



Facultad de Ingeniería

Escuela de Ingeniería Civil en Bioinformática

---

# BORRADOR Informe

## Administración de Sistemas

---

*Alumno:*

Kevin Aguilar Valdés

2015430842

kaguilar15@alumnos.utalca.cl

*Profesor:*

Alejandro Valdés Jiménez

*Fecha:*

26 de noviembre de 2020

# Índice

<b>1. Compilado de programas, librerías, kernel</b>	<b>5</b>
1.1. Compilado de programas . . . . .	5
1.2. Librerías . . . . .	7
1.3. Kernel . . . . .	7
1.3.1. Conexión remota a Maquina Virtual . . . . .	7
1.3.2. screen . . . . .	7
1.3.3. Kernel . . . . .	7
1.3.4. Time . . . . .	8
<b>2. Acceso remoto</b>	<b>9</b>
2.1. Usuarios . . . . .	9
2.2. Telnet / Teletype Network Protocol . . . . .	9
2.2.1. Conexiones activas . . . . .	10
2.3. SSH / Secure Shell . . . . .	10
2.4. Secure Copy Protocol - SCP . . . . .	10
2.5. Secure Shell Filesystem - SSHFS . . . . .	11
2.6. Relación de confianza . . . . .	11
<b>3. Administración de servicios - Transferencia de datos</b>	<b>12</b>
3.1. Trivial File Transfer Protocol - TFTP . . . . .	12

3.2. File Transfer Protocol - FTP . . . . .	12
3.3. Implementación - Servidor . . . . .	12
3.4. Implementación - Cliente . . . . .	13
<b>4. DNS</b>	<b>15</b>
4.1. Consulta desde otro equipo . . . . .	16
<b>5. Network File System - NFS</b>	<b>17</b>
5.1. Implementación - Servidor . . . . .	17
5.2. Implementación - Cliente . . . . .	17

# Índice de figuras

1. Pipeline instalación librerías . . . . .	6
---	---

# 1. Compilado de programas, librerías, kernel

GNU Compiler Collection (GCC-GNU) es una suite de compiladores desarrollados por el proyecto GNU. GCC es un software libre y permite a los usuarios generar aportes a este, incentivando las mejoras en el compilador. GCC soporta una amplia gama de plataformas, este fomenta el uso de un compilador de alcance global, que atraiga a desarrolladores para asegurar el buen funcionamiento de GCC y de los sistemas GNU en las diferentes arquitecturas y entornos a los cuales puede acceder, para así extender la funcionalidad de esta herramienta.

En esta primera lección se trabajó con la instalación y compilado de programas necesarios para la implementación de GCC, así como la instalación de kernel para el equipo.

## 1.1. Compilado de programas

En una primera aproximación al compilado de programas se requiere realizar la instalación de una serie de dependencias necesarias para la implementación del compilador GCC. Estas corresponden a las librerías: GMP - MPFR - MPC - ISL.

Para realizar la implementación se utilizó un pipeline de comandos descritos en Figura 1. Se siguió el mismo procedimiento en los cuatro casos, finalizando con la confirmación en la instalación de las librerías. Esto ultimo se logra consultando el directorio `/opt/util_compiler`, en este caso con la utilización del comando `tree`.

```
/opt/util_compiler$ tree lib/
lib/
  libgmp.a
  libgmp.la
  libisl.a
  libisl.la
  libmpc.a
  libmpc.la
  libmpfr.a
  libmpfr.la
  pkgconfig
  isl.pc

1 directory, 9 files
```

Una vez concluida la implementación de las librerías se procede a compilar GCC. Este procedimiento sigue la misma lógica expuesta anteriormente.

