

**INTRODUCCIÓN A OPENFLOW Y PRÁCTICAS DE LABORATORIO
MEDIANTE LA TARJETA NETFPGA**

CHRISTIAN DUVÁN MOTTA BARRAGÁN

EDISSON BADILLO GUALDRÓN

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA ELECTRÓNICA
ESCUELA DE INGENIERÍAS
SECCIONAL BUCARAMANGA
2013**

**INTRODUCCIÓN A OPENFLOW Y PRÁCTICAS DE LABORATORIO
MEDIANTE LA TARJETA NETFPGA**

CHRISTIAN DUVÁN MOTTA BARRAGÁN

EDISSON BADILLO GUALDRÓN

PROYECTO DE GRADO

**JHON JAIRO PADILLA AGUILAR, PhD
DIRECTOR DEL PROYECTO**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA ELECTRÓNICA
ESCUELA DE INGENIERÍAS
SECCIONAL BUCARAMANGA
2013**

NOTA DE ACEPTACIÓN

FIRMA DEL PRESIDENTE DEL JURADO

FIRMA DEL JURADO

FIRMA DEL JURADO

BUCARAMANGA, FEBRERO DE 2013

DEDICATORIA

**Este proyecto va dedicado a mi familia,
en especial a mis padres y hermanos
por tanto sacrificio y esfuerzo.**

**A Andrea Prada, por ser mi apoyo
incondicional en esta etapa
y esperarme para graduarnos juntos.**

Muchas gracias a todos y por todo.

CHRISTIAN MOTTA

**Dedico este proyecto a mis padres y
mi hermano que me han brindado
todo su tiempo, esfuerzo y apoyo
durante estos años.**

**Especialmente a mi hermana
que ha sido otra madre para mí, sin ti
no hubiera sido posible este proyecto.**

**Para mi bella esposa que me ha
apoyado siempre y que me impulsó en
todo momento a luchar por esta meta.
Los amo a todos.**

EDISSON BADILLO G.

AGRADECIMIENTOS

Primero que todo dar gracias a Dios por darnos sabiduría y salud para lograr culminar este capítulo de nuestras vidas.

A nuestras familias, por darnos el apoyo tanto moral como económico, ya que sin ellos, nada de esto hubiera sido posible. Por el esfuerzo que realizaron día a día junto con nosotros.

A la Universidad Pontificia Bolivariana Seccional Bucaramanga por permitirnos ingresar a tan prestigiosa Universidad. Por llenarnos de conocimiento y permitirnos vivir la mejor época de nuestras vidas.

Al ingeniero Jhon Jairo Padilla y al estudiante Oscar Maestre, por brindarnos su conocimiento y todo el tiempo solicitado para nuestras consultas.

Y en general, a todos los profesores que hicieron parte de nuestra formación tanto académica como humana.

LISTA DE TABLAS

	pág
TABLA 1.OPCIONES DEL COMANDO MKDIR	63
TABLA 2.OPCIONES DEL COMANDO RMDIR	65
TABLA 3.OPCIONES DEL COMANDO LS.	66
TABLA 4. OPCIONES DEL COMANDO CP.	67
TABLA 5. OPCIONES DEL COMANDO MV.	68
TABLA 6. OPCIONES DEL COMANDO RM.	69
TABLA 7. OPCIONES DEL COMANDO CAT.	70
TABLA 8. OPCIONES DEL COMANDO TAIL.	70
TABLA 9. OPCIONES DEL COMANDO STAT.	71
TABLA 10. OPCIONES DEL COMANDO WC.	72
TABLA 11. OPCIONES DEL COMANDO DU.	74
TABLA 12. OPCIONES DEL COMANDO FIND.	75

LISTA DE FIGURAS

	pág
FIGURA 1. NETFPGA TIPO RACK.	5
FIGURA 2. NETFPGA TIPO CUBO	6
FIGURA 3. PLANOS DE UNA COMUNICACIÓN DE DATOS EN UNA RED.	8
FIGURA 4. TARJETA NETFPGA	13
FIGURA 5. CREACIÓN DE MÁQUINA VIRTUAL.	17
FIGURA 6. INGRESAR EL NOMBRE, SISTEMA OPERATIVO Y VERSIÓN DE LA MÁQUINA VIRTUAL.	18
FIGURA 7. CANTIDAD DE MEMORIA RAM PARA LA MÁQUINA VIRTUAL.	19
FIGURA 8. SELECCIONAR DISCO DURO VIRTUAL EXISTENTE O NUEVO.	20
FIGURA 9. SELECCIONAR EL TIPO DE ARCHIVO A USAR PARA EL NUEVO DISCO VIRTUAL.	21
FIGURA 10. ALMACENAMIENTO DE DISCO VIRTUAL.	22
FIGURA 11. NOMBRE Y TAMAÑO DEL DISCO VIRTUAL.	23
FIGURA 12. RESUMEN DEL NUEVO DISCO VIRTUAL A CREAR.	24
FIGURA 13. MÁQUINA VIRTUAL CREADA.	25
FIGURA 14. ASISTENTE DE AYUDA PARA LA INSTALACIÓN DEL SISTEMA OPERATIVO.	26
FIGURA 15. MEDIO DE INSTALACIÓN DEL CENTOS.	27
FIGURA 16. RESUMEN DEL ARCHIVO IMÁGEN.	28

FIGURA 17. DIFERENTES TIPOS DE INSTALACIÓN DE CENTOS.	29
FIGURA 18. TEST DE COMPROBACIÓN DE DVD O USB.	30
FIGURA 19. PANTALLA DE BIENVENIDA DE CENTOS 5.	31
FIGURA 20. ELECCIÓN DEL IDIOMA DE INSTALACIÓN.	32
FIGURA 21. ELECCIÓN DEL TECLADO.	33
FIGURA 22. AVISO DE PARTICIÓN.	34
FIGURA 23. PARTICIÓN DEL DISCO DURO.	35
FIGURA 24. CONFIGURACIÓN DE LOS DISPOSITIVOS DE RED.	36
FIGURA 25. CONFIGURACIÓN ZONA HORARIA.	37
FIGURA 26. CONTRASEÑA DEL USUARIO ROOT O SUPER USUARIO.	38
FIGURA 27. APLICACIONES PARA USO GENERAL DE INTERNET.	39
FIGURA 28. VENTANA DE INICIO DE LA INSTALACIÓN DEL CENTOS 5.	40
FIGURA 29. VENTANA DE INICIO DE LA INSTALACIÓN DEL CENTOS 5.	41
FIGURA 30. TRANSFERENCIA DE DATOS DESDE LA IMAGEN DEL PROGRAMA DE INSTALACIÓN HACIA EL DISCO DURO.	42
FIGURA 31. INSTALACIÓN DEL SOFTWARE.	43
FIGURA 32. INSTALACIÓN COMPLETA DE CENTOS 5.	44
FIGURA 33. MENSAJE DE BIENVENIDA DE CENTOS 5.	45
FIGURA 34. CONFIGURACIÓN DE CORTAFUEGOS.	46
FIGURA 35. CONFIGURACIÓN SELINUX.	47
FIGURA 36. CONFIGURACIÓN FECHA Y HORA.	48
FIGURA 37. CREACIÓN DE USUARIO.	49
FIGURA 38. CONFIGURACIÓN DE LA TARJETA DE SONIDO.	50

FIGURA 39. CONFIGURACIÓN DE SOFTWARE ADICIONAL.	51
FIGURA 40. NOMBRE DE USUARIO.	52
FIGURA 41. CONTRASEÑA A USUARIO ROOT.	53
FIGURA 42. ENTORNO GRÁFICO CENTOS 5.6	54
FIGURA 43. EJEMPLO DE ÁRBOL JERÁRQUICO DE LINUX.	55
FIGURA 44. RUTA POR MEDIO GRÁFICO HACIA EL DIRECTORIO HOME 1.	56
FIGURA 45. RUTA POR MEDIO GRÁFICO HACIA EL DIRECTORIO HOME 2.	57
FIGURA 46. RUTA POR MEDIO GRÁFICO HACIA EL DIRECTORIO HOME 3.	58
FIGURA 47. FUNCIÓN TECLA TAB.	59
FIGURA 48. EJEMPLO DE ESCRITURA PARA DIRECTORIOS CON NOMBRES LARGOS.	60
FIGURA 49. EJEMPLO COMANDO PWD.	61
FIGURA 50. CREAR MÚLTIPLES DIRECTORIOS CON MKDIR.	62
FIGURA 51. CREAR DIRECTORIOS HIJOS DESDE EL DIRECTORIO PADRE CON MKDIR.	63
FIGURA 52. EJEMPLO COMANDO RMDIR.	64
FIGURA 53. EJEMPLO COMANDO LS.	65
FIGURA 54. COMANDO LS.	66
FIGURA 55. EJEMPLO DEL COMANDO RM.	69
FIGURA 56. PERMISOS DE ACCESO.	72
FIGURA 57. PRIVILEGIOS SUPER USUARIO.	76
FIGURA 58. FICHERO ENV_VARS.	79
FIGURA 59. FICHERO DE RED.	81

CONTENIDO

	pág
INTRODUCCIÓN	3
1. OBJETIVOS DEL PROYECTO	4
1.1 OBJETIVO GENERAL	4
1.2 OBJETIVOS ESPECÍFICOS	4
2. METODOLOGÍA	5
2.1 EQUIPOS Y SOFTWARE DE LABORATORIO	5
2.2 ACTIVIDADES REALIZADAS	6
3. MARCO TEÓRICO	8
3.1 OPENFLOW	8
3.1.1 Redes Definidas por Software o SDN´s	8
3.1.2 Proyecto OpenFlow	10
3.2 NETFPGA	12
3.3 GNU/LINUX CENTOS	14
3.3.1 Definición software libre o GNU	14
3.3.2 Definición e historia GNU/LINUX	14
3.3.3 Definición Centos 5	15
4. RESULTADOS DEL PROYECTO	16
4.1 TUTORIAL DE INSTALACIÓN CENTOS 5 EN MÁQUINA VIRTUAL <i>VIRTUALBOX</i>	16
4.2 COMANDOS BÁSICOS DE LINUX	54
4.2.1 Manipular ficheros y directorios	54
4.2.1.1 Comando cd	54
4.2.1.2 Comando pwd	60
4.2.1.3 Comando mkdir	61
4.2.1.4 Comando rmdir	64
4.2.1.5 Comando ls	65
4.2.1.6 Comando cp	67
4.2.1.7 Comando mv	68
4.2.1.8 Comando rm	68

4.2.2	Visor de ficheros	69
4.2.2.1	Comando cat	69
4.2.2.2	Comando tail	70
4.2.3	Propiedades de los ficheros	71
4.2.3.1	Comando stat	71
4.2.3.2	Comando wc	71
4.2.3.3	Permisos de acceso	72
4.2.3.4	Comando chmod	73
4.2.3.5	Comando du	74
4.2.4	Localización de ficheros	74
4.2.4.1	Comando find	74
4.2.4.2	Comando locate	75
4.3	INSTALACIÓN OPENFLOW	75
4.4	COMANDOS DEL OPENFLOW	82
4.4.1	Comando dpctl	82
4.5	PRÁCTICAS DE LABORATORIO	93
4.5.1	Tarjeta NetFPGA como switch Openflow	93
4.5.2	Creación de VLAN mediante la tarjeta NetFPGA como switch Openflow 103	
4.5.3	Compartir a internet por medio del switch Openflow	118
4.5.4	Servidor DHCP dinámico con dos puertos	133
4.5.5	Laboratorio firewall por medio del switch Openflow	150
5.	GLOSARIO	160
6.	CONCLUSIONES	162
7.	REFERENCIAS BIBLIOGRÁFICAS	163

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: INTRODUCCION A OPENFLOW Y PRACTICAS DE LABORATORIO MEDIANTE LA TARJETA NETFPGA

AUTOR(ES): CHRISITAN DUVAN MOTTA BARRAGAN
EDISSON BADILLO GUALDRON

FACULTAD: FACULTAD DE INGENIERÍA ELECTRÓNICA

DIRECTOR(A): JHON JAIRO PADILLA AGUILAR

RESUMEN

Las redes actuales se construyen con switches, routers y otros dispositivos que se han hecho cada vez más complejos, ya que manejan más tráfico de datos y nuevos servicios más exigentes. Otra característica de los routers actuales es que, además de interfaces propietarias para su administración, estos son cerrados a la creación de nuevos algoritmos de procesamiento de los paquetes de información. Para poder hacer una variación en la red, se necesita que los fabricantes de los dispositivos de red expongan el funcionamiento interno de los mismos. Ante este escenario, es casi imposible que los operadores de red y los investigadores puedan innovar. Las líneas de código de los routers actuales son tan extensas que inclusive un programador puede tardar días para hacer cambios en la red. Para dar solución a esta problemática, se estudió una nueva manera de crear redes actualmente utilizadas por las grandes empresas y universidades llamada Redes Definidas por Software (SDN- Software Defined Networking). El concepto de SDN busca tener equipos comerciales con aplicaciones abiertas utilizando como base la herramienta Openflow, la cual permite crear o modificar redes de manera rápida, fácil, segura y abierta. Esta herramienta se desarrolló en una NetFPGA, la cual se configuró como un switch Openflow. Esta tarjeta es especializada para la creación de las redes y viene configurada de fábrica para trabajar con Linux CentOS. En este proyecto se desarrollaron manuales de comandos, instalación y compilación de CentOS y OpenFlow. Se realizaron 5 prácticas de laboratorio cuyo objetivo es introducir a los estudiantes en el nuevo mundo de las Redes Definidas por Software.

PALABRAS CLAVES: OPENFLOW, REDES DEFINIDAS POR SOFTWARE, LINUX CENTOS, NETFPGA, REDES, SWITCH.

V° B° DIRECTOR DE TRABAJO DE GRADO

ABSTRACT

TITTLE: INTRODUCTION TO OPENFLOW AND LAB GUIDES USING NETFPGA CARD.

AUTHOR(S): CHRISTIAN DUVAN MOTTA BARRAGAN
EDISSON BADILLO GUALDRON

FACULTY: FACULTY OF ELECTRONIC ENGINEERING

DIRECTOR: JHON JAIRO PADILLA AGUILAR

Today's data networks are built with switches, routers and other devices that have become increasingly complex due to they handle more data traffic and new services. Current Routers have proprietary interfaces for its administration, besides they have a closed architecture. Thus, to modify networks, it requires that network devices manufacturers expose internal functions of such devices. Given this scenario, it's almost impossible for network operators and researchers to innovate. The code lines of current routers are so extensive and even a programmer can take days to make changes to the network. To solve this problem, we studied a way to create new networks currently deployed by large enterprises and universities called Software Defined Networking, which principal objective is to have commercial devices with open applications using OpenFlow tool. Openflow allows creating or modifying networks, by means of a fast, easy, safe and open mode. This tool was implemented on a NetFPGA, which was configured as an OpenFlow switch. This device is dedicated to the creation of networks and it is configured to work with Linux CentOS. In this project, we made tutorials of commands, installation and compiling the CentOS and OpenFlow software. There were written 5 lab guides that introduce students to the new world of Software-Defined Networking.

KEYWORDS: OPENFLOW, SOFTWARE DEFINED NETWORKING, LINUX
CENTOS, NETFPGA, NETWORKS, SWITCH.

V° B° DIRECTOR OF GRADUATE WORK

INTRODUCCIÓN

Openflow es un concepto que fué creado con el fin de permitir el diseño de nuevas formas de redes, teniendo como característica principal que el programador crea sus redes de acuerdo a su gusto e intereses, y no se ve reducido a los dispositivos actuales de redes que tienen sus sistemas cerrados, tales como los *switches* y *routers*, que operan sin que los usuarios puedan modificarlos para dar prioridades en la transmisión de datos según la conveniencia del cliente.

Openflow permite a las redes dar respuesta a las necesidades tanto investigativas como empresariales de forma eficaz, ya que es un nuevo concepto de enrutamiento que opera como un estándar abierto, en el que los usuarios son los encargados de experimentar con el protocolo y crear e implementar nuevas características y funciones que éste tiene para dar. Por lo tanto, para las universidades sería de gran importancia que sus estudiantes investiguen y pongan en práctica una manera más fácil de crear y administrar las redes.

