



Universidad Luterana Salvadoreña

Cátedra: Redes II

Docente: Ing. Manuel de Jesús Flores

Evaluación: Proyecto Final
Fecha de entrega: 14/11/2015

INSTALADOR AUTOMÁTICO DE GNU/LINUX

Estudiante:

Carnet	Apellidos	Nombres	Participación
DP02110151	Domínguez Parada	Luis Antonio	100%
GM02110987	Granados Maradiaga	Elmer Alexander	100%

Índice

Introducción	4
Objetivos	5
Objetivo general	5
Objetivo específico	5
Marco teórico.....	6
FAI	6
DHCP	7
Puertos	7
Funciones de DHCP	7
TFTP.....	8
NFS	8
PXE	9
Funcionamiento	9
Protocolo.....	10
Servidor de arranque.....	10
Descripción del proyecto.....	11
Diagrama de red.....	11
Lista de actividades	12
Diagrama de Gantt	13
Factibilidad del proyecto.....	14
Factibilidad económica.....	14
Presupuesto	14
Factibilidad Operativa	15
Factibilidad Técnica.	15
Factibilidad Legal.....	15
Construcción del proyecto	16
Instalación FAI	16
Requerimientos	16
Crear una réplica local de Debian.....	17
Configuración de la FAI.....	18
Configuración de interfaces	20

Escenarios de prueba	25
Conclusión.....	30
Recomendaciones	30
Bibliografía	31
Anexo	32

Índice de ilustraciones

Ilustración 1. Diagrama DHCP	7
Ilustración 2. Diagrama FTP.....	8
Ilustración 3. Diagrama NFS	8
Ilustración 4. Diagrama PXE	9
Ilustración 5. Diagrama de red	11
Ilustración 6. Diagrama de Gantt	13
Ilustración 7: repositorio de proyecto fai	20
Ilustración 8: actualizar lista de paquetes	20
Ilustración 9: configuración FAI	21
Ilustración 10: configuración nfsroot	21
Ilustración 11: archivo hosts	22
Ilustración 12: configuración DHCP	22
Ilustración 13: archivo hostname	23
Ilustración 14: creación de archivos FAI	23
Ilustración 15: repositorios locales.....	23
Ilustración 16: arranque PXE	25

Introducción

El presente proyecto contendrá la información necesaria para la implementar el proyecto de auto instalación de GNU/LINUX por medio de una red LAN, la instalación de nuestro sistema operativo (Debian Jessie) deberá ser completamente desatendida

El proyecto comprende en la instalación desatendida de un sistema GNU/LINUX en todas más computadoras conectadas a la misma red LAN (centro de cómputo), instalando el sistema operativo sin la necesidad de configuración manual como zona horaria, fecha y hora, teclado, idioma, etc.

En el transcurso e implementación de este proyecto se ha utilizado la herramienta FAI (fully automatic installation), esta herramienta nos ayuda a instalar el sistema operativo en varias PC's a la vez de forma desatendida, dando como resultado un gran ahorro de tiempo en configuración del centro computo.

FAI instala un sistema base en nuestras maquinas clientes creando las particiones en el disco duro de cada PC.

Además de esto descarga los paquetes necesarios para que nuestro sistema operativo funciones correctamente, por lo cual se ha implementado un espejo local, para que no se tenga que descargar de internet los paquetes para cada PC cliente.

Objetivos

Objetivo general

- Implementar el proyecto de **“Auto instalación de GNU/LINUX Debían desde red”**, configuración, demostración e implementación de clientes.

Objetivo específico

- Mostrar las tecnologías y protocolos necesarios para la implementación del proyecto.
- Dar a conocer las ventajas de la Auto Instalación por medio de red LAN y en qué caso se podría utilizar.
- Demostrar el funcionamiento del proyecto

Marco teórico

FAI

El proyecto *de la FAI*, o "Instalación completamente automática" (<http://fai-project.org/>), es un proyecto creado hace más de 10 años en la Universidad de Colonia en Alemania, que está diseñado para un rápido despliegue de servidores o sistemas *Debian* y otras distribuciones de forma no interactiva.

En este modelo, la instalación es totalmente diferente de una instalación predeterminada de *Debian* y sólo se recomienda para las nuevas máquinas (reset) en máquinas virtuales o máquinas que contienen ningún sistema preinstalado, excepto, por supuesto, en el caso de reemplazo.

A medida que la orientación dada en el artículo original , no pido para instalar discos que ya contienen datos, bajo pena de pérdida de la misma!

Es un sistema muy interesante que carga un script todos los ajustes necesarios para la instalación de un servidor *Debian* o un escritorio *XFCE* automáticamente! ¿Dónde se lleva a cabo sin duda durante el proceso, un proceso que en *VirtualBox* tomó menos de seis minutos desde el CD de arranque a la pantalla de *GRUB* instalado.

La *FAI* se utiliza en muchos sistemas grandes, como en grandes grupos , especialmente por su rápida instalación y no requiere acceso a Internet para completar el proceso, se utiliza por ejemplo en el Proyecto *Beowulf* , incluyendo el manual hay un capítulo los dedicados este tipo de instalación.

También fue utilizado en el *SS* (super-servidor) 200 Teraflops, <http://i.top500.org/system/10804> , Electricité de France SA, que en junio de 2011 ocupó el lugar número 43 en el ranking del *Top500*, el sistema utilizado fue el *Squeeze Debian* .

Más allá de lo que he mencionado anteriormente, creo que un modelo interesante, ya sea para Administradores (estudiantes de TI, que recibieron un servidor *Debian* listo en poco tiempo), o para los que no conocen *Debian* y las damas y caballeros que no pierden una novedad (respetando por supuesto la orientación sobre dónde instalar!).

DHCP

"DHCP (siglas en inglés de Dynamic Host Configuration Protocol - Protocolo de configuración dinámica de host) es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después."(Wikipedia)"

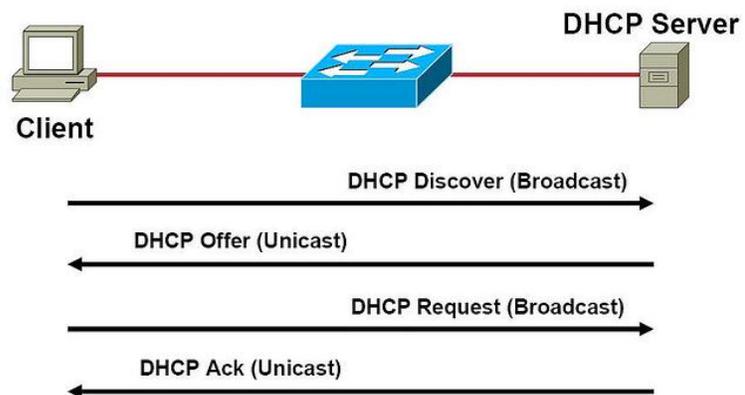


Ilustración 1. Diagrama DHCP

Puertos 67/UDP (servidor), 68/UDP (cliente)

Funciones de DHCP

Un servidor DHCP puede proveer de una configuración opcional al dispositivo cliente.

- Dirección del servidor DNS
- Nombre DNS
- Puerta de enlace de la dirección IP
- Dirección de Publicación Masiva (*broadcast address*)
- Máscara de subred
- Tiempo máximo de espera del ARP (*Protocolo de Resolución de Direcciones* según siglas en inglés)
- MTU (*Unidad de Transferencia Máxima* según siglas en inglés) para la interfaz
- Servidores NIS (*Servicio de Información de Red* según siglas en inglés)
- Dominios NIS
- Servidores NTP (*Protocolo de Tiempo de Red* según siglas en inglés)
- Servidor SMTP
- Servidor TFTP
- Nombre del servidor WINS

TFTP

"TFTP son las siglas de Trivial file transfer Protocol (Protocolo de transferencia de archivos trivial). Es un protocolo de transferencia muy simple semejante a una versión básica de FTP. TFTP a menudo se utiliza para transferir pequeños archivos entre ordenadores en una red, como cuando un terminal X Window o cualquier otro cliente ligero arranca desde un servidor de red." (Wikipedia).