Para finalizar con la instalación de GCC v.10.2.0

\$time make	\$time make install
real 237m56.823s	real 0m42.660s
user 228m11.343s	user 0m13.406s
sys 8m56.523s	sys 0m8.862s

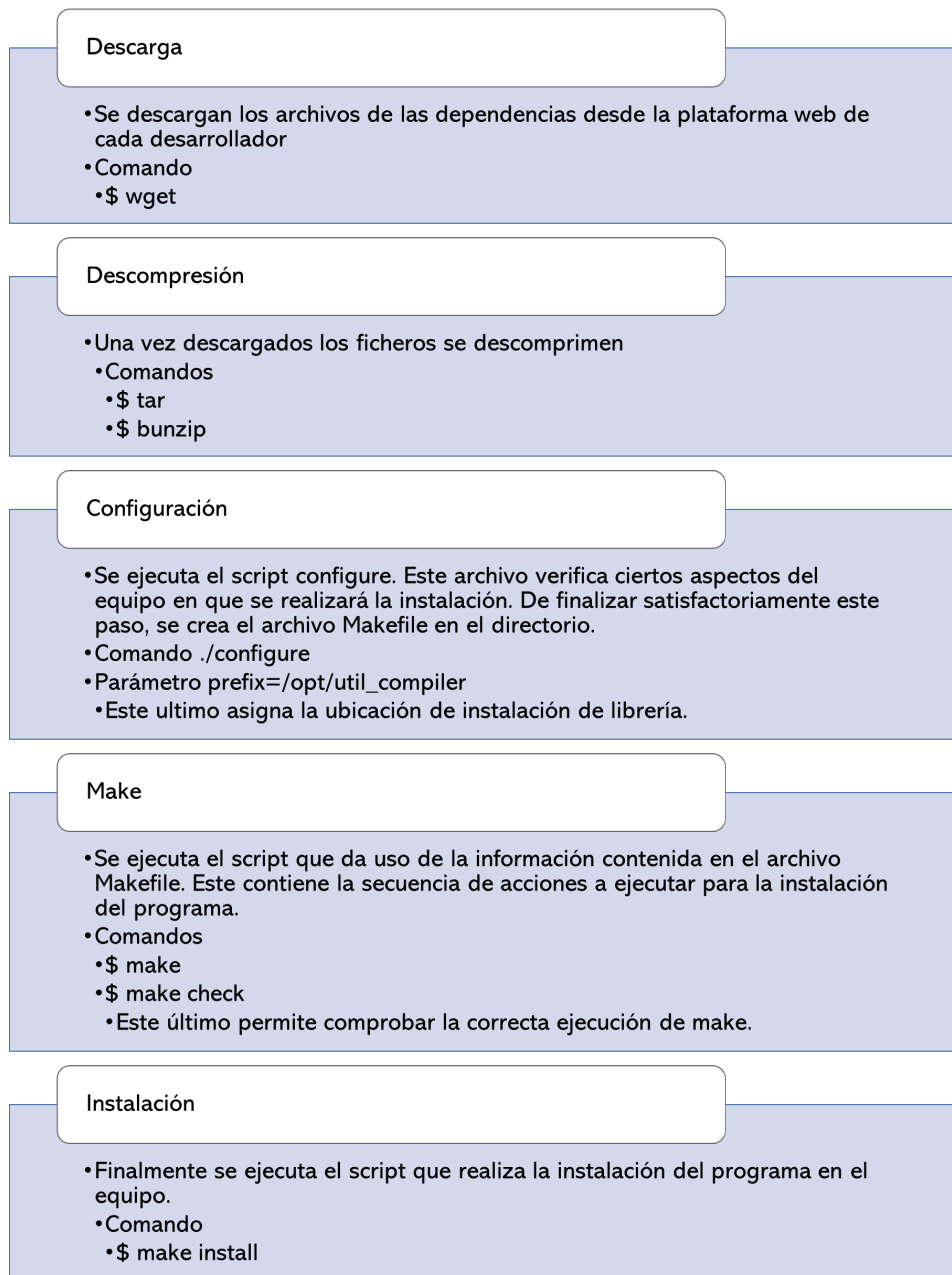


Figura 1: Pipeline instalación librerías  
Se muestra la serie de pasos seguidos para implementar las dependencias de las librerías en el equipo.

## 1.2. Librerías

Para la instalación de librerías se decidió implementar la librería kd-tree.<sup>1</sup>. Esta es una dependencia de C trabaja con arboles de decisión de dimensión k. La estructura de datos permite asignar búsquedas rápidas entre nodos estableciendo distancia entre la información que lo compone.