Para la universidad es de gran importancia la introducción de esta nueva tecnología en los laboratorios, pues se introduce el concepto de Redes Definidas por software, el cual es una tendencia muy importante en el área de Redes de telecomunicaciones actualmente. El grupo de investigación en telecomunicaciones y las Facultades de Ingeniería Electrónica e informática se verán beneficiados, pues esta herramienta permitirá la capacitación de los estudiantes tanto de pregrado como de maestría en esta nueva tendencia de las telecomunicaciones.

1. OBJETIVOS DEL PROYECTO

1.1 OBJETIVO GENERAL

Introducir a los estudiantes en el nuevo mundo de las Redes Definidas por Software utilizando el proyecto Openflow de la Universidad de Stanford. Se realizarán tutoriales y guías de laboratorio para tal fin.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar manuales en los cuales los estudiantes de las facultades de ingeniería electrónica e informática de la universidad Pontificia Bolivariana se guíen para una mejor comprensión en cuanto a procesos de instalación y compilación de los software utilizados, ya que tienden a ser complejos o no conocidos.
- Elaborar guías de laboratorio donde los estudiantes de las facultades de ingeniería electrónica e informática de la universidad Pontificia Bolivariana tengan una introducción acerca de algunas configuraciones básicas de Openflow.
- Introducir en las Facultades de Ingeniería Electrónica e informática el concepto de Redes Definidas por Software.

2. METODOLOGÍA

Durante el desarrollo del proyecto se realizaron varias actividades, las cuales se pueden verificar más detalladamente en el capítulo llamado resultados del proyecto. Se usaron los siguientes equipos de laboratorio.

2.1 EQUIPOS Y SOFTWARE DE LABORATORIO

Los equipos que se utilizaron para que estas prácticas de laboratorio fueran realizadas son: una NetFPGA tipo *rack* que funcionó como *switch* Openflow, un monitor, un mouse y un teclado que se puede apreciar en la figura 1. Dos NetFPGA tipo cubo que funcionaron como clientes (figura 2), para analizar los distintos resultados que arrojaron las prácticas de laboratorio.

FIGURA 1. NETFPGA TIPO RACK.



FIGURA 2. NETFPGA TIPO CUBO



Tanto la NetFPGA tipo rack y las tipo cubo, cuentan con el sistema operativo Linux CentOS 5.6. Se utilizó este sistema operativo debido a que estos equipos vinieron con el paquete ya instalado de fábrica.

2.2 ACTIVIDADES REALIZADAS

Como primera medida, se tuvo que aprender a manejar el sistema operativo Linux CentOS, puesto que, sin él no se hubieran podido instalar los programas y librerías necesarias para que el protocolo Openflow funcionara.

Se trabajó con la versión 5.6 debido a que, al instalar y compilar con distribuciones anteriores o superiores, presentaba algún tipo de problema de compatibilidad que no permitía el desarrollo del proyecto.

Todas estas instalaciones se hicieron por código en el terminal del sistema operativo CentOS, puesto que, este hardware venía de fábrica con esta distribución de Linux.

Posterior a eso, se instaló en la NetFPGA la herramienta Openflow, que contenía el controlador utilizado llamado *controller*.

Como última medida, después de haber tenido instalado y compilado todos los programas, se desarrollaron 5 prácticas de laboratorio, las cuales sirvieran de base para la investigación de los estudiantes de la facultad de ingeniería electrónica e informática de la Universidad Pontificia Bolivariana.

Se involucra tanto a los estudiantes de ingeniería electrónica como de informática; debido a que recibimos asesoría por parte del estudiante de ingeniería informática Oscar Maestre, quien nos brindó su tiempo y conocimiento.

Se involucra también al docente de ingeniería informática Lennin Serrano, quien fue uno de nuestros calificadores del proyecto.

Se involucra a la ingeniera Line Yasmin Becerra, quien es actualmente docente de la Universidad Católica de Pereira y estudiante de doctorado de la Universidad Pontificia Bolivariana con su sede en Medellín, la cual se expuso este proyecto debido a que esta es de apoyo para su doctorado.

3. MARCO TEÓRICO

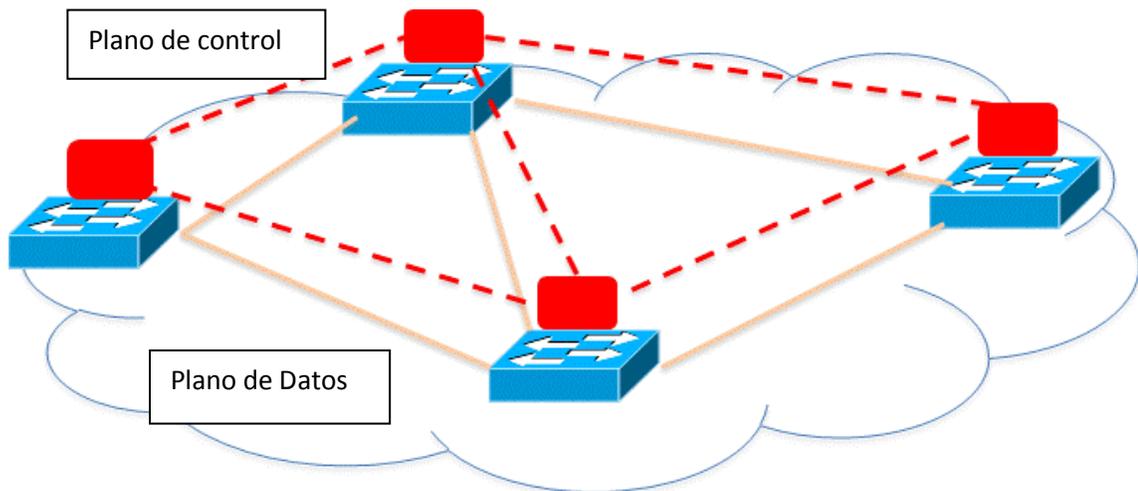
3.1 OPENFLOW

3.1.1 Redes Definidas por Software o SDN's

Las redes definidas por software, o por sus siglas en inglés SDN (Software Defined Networks), son un enfoque para la creación de redes, en el cual, el control es retirado del hardware y es dado a una aplicación de software llamado controlador.

Las redes de computadores tradicionales hacen la transmisión de paquetes utilizando dos planos diferentes: el plano de datos y el plano de control (ver figura 3).

FIGURA 3. PLANOS DE UNA COMUNICACIÓN DE DATOS EN UNA RED.



El plano de datos envía, filtra y almacena temporalmente la información mientras se procesa (*buffer*); también se hacen operaciones como marcación de paquetes, limitación de la velocidad de envío de los paquetes y la medición del tráfico. De otra parte, en el plano de control, los algoritmos son distribuidos y se encargan de controlar los cambios de topología, el cálculo de las rutas, y se establecen las reglas de re-envío.

Un tercer plano podría denominarse como el plano de Gestión. En el plano de gestión, en la escala de tiempo humana, se recogen las mediciones y se configuran los equipos.

Debido a que Internet fue un éxito total, pues inició como un experimento de investigación, y llegó a convertirse en una infraestructura global de telecomunicaciones. Internet es una red que hace el mejor esfuerzo en la entrega de paquetes y los *hosts* tienen múltiples aplicaciones. En este entorno se ha innovado mucho en aplicaciones como: P2P (*Peer to Peer*), Web, redes sociales y mundos virtuales. Sin embargo, dentro de la red, la historia es diferente, puesto que, los equipos son cerrados, el software está incluido en el hardware, la estandarización de los protocolos es lenta, pocas personas pueden innovar, los fabricantes de los equipos son los que escriben el código. También hay largas demoras para introducir nuevas características, las redes son muy complejas de administrar y muy costosas.

Por estas razones, se vió la necesidad de crear una fundación para buscar la flexibilización de los desarrollos de las redes de Telecomunicaciones. A esta fundación se le conoce como Open Networking Foundation (www.opennetworking.org), y se encarga de velar por el desarrollo de las SDNs. Las SDNs permiten construir redes de comunicaciones de una manera digna de confianza para la sociedad. Mediante las SDNs se simplifica la gestión, no hay necesidad de invertir las operaciones del plano de control, hay un ritmo de innovación más rápido, hay menos dependencia de los proveedores y las normas. Por tanto, en las SDNs los equipos son más baratos por ser más simples y con un mínimo de software. Al tener un control lógico centralizado, el control es más lento, pero con mayor inteligencia. De otro lado, los routers son más rápidos pero con menos inteligencia. El controlador cuenta con una interfaz de programación de aplicaciones, o en sus siglas en inglés, API, para el plano de datos. Un ejemplo de este tipo de Routers para SDNs es el proyecto Openflow. Las redes con Openflow tienen un manejo simple de las reglas de tratamiento de los paquetes en el plano de datos, pues el router posee una tabla de reglas que permiten hacer operaciones como enlazar bits de cabecera de paquetes, eliminar paquetes, enviar por un puerto de salida específico, modificar datos de las cabeceras y enviar paquetes hacia el controlador para un tratamiento especial.

“Cuando un paquete llega al *switch* en una red convencional, las reglas incorporadas en el *firmware* propietario del dispositivo le dicen donde enviar el paquete. El *switch* envía cada paquete que va al mismo destino a través del mismo camino y trata todos los paquetes de la misma manera. En las empresas, pueden usarse *switches* inteligentes diseñados con aplicaciones específicas de circuitos integrados, o por sus siglas en inglés ASICs, que son lo suficientemente sofisticados para reconocer diferentes tipos de paquetes y tratarlos de una manera diferente. Estos tipos de *switches* son bastante caros.

En una red definida por *software*, un administrador de red puede ajustar el tráfico desde una consola de control centralizado sin tener que tocar los *switches*. El administrador puede cambiar las reglas de cualquier *switch* de red cuando sea necesario; dándole prioridad o no, incluso bloquear tipos específicos de paquetes con un nivel de control muy bajo. Esto es especialmente útil en la arquitectura de una nube, porque le permite al administrador gestionar las cargas de tráfico de una manera flexible y más eficiente. Esencialmente, le permite al administrador usar *switches* menos caros, cómodos y tener más control sobre el flujo del tráfico de la red más que nunca.

Actualmente, una de las aplicaciones más populares para crear una SDN es un estándar abierto llamado Openflow, que permite a los administradores de red controlar de forma remota las tablas de enrutamiento¹.

3.1.2 Proyecto OpenFlow

“Openflow es la primera interfaz de comunicaciones estándar definido entre el control y reenvío de las capas de una arquitectura SDN. Openflow permite acceso directo y la manipulación al plano de reenvío de dispositivos de red tales como *switches* y *routers*, tanto físicos como virtuales. Es la ausencia de una interfaz abierta para el plano de reenvío que ha llevado a la caracterización de dispositivos de red hoy en día como monolítico, cerrado y *mainframe*. Ningún otro protocolo estándar hace lo que Openflow hace, y un protocolo como Openflow es necesario para mover el control de la red fuera de los *switches* de red hacia el software de control de lógica centralizado.

El protocolo Openflow se implementa en ambos lados de la interfaz entre dispositivos de infraestructura de red y el software de control SDN. Openflow utiliza el concepto de flujos para identificar el tráfico de red basado en reglas predefinidas de coincidencia que puede ser estática o dinámicamente programadas por el software de control SDN. Además, permite definir cómo el tráfico puede fluir a través de dispositivos de red sobre la base de parámetros tales como los patrones de uso, aplicaciones y recursos de la nube. Desde que Openflow le permite a la red ser programada sobre una base por flujo, una arquitectura SDN basada en Openflow proporciona un control extremadamente granular, permitiendo a la red responder a cambios en tiempo real a nivel de la aplicación y usuario.

¹ GIBILISCO, Stan. *Software-Defined networking (SDN)*. [en línea]. Junio de 2012. [Consultado 20 de Julio de 2012]. Disponible en: < <http://whatis.techtarget.com/definition/software-defined-networking-SDN> >

El protocolo Openflow es un factor clave para las Redes Definidas por Software o SDN, y en la actualidad es el único protocolo SDN estandarizado que permite la manipulación del plano de reenvío de los dispositivos de red.

Openflow basado en SDNs se pueden implementar en las redes existentes, tanto físicas como virtuales. Los dispositivos de red pueden soportar envíos basados en Openflow, como también, envío tradicional, lo que hace que sea más fácil para empresas y operadores introducir progresivamente tecnologías SDN basadas en Openflow, incluso en entornos de red de múltiples proveedores².

Openflow está siendo actualmente implementado por los principales proveedores como: *HP, Google, Verizon, Yahoo, Microsoft, Facebook, Cisco*, entre muchos más.

Un switch Openflow consiste de una o más tablas de flujo y una tabla de grupo, los cuales realizan búsquedas de paquetes y re-envíos, y un canal Openflow hacia un controlador externo llamado *controller*. Este controlador se apropia del switch via protocolo-Openflow. Usando este protocolo, el controlador puede adicionar, actualizar y borrar flujos de entrada.

Cada tabla de flujo en el switch contiene un conjunto de flujos de entrada; cada flujo de entrada consta de campos de enlace o de comparación, contadores y un conjunto de instrucciones para aplicar a los paquetes enlazados o comparados.

La comparación comienza en la primera tabla de flujo y puede continuar a tablas de flujo adicionales. Los flujos de entrada comparan paquetes en orden de prioridad, con la primera entrada comparada en cada tabla que se utilice. Si una entrada comparada coincide, las instrucciones asociadas con la entrada de flujo especificada son ejecutadas. Si no hay coincidencia en una tabla de flujo, la salida depende de la configuración del switch: el paquete puede ser enviado al controlador a través del canal Openflow, puede ser eliminado o puede continuar a la siguiente tabla de flujo.

Las instrucciones asociadas con cada flujo de entrada describen el re-envío del paquete, la modificación del paquete, el procesamiento de la tabla de grupo y el procesamiento pipeline.

² *Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. ONF White Paper.* Abril 13 de 2012. [en línea]. Consultado julio 14 de 2012. Disponible en:
< <https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf> >

Las instrucciones del procesamiento pipeline (este procesamiento contiene múltiples tablas de flujo, cada tabla de flujo contiene múltiples flujos de entrada, por lo tanto, este procesamiento define cómo los paquetes interactúan con esas tablas) permiten que los paquetes sean enviados a tablas siguientes para el procesamiento adicional y permiten que la información, en forma de *metadata*, para que se comunique con entre tablas. El procesamiento de tablas pipeline se detiene cuando el conjunto de instrucciones asociadas con una comparación de flujo de entrada no especifica una tabla próxima; en este punto, el paquete usualmente se modifica y se re-envía.

Los flujos de entrada pueden enviarse a un puerto. Usualmente un puerto físico, pero también puede ser un puerto virtual definidos por el switch o un puerto virtual reservado pueden especificar acciones genéricas de re-envío, así como las que se envían al controlador.

Los flujos de entrada pueden apuntar a un grupo, lo cual, especifica procesamiento adicional. Los grupos representan conjuntos de acciones para las inundaciones o *flooding*, así como más complejos re-envíos como: *multipath*, re direccionamiento rápido y agregación de enlaces. Los grupos también permiten enviar múltiples flujos a un solo identificador, como por ejemplo el re-envío IP para un siguiente salto común. Esta abstracción permite acciones de salida comunes a través de los flujos para ser cambiados de manera eficiente.

Los diseñadores de switches son libres de implementar las partes internas de cualquier forma, siempre que la comparación correcta y la semántica de las instrucciones se conservan. Por ejemplo, mientras que un flujo puede usar un grupo entero para transmitir paquetes a múltiples puertos, un programador de switches puede escoger implementar lo anteriormente mencionado como una máscara de bits única dentro de la tabla de re-envío de hardware. Otro ejemplo es la comparación; el pipeline expuesto por un switch Openflow puede ser implementado físicamente con un número diferente de tablas de hardware.

3.2 NETFPGA

“La NetFPGA es una plataforma de hardware reconfigurable de bajo costo, especializada para diseñar dispositivos de redes de alta velocidad. La tarjeta incluye una FPGA (*Field Programmable Gate Array*), específicamente la *Xilinx Virtex-II Pro 50* la cual cuenta con más de 53000 células lógicas para reconfigurar, una memoria RAM de 4.5 Mb, una DDR2 SDRAM de 64 Mb y de interfaces, un puerto PCI para conectar a un PC y además 4 puertos *Gigabit Ethernet* con los cuales se puede diseñar cualquier dispositivo de red como una tarjeta de red, un *switch* o incluso un *router*.