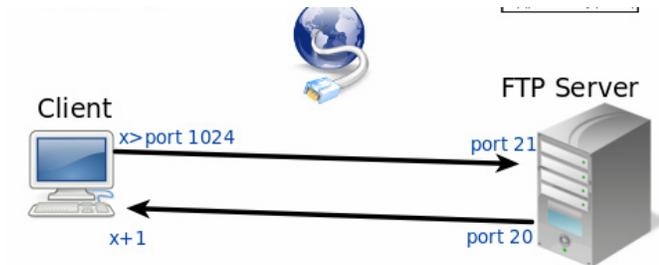


Ilustración 2. Diagrama FTP

- Utiliza UDP (en el puerto 69) como protocolo de transporte (a diferencia de FTP que utiliza los puertos 20 y 21 TCP).
- No puede listar el contenido de los directorios.
- No existen mecanismos de autenticación o cifrado.
- Se utiliza para leer o escribir archivos de un servidor remoto.
- Soporta tres modos diferentes de transferencia, "netascii", "octet" y "mail", de los que los dos primeros corresponden a los modos "ascii" e "imagen" (binario) del protocolo FTP.
-

NFS

El Network File System (*Sistema de archivos de red*), o NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. Originalmente fue desarrollado en 1984 por Sun Microsystems, con el objetivo de que sea independiente de la máquina, el sistema operativo y el protocolo de transporte, esto

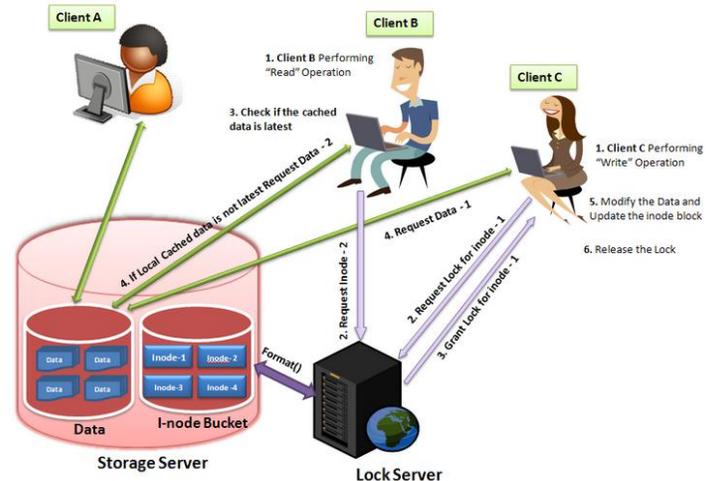


Ilustración 3. Diagrama NFS

fue posible gracias a que está implementado sobre los protocolos XDR (presentación) y ONC RPC (sesión).¹ El protocolo NFS está incluido por defecto en los Sistemas Operativos UNIX y la mayoría de distribuciones Linux.

Puertos que utiliza:

nfs 2049/tcp nfsd
nfs 2049/udp nfsd

PXE

"Preboot eXecution Environment (PXE) (Entorno de ejecución de prearranque), es un entorno para arrancar e instalar el sistema operativo en ordenadores a través de una red, de manera independiente de los dispositivos de almacenamiento de datos disponibles (como discos duros) o de los sistemas operativos instalados.

En nuestro caso PXE se utiliza para cargar una minimagen de Linux que no se instalara en el equipo destino. Etherboot o Yaboot se utilizan para lo mismo en este sistema para obtener del servidor la minimagen de Linux.

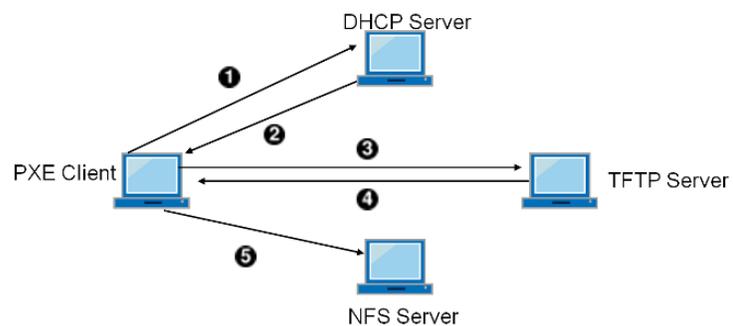


Ilustración 4. Diagrama PXE

Funcionamiento

El firmware del cliente trata de encontrar un servicio de redirección PXE en la red para recabar información sobre los servidores de arranque PXE disponibles. Tras analizar la respuesta, el firmware solicitará al servidor de arranque apropiado el *file path* de un *network bootstrap program* (NBP), lo descargará en la memoria RAM del ordenador mediante TFTP, probablemente lo verificará, y finalmente lo ejecutará. Si se utiliza un único NBP para todos los clientes PXE se puede especificar mediante BOOTP sin necesidad de un proxy DHCP, pero aún será necesario un servidor TFTP.

Protocolo

El protocolo PXE consiste en una combinación de los protocolos DHCP y TFTP con pequeñas modificaciones en ambos. DHCP es utilizado para localizar el servidor de arranque apropiado, con TFTP se descarga el programa inicial de bootstrap y archivos adicionales.

Para iniciar una sesión de arranque con PXE el firmware envía un paquete de tipo DHCPDISCOVER extendido con algunas opciones específicas de PXE al puerto 67/UDP (puerto estándar del servicio DHCP). Estas opciones indican que el firmware es capaz de manejar PXE, pero serán ignoradas por los servidores DHCP estándar.

Servidor de arranque

Para contactar con cualquier servidor de arranque PXE el firmware debe obtener una dirección IP y el resto de información de un único paquete DHCP OFFER extendido. Tras elegir el servidor de arranque PXE apropiado el firmware envía un paquete DHCPREQUEST extendido mediante multicast o unicast al puerto 4011/UDP o broadcast al puerto 67/UDP. Este paquete contiene el servidor de arranque PXE y la capa de arranque PXE, permitiendo ejecutar múltiples tipos de servidores de arranque mediante un único *daemon* (o programa) de arranque. El paquete DHCPREQUEST extendido también puede ser un paquete DHCPINFORM.

Si un servidor de arranque PXE recibe un paquete DHCPREQUEST extendido como el descrito anteriormente y si está configurado para el tipo de servidor de arranque PXE y la arquitectura de cliente solicitado, debe responder devolviendo un paquete DHCPACK extendido con opciones específicas de PXE.

El contenido más importante de un paquete DHCPACK extendido es:

- el *file path* completo para descargar el NBP vía TFTP
- el tipo de servidor de arranque PXE y la capa de arranque PXE
- la configuración multicast TFTP, si debe utilizarse multicast TFTP

Un servidor de arranque PXE debe soportar *Boot Integrity Services* (BIS). BIS permite al cliente PXE verificar los NBPs descargados mediante un archivo de checksum que es descargado desde el mismo servidor de arranque que el NBP.

Descripción del proyecto

Este proyecto se basará en la implementación de un servidor de Instalación de GNU/LINUX Debian por medio de red LAN, usando el sistema operativo Debian Jessie 8, el proyecto estará enfocado en que todas las computadoras de un centro de cómputo dentro de una misma red LAN, puedan ser formateadas, reinstalado y configurado el sistema operativo Debian Jessie.

