje solo los caracteres del alfabeto en mayúsculas y los números.

**ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789**

un posible diccionario para la url [www.google.com](http://www.google.com) para desencriptar podría ser.

**KLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ**

*Por lo que el mensaje **HOLA9** para este dominio se encriptará como **RXUKJ***

*Al crear un nuevo dominio en el DNS se deberá generar al azar el diccionario de desencriptación para el dominio creado.*