Ya que todo el diseño de los dispositivos de red se realiza en hardware, la velocidad de procesamiento es en *Gigabit*, por lo cual no hay retrasos ni pérdidas de paquetes dentro de la plataforma. La tarjeta NetFPGA va conectada a un PC *Host* por medio del puerto PCI desde donde se programa la tarjeta³.

FIGURA 4. TARJETA NETFPGA



Esta tarjeta contiene:

- Conector PCI: Bus estándar de 32 bits, 33 Mhz. Este es el que permite conectar la tarjeta a un PC.
- 4 Interfaces Gbit Ethernet (1Gbps).
- 1 chip FPGA Virtex-2.
- 4 Bancos de memoria SRAM y DRAM.

³ PADILLA AGUILAR, Jhon Jairo; BECERRA SÁNCHEZ, Line Yasmin; MORALES GONZÁLEZ, Alvaro Ignacio. U2-ROUTE: UNIVERSITY UNIVERSAL ROUTER. Guía práctica. Primera edición. Bucaramanga: Sic Editorial Ltda. Diciembre de 2011. 117p.

3.3 GNU/LINUX CENTOS

3.3.1 Definición software libre o GNU

GNU significa, de sus siglas en Inglés (GNU is Not Unix) que quiere decir en español GNU No es Unix. Es un proyecto creado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre y sin ningún tipo de limitación. A diferencia del software propietario, éste se puede redistribuir a cuantos computadores se deseen sin tener ningún tipo de preocupación por lo anteriormente mencionado.

“Una de las características más interesantes de tener un software libre es que el código fuente del programa está presente en él, en el cual se podrán hacer modificaciones del mismo y adecuarlo a las necesidades que requiera el usuario⁴”.

3.3.2 Definición e historia GNU/LINUX

Es un sistema operativo multiusuario de software libre basado en UNIX, y cualquier persona puede hacer parte de él, puesto que por ser libre, muchos programadores de todo el mundo unen todos sus componentes desarrollados y forman este gran sistema operativo, en el cual, se pueden hacer las más diversas tareas, desde aplicaciones multimedia hasta montaje de servidores de red.

“Toda esta unión empezó a principios de la década del ‘80, cuando un científico del área de inteligencia artificial del Instituto de Tecnología de *Massachusetts* (MIT), llamado Richard Stallman, decidió comenzar a desarrollar un sistema operativo libre, ya que en esa época la única opción que tenían los usuarios de computadoras era adquirir un software propietario. El nombre de este sistema es GNU, acrónimo de las palabras “¡GNU No es Unix!”. La idea principal era que el sistema mantuviera un grado de similitud con el ya conocido UNIX sin compartir una sola línea de código fuente. Idea que, obviamente, fue hecha realidad en algunos años.

Para finales de los ‘80, el sistema estaba prácticamente completo. Disponía de editores de texto, intérpretes de comandos, compiladores, *debuggers*, etc. Sólo faltaba un componente: el núcleo (*kernel*).

Casualmente, en el otro lado del mundo (Finlandia), un estudiante llamado Linus B. Torvalds desarrolló un núcleo compatible con UNIX, sin saber que iba a formar parte del sistema operativo más revolucionario del mundo de la informática. Así fue como apareció Linux, el núcleo del sistema GNU. Y es así como hoy tenemos un sistema completo a nuestra disposición, el sistema GNU/Linux.

Paralelamente con el desarrollo de este sistema operativo, surgió la Fundación del Software Libre⁵, la cual fomenta, entre otras cosas, la utilización de herramientas de Software Libre en las computadoras de todo el mundo.

GNU es uno de los proyectos más grandes de Software Libre. En él participan miles de personas de todo el mundo colaborando con el desarrollo de aplicaciones para el sistema, documentación, mantenimiento del servidor web y muchísimo más⁴.

3.3.3 Definición Centos 5

“CentOS es una distribución Linux de clase empresarial derivada de fuentes libremente ofrecidas al público por un destacado proveedor: North American Enterprise Linux. CentOS se ajusta plenamente a la política de redistribución del proveedor anteriormente mencionado, y aspira a ser 100% compatible a nivel binario.

CentOS es desarrollado por un equipo pequeño, pero con un creciente número de desarrolladores del núcleo. A su vez, los desarrolladores principales son apoyados por una comunidad de usuarios activa, incluyendo los administradores de sistemas, administradores de red, los usuarios empresariales, gerentes, colaboradores de núcleo de Linux y los entusiastas de Linux de todo el mundo⁶”.

⁴ FACUNDO ARENA, Héctor. La biblia del Linux. Ciudad de Buenos Aires: MP Ediciones. Enero de 2003. 264 p.

⁵ Página oficial GNU. Disponible en:< <http://www.gnu.org>>

⁶ Página oficial CentOS. Disponible en:< <http://www.centos.org>>

4. RESULTADOS DEL PROYECTO

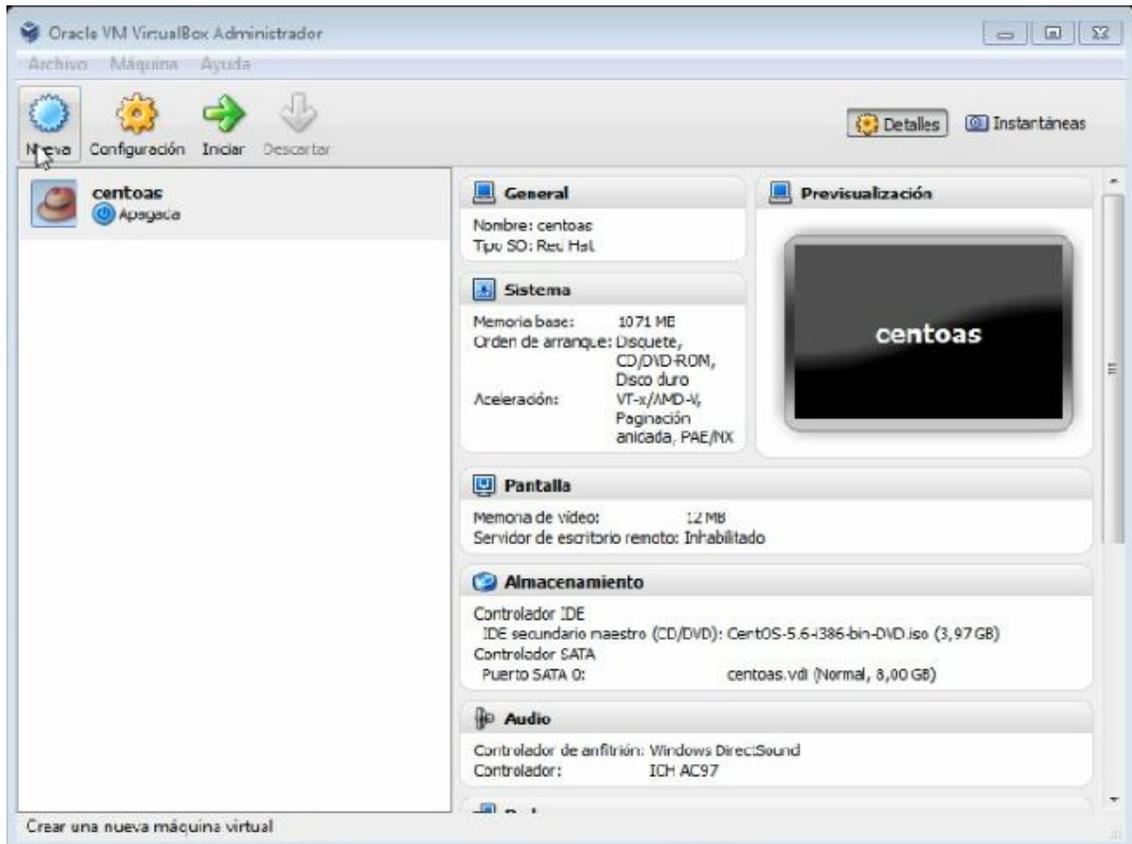
4.1 TUTORIAL DE INSTALACIÓN CENTOS 5 EN MÁQUINA VIRTUAL *VIRTUALBOX*

Como primera medida, se requiere tener instalada la aplicación Virtualbox. También, se debe tener a la mano una imagen iso (es una copia exacta de algún fichero o unidad, ya sea CD, DVD) del sistema operativo Linux CentOS 5.6 para su instalación.

La distribución de CentOS 5 u otras versiones se pueden descargar de forma gratuita, desde la página web principal de CentOS (<http://www.centos.org>).

Este tutorial se realizó en una máquina virtual llamada *VirtualBox*, por ende, todo lo que se instale va a quedar guardado en un disco duro virtual, sin afectar el disco duro físico del computador. La idea de realizarlo de esta forma es para que los estudiantes que por primera vez van a instalar esta distribución de Linux se familiaricen y aprendan a utilizarlo, para luego instalarlo en un disco duro físico. A continuación, se mostrará todo el proceso tanto de la creación del disco duro virtual como de la instalación de la distribución CentOS.

FIGURA 5. CREACIÓN DE MÁQUINA VIRTUAL.



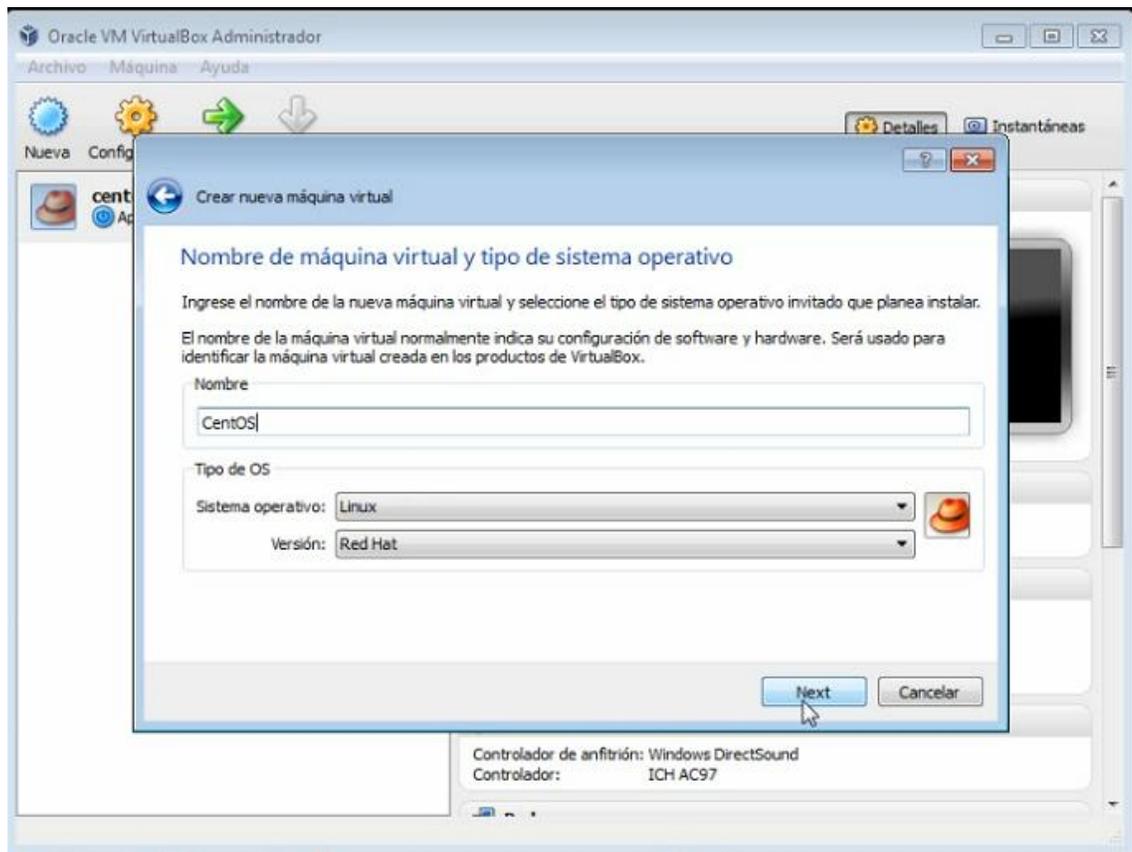
La figura 5, muestra la aplicación *VirtualBox* de *Oracle*⁷, para crear una máquina virtual se procede a dar *click* en nueva, como la figura anterior.

La ventana que aparece luego del paso anterior es la de la figura 6, en la cual, se ingresa el nombre de la máquina virtual a crear y se elige el tipo de sistema operativo que se planea instalar. En este caso, el sistema operativo es Linux y la versión es *Red Hat*⁸.

⁷ Página oficial de Oracle. Disponible en:< <http://www.oracle.com>>

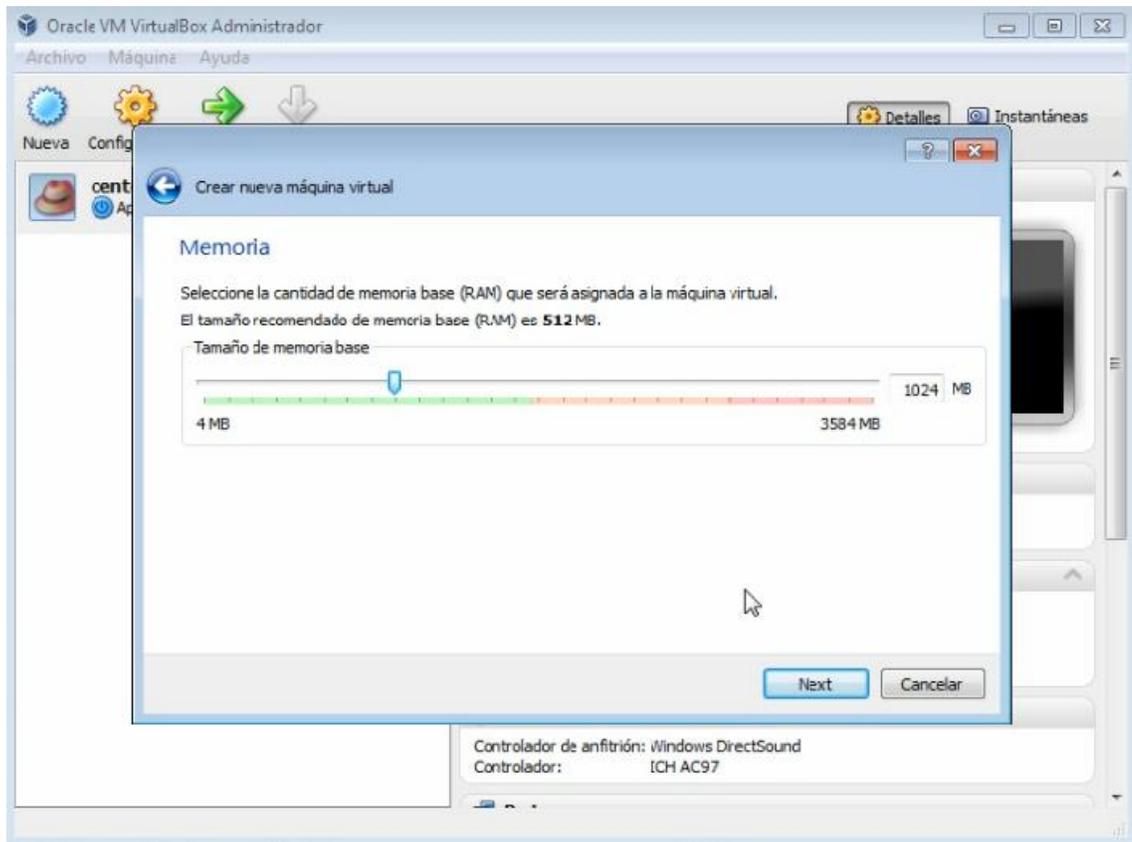
⁸ Página oficial de la distribución de Linux *red hat*. Disponible en:< <http://redhat.com>>

FIGURA 6. INGRESAR EL NOMBRE, SISTEMA OPERATIVO Y VERSIÓN DE LA MÁQUINA VIRTUAL.