Diagrama de red

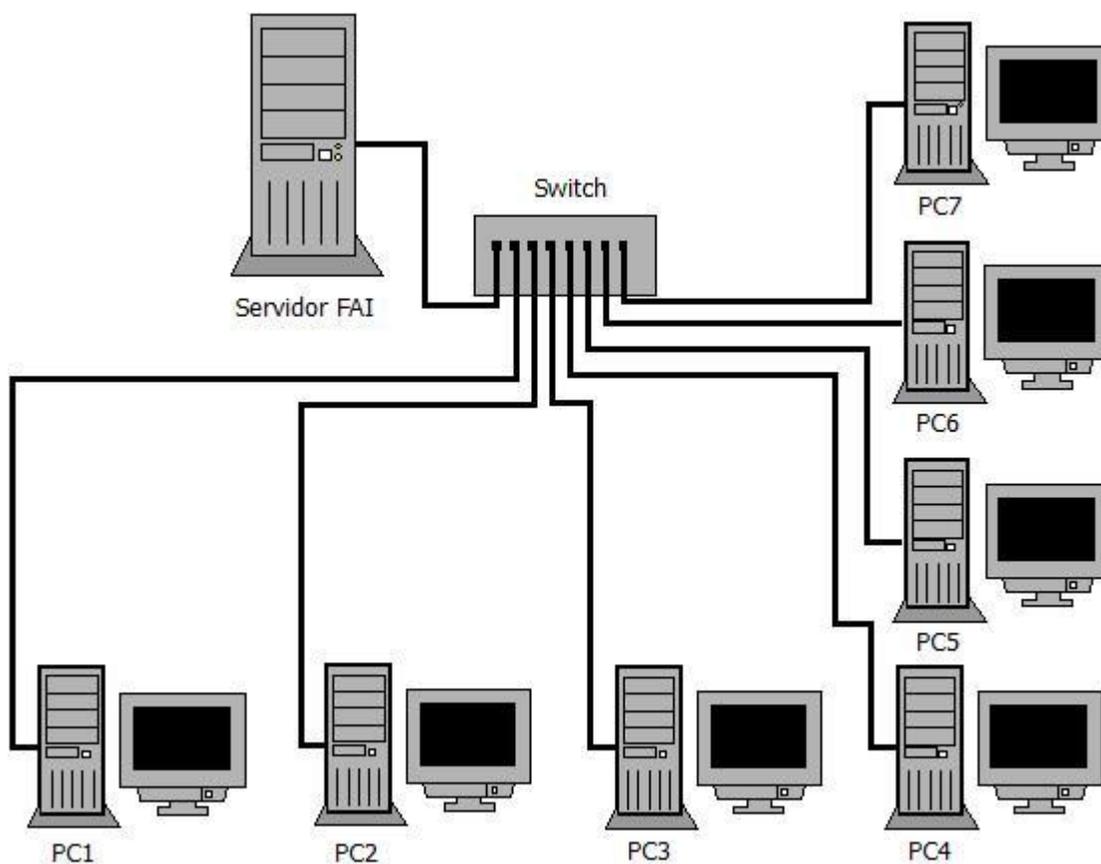


Ilustración 5. Diagrama de red

Lista de actividades

- Recopilar información para implementación del proyecto
- Creación del perfil de proyecto
- Analizar requerimientos y operaciones
- Instalación de Virtual Box
- Instalación de sistema operativo Debian Jessie
- Creación de la red física
- Instalación de paquete FAI
- Instalación de servidor DHCP
- Configuración de servidor DHCP
- Instalación y configuración de TFTP
- Instalación y configuración de NFS
- Pruebas de red
- Pruebas de booteo de los clientes
- Crear mirror local

Diagrama de Gantt

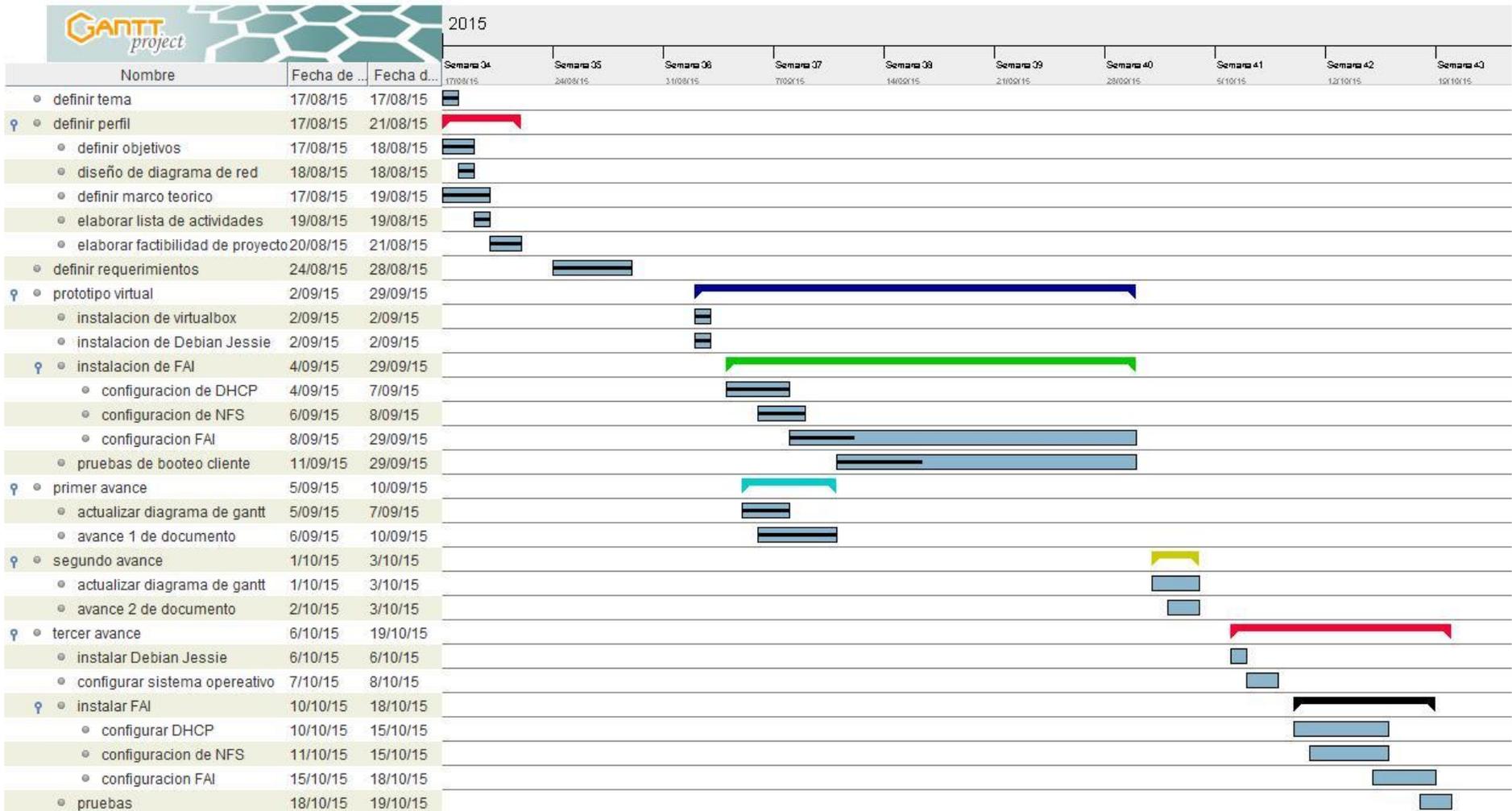


Ilustración 6. Diagrama de Gantt

Factibilidad del proyecto

Factibilidad económica

Se presenta una tabla de costos de la implementación del proyecto de “auto instalación de Debian por medio de red LAN”

Tomando en cuenta que una PC con las características que se necesitan para la implementación del proyecto cuesta \$300.00 en el Salvador.

Además que un profesional de la informática cobra \$1,200.00 al mes por prestar sus servicios en el Salvador.