## 1.3. Kernel

### 1.3.1. Conexión remota a Maquina Virtual

Para realizar esta actividad se dio acceso a una maquina virtual implementada en un servidor de la Universidad. Para lograr la conexión se utilizó el protocolo ssh. Esta herramienta permite la administración remota de los usuarios para controlar o modificar servidores remotos a través de internet. En este caso se modificó el archivo de configuración ssh con los datos necesarios para acceder a la maquina virtual.

### 1.3.2. screen

Screen es un multiplexor de terminales, esto significa que es posible generar sesiones en una pantalla y luego iniciar varios procesos en terminales virtuales, dentro de esta sesión. Los procesos en esta sesión seguirán ejecutándose aun cuando la ventana no sea visible, e incluso cuando se interrumpa la conexión.

El paquete Screen fue implementado en la maquina virtual a través del comando.

```
$ apt install screen
$ screen --version
screen version 4.06.02 (GNU) 23-Oct-17
```

### 1.3.3. Kernel

Para la implementación de kernel el primer paso es descargar el archivo. Se accede a la pagina de Linux Kernel Archives<sup>2</sup> Se descargo la ultima version: 5.8.14.

---

<sup>1</sup><http://nuclear.mutantstargoat.com/sw/kdtree/>

<sup>2</sup><https://www.kernel.org>.

```
$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.8.14.tar.xz
$ tar axvf linux-5.8.14.tar.xz
```

Al finalizar el proceso se comprobó que los archivos creados correspondían a los requeridos.

```
root@admSist1:/usr/src# ls -lhtr --color
total 342M
-rw-r--r-- 1 root root 110M Oct  7 03:42 linux-5.8.14.tar.xz
-rw-r--r-- 1 root root 177M Oct 11 11:39 linux-5.8.14_5.8.14.orig.tar.gz
-rw-r--r-- 1 root root 242K Oct 11 11:39 linux-5.8.14_5.8.14-1.diff.gz
-rw-r--r-- 1 root root 1.1K Oct 11 11:40 linux-5.8.14_5.8.14-1.dsc
-rw-r--r-- 1 root root 7.5M Oct 11 14:34 linux-headers-5.8.14_5.8.14-1_amd64.deb
drwxrwxr-x 25 root root 4.0K Oct 11 14:34 linux-5.8.14
-rw-r--r-- 1 root root 1.1M Oct 11 14:34 linux-libc-dev_5.8.14-1_amd64.deb
-rw-r--r-- 1 root root 47M Oct 11 14:36 linux-image-5.8.14_5.8.14-1_amd64.deb
-rw-r--r-- 1 root root 5.0K Oct 11 14:36 linux-5.8.14_5.8.14-1_amd64.buildinfo
-rw-r--r-- 1 root root 2.6K Oct 11 14:36 linux-5.8.14_5.8.14-1_amd64.changes
```

### 1.3.4. Time

Adicionalmente se utilizó la herramienta en terminal time para determinar el lapso de tiempo en que se generó la compilación.

```
real    178m24.349s
user    167m49.970sGIT
sys     19m48.469s
```



## 2. Acceso remoto

EL acceso remoto es la capacidad de acceder a un dispositivo desde un lugar remoto. Esta acción viene dada por la participación de un software que instalado en el equipo permite el acceso a otra maquina, teniendo la libertad de enlazarse a otro dispositivo. Una vez establecida la conexión es posible controlar el dispositivo, ejecutar aplicaciones, manipular archivos, etc.

Como se menciona anteriormente es necesario el uso de un software que permita la comunicación entre los equipos. En el caso de este practico se utilizarán dos protocolos: SSH<sup>3</sup> y Telnet<sup>4</sup>.

### 2.1. Usuarios

Para acceder a un servidor de forma remota es preciso manejar la información de autenticación del usuario que se desea acceder. En este en particular se estableció un nuevo usuario para permitir a un compañero establecer conexión a la maquina virtual propia. Para lograr esto fue necesario crear un nuevo usuario dentro del equipo. Esto se logró mediante el comando adduser, que permite añadir usuarios (o grupos en su variante addgroup) al sistema de acuerdo a los parametros ingresados en la linea de comandos.