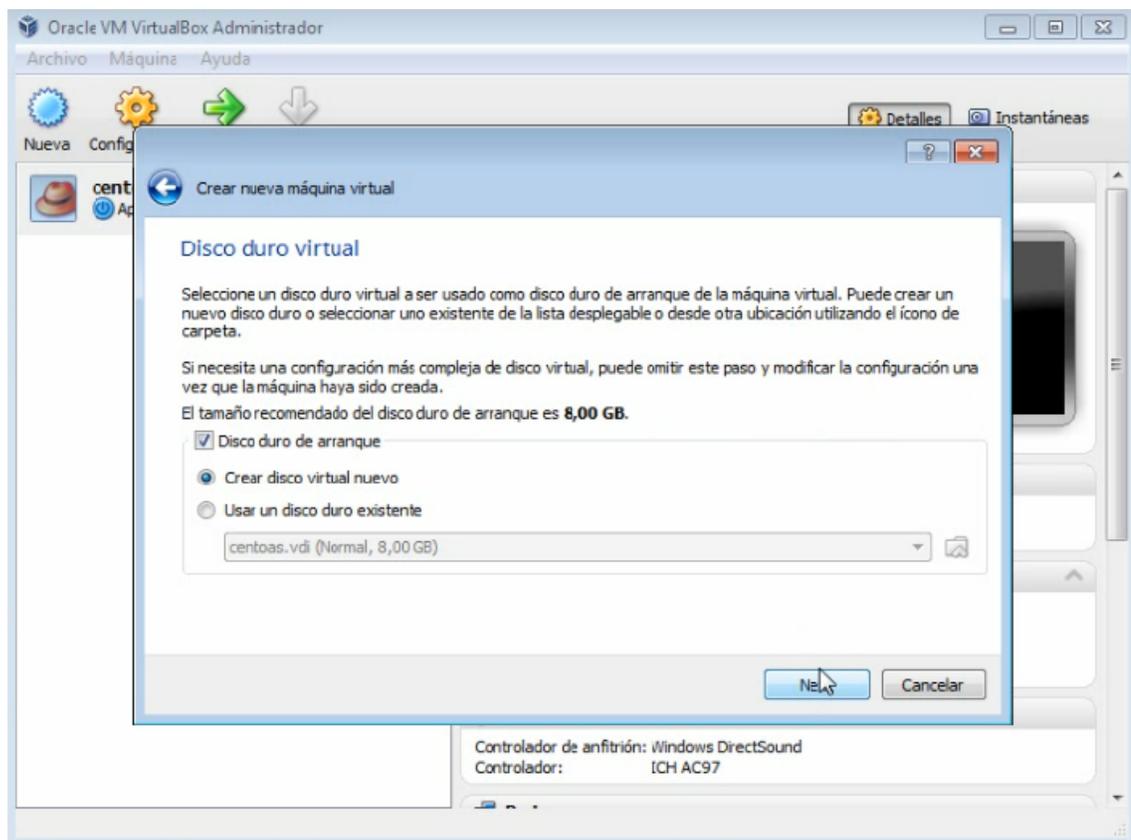
En la figura 7, se procede a seleccionar la cantidad de memoria RAM que será asignada a la máquina virtual. Como esta máquina virtual es para uso de aprendizaje, se opta por escoger 1024 MB.

FIGURA 7. CANTIDAD DE MEMORIA RAM PARA LA MÁQUINA VIRTUAL.



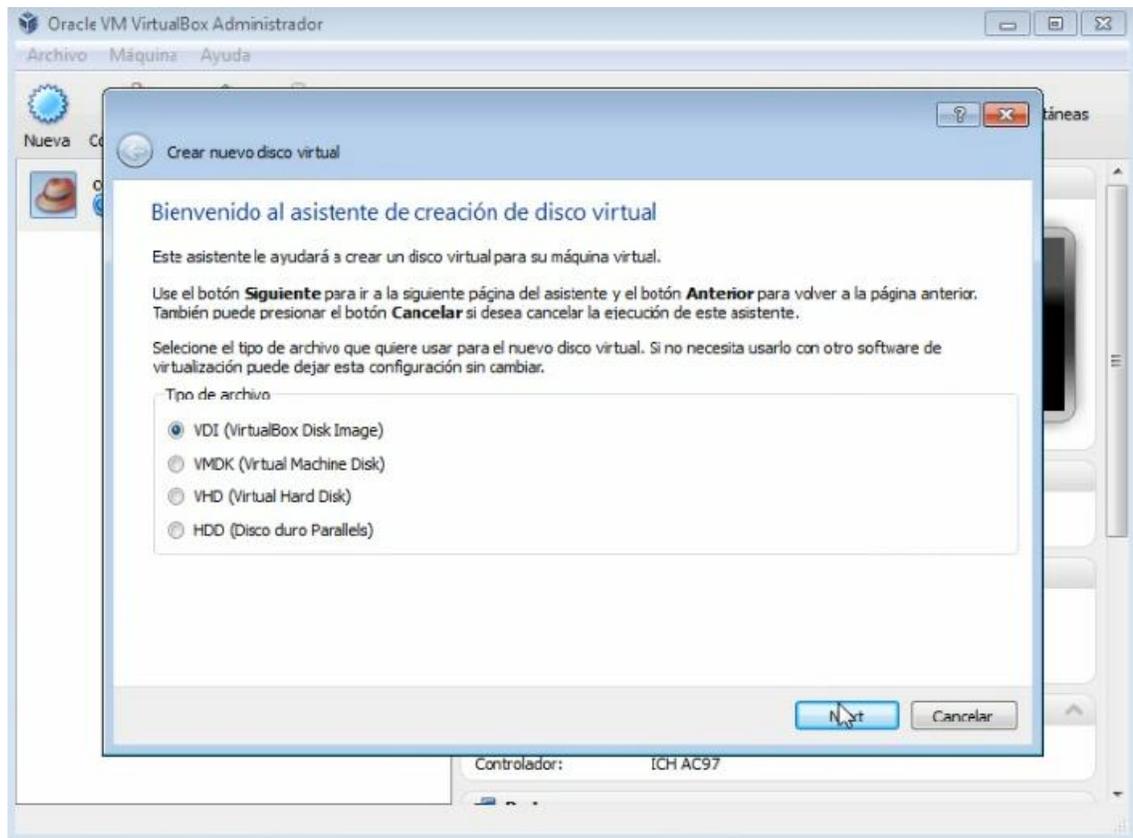
En la figura 8, se escoge la opción de crear un disco duro virtual nuevo y posteriormente, se dejan las opciones tal cual como están.

FIGURA 8. SELECCIONAR DISCO DURO VIRTUAL EXISTENTE O NUEVO.



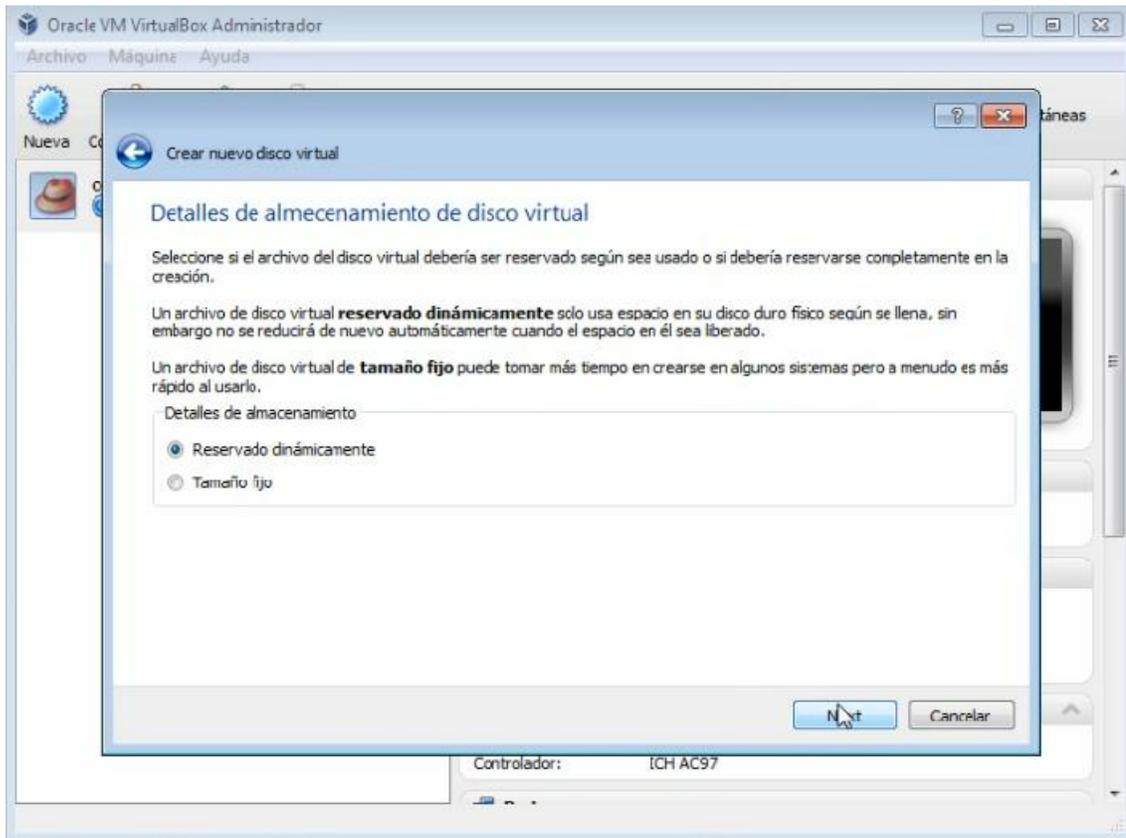
En la figura 9, se debe seleccionar la opción VDI (*VirtualBox Disk Image*) para el tipo de archivo que se quiere usar para el nuevo disco duro virtual.

FIGURA 9. SELECCIONAR EL TIPO DE ARCHIVO A USAR PARA EL NUEVO DISCO VIRTUAL.



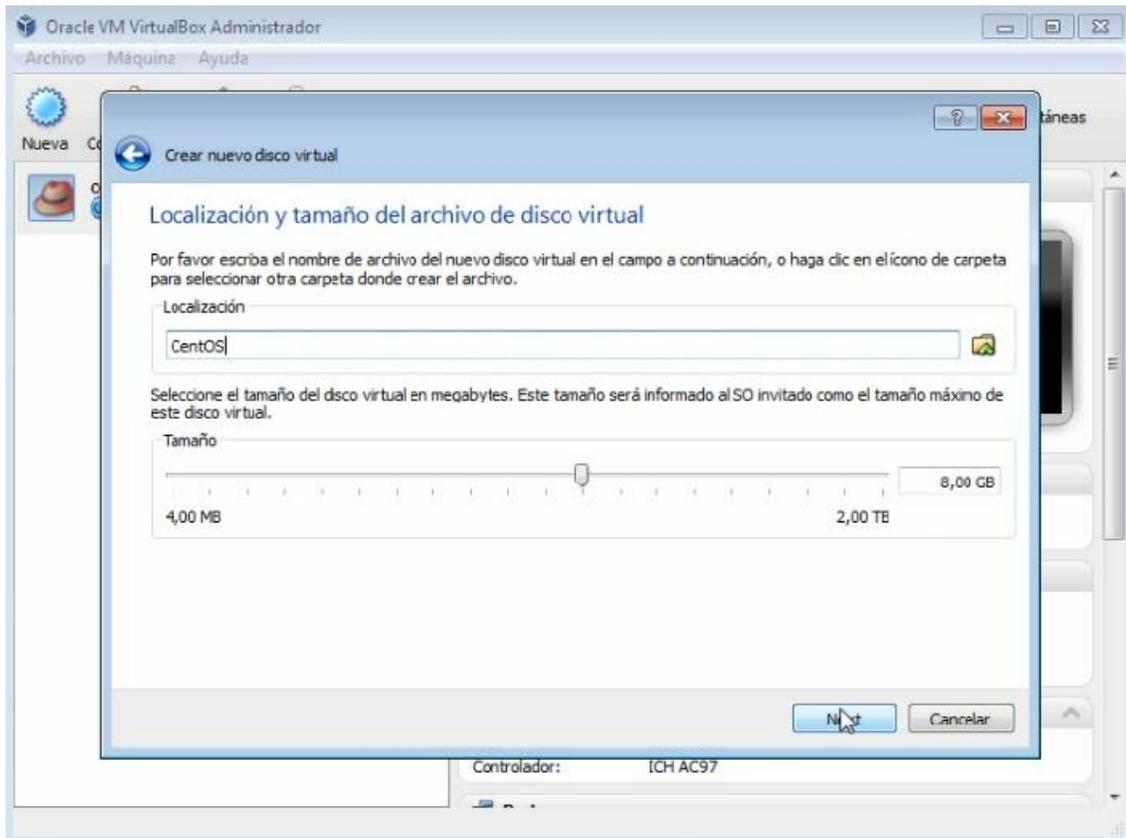
En la figura 10, para el almacenamiento, se recomienda escoger la opción reservado dinámicamente para no malgastar espacio del disco duro físico.

FIGURA 10. ALMACENAMIENTO DE DISCO VIRTUAL.



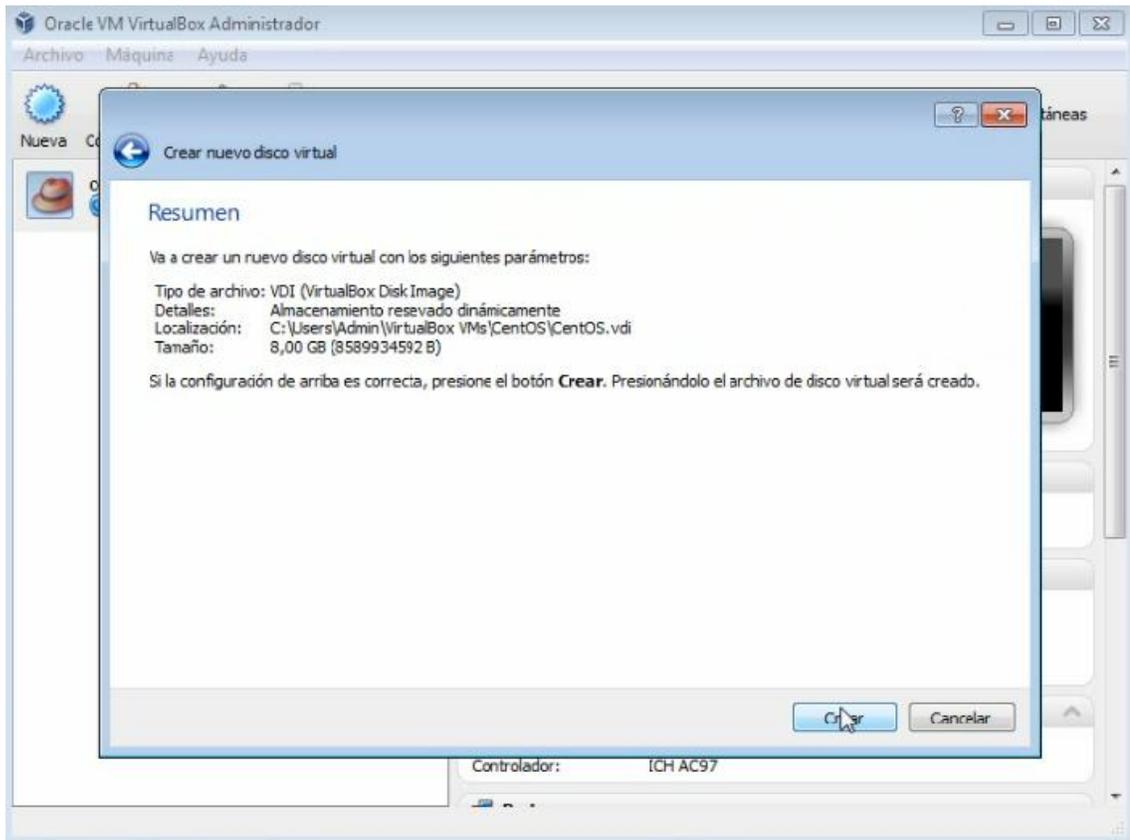
En la figura 11, se debe dejar el tamaño por defecto, 8 GB para el tamaño del disco virtual.

FIGURA 11. NOMBRE Y TAMAÑO DEL DISCO VIRTUAL.