Presupuesto

cantidad	Descripción	costo
1	Computadora servidor (microprocesador Dual core 2.0 GHZ o más, RAM de 4 GB o más)	\$300.00
1	Sistema operativo para maquina servidor(debian)	\$0.00
1	Switch de 8 puertos	\$10.00
5	Metros de cable UTP por pc	\$2.50
2	Conectores RJ45 por pc	\$5.00
50	Impresiones de (documento)	\$10.00
2	Honorarios de un mes de \$1200	\$2,400.00
2	Capacitación de encargados de centro de computo	\$800
TOTAL		\$3,522.50

Observando los datos anteriores y valorando la importancia que aportara este proyecto en cualquier tipo de entidad que lo sugiera, es relativamente accesible y cómodo.

Ya que se puede implementar en un laboratorio de cómputo o empresa que formatee todas las computadora de una misma red, ahorrando mucho tiempo en configuración de equipos.

Factibilidad Operativa

La factibilidad operacional del proyecto es la instalación y configuración del sistema operativo de todas las computadoras de un centro de cómputo, al tener la instalación del sistema operativo y todas las aplicaciones y configuración de la instalación en un servidor se optimiza el tiempo de instalación y al optimizar el tiempo se convierte en eficiente el proceso de instalación

Factibilidad Técnica.

Se pretende utilizar una PC con las siguientes características:

Procesador: i5 a 3.3 GHZ

Memoria RAM: DDR3 de 8 GB

Red: Ethernet 10/100

Además de la conexión extra para dicho servidor para el cual será necesario un cable UTP de 10 metros y dos conectores RJ45

Se tendrá que capacitar al personal de IT, ya que se asume que no tienen conocimiento de este tipo de herramienta tecnológica, la cual consistirá en el mantenimiento y configuración del servidor de instalación y los pasos a seguir para la instalación por medio de la red LAN.

Al estar correctamente instalado y con los servicios ejecutándose correctamente, el personal de mantenimiento solo tendrá que arrancar las computadoras del centro de cómputo por medio de red (PXE, Netboot)

Factibilidad Legal

El software que se utilizara será en su totalidad libre lo que quiere decir que no se violara ningún contrato de licencia. No habrá ningún contrato de alguna licencia.

Construcción del proyecto

Instalación FAI

Requerimientos

Los siguientes elementos son necesarios para una instalación a través de la FAI.

Un ordenador

El equipo debe tener una tarjeta de interfaz de red. A menos que se debe realizar una instalación sin disco también se necesita un disco duro local. No se necesita un disquete, CD-ROM, teclado o adaptador de gráficos.

Servidor DHCP

Los clientes necesitan uno de estos demonios para obtener información de arranque.

Servidor TFTP

El demonio TFTP se utiliza para transferir el núcleo de los clientes. Es sólo necesario cuando arranque desde la tarjeta de red con un PROM de arranque.

NFS-Root

Es un directorio que contiene todo el sistema de archivos para los clientes instalar durante la instalación. Debe ser exportado a través de NFS, por lo que los clientes pueden instalar montarlo. Se creó durante la instalación del paquete de la FAI y también se llama **nfsroot**.

Debian espejo

Se necesita acceso a una réplica de Debian. Un espejo local de todos los paquetes de Debian o un apt-proxy(8) se recomienda si instala varias computadoras.

Espacio de configuración

Este árbol de directorios, que contiene los datos de configuración, está montado a través de NFS de forma predeterminada. Pero también se puede obtener este directorio de un sistema de control de versiones como CVS, subversión o Git.

Crear una réplica local de Debian

El guión `mkdebmirror` se puede utilizar para crear su propia réplica local de Debian. Este script utiliza el guión `debmirror` Un espejo parcial Debian sólo para la arquitectura `i386` para Debian 8.0 sin los paquetes de código necesita alrededor de 22GB de espacio en disco. Acceso al espejo a través de HTTP será la forma por defecto en la mayoría de los casos. Para ver más producción de la llamada escritura `mkdebmirror -v` Una cuenta de root no es necesario para crear y mantener la réplica de Debian.

Se puede utilizar el comando `fai-mirror` para la creación de un espejo parcial que sólo contiene los paquetes de software que se utilizan en las clases en su espacio de configuración. Un espejo parcial que contiene todo el paquete para los ejemplos simples del paquete `fai-doc` sólo necesitará aproximadamente 440MB de espacio en disco.

Otra opción es copias los paquetes directamente de los DVD's.

Para ello digitamos los siguientes comandos:

```
# cp -R /media/cdrom0/pool/* /repo
```

Creamos el archivo que contiene la información de paquetes para apt. Como root y despues de un rato se crea el archivo `Packages.gz`.

```
# apt-ftpparchive packages . /dev/null | gzip -c -9 > Packages.gz
```

Para utilizar el acceso HTTP a la réplica local de Debian, instalar un servidor web y crear un enlace simbólico al directorio local donde se encuentra su espejo:

```
# apt-get install apache2  
# ln -s /repo/ /var/www/html/repo
```

Cree un archivo `sources.list` en `/etc/fai/apto` que da acceso a su réplica. Un ejemplo se puede encontrar en `/usr/share/doc/fai-doc/examples/etc`. También agregue la dirección IP del servidor HTTP para las variables `$NFSROOT_ETC_HOSTS` en maquillaje `fai-nfsroot.conf` cuando los clientes instalar no tienen acceso DNS.

Ejemplo: deb <http://192.168.1.250/repo/>

Configuración de la FAI

Para configurar un servidor de instalación de la FAI se necesitan al menos los paquetes *fai-servidor* y *fai-doc*. El paquete *fai-quickstart* contiene dependencias en todos los paquetes necesarios para un servidor de instalación. No instalar el paquete *fai-nfsroot* en un sistema normal. Este paquete sólo se puede instalar en el interior del *nfsroot*.

Si desea instalar todos los paquetes que son útiles para un FAI instalar el servidor, utilice el siguiente comando

```
# aptitude install fai-quickstart
```

```
Reading Package Lists... Done
```

```
Building Dependency Tree
```

```
Reading extended state information
```

```
Initializing package states... Done
```

```
Reading task descriptions... Done
```

```
The following NEW packages will be installed:
```

```
apt-move{a} isc-dhcp-server{a} fai-doc{a} fai-quickstart fai-server{a}
```

```
genisoimage{a} inetutils-inetd{a} nfs-kernel-server{a}
```

```
openssh-server{a} syslinux-common{a} tftpd-hpa{a}
```

```
0 packages upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
```

```
Need to get 2593kB of archives. After unpacking 8561kB will be used.
```

```
Do you want to continue? [Y/n/?]
```

La configuración para el paquete de la FAI (no los datos de configuración para los clientes instalar) se define en *fai.conf*. Definiciones que sólo se utilizan para la creación de la *nfsroot* se encuentran en *nfsroot.conf*. Edite estos archivos antes de llamar *fai-setup*. Estas son variables importantes en *nfsroot.conf*:

FAI_DEBOOTSTRAP

Para la construcción de la *nfsroot* está el comando llamado *debootstrap*. Necesita la ubicación de una réplica de Debian y el nombre de la distribución, para los que el sistema básico de Debian debe ser construido.

NFSROOT_ETC_HOSTS

Si utiliza HTTP o FTP de acceso a la réplica de Debian, añada su dirección IP y el nombre de esta variable. Para un nodo maestro Beowulf, añada el nombre y la dirección IP de ambas redes a la misma. No se necesita esta variable cuando los clientes tienen acceso a un servidor DNS.