### 2.2. Telnet / Teletype Network Protocol

El protocolo Telnet proporciona una interfaz en linea de comandos para establecer comunicación entre dispositivos remotos. Al ser este un desarrollo muy antiguo, no cuenta con ninguna forma de cifrado de información, ya que todos los datos de nombres de usuario y contraseñas necesarias para acceder a las maquinas se intercambian por la red como texto plano. Sin embargo, existen software que trabajan en paralelo a telnet para poder otorgar ciertas garantías de seguridad al momento de manipular la información. Este es el caso de Kerberos<sup>5</sup>, un protocolo de autenticación de red diseñado para proporcionar acceso acreditado para aplicaciones cliente/servidor mediante la encriptación de las claves de acceso.

---

<sup>3</sup><https://www.ssh.com/ssh/>

<sup>4</sup><https://www.extrahop.com/resources/protocols/telnet/>

<sup>5</sup><https://web.mit.edu/kerberos/>

### 2.2.1. Conexiones activas

Para verificar la conexión del compañero se utilizó el comando `w`. Éste muestra información sobre los usuarios que se encuentran actualmente en la máquina y los procesos que realiza.

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
alumno	pts/0	10.1.1.71	17:26	26:39	0.02 s	0.02 s	-bash
alumno	pts/1	10.1.1.71	17:30	2:00 s	0.09 s	0.01 s	sshd: alumno [priv]
acofre	pts/2	stgo-10.1.1.32.i	17:45	5:42	0.04 s	0.03 s	-bash

## 2.3. SSH / Secure Shell

El protocolo SSH utiliza encriptación para garantizar una conexión segura entre cliente y servidor. Toda la información del usuario, ya sea autenticación, comandos, archivos están encriptados para proteger dicha información de cualquier agente externo. Una vez se establece la conexión entre el cliente y el servidor, estos acuerdan el algoritmo de cifrado a utilizar para así proteger la información.

Conexión remota mediante protocolo ssh. Se accede mediante la identificación del usuario y la dirección ip del equipo.

```
ssh kevin@10.1.1.35
```

## 2.4. Secure Copy Protocol - SCP

Es un protocolo de transferencia segura de archivos informáticos entre un servidor local y otro remoto, o entre dos hosts remotos. Todo esto por medio del protocolo Secure Shell (SSH). Se utilizó el comando para transferir archivos desde el servidor de un compañero mediante el protocolo.

```
scp The_Hobbit.txt kevin@10.1.1.35:/var/kevin/
kevin@10.1.1.35's password:
The_Hobbit.txt

scp kevin@10.1.1.35:/var/kevin/The_Hobbit.txt .
kevin@10.1.1.35's password:
The_Hobbit.txt
519KB 91.1MB/s 00:00
```

100 %

## 2.5. Secure Shell Filesystem - SSHFS

Este es un sistema de archivos utilizado en LINUX, u otros sistemas operativos, que opera sobre archivos en una computadora remota utilizando un entorno seguro de acceso. Este servicio se monta en la computadora local mediante el comando SSHFS. Los efectos prácticos de este servicio es que el usuario puede interactuar mediante la interfaz gráfica con los archivos remotos que se encuentran en el servidor, viendolos como si estos se encontraran en su maquina local.

```
$ sshfs admsist:/home/alumno/remote share-sshfs /  
  
Archivos en servidor remoto, directorio con ruta /home/alumno/remote  
  
$ tree remote/  
remote/  
    file.txt  
    The.Hobbit.txt  
  
Archivos en equipo local mediante enlace por protocolo sshfs  
/home/kevin/share-sshfs  
tree share-sshfs/  
share-sshfs/  
    file.txt  
    The.Hobbit.txt
```

## 2.6. Relación de confianza

La relación de confianza es una conexión establecida entre dos dominios de forma que se permite a los usuarios de una dirección ser reconocidos por el otro dominio sin necesidad de acceder utilizando contraseña. Este procedimiento se logró generando claves públicas utilizando el protocolo ssh-keygen. Posterior a esto se utilizó el comando cat para ubicar las claves creadas en los directorios creados anteriormente. Para efectos de este trabajo se utilizó la maquina virtual facilitada para la realización del curso.

```
#PC1  
ssh-keygen -t rsa  
sh admsist mkdir -p .ssh  
cat .ssh/id_rsa.pub | ssh usuario2@S2 'cat >> .ssh/authorized_keys'
```

Este procedimiento se realizó dos veces debido a que el acceso al servidor incluye un túnel para acceder a la maquina virtual. <sup>6</sup>