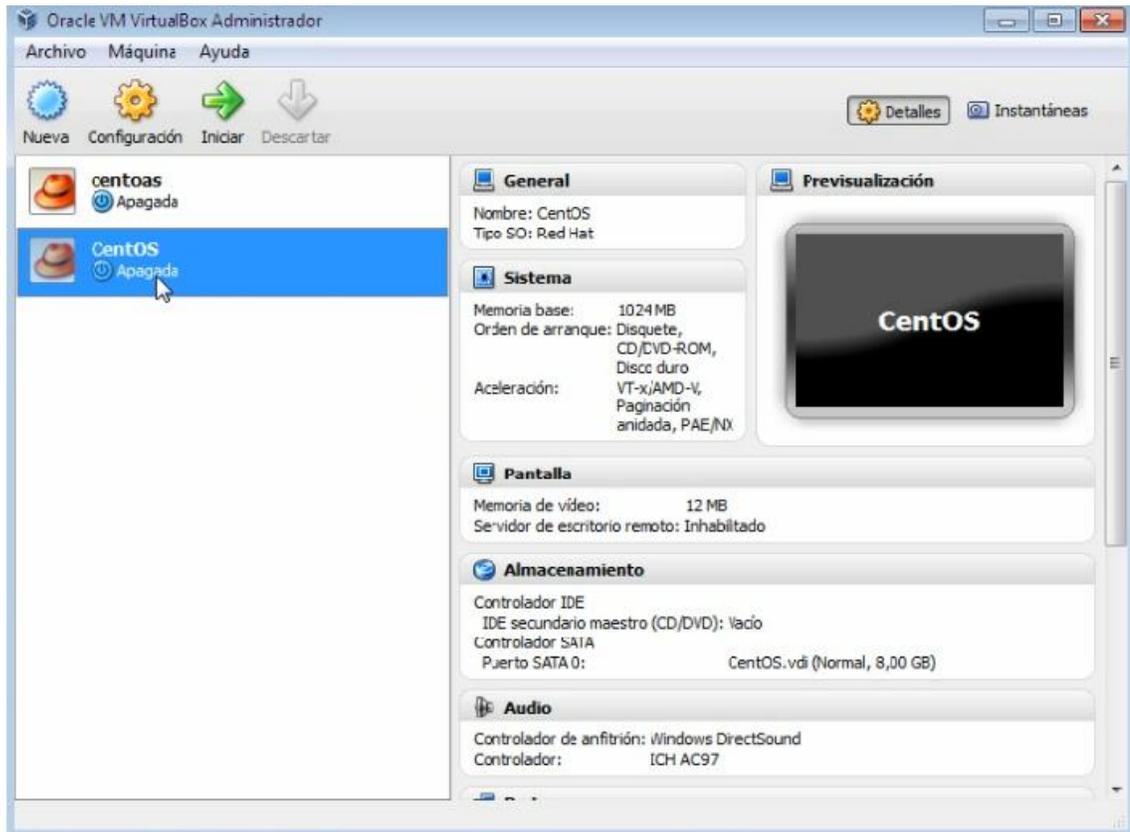
La figura 12, muestra un resumen de las características que va a tener el nuevo disco virtual, por lo que se procede a dar *click* en crear.

FIGURA 12. RESUMEN DEL NUEVO DISCO VIRTUAL A CREAR.



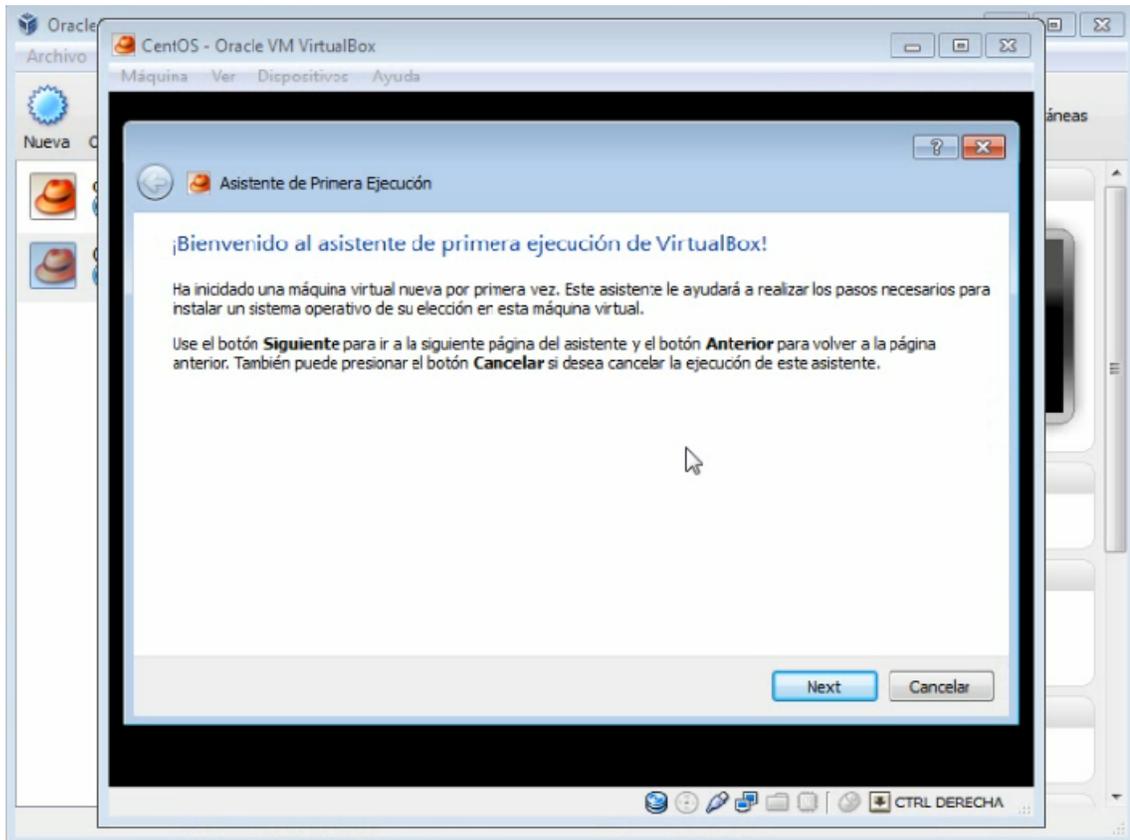
La figura 13, muestra ya la máquina virtual creada, ahora para instalar el CentOS, se procede a dar doble *click* en la máquina virtual o seleccionarla y dar *click* en la opción iniciar.

FIGURA 13. MÁQUINA VIRTUAL CREADA.



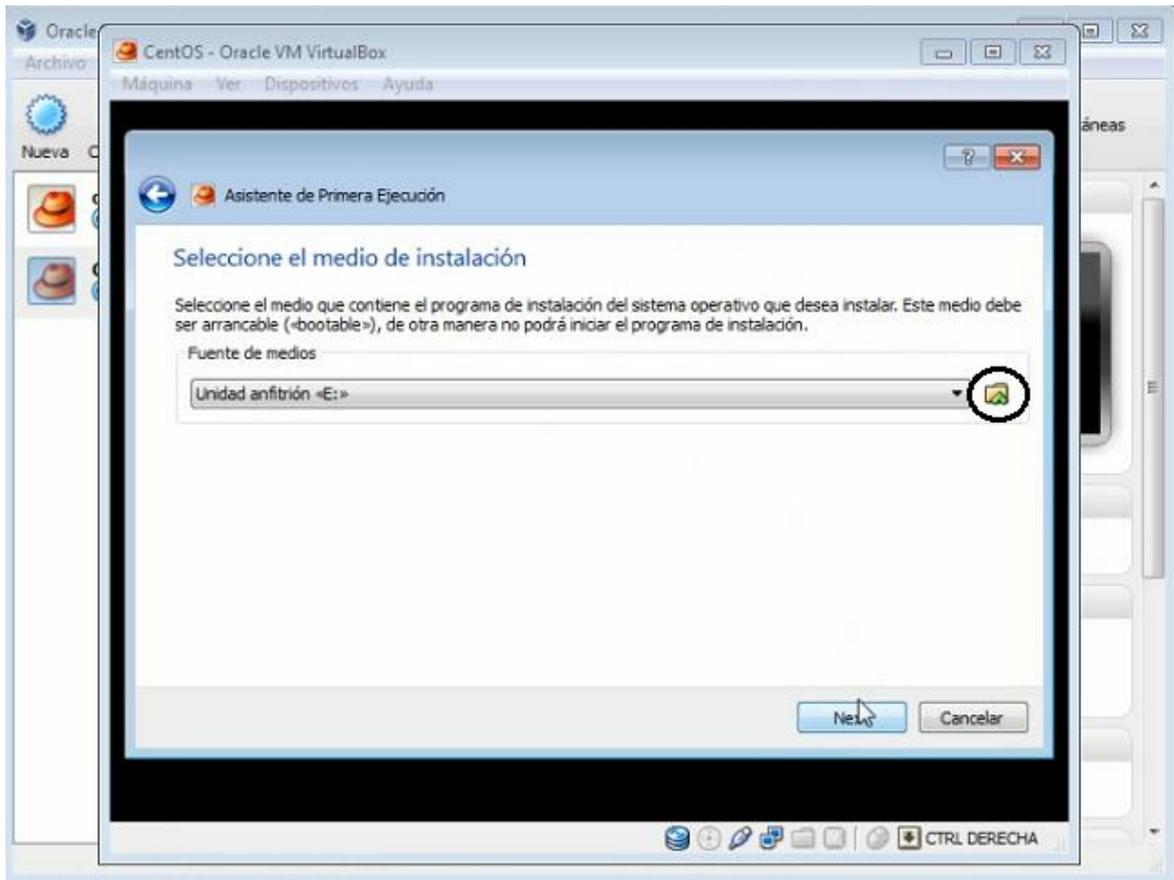
En la figura 14, se puede ver un asistente de ayuda para instalar el sistema operativo, en este caso el CentOS 5.6. A continuación, se debe dar *click* en *next* para iniciar el proceso de instalación.

FIGURA 14. ASISTENTE DE AYUDA PARA LA INSTALACIÓN DEL SISTEMA OPERATIVO.



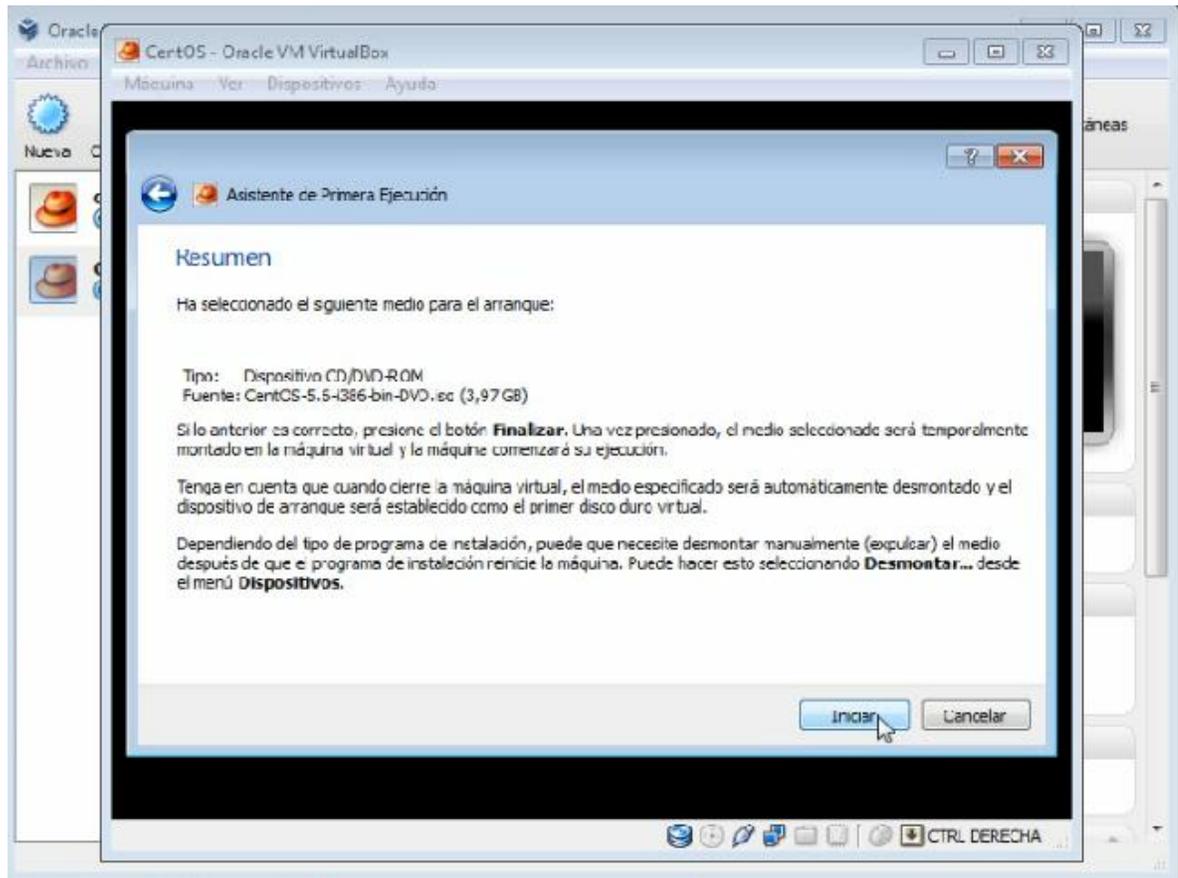
En la figura 15, se escoge el medio de instalación del CentOS, cabe resaltar que el DVD o USB a utilizar debe ser de arranque (comúnmente conocido como booteable) que significa, la secuencia de arranque que tienen los computadores cuando se enciende el mismo, puede ser DVD, disco duro, CD, etc. Para explorar las carpetas donde se encuentra el archivo imagen, se procede a dar *click* en la carpeta encerrada por el círculo.

FIGURA 15. MEDIO DE INSTALACIÓN DEL CENTOS.



La figura 16, muestra un resumen en el cual se pregunta si el archivo imagen escogido en el paso anterior es el correcto. De ser así, se procede a dar *click* en iniciar, de lo contrario, en cancelar.

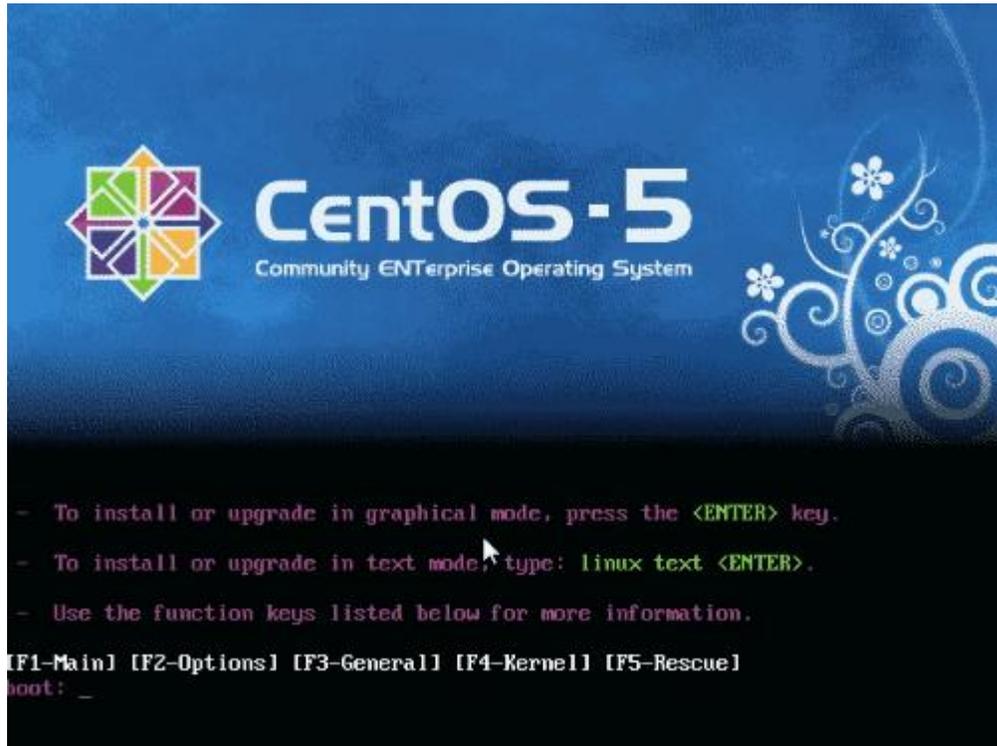
FIGURA 16. RESUMEN DEL ARCHIVO IMÁGEN.



Si la instalación se va a realizar en un disco duro rígido, antes de empezar con el proceso de instalación, se procede a mirar si la BIOS del computador en el cual se va a instalar el sistema operativo está configurado de manera adecuada para arrancar, ya sea desde una unidad de DVD o una unidad USB, dependiendo de dónde se haya grabado la imagen .iso del CentOS para su posterior instalación.