FAI_CONFIG_SRC

Estas variables describen cómo acceder al espacio de configuración en los clientes instalar. Es un Universal Resource Identifier (URI). Métodos soportados actualmente son:

- `nfs://host/path/to/exported/config` El espacio de configuración se monta desde el host a través de NFS.
- `cvs[+ssh]://user@host/path/to/cvsroot module[=tag]` Se recibe el espacio de configuración de una salida cvs.
- `svn://user@host/svnpath` El espacio config desprotegido de un repositorio de subversión. También con el apoyo son `svn + archivo`, `svn + http`, `svn + ssh`, `svn + https` y cajas sin un nombre de usuario.
- `git://host/path` El espacio config desprotegido de un repositorio git, el anfitrión puede estar vacío. También apoyaron es `git + http`.
- `http://host/path/tarfile` El espacio de configuración se puede descargar desde el lugar dado a través de HTTP. Además `path/tarfile.md5` se descargará y la suma md5 será revisado. `tarfile` será extraído por `ftar` y por lo tanto debe tener un sufijo reconocido, como `.tar.gz`.

`$FAI_CONFIG_SRC` se define normalmente por `fai-chboot` con la opción `-u`.

Recuerde que este directorio debe ser exportado a todos los clientes a instalar, si se utiliza el protocolo NFS.

FAI_DEBMIRROR

Si tiene acceso NFS a la réplica local de Debian, especifique el sistema de archivos remoto. Se monta en `$MNTPOINT` que también debe ser definido. No es necesario si utiliza el acceso a través de FTP o HTTP.

El contenido de `/etc/fai/apt/sources.list` y `$FAI_DEBMIRROR` son utilizados por el servidor de instalación y también por los clientes. Si el servidor de instalación tiene tarjetas de red múltiples y diferentes nombres de host para cada tarjeta (como para un servidor Beowulf), utilizar el nombre de servidor de instalación que es conocida por los clientes instalar.

FAI utiliza `debootstrap` y `apt-get` para crear el sistema de archivos `nfsroot` en `/srv/fai/nfsroot`. Se necesita alrededor de 380MB de espacio libre en disco. Después de editar `fai.conf` y `make-fai-nfsroot.conf` llamada `fai-setup`

```
# fai-setup
# /etc/init.d/nfs-kernel-server recarga
```

Configuración de interfaces

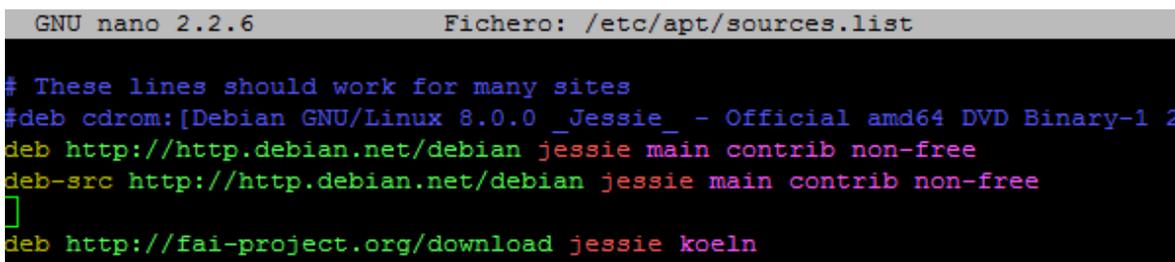
A) Interface eth1

Ip= 192.168.1.250

Mascara de red= 255.255.255.0

B) Verificamos el source.list del servidor fai y añadimos el repositorio oficial de fai para obtener los paquetes más recientes

```
deb http://fai-project.org/download jessie koeln
```

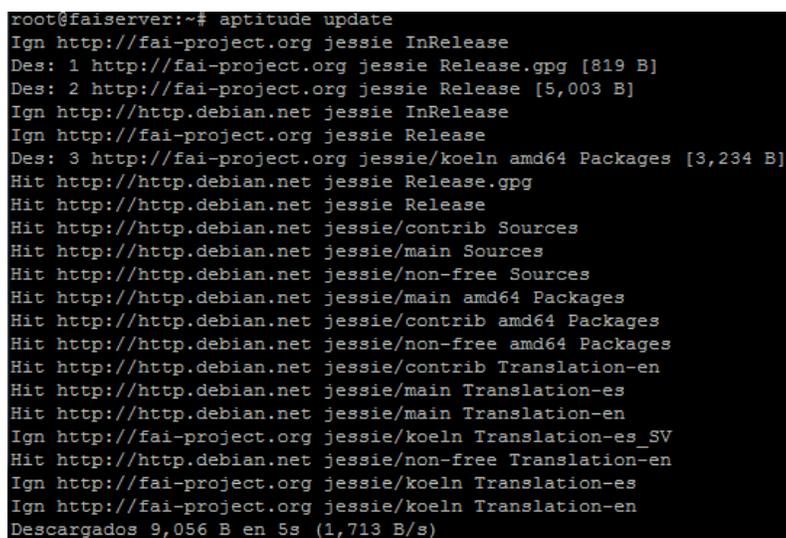


```
GNU nano 2.2.6          Fichero: /etc/apt/sources.list
# These lines should work for many sites
#deb cdrom:[Debian GNU/Linux 8.0.0 _Jessie_ - Official amd64 DVD Binary-1 2
deb http://http.debian.net/debian jessie main contrib non-free
deb-src http://http.debian.net/debian jessie main contrib non-free
]
deb http://fai-project.org/download jessie koeln
```

Ilustración 7: repositorio de proyecto fai

C) Actualizamos la lista de paquetes

```
# aptitude update
```



```
root@faiserver:~# aptitude update
Ign http://fai-project.org jessie InRelease
Des: 1 http://fai-project.org jessie Release.gpg [819 B]
Des: 2 http://fai-project.org jessie Release [5,003 B]
Ign http://http.debian.net jessie InRelease
Ign http://fai-project.org jessie Release
Des: 3 http://fai-project.org jessie/koeln amd64 Packages [3,234 B]
Hit http://http.debian.net jessie Release.gpg
Hit http://http.debian.net jessie Release
Hit http://http.debian.net jessie/contrib Sources
Hit http://http.debian.net jessie/main Sources
Hit http://http.debian.net jessie/non-free Sources
Hit http://http.debian.net jessie/main amd64 Packages
Hit http://http.debian.net jessie/contrib amd64 Packages
Hit http://http.debian.net jessie/non-free amd64 Packages
Hit http://http.debian.net jessie/contrib Translation-en
Hit http://http.debian.net jessie/main Translation-es
Hit http://http.debian.net jessie/main Translation-en
Ign http://fai-project.org jessie/koeln Translation-es_SV
Hit http://http.debian.net jessie/non-free Translation-en
Ign http://fai-project.org jessie/koeln Translation-es
Ign http://fai-project.org jessie/koeln Translation-en
Descargados 9,056 B en 5s (1,713 B/s)
```

Ilustración 8: actualizar lista de paquetes

- D) Instalamos los paquetes necesarios para nuestro servidor FAI (DHCP, TFTP, NFS).
Para instalar todos los paquetes necesarios ejecutamos el siguiente comando

```
# aptitude install fai-quickstart
```

Luego de la instalación se creara el directorio `/etc/fai`, donde se encuentran los archivos de configuración

- E) Configuramos el archivo `fai.conf`

```
GNU nano 2.2.6 Fichero: /etc/fai/fai.conf
# See fai.conf(5) for detailed information.

# Account for saving log files and calling fai-chboot.
LOGUSER=fai
LOGSERVER=faiserver
FAI_LOGPROTO=ssh
# URL to access the fai config space
FAI_CONFIG_SRC=nfs://faiserver/srv/fai/config
```

Ilustración 9: configuración FAI

- F) Configuramos el archivo `nfsroot.conf`

```
# For a detailed description see nfsroot.conf(5)

# "<suite> <mirror>" for debootstrap

FAI_DEBOOTSTRAP="jessie http://http.debian.net/debian"
FAI_ROOTPW='$1$kBnWcO.E$djxB128U7dMkrltJHPf6d1'

NFSROOT=/srv/fai/nfsroot
TFTPROOT=/srv/tftp/fai
NFSROOT_HOOKS=/etc/fai/nfsroot-hooks/
FAI_DEBOOTSTRAP_OPTS="--exclude=info --include=aptitude"

# Configuration space
NFSROOT_ETC_HOSTS="192.168.1.250 faiserver"
FAI_CONFIGDIR=/srv/fai/config
```