---

<sup>6</sup><https://www.activasistemas.com/blog/2014/06/30/acceso-ssh-sin-contrasena-a-servidores-linux/>

## 3. Administración de servicios - Transferencia de datos

### 3.1. Trivial File Transfer Protocol - TFTP

El protocolo TFTP regula transferencia de archivos en redes informáticas. Este protocolo servidor-cliente se basa en la solicitud de acceso del cliente, ya sea para lectura o escritura. Esta solicitud opera como petición de conexión concedida en el momento en que se pide acceso al servidor.

```
/srv/tftp# ll
total 1016K
-rw-r--r-- 1 alumno alumno 43 Nov 5 09:40 file.txt
-rw-r--r-- 1 alumno alumno 520K Nov 5 09:50 the_hobbit.txt
-rw-rw-rw- 1 tftp tftp 489K Nov 12 09:46 El.pdf

$ tftp 10.1.1.31
tftp> get the_hobbit.txt
Received 542147 bytes in 0.1 seconds
tftp> put The_Hobbit.txt
Sent 542147 bytes in 0.1 seconds
```

### 3.2. File Transfer Protocol - FTP

El protocolo de transferencia de archivos es un servicio de red para la transferencia de archivos entre sistemas utilizando una red TCP. Este protocolo se basa en la arquitectura cliente-servidor, pudiendo conectar desde un equipo cliente a un servidor para descargar o enviar archivos. Todo esto, independiente del sistema operativo utilizado en cada equipo. Un problema asociado a FTP es que está diseñado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, esto debido a que desde la información de sesión hasta la transferencia de archivos se realiza en texto plano, con lo que es posible poder capturar el tráfico, acceder al servidor o apropiarse de archivos. Para la realización de este practico se utilizó el tutorial disponible en la web <https://www.redeszone.net/tutoriales/servidores/proftpd/>.

### 3.3. Implementación - Servidor

Primeramente se instala el servidor del servicio FTP.

```
# apt upgrade
# apt install proftpd-basic
```

Posteriormente a la instalación del software se procede a configurar el servidor FTP. Esto incluye configuraciones como a especificación de los nombres de host del servidor o

los directorios que se desea compartir para la carga y descarga de archivos. Los archivos de configuración corresponden a `/etc/proftpd/proftpd.conf` y `/etc/default/proftpd`.

Para manejar el estado del servicio se consulta a través de los siguientes comandos

```
# systemctl status proftpd.service
# systemctl start proftpd.service
# systemctl restart proftpd.service
# systemctl stop proftpd.service
```

Con el fin de proteger la información se le añade una capa TLS. Para implementar el protocolo TLS/SSL se debe configurar en el servidor un certificado RSA (o similar), con su clave publica y privada, y de esta forma poder establecer correctamente el servicio FTP. En primer lugar se debe activar la directiva en el archivo de configuración `/etc/proftpd/tls.conf`. Luego se crea el certificado digital RSA, generando una clave privada con 4096 bits RSA. Posteriormente se importa la clave y exporta el certificado al directorio `/etc/ssl/certs/proftpd.crt`. Para finalmente editar el archivo `/etc/proftpd/tls.conf`.

```
# openssl genrsa -out /etc/ssl/private/proftpd.key 4096
# openssl req -new -x509 -days 1460 -key /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt
```

ARCHIVO `/etc/proftpd/tls.conf`

```
<IfModule mod_tls.c>
    TLSEngine                 on
    TLSLog                    /var/log/proftpd/tls.log
    TLSProtocol                TLSv1 TLSv1.1 TLSv1.2

    TLSRSACertificateFile      /etc/ssl/certs/proftpd.crt
    TLSRSACertificateKeyFile    /etc/ssl/private/proftpd.key

    TLSVerifyClient            on
    TLSVerifyClient            on
```

### 3.4. Implementación - Cliente

Para lograr la conexión al servidor es preciso contar con la implementación del cliente FTP (o bien LFTP) en el equipo de destino. En este caso se trabajó con el cliente LFTP. Se ejecuta el siguiente comando:

```
apt install lftp
```