La primera imagen que debe aparecer es la de la figura 17, la cual, tiene diferentes opciones de instalación o actualización de tipo: modo gráfico o modo de texto. Se procede a dar *intro* o *enter* en la opción instalar o actualizar en modo gráfico para una mayor facilidad de instalación para las personas que apenas van a aprender a manejar el CentOS.

FIGURA 17. DIFERENTES TIPOS DE INSTALACIÓN DE CENTOS.



Después de lo anterior, sale en pantalla la posibilidad de comprobar si la imagen del DVD o USB se grabó de forma adecuada (figura 18). Esta acción requiere de cierto tiempo de comprobación. Si se quiere saber si el DVD o USB se grabó de forma correcta, se procede a explorar las opciones con la tecla “tab” o tabulador, que se encuentra una posición arriba de la tecla mayúscula, ya que, el ratón en este proceso queda inservible.

Al quedar la opción sombreada, lo siguiente es pulsar la tecla *enter* o *intro* para hacer el proceso de comprobación. Si se sabe que el DVD o USB está bien grabado, debido a que, anteriormente se había instalado la distribución CentOS en otro computador con el mismo DVD o USB, se procede a pulsar *intro* en la opción *skip* cuando ella se encuentre sombreada.

FIGURA 18. TEST DE COMPROBACIÓN DE DVD O USB.



Se asume que el DVD o USB están bien instalados, por ende, no se muestran los procesos siguientes a la comprobación.

Luego del paso anterior, se muestra en la figura 19, la pantalla de bienvenida de CentOS 5. En este momento, ya se puede usar el ratón y se procede a pulsar en la opción *next*, para continuar con el proceso de instalación.

FIGURA 19. PANTALLA DE BIENVENIDA DE CENTOS 5.



En la figura 20, aparece una ventana, la cual, es la de la elección del idioma por el cual se desea usar durante el resto de la instalación. A conveniencia, se opta por elegir el idioma español.

FIGURA 20. ELECCIÓN DEL IDIOMA DE INSTALACIÓN.



La figura 21, muestra la selección del teclado apropiado para el sistema. A continuación, se elige el teclado español.

FIGURA 21. ELECCIÓN DEL TECLADO.



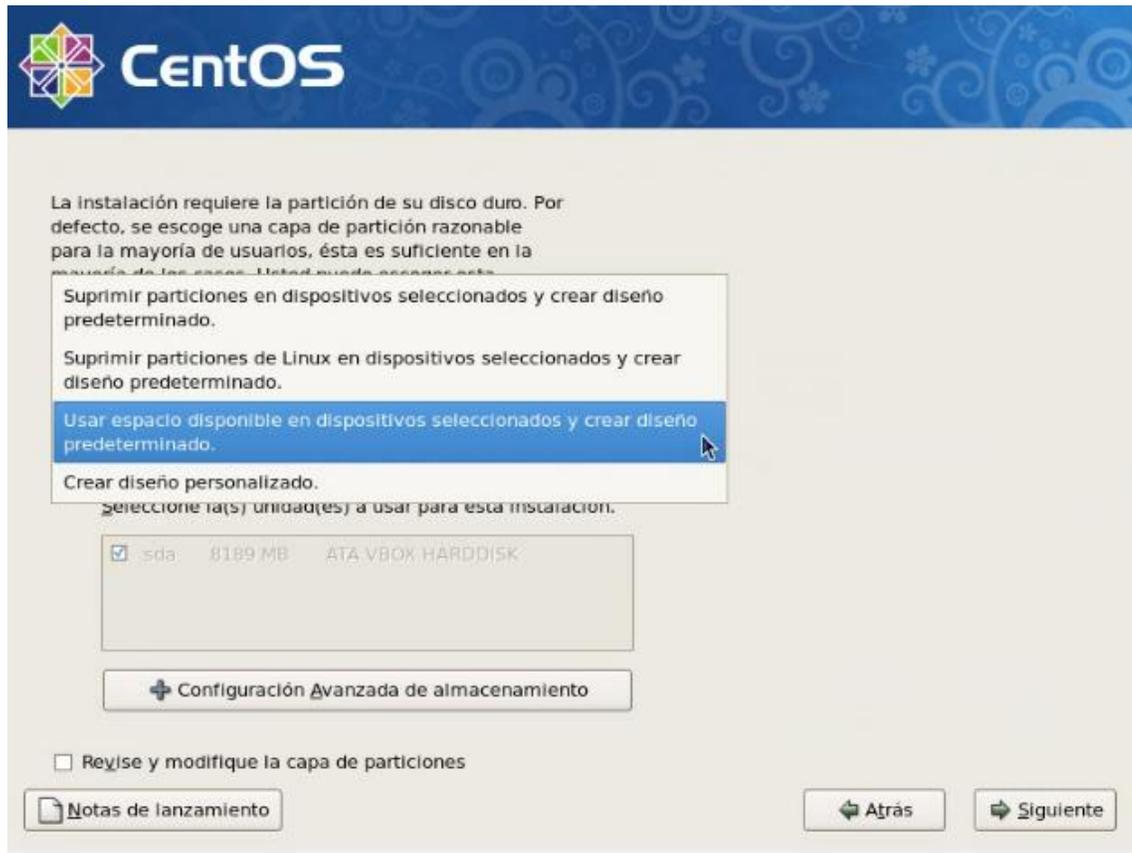
FIGURA 22. AVISO DE PARTICIÓN.



Como la instalación se está realizando por medio de una máquina virtual, no hay ningún problema de inicializar la unidad y eliminar todos los datos (figura 22), puesto que, el disco virtual que se creó anteriormente no contiene ningún otro sistema operativo instalado, lo que quiere decir que el disco está vacío. Por lo tanto, se procede a dar *click* en la opción sí.

La figura 23, muestra ciertas opciones en cuanto a la partición del disco duro se refiere.

FIGURA 23. PARTICIÓN DEL DISCO DURO.



En esta etapa se supone que el disco duro virtual se encuentra vacío, ya que, el proceso anterior lo formateó, por lo tanto, se escoge la opción: usar espacio disponible en dispositivos seleccionados y crear diseño predeterminado y se debe dar *click* en siguiente.

En la figura 24, se muestra la configuración de los dispositivos de red. Como se puede apreciar, el dispositivo, que en este caso es un eth0, se detecta automáticamente, por lo cual, se da *click* en siguiente.

FIGURA 24. CONFIGURACIÓN DE LOS DISPOSITIVOS DE RED.



The screenshot shows the CentOS network configuration window. At the top is the CentOS logo and name. Below it, the section "Dispositivos de red" contains a table with columns for "Activar al inicio", "Dispositivo", "IPv4/Máscara de red", and "IPv6/Prefijo". A "Modificar" button is to the right. The table has one row with a checked checkbox, "eth0", "DHCP", and "Auto". Below this is the "Nombre del Host" section with a "Configurar el nombre del host:" label. It has two radio buttons: "de forma automática a través de DHCP" (selected) and "manualmente" with a text input field containing "localhost.localdomain" and an example "(ej. 'mipc.dominio.com.ar')". The "Configuración miscelánea" section has three text input fields for "Puerta de enlace:", "DNS Primario:", and "DNS Secundario:". At the bottom, there is a "Notas de lanzamiento" button, an "Atrás" button, and a "Siguiente" button with a mouse cursor over it.

Activar al inicio	Dispositivo	IPv4/Máscara de red	IPv6/Prefijo
<input checked="" type="checkbox"/>	eth0	DHCP	Auto

A continuación, la figura 25, muestra una ventana en la cual, se debe elegir la configuración horaria, de acuerdo al país donde se encuentre la persona en ese momento. Por ende, se escoge la opción Bogotá-América, dando *click* correspondiendo al mapa de Colombia, o en su defecto buscar en el cuadro de selección Bogotá.

FIGURA 25. CONFIGURACIÓN ZONA HORARIA.



La elección de la contraseña del super usuario se puede apreciar en la figura 26, ese super usuario es el encargado de administrar el sistema. Luego de suministrar la debida contraseña, se da *click* en la opción siguiente.

FIGURA 26. CONTRASEÑA DEL USUARIO ROOT O SUPER USUARIO.



La figura 27, muestran las aplicaciones a instalar por defecto, en este caso se elige la opción *Desktop-Gnome* para tener un entorno gráfico el cual provee ventanas, menús, íconos y muchas más aplicaciones gráficas así como las que tiene *Windows*. Posterior a eso, se debe dar *click* en siguiente.

FIGURA 27. APLICACIONES PARA USO GENERAL DE INTERNET.



Luego del paso anterior, se muestra la figura 28, la cual inicia la instalación de CentOS, y por tanto, se debe dar *click* en la opción siguiente.

FIGURA 28. VENTANA DE INICIO DE LA INSTALACIÓN DEL CENTOS 5.



La figura 29 y figura 30 muestran respectivamente el formateo de las particiones creadas previamente y la transferencia de los datos desde el DVD o USB usada, hacia el disco virtual del sistema.

FIGURA 29. VENTANA DE INICIO DE LA INSTALACIÓN DEL CENTOS 5.



FIGURA 30. TRANSFERENCIA DE DATOS DESDE LA IMAGEN DEL PROGRAMA DE INSTALACIÓN HACIA EL DISCO DURO.



La instalación de los programas elegidos en todo el proceso de configuración se muestra en la figura 31, en los cuales se puede apreciar una barra de llenado con los programas que se están instalando en ese momento y una breve explicación de para qué sirven dichos programas.

FIGURA 31. INSTALACIÓN DEL SOFTWARE.



La etapa de instalación ha finalizado, y se requiere reiniciar el sistema para cargar el CentOS por primera vez en el equipo, como se puede ver en la figura 32. En ese momento, se debe retirar el DVD o USB usado para la instalación.

FIGURA 32. INSTALACIÓN COMPLETA DE CENTOS 5.



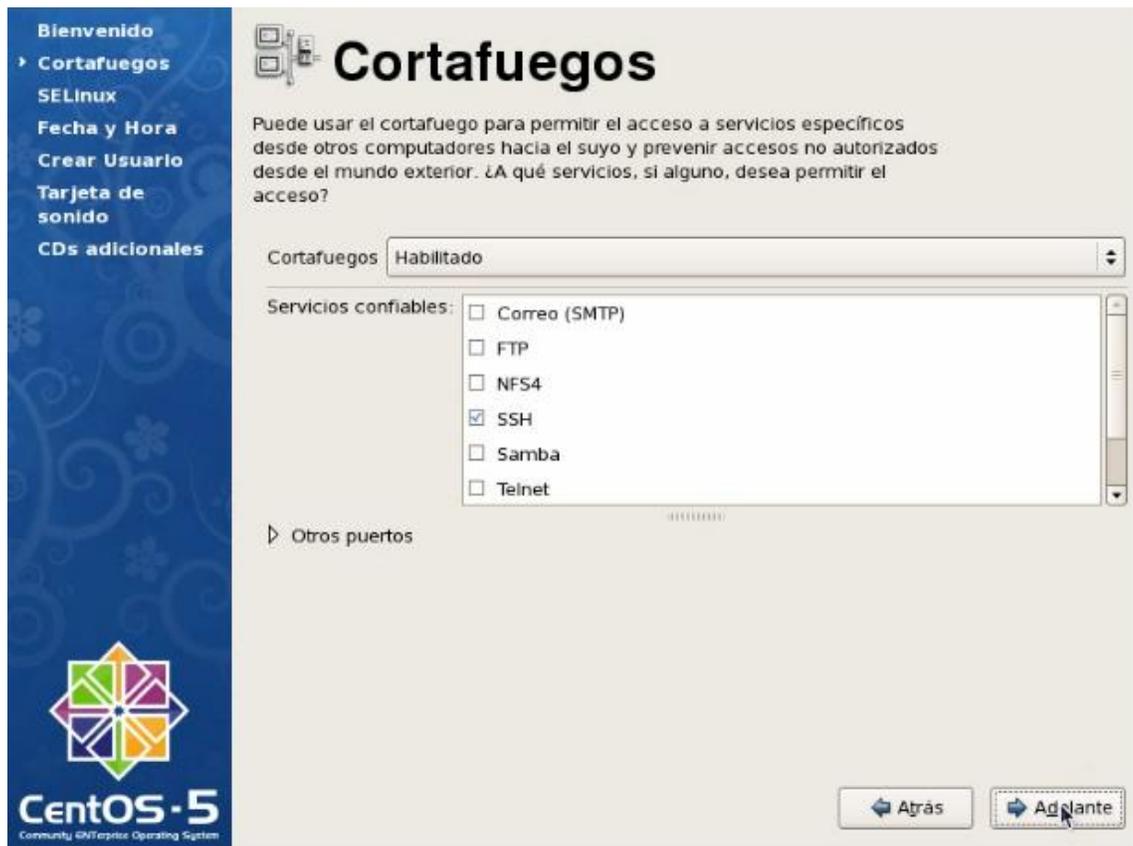
La siguiente ventana muestra el mensaje de bienvenida y algunos pasos más que se deben realizar antes que el sistema esté listo para ser usado. Lo anteriormente mencionado se puede apreciar en la figura 33. Se procede a dar *click* en la opción adelante.

FIGURA 33. MENSAJE DE BIENVENIDA DE CENTOS 5.



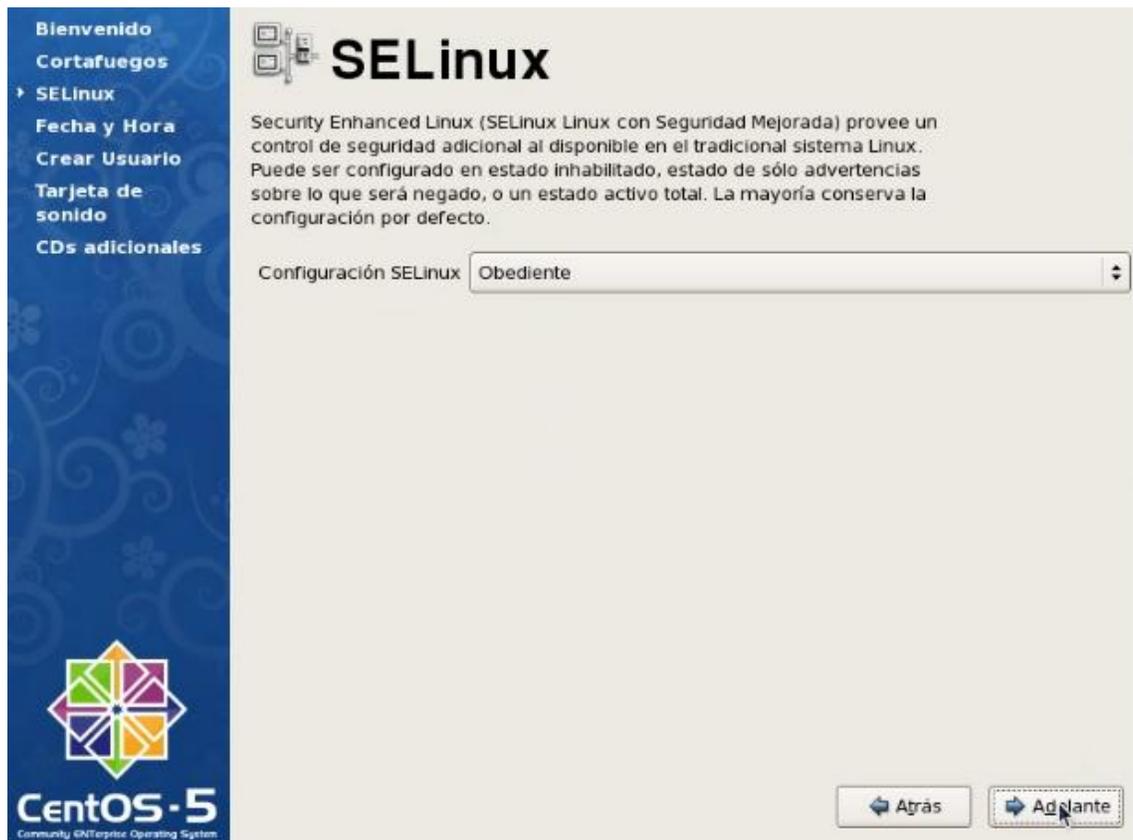
La configuración del cortafuego puede verse en la figura 34, en la cual se escoge la opción "habilitado" y como servicios confiables se escoge la opción SSH. Luego, se da *click* en adelante.