Ilustración 10: configuración nfsroot

- G) Agregamos la dirección IP de nuestro servidor y la de nuestros clientes al archivo /etc/hosts

```
127.0.0.1    localhost
#127.0.1.1   server

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

192.168.1.250  faiserver

192.168.1.10  cliente1
192.168.1.11  cliente2
192.168.1.12  cliente3
192.168.1.13  cliente4
192.168.1.14  cliente5
192.168.1.15  cliente6
192.168.1.16  cliente7
```

Ilustración 11: archivo hosts

- H) Configuramos nuestro servidor dhcp teniendo en cuenta que nuestro servidor permite NAT en la interface eth1 (192.168.1.250).

```
GNU nano 2.2.6      Fichero: /etc/dhcp/dhcpd.conf

deny unknown-clients;
option dhcp-max-message-size 2048;
use-host-decl-names on;

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.250;
    option domain-name "fai";
    option domain-name-servers 8.8.8.8;
    option time-servers faiserver;
    server-name faiserver;
    next-server faiserver;
    filename "pxelinux.0";
    allow unknown-clients;
    pool {
        range 192.168.1.10 192.168.1.110;
    }
}
```

Ilustración 12: configuración DHCP

- I) Agregamos el nombre de nuestro servidor (faiserver) al archivo /etc/hostname.

```
GNU nano 2.2.6 Fichero: /etc/hostname
faiserver
```

Ilustración 13: archivo hostname

- J) Luego de configurar los archivos anteriores ejecutamos el comando
fai-setup

```
root@faiserver:~# fai-setup
Account $LOGUSER=fai already exists.
Make sure that all install clients can
log into this account without a password.
/var/log/fai/.ssh/known_hosts remained unchanged.
/var/log/fai/.ssh/authorized_keys created.
User account fai set up.
-----
Error: /srv/fai/nfsroot exists already.
Neither force nor update option present, exiting.
-----
NOTE: Use -f option to force overwriting an existing nfsroot.
      Use -k option to update/install packages defined in NFSROOT config.
Log file written to /var/log/fai/fai-make-nfsroot.log
Log file written to /var/log/fai/fai-setup.log
```

Ilustración 14: creación de archivos FAI

La descarga dependerá del ancho de banda, la cual bajara todos los paquetes del sistema base que se utilizaran para la instalación de los clientes.

- K) Copiamos los archivos de ejemplo que trae la documentación de FAI al directorio de configuración.

```
# cp -a /usr/share/doc/fai-doc/examples/simple/* /srv/fai/config/
```

- L) Comentamos los repositorios por defecto y ponemos el repositorio local

```
GNU nano 2.2.6 Fichero: /srv/fai/nfsroot/etc/apt/sources.list Modificado
# These lines should work for many sites
#deb http://http.debian.net/debian jessie main contrib non-free
#deb http://security.debian.org/debian-security jessie/updates main contrib non$
# repository that may contain newer fai packages for jessie
#deb http://fai-project.org/download jessie koeln
# repositorio local
deb http://192.168.1.250/repo /
```

Ilustración 15: repositorios locales

M) Añadimos los clientes FAI especificando el directorio de configuración

Para la configuración default

```
# fai-chboot -IBv -u nfs://192.168.1.250/srv/fai/config default
```

```
root@faiserver:~# fai-chboot -IBv -u nfs://192.168.1.250/srv/fai/config default
Booting kernel vmlinuz-3.16.0-4-amd64
append initrd=initrd.img-3.16.0-4-amd64 ip=dhcp
    FAI_FLAGS=verbose,sshd,reboot FAI_CONFIG_SRC=nfs://192.168.1.250/srv/fai/conf
ig

default has no IP in hex default
Writing file /srv/tftp/fai/pxelinux.cfg/default for default
root@faiserver:~# █
```

Para la configuración de los clientes

```
# fai-chboot -IBv -u nfs://192.168.1.250/srv/fai/config cliente1
```

```
root@faiserver:~# fai-chboot -IBv -u nfs://192.168.1.250/srv/fai/config cliente1
Booting kernel vmlinuz-3.16.0-4-amd64
append initrd=initrd.img-3.16.0-4-amd64 ip=dhcp
    FAI_FLAGS=verbose,sshd,reboot FAI_CONFIG_SRC=nfs://192.168.1.250/srv/fai/conf
ig

cliente1 has 192.168.1.10 in hex COA8010A
Writing file /srv/tftp/fai/pxelinux.cfg/COA8010A for cliente1
root@faiserver:~# █
```

N) Reiniciar los servicios (DHCP, TFTP y NFS).

```
/etc/init.d/nfs-kernel-server restart
```

```
/etc/init.d/isc-dhcp-server restart
```

```
/etc/init.d/tftpd-hpa restart
```

Escenarios de prueba

Iniciar la pc, con la configuración para iniciar desde la red la primer ves (presionar f6, f8, f9 o f11 dependerá la motherboard de nuestras PC).

```
iPXE (PCI 00:0D:4) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+ -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS TFTP HTTP PXE PXEXT Menu

net0: 08:00:27:6e:ab:68 using 82540em on PCI00:03.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
DHCP (net0 08:00:27:6e:ab:68)..... ok
net0: 192.168.1.14/255.255.255.0 gw 192.168.1.250
Next server: 192.168.1.250
Filename: pxelinux.0
tftp://192.168.1.250/pxelinux.0... ok

PXELINUX 6.03 PXE 20150819 Copyright (C) 1994-2014 H. Peter Anvin et al
Loading vmlinuz-3.16.0-4-amd64... ok
Loading initrd.img-3.16.0-4-amd64..._
```

Ilustración 16: arranque PXE

Al iniciar desde red no es necesario hacer nada más ya que el servidor FAI se encargara de todo.

A continuación se presentan capturas de la instalación.

```
[ 1.892093] usb 1-1: new full-speed USB device number 2 using ohci-pci
[ 2.140121] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021
[ 2.141086] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 2.141991] usb 1-1: Product: USB Tablet
[ 2.142737] usb 1-1: Manufacturer: VirtualBox
[ 2.150312] hidraw: raw HID events driver (C) Jiri Kosina
[ 2.155836] usbcore: registered new interface driver usbhid
[ 2.156537] usbhid: USB HID core driver
[ 2.160497] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/
usb1/1-1/1-1:1.0/0003:80EE:0021.0001/input/input2
[ 2.162145] hid-generic 0003:80EE:0021.0001: input,hidraw0: USB HID v1.10 Mou
se [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
Loading /etc/console-setup/cached_ISO-8859-15_del.kmap
[ 2.484030] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 2.486540] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: R
X
[ 2.487717] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
Starting dhcp for interface eth0
dhcp: PREINIT eth0 up
dhcp: BOND setting eth0
[ 2.692185] Switched to clocksource tsc
```

```

. ok
[ ok ] Synthesizing the initial hotplug events...done.
[ ok ] Waiting for /dev to be fully populated...done.
[ 6.804013] ACPI: AC Adapter [AC] (on-line)
[ 6.849329] sd 0:0:0:0: Attached scsi generic sg0 type 0
[ 6.849376] sr 2:0:0:0: Attached scsi generic sg1 type 5
[ 6.862949] _nachoctl_pc.00:02: reported hu Plus and Plus ACPI
[ 6.921122] input: PC Speaker as /devices/platform/pcspkr/input/input4
[ 6.922898] input: Power Button as /devices/LNXSYSTM:00/LNXPWRBN:00/input/input5
[ 6.923848] ACPI: Power Button [PWRF]
[ 6.924271] input: Sleep Button as /devices/LNXSYSTM:00/LNXSLPBN:00/input/input6
[ 6.925048] ACPI: Sleep Button [SLPF]
[ 6.926329] ACPI: Video Device [GFX0] (multi-head: yes rom: no post: no)
[ 6.927491] input: Video Bus as /devices/LNXSYSTM:00/LNXXSYBUS:00/PNPOA03:00/LNXXVIDEO:00/input/input7
[ 6.962052] intel_rapl: no valid rapl domains found in package 0
[ 6.994776] ppdev: user-space parallel port driver
[ 7.117465] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3