Acceso a protocolo FTP en la máquina IP 10.1.1.35

```
lftp sftp://alumno@10.1.1.35
Password:
lftp alumno@10.1.1.35:~> ls
drwxr-xr-x  11 alumno  alumno    4096 Nov 25 23:38 .
drwxr-xr-x   5 root    root      4096 Nov  5 19:37 ..
-rw-r--r--   1 alumno  alumno    6121 Nov 25 23:38 .bash_history
-rw-r--r--   1 alumno  alumno     220 Sep 24 20:38 .bash_logout
-rw-r--r--   1 alumno  alumno    3526 Sep 24 20:38 .bashrc
drwxr-xr-x   3 alumno  alumno    4096 Nov 12 09:20 .config
drwx-----  3 alumno  alumno    4096 Oct  8 15:41 .gnupg
drwxr-xr-x   3 alumno  alumno    4096 Oct 16 12:30 .local
```

-rw-r--r--	1	alumno	alumno	807	Sep	24	20:38	.profile
drwx-----	2	alumno	alumno	4096	Nov	5	15:38	.ssh
drwxr-xr-x	2	alumno	alumno	4096	Nov	12	08:55	.vim
-rw-----	1	alumno	alumno	11164	Nov	25	23:38	.viminfo
-rw-r--r--	1	alumno	alumno	4	Nov	5	15:52	1.txt
-rw-r--r--	1	alumno	alumno	499759	Aug	16	2017	El.pdf
-rw-r--r--	1	alumno	alumno	531525	Nov	12	09:39	TheHobbit.txt
-rw-r--r--	1	alumno	alumno	25	Nov	5	18:31	archivo1.txt
drwxr-xr-x	2	alumno	alumno	4096	Oct	16	13:05	carpeta_personal
-rw-r--r--	1	alumno	alumno	25	Oct	26	15:00	prueba_sshfs.txt
drwxr-xr-x	2	alumno	alumno	4096	Nov	26	04:48	servidor_Nico
drwxr-xr-x	2	alumno	alumno	4096	Nov	25	13:08	shared_Nico
drwxr-xr-x	2	alumno	alumno	4096	Nov	25	13:27	shared_willy
-rw-r--r--	1	alumno	alumno	13	Oct	16	13:07	top_secret.txt

De la misma forma se realizó la conexión desde un cliente externo al servidor local, confirmando su implementación.

\$ lftp sftp://alumno@10.1.1.31								
Password:								
lftp alumno@10.1.1.31:~> ls								
drwxr-xr-x	9	alumno	alumno	4096	Nov	25	11:46	.
drwxr-xr-x	5	root	root	4096	Oct	29	19:21	..
-rw-----	1	alumno	alumno	5399	Nov	26	11:57	.bash_history
-rw-r--r--	1	alumno	alumno	220	Sep	24	20:38	.bash_logout
-rw-r--r--	1	alumno	alumno	3555	Oct	12	12:21	.bashrc
drwx-----	5	alumno	alumno	4096	Nov	12	08:52	.config
drwx-----	3	alumno	alumno	4096	Oct	9	11:55	.gnupg
drwxr-xr-x	3	alumno	alumno	4096	Nov	12	08:52	.local
-rw-r--r--	1	alumno	alumno	807	Sep	24	20:38	.profile
drwx-----	2	alumno	alumno	4096	Nov	5	10:51	.ssh
-rw-----	1	alumno	alumno	9493	Nov	25	09:44	.viminfo
drwxr-xr-x	3	alumno	alumno	4096	Oct	15	08:52	admsist
drwxr-xr-x	2	root	root	4096	Nov	5	11:59	remote
drwxr-xr-x	2	alumno	alumno	4096	Nov	25	11:50	shared-nfs
lftp alumno@10.1.1.31:~>								

## 4. DNS

### Instalación

```
# cat /etc/bind/db.kav.cl
;
; BIND data file for local loopback interface
;
$TTL      604800
$ORIGIN   kav.cl.
@         IN      SOA      kav.cl. hostmaster.kav.cl. (
        2020011181      ; Serial
        3600            ; Refresh [1h]
        600             ; Retry [10m]
        86400           ; Expire [1d]
        600 )           ; Negative Cache TTL [1h]
;
;
@         IN      NS       kav.cl.
@         IN      A        10.1.1.31
;
test1     IN      A        10.1.1.31
```