FIGURA 34. CONFIGURACIÓN DE CORTAFUEGOS.



Como se puede ver en la figura 35, SELinux es utilizado para proveer un control de seguridad adicional al disponible en el sistema tradicional de Linux. Por defecto, se escoge la opción “obediente” y se procede a dar *click* en la opción adelante.

FIGURA 35. CONFIGURACIÓN SELINUX.



La figura 36 muestra la configuración de la fecha y hora para el sistema. Luego de verificar la configuración se dá *click* en la opción adelante.

FIGURA 36. CONFIGURACIÓN FECHA Y HORA.



Luego, en la figura 37, se procede a crear un usuario normal (no super usuario, anteriormente creado). Al tener todos los campos llenos, se dá *click* en la opción adelante. Los usuarios normales pueden ser cuantos quiera a diferencia del super usuario. Estos se pueden crear una vez cargado completamente el CentOS, más adelante se explicará las características que tiene cada usuario.

FIGURA 37. CREACIÓN DE USUARIO.

Bienvenido
Cortafuegos
SELinux
Fecha y Hora
▶ Crear Usuario
Tarjeta de sonido
CDs adicionales

Crear Usuario

Se recomienda que cree un nombre de usuario para uso normal (no administrativo) del sistema. Para crear un nombre de usuario del sistema, proporcione la información requerida a continuación.

Nombre de usuario:

Nombre completo:

Contraseña:

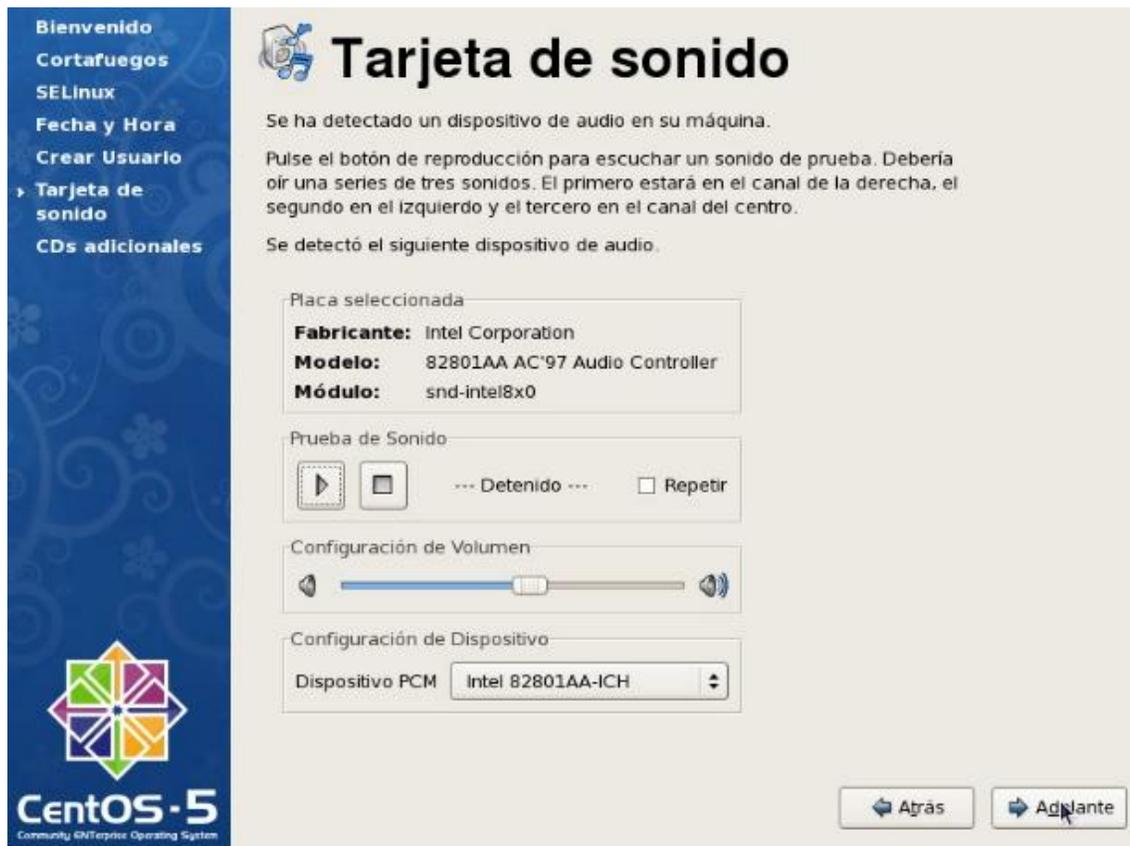
Confirmar contraseña:

Si necesita utilizar una autenticación de red tal como Kerberos o NIS, por favor pulse en el botón Utilizar conexión de red.

CentOS-5
Community Enterprise Operating System

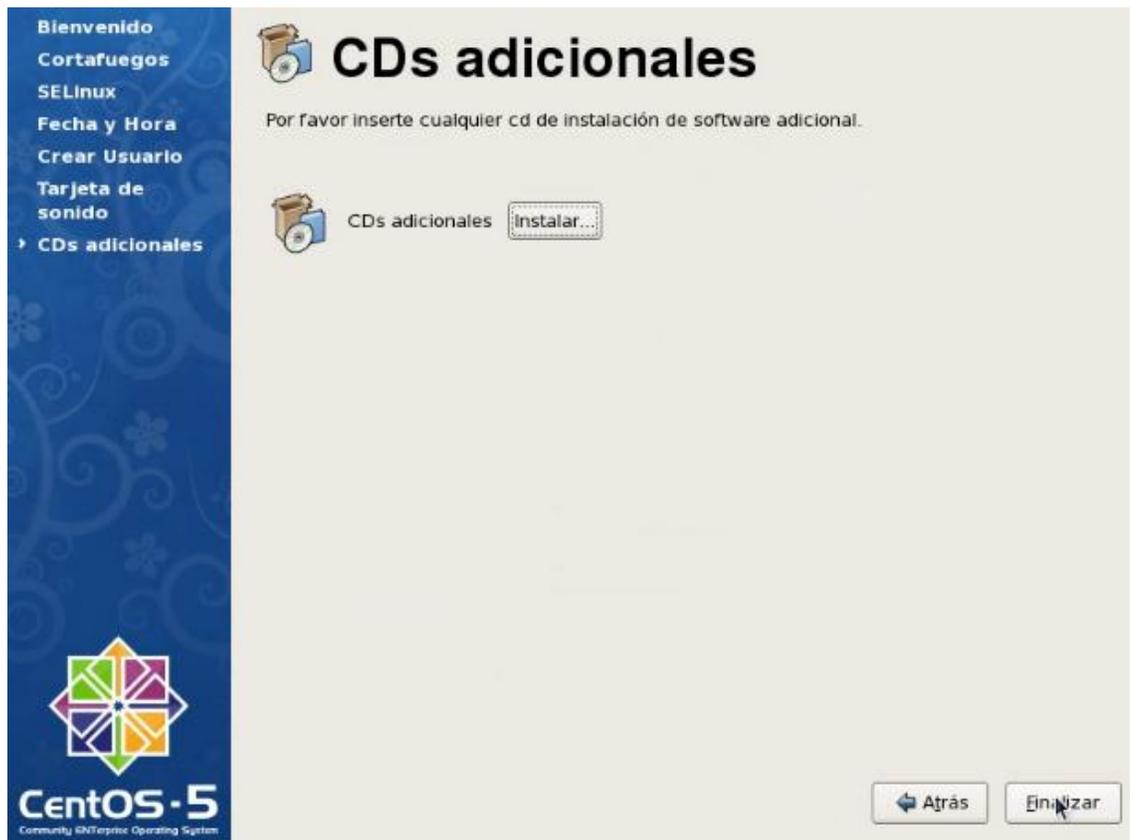
La figura 38, muestra la configuración de la tarjeta de sonido, la cual, detecta automáticamente la existencia o no de la misma. Independientemente del resultado se debe dar *click* en siguiente.

FIGURA 38. CONFIGURACIÓN DE LA TARJETA DE SONIDO.



En la figura 39, se puede ver la instalación de software adicional por medio de la unidad de CD. En este caso, no se requiere la instalación de software adicional y por ende, se procede a dar *click* en finalizar.

FIGURA 39. CONFIGURACIÓN DE SOFTWARE ADICIONAL.



Las figuras 40 y 41, muestran respectivamente la pantalla de acceso a CentOS, donde se escribe el usuario al cual se quiere acceder y la contraseña correspondiente para dicho usuario.

FIGURA 40. NOMBRE DE USUARIO.

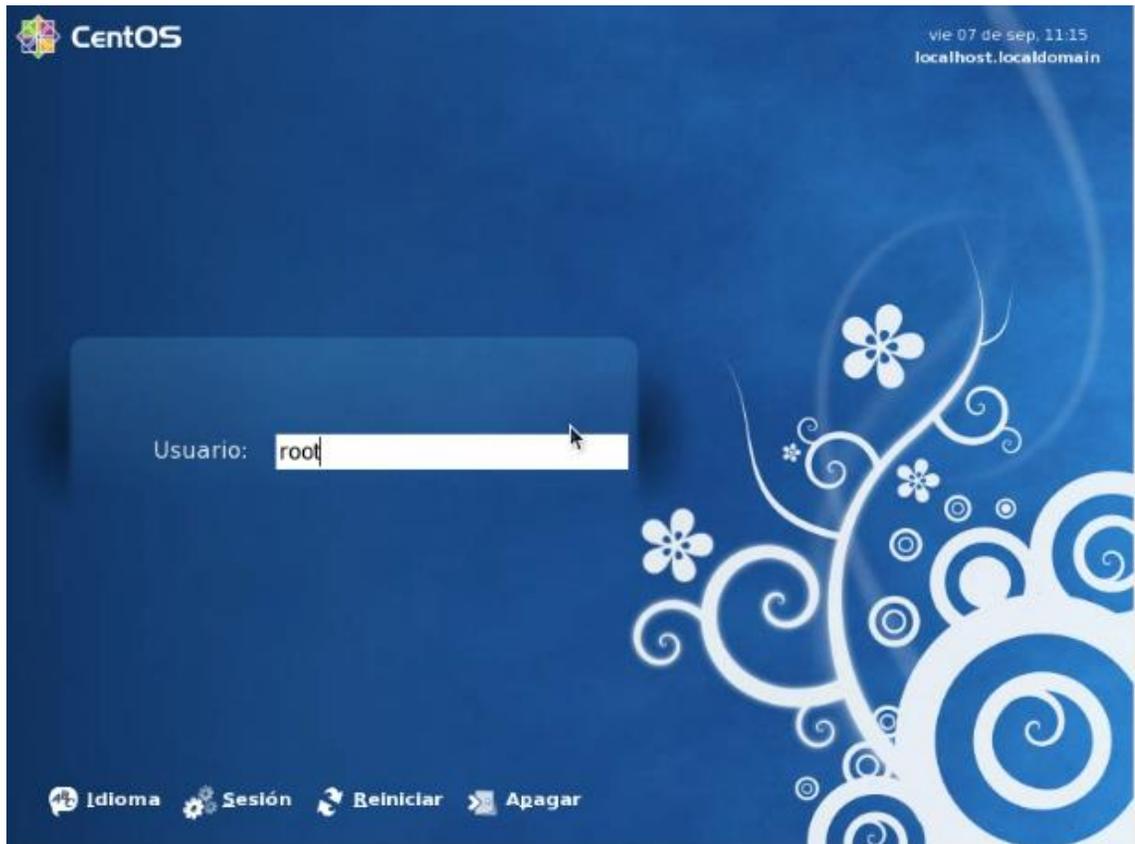
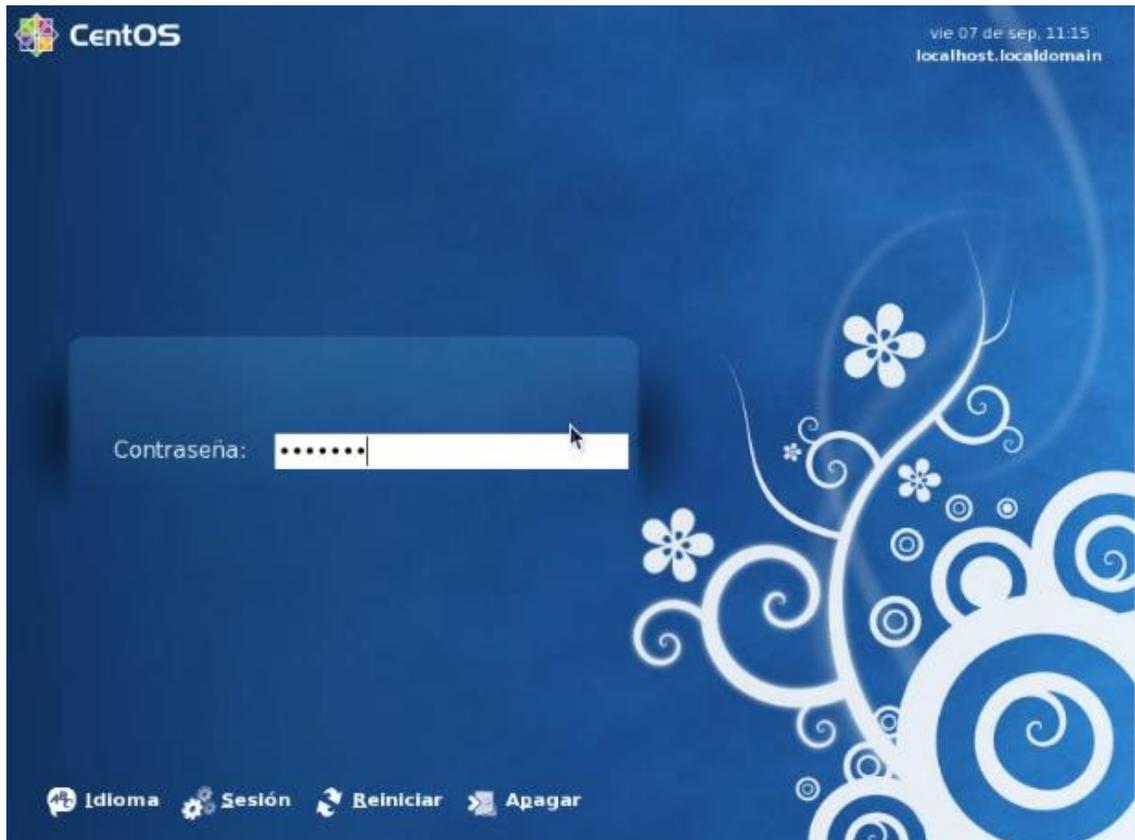
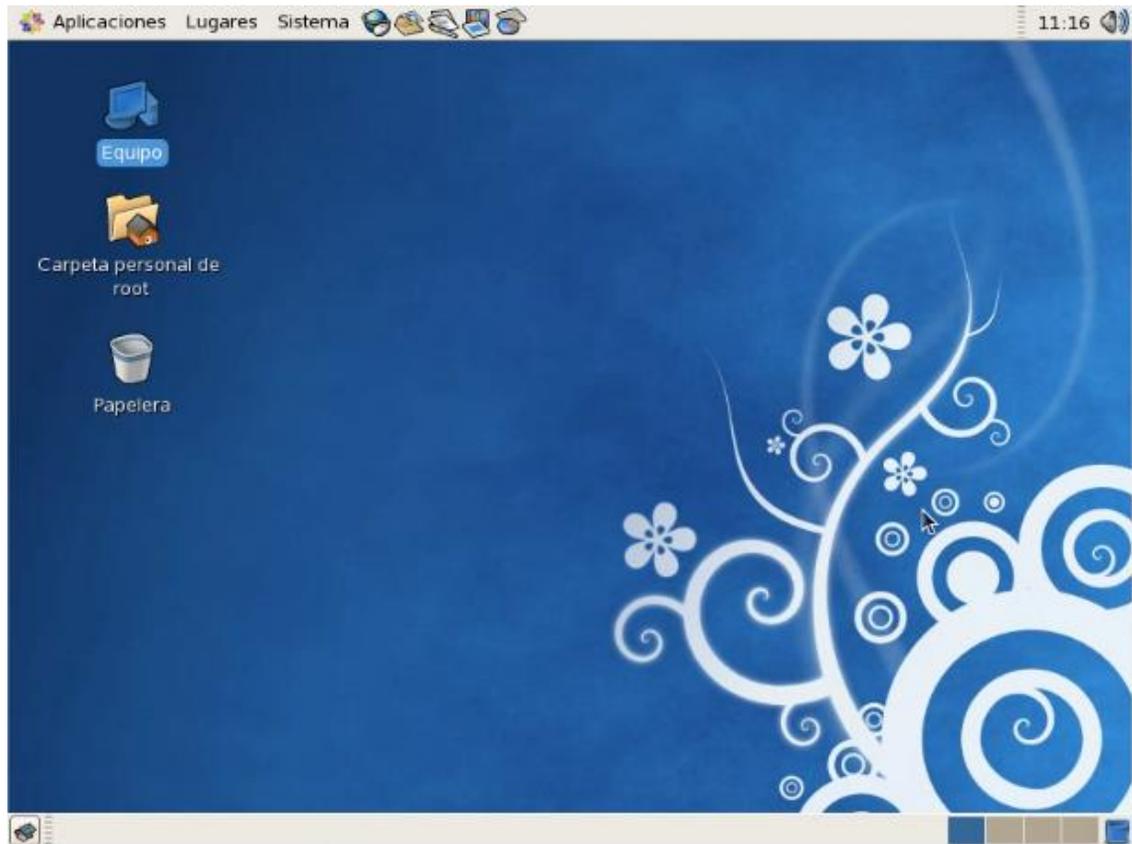


FIGURA 41. CONTRASEÑA A USUARIO ROOT.