```

```

[ 8.076146] snd_intel8x0 0000:00:05.0: intel8x0_measure_ac97_clock: measured 55398 usecs (12625 samples)
[ 8.076599] snd_intel8x0 0000:00:05.0: measured clock 227896 rejected
[ 8.076945] snd_intel8x0 0000:00:05.0: clocking to 48000
-----
Fully Automatic Installation - FAI
4.3.1+deb8u1 (c) 1999-2015
Thomas Lange <lange@informatik.uni-koeln.de>
-----
Calling task_confdir
Kernel currently running: Linux 3.16.0-4-amd64 x86_64 GNU/Linux
Kernel parameters: BOOT_IMAGE=vmlinuz-3.16.0-4-amd64 initrd=initrd.img-3.16.0-4-amd64 ip=dhcp root=/srv/fai/nfsroot aufs FAI_FLAGS=verbose,sshd,reboot FAI_CONFIG_SRC=nfs://192.168.1.250/srv/fai/config FAI_ACTION=install
Reading /tmp/fai/boot.log

```

```

[ 8.076146] snd_intel8x0 0000:00:05.0: intel8x0_measure_ac97_clock: measured 55398 usecs (12625 samples)
[ 8.076599] snd_intel8x0 0000:00:05.0: measured clock 227896 rejected
[ 8.076945] snd_intel8x0 0000:00:05.0: clocking to 48000
-----
Fully Automatic Installation - FAI
Executing: tune2fs -c 0 -i 0 /dev/sda6
Executing: mkfs.ext4 -L home -m 1 /dev/sda7
Executing: tune2fs -c 0 -i 0 /dev/sda7
/dev/sda6 UUID=edf1383a-5202-48f1-b265-899c5af122ea
/dev/sda6 LABEL=tmp
/dev/sda7 UUID=cb8790a1-ca49-431a-8711-9462ca6064fc
/dev/sda7 LABEL=home
/dev/sda1 UUID=717042b4-5307-4e45-9eb0-2765e7f77b91
/dev/sda5 UUID=0d41383d-e1d4-41b1-b1ea-163eb7d9a468
Calling task_mountdisks
Enable swap device /dev/sda5
Mounting UUID=717042b4-5307-4e45-9eb0-2765e7f77b91 to /target/
Mounting UUID=cb8790a1-ca49-431a-8711-9462ca6064fc to /target/home
Mounting UUID=edf1383a-5202-48f1-b265-899c5af122ea to /target/tmp
Calling task_extrbase
Unpacking Debian base archive
ftar: No matching class found in /var/lib/fai/config/basefiles//
ftar: extracting /var/tmp/base.tar.xz to /target/

```

```

8.076146] snd_intel8x0 0000:00:05.0: intel8x0_measure_ac97_clock: measured
5398 usecs (12625 samples)
8.076599] snd_intel8x0 0000:00:05.0: measured clock 227896 rejected
8.076945] snd_intel8x0 0000:00:05.0: clocking to 48000
-----
Fully Automatic Installation - FAI
et: 21 http://192.168.1.250/repo/ console-setup-linux 1.123 [984 kB]
et: 22 http://192.168.1.250/repo/ xkb-data 2.12-1 [597 kB]
et: 23 http://192.168.1.250/repo/ console-setup 1.123 [128 kB]
et: 24 http://192.168.1.250/repo/ libdevmapper-event1.02.1 2:1.02.90-2.2 [35.8
kB]
et: 25 http://192.168.1.250/repo/ libpng12-0 1.2.50-2+b2 [172 kB]
et: 26 http://192.168.1.250/repo/ libfreetype6 2.5.2-3 [466 kB]
et: 27 http://192.168.1.250/repo/ libfuse2 2.9.3-15+b1 [135 kB]
et: 28 http://192.168.1.250/repo/ dmeventd 2:1.02.90-2.2 [50.0 kB]
et: 29 http://192.168.1.250/repo/ liblvm2cmd2.02 2.02.111-2.2 [522 kB]
et: 30 http://192.168.1.250/repo/ libusb-1.0-0 2:1.0.19-1 [48.1 kB]
et: 31 http://192.168.1.250/repo/ libuuid-perl 0.05-1+b1 [10.4 kB]
et: 32 http://192.168.1.250/repo/ linux-base 3.5 [34.3 kB]
et: 33 http://192.168.1.250/repo/ libklibc 2.0.4-2 [51.7 kB]
et: 34 http://192.168.1.250/repo/ klibc-utils 2.0.4-2 [108 kB]
et: 35 http://192.168.1.250/repo/ initramfs-tools 0.120 [95.7 kB]
et: 36 http://192.168.1.250/repo/ linux-image-3.16.0-4-amd64 3.16.7-ckt9-2 [33
8 MB]

```

```

[ 8.076146] snd_intel8x0 0000:00:05.0: intel8x0_measure_ac97_clock: measured
55398 usecs (12625 samples)
[ 8.076599] snd_intel8x0 0000:00:05.0: measured clock 227896 rejected
[ 8.076945] snd_intel8x0 0000:00:05.0: clocking to 48000
-----
Fully Automatic Installation - FAI
192.168.1.250:/srv/fai/nfsroot nfs4      92G   17G   71G   20% /live/image
192.168.1.250:/srv/fai/config nfs4     92G   17G   71G   20% /var/lib/fai/confi
g
/dev/sda1          ext4      5.8G  526M  4.9G   10% /target
/dev/sda7          ext4      1.3G  2.0M  1.2G    1% /target/home
/dev/sda6          ext4      359M  2.1M  353M    1% /target/tmp
eth0 rx_bytes 165.82 Mbytes
eth0 tx_bytes 1.51 Mbytes
Ramdisk on /target/var/lib/dpkg/umounted
Fri Nov 13 08:50:11 CST 2015
The install took 109 seconds.
Calling task_chboot
Source hook: savelog.LAST.source
ERRORS found in log files. See /tmp/fai/error.log
savelog.LAST.source OK.
Calling task_savelog
$LOGUSER is undefined. Not saving log files to remote.
Calling task_faiend

```

```

GNU GRUB  version 2.02~beta2-22

*Debian GNU/Linux
Advanced options for Debian GNU/Linux

```

```
Debian GNU/Linux 8 none tty1
none login: _
```

El password de usuario root es "fai"

Accedemos como usuario root y creamos nuestro usuario, además instalamos los paquetes que necesite el usuario, podemos instalar "xfce4" que trae los paquetes necesarios para trabajar en nuestra PC.

```
root@none:/home/cliente1# aptitude install xfce4_
```