### Creación y configuración de archivo con opciones de zona

```
# vim /etc/bind/named.conf.kav.cl
# cat /etc/bind/named.conf.kav.cl
zone "kav.cl" {
    type master;
    file "/etc/bind/db.kav.cl";
};
```

### Incluir zona al servicio /etc/bind/named.conf

```
# vim /etc/bind/named.conf
root@admSist1:~# cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
include "/etc/bind/named.conf.kav.cl";
```

### Se ingresó la ip a la resolución de nombres: /etc/resolv.conf

```
# vim /etc/resolv.conf
# cat /etc/resolv.conf
domain icb.local
search icb.local
nameserver 10.1.1.31
nameserver 192.168.20.2
```

Posteriormente se utiliza el comando nslookup para determinar si la configuración fue adecuada y se presentó el siguiente problema.

```
# nslookup kav.cl
Server:          192.168.20.2
Address:         192.168.20.2#53
** server can't find kav.cl: NXDOMAIN
```

El problema se presentó debido a que no se reinició el servicio una vez configurado el archivo de resolución de nombres /etc/resolv.conf. Se procedió a recargar el servicio y se consultó nuevamente, siendo exitoso este intento.

```
# systemctl reload bind9
# nslookup kav.cl
Server:      10.1.1.31
Address:     10.1.1.31#53

Name:   kav.cl
Address: 10.1.1.31
```

## 4.1. Consulta desde otro equipo

Para verificar la implementación del servicio se realizó lo siguiente:

```
$ nslookup kav.cl
Server:      192.168.20.2
Address:     192.168.20.2#53

** server can't find kav.cl: NXDOMAIN
```

---

```
# vim /etc/resolv.conf
# cat /etc/resolv.conf
domain icb.local
search icb.local
nameserver 10.1.1.31
nameserver 192.168.20.2
```

---

```
# nslookup kav.cl
Server:      10.1.1.31
Address:     10.1.1.31#53

Name:   kav.cl
Address: 10.1.1.31
```

```
# nslookup www.kav.cl
Server:      10.1.1.31
Address:     10.1.1.31#53
```

```
Name:   www.kav.cl
Address: 10.1.1.31
```

```
# nslookup sub.kav.cl
Server:      10.1.1.31
Address:     10.1.1.31#53
```

```
Name:   sub.kav.cl
Address: 10.1.1.31
```



## 5. Network File System - NFS

El sistema de archivos de red es un protocolo utilizado para sistemas de archivos distribuidos en un entorno de red. Éste posibilita a distintos sistemas conectados a una misma red para que accedan a ficheros de forma remota como sería de forma local. El sistema NFS está dividido en dos partes principales: un servidor y uno (o más) clientes. Estos últimos acceden de forma remota a los datos que se encuentran almacenados en el servidor. Los archivos almacenados en los clientes utilizan menor espacio de disco debido a que los datos se encuentran centralizados en el servidor, de donde pueden ser accedidos y modificados por los diferentes clientes, de esta manera no es necesario replicar la información.

### 5.1. Implementación - Servidor

Para habilitar el sistema en el equipo local es necesario instalar el software del servidor.

```
apt install nfs-kernel-server portmap
```

Posterior a esto se crea una carpeta local que será designada como la contenedora de los archivos a compartir.

```
$mkdir shared-nfs
$readlink -f
/home/alumno/shared-nfs
```

Para dar al servicio NFS el acceso a al directorio recién creado se configuran los archivos a compartir en el archivo */etc/exports*. y posteriormente se reinicia el servicio para volver a cargar la configuración recién modificada.

```
# vim /etc/exports
root@admSist1:/home/alumno# cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/alumno/shared-nfs 10.1.1.35/24(rw)
# /etc/init.d/nfs-kernel-server restart
```

### 5.2. Implementación - Cliente

En primera instancia se instala el software del cliente.

```
$ apt install nfs-common
```

Posteriormente se crea un directorio para enlazarlo al creado anteriormente en el servidor y se monta el recurso en dicho directorio.

```
# mkdir shared  
# mount -t nfs4 10.1.1.31:/home/alumno/shared-nfs shared/
```

7

---

<sup>7</sup><https://www.ionos.es/digitalguide/servidores/know-how/tftp/>