

El servicio DNS es muy importante para el funcionamiento de Internet porque es quien convierte nombres de dominio en direcciones IP.

Por ejemplo, al escribir en un navegador la dirección: www.google.com el cliente primero debe averiguar cuál es la dirección IP correspondiente a ese nombre de dominio, para que luego si se pueda establecer la comunicación.

Su tarea consiste en escribir un servidor DNS, al cual llamaremos “*ServidorParcial*” que permita a los computadores de una red preguntarle por la dirección IP de un dominio:

- El *ServidorParcial* debe tener una comunicación confiable con el cliente, se ha establecido que el servidor escuchará por el puerto 10001. El cliente debe enviarle al *ServidorParcial* el nombre de dominio que quiere buscar, como una cadena de texto. El *ServidorParcial* debe poder atender a varios clientes simultáneamente.
- El *ServidorParcial* tiene un archivo llamado “hosts.txt”, que contiene los nombres de dominio de los cuales ya conoce su dirección IP, de forma que si el cliente le envía un nombre que está en ese archivo, el *ServidorParcial* le responde con la dirección IP ahí guardada. El formato de este archivo es el siguiente: en cada línea va un nombre de dominio junto con la dirección IP correspondiente.

nombreDominio;direcciónIP

- El cliente espera que el *ServidorParcial*, en respuesta a la solicitud, le envíe cuatro (4) números enteros que corresponden a cada uno de los octetos de la dirección IP. Si no se puede resolver ese nombre, le envía cuatro veces el número -1. En ambos casos, luego de enviar la respuesta, el *ServidorParcial* corta la conexión.
- Si en el archivo “hosts.txt” no se encuentra ese nombre de dominio, el *ServidorParcial* le preguntará a otros servidores DNS si ellos tienen esa información. Para esto, el *ServidorParcial* cuenta con un archivo llamado “servidores.txt” con la lista de las direcciones IP de estos servidores DNS, cada una en una línea de texto diferente. Estos servidores escuchan por el puerto 53. La idea es que se les pregunta uno por uno si conocen ese nombre de dominio, ellos pueden responder con la dirección IP (en formato texto), o con la cadena “NO” si ellos tampoco conocen ese nombre de dominio. Si algún servidor contesta con la dirección, ya no hay necesidad de preguntarle a los demás. No se garantiza que la comunicación entre el *ServidorParcial* y los demás servidores sea confiable: puede que los mensajes lleguen o se pierdan en el camino.
- Si ninguno de estos servidores es capaz de resolver ese nombre de dominio, el protocolo DNS establece que existen unos servidores RAIZ, uno por cada TLD (top level domain). Por ejemplo, hay un servidor raíz para el dominio .edu, otro para el .com y así para todos. Cada servidor raíz conoce las direcciones IP de TODOS los nombres de su dominio y el *ServidorParcial* sólo debe consultarlo si ni él ni ningún otro conoce esa

información. Este servidor raíz funciona **de la misma forma** que los citados en el punto anterior, sólo que si él responde “NO” es porque ese dominio definitivamente no existe. La lista de los servidores raíz está en el archivo “raices.txt”. En este archivo cada línea tiene el TLD (edu, com, org, mil, net, ...) separado por un punto y coma de la dirección IP del servidor.

- Cada vez que el *ServidorParcial* resuelva una dirección IP que no conocía antes, debe escribirla en el archivo “hosts.txt” ANTES de enviarle la respuesta al cliente. De esta forma la siguiente vez que alguien pregunte por ese dominio se le responde directamente sin necesidad de consultar a nadie.
- NOTA: Para resolver el nombre de dominio que pregunta el cliente, debe usarse el protocolo DNS tal como se está explicando. De ninguna forma se acepta que se utilice la clase `InetAddress` como mecanismo para resolver los nombres. Esta clase sólo se podrá usar en los constructores a que haya lugar.
- Asuma que existe una clase llamada “Utilidades”, que tiene un método con la siguiente especificación:

```
public static ArrayList cargarArchivo(String archivo)
```

Este método recibe el nombre de un archivo texto, y devuelve un `ArrayList` de cadenas, donde cada una de ellas corresponde a una línea del archivo.

Criterios de Evaluación

- (60%) Análisis del *ServidorParcial*
 - Diagrama de clases
 - Diagrama donde se especifique claramente el protocolo
 - Descripción del funcionamiento de los métodos: Esta descripción debe ser clara y concisa, NO debe ser la traducción en palabras del código Java.
- (40%) Código del *ServidorParcial*