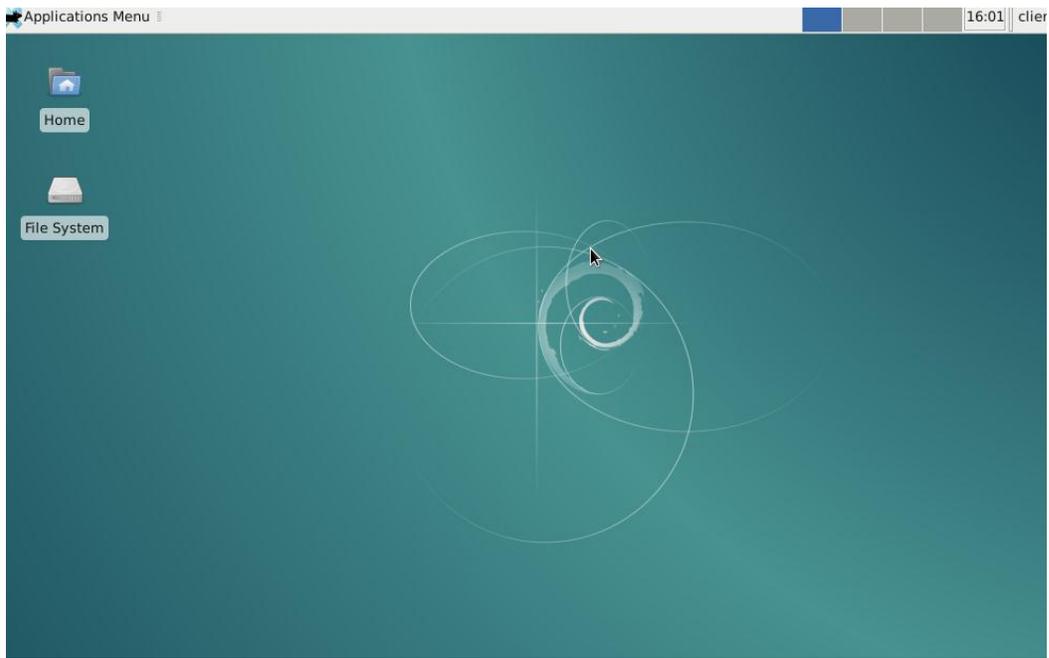
```
1,030 kB]
Get: 148 http://192.168.1.250/repo/ libgstreamer1.0-0 1.4.4-2 [1,659 kB]
Get: 149 http://192.168.1.250/repo/ libgstreamer-plugins-base1.0-0 1.4.4-2 [1,297 kB]
Get: 150 http://192.168.1.250/repo/ libgudev-1.0-0 215-17 [39.8 kB]
Get: 151 http://192.168.1.250/repo/ libplist2 1.11-3 [26.4 kB]
Get: 152 http://192.168.1.250/repo/ libusbmuxd2 1.0.9-1 [14.6 kB]
Get: 153 http://192.168.1.250/repo/ libimobiledevice4 1.1.6+dfsg-3.1 [56.5 kB]
Get: 154 http://192.168.1.250/repo/ libmpfr4 3.1.2-2 [527 kB]
Get: 155 http://192.168.1.250/repo/ libmtdev1 1.1.5-1 [22.2 kB]
Get: 156 http://192.168.1.250/repo/ libnotify4 0.7.6-2 [28.3 kB]
Get: 157 http://192.168.1.250/repo/ libxft2 2.3.2-1 [55.4 kB]
Get: 158 http://192.168.1.250/repo/ libpangoxft-1.0-0 1.36.8-3 [195 kB]
Get: 159 http://192.168.1.250/repo/ libparted2 3.2-7 [269 kB]
Get: 160 http://192.168.1.250/repo/ libpolkit-gobject-1-0 0.105-8 [40.8 kB]
Get: 161 http://192.168.1.250/repo/ libopenjpeg5 1:1.5.2-3 [111 kB]
Get: 162 http://192.168.1.250/repo/ libpoppler46 0.26.5-2 [1,210 kB]
Get: 163 http://192.168.1.250/repo/ libpoppler-glib8 0.26.5-2 [121 kB]
Get: 164 http://192.168.1.250/repo/ libtheora0 1.1.1+dfsg.1-6 [171 kB]
Get: 165 http://192.168.1.250/repo/ thunar-data 1.6.3-2 [1,005 kB]
Get: 166 http://192.168.1.250/repo/ libthunarx-2-0 1.6.3-2 [62.8 kB]
Get: 167 http://192.168.1.250/repo/ libtumbler-1-0 0.1.30-1+b1 [70.1 kB]
Get: 168 http://192.168.1.250/repo/ libupower-glib3 0.99.1-3.2 [47.2 kB]
Get: 169 http://192.168.1.250/repo/ libvisual-0.4-0 0.4.0-6 [121 kB]
60% [Working] 13.0 MB/s 4s
```

```
38) udisks2 [Not Installed]
39) update-inetd [Not Installed]
40) xml-core [Not Installed]
41) xscreensaver [Not Installed]
42) xscreensaver-data [Not Installed]

Leave the following dependencies unresolved:
43) libexo-1-0 recommends liburi-perl
44) thunar recommends gvfs
45) libhtml-tree-perl recommends libwww-perl
46) libio-socket-ssl-perl recommends perl (>= 5.15.6) | libsocket-perl (>= 1
47) libio-socket-ssl-perl recommends perl (>= 5.19.8) | libio-socket-ip-perl
48) libxml2 recommends xml-core
49) libcolord2 recommends colord
50) xscreensaver recommends perl5
51) xfce4-session recommends libpam-systemd
52) xfce4-session recommends xscreensaver
53) upower recommends policykit-1
54) gvfs-common recommends gvfs
55) libsane recommends sane-utils (>= 1.0.24-8)
56) libnet-http-perl recommends libio-socket-ssl-perl (>= 1.38)
57) libnet-smtp-ssl-perl recommends libauthen-sasl-perl

Accept this solution? [Y/n/q/?]
```

A finalizar digitamos el comando: startx



Conclusión

La conclusión de este documento, es que el proyecto de auto instalación de Debian por medio de una red LAN es una herramienta que mejorara considerablemente el manteniendo de un centro de cómputo, mejorando y estableciendo una misma configuración en todas las computadoras de una misma red, además de optimizar el tiempo que se utiliza para el mantenimiento del software del centro de computo

Recomendaciones

- 1- Se recomienda crear un mirror local con los paquetes necesarios para que no se tengan que descargar de internet para cada cliente.
- 2- Usar la configuración de ejemplo que trae la documentación de FAI.
- 3- Tener dos interfaces de red; una para que nuestro servidor esté conectado a internet y la otra para la red local de instalación.
- 4- Configurar el servidor para permitir NAT en la red local.

Bibliografía

- «D.3. Instalar Debian GNU/Linux desde un sistema Unix/Linux.» [En línea]. Disponible en: <https://www.debian.org/releases/stable/i386/apds03.html.es>. [Accedido: 21-ago-2015]. [1]
- «FAI - Debian Wiki.» [En línea]. Disponible en: <https://wiki.debian.org/FAI>. [Accedido: 21-ago-2015]. [2]
- «FAI - Fully Automatic Installation.» [En línea]. Disponible en: <http://fai-project.org/>. [Accedido: 21-ago-2015]. [3]
- «Instalar Debian desde red con PXE (nunca fue tan fácil) | CRySoL.» [En línea]. Disponible en: <http://crysol.org/es/node/1080>. [Accedido: 21-ago-2015]. [4]
- chapu79, «Instalar Linux Ubuntu desde la red, ideal para tu netbook», 25-sep-2011. [En línea]. Disponible en: <http://www.taringa.net/posts/linux/12620912/Instalar-Linux-Ubuntu-desde-la-red-ideal-para-tu-netbook.html>. [Accedido: 21-ago-2015]. [5]
- «Como crear un repositorio local de Debian desde los DVD's», *ADMINISTRACION DE REDES*. . [6]

Anexo

FAI (Fully Automatic Installation)

Un sistema de instalación con guión para administrar máquinas basadas Debian. Las máquinas están asignadas a "clases" que definen un conjunto de paquetes a instalar. todas las máquinas realizan un arranque desde red para recuperar comandos del FAI-INSTALADOR EL SERVIDOR. Si la maquina es debido para la re-instalación, se montará es kernel y la raíz del maestro y volver a instalar en sí sobre la base de la configuración especificada para cada una de las clases que se le ha asignado