La figura 42 muestra el entorno gráfico de la distribución CentOS 5, en donde ya se puede acceder a las distintas aplicaciones que tiene este sistema operativo.

FIGURA 42. ENTORNO GRÁFICO CENTOS 5.6



4.2 COMANDOS BÁSICOS DE LINUX

La ayuda de los siguientes comandos se puede encontrar en el del terminal digitando **man** comando a averiguar o comando a averiguar **--help**.

4.2.1 Manipular ficheros y directorios

4.2.1.1 Comando cd

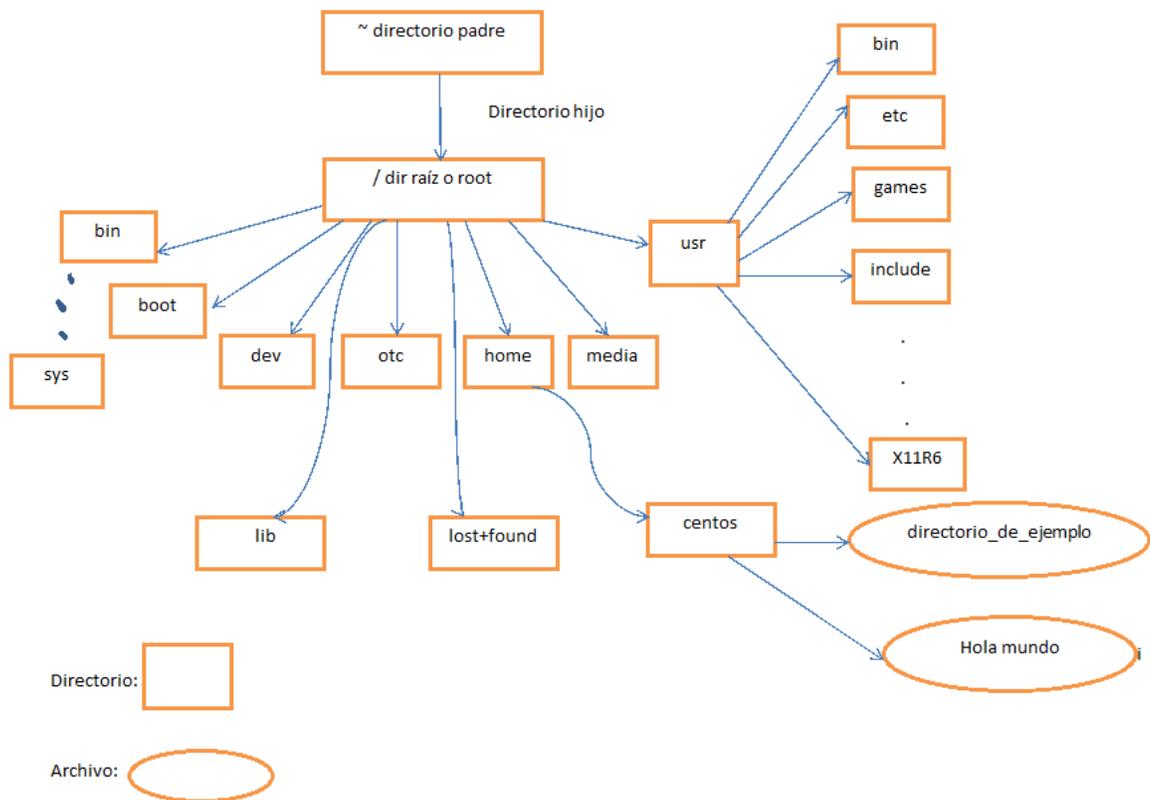
Este comando sirve para explorar o entrar en un directorio. El modelo a seguir este comando es:

```
cd /ruta del destino
```

```
[root@localhost ~]# cd /home
[root@localhost home]#
```

Todos los ficheros de Linux están guardados en una especie de cajas llamadas directorios. Un directorio padre puede contener ficheros y otros directorios llamados subdirectorios o directorios hijos, los que a su vez pueden tener más ficheros y más subdirectorios. La figura 43 muestra un árbol jerárquico, en el cual se desprenden algunos de los directorios y subdirectorios que hay en Linux.

FIGURA 43. EJEMPLO DE ÁRBOL JERÁRQUICO DE LINUX.



Para una mejor comprensión, en las figuras 44, 45 y 46, se muestran la ruta que se siguió en el ejemplo anterior, por medio de la interfaz gráfica de Linux.

FIGURA 44. RUTA POR MEDIO GRÁFICO HACIA EL DIRECTORIO HOME 1.

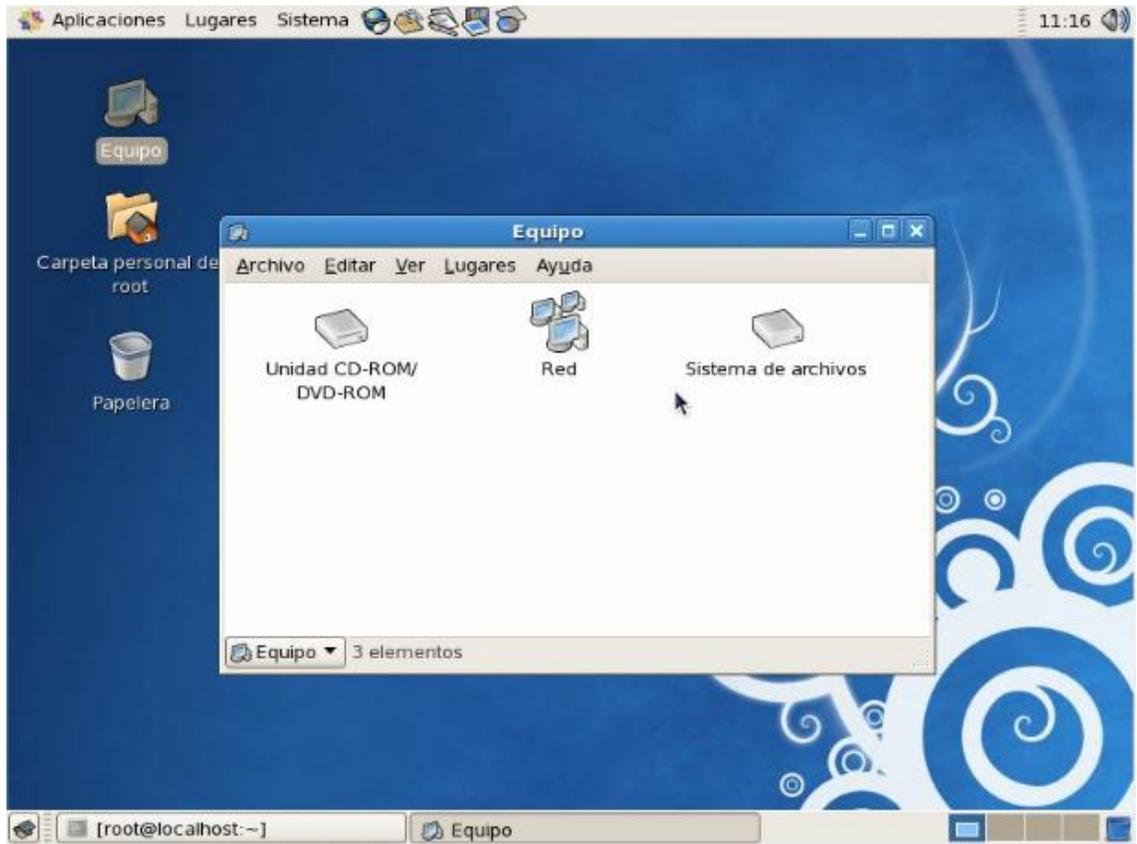


FIGURA 45. RUTA POR MEDIO GRÁFICO HACIA EL DIRECTORIO HOME 2.

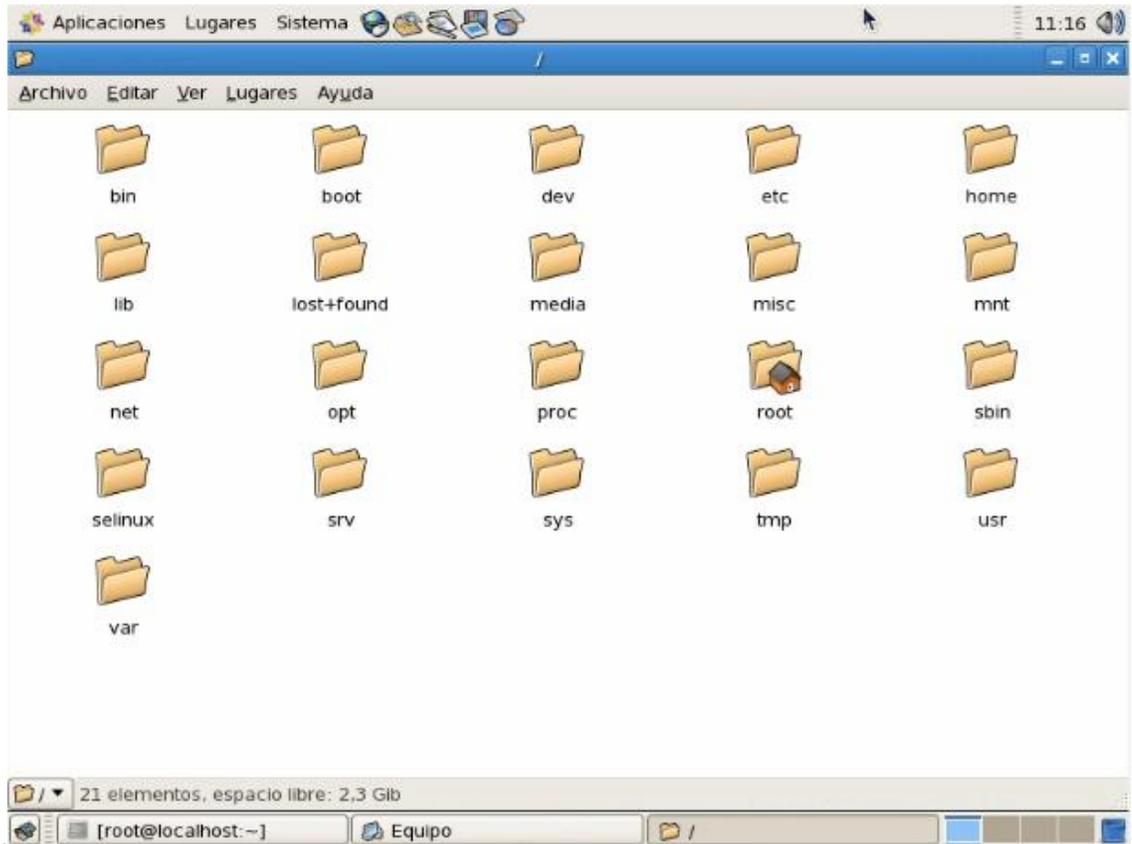
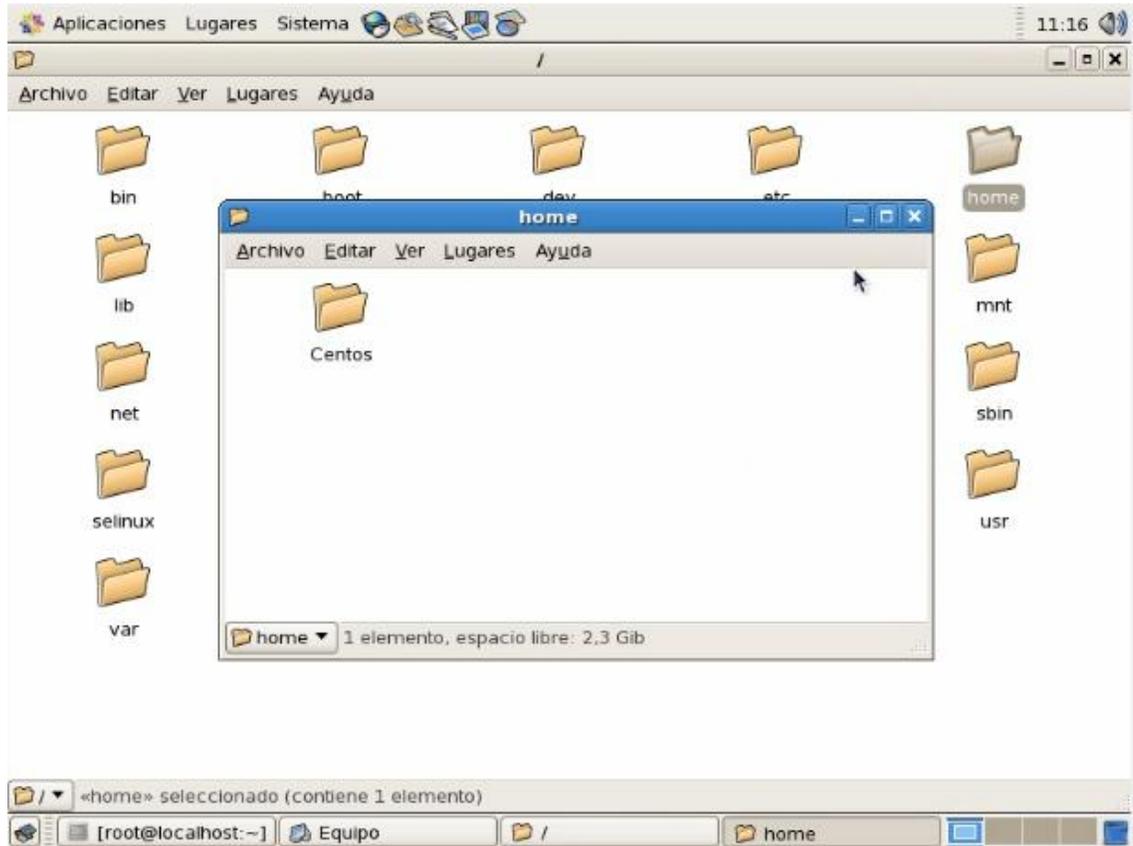


FIGURA 46. RUTA POR MEDIO GRÁFICO HACIA EL DIRECTORIO HOME 3.



Si el usuario se sabe la ruta completa desde cualquier directorio origen hacia un directorio o archivo destino, se digita el siguiente comando En este caso el origen es el directorio *root* y el directorio destino es Centos.

```
[root@localhost ~]# cd /home/Centos  
[root@localhost Centos]#
```

O por el contrario, si no se recuerda la ruta a seguir, se pulsa la tecla *tab* para ver los distintos directorios hijos o archivos existentes en el directorio que se encuentra actualmente para seguir con la ruta final. Un ejemplo de esto se encuentra en la figura 47.

FIGURA 47. FUNCIÓN TECLA TAB.

```
[root@localhost ~]# cd /root/
anaconda-ks.cfg      .dmrc                .gtkrc-1.2-gnome2    .recently-used
.bash_history        .eggccups/           .ICEauthority         .recently-used.xbel
.bash_logout         .gconf/              install.log           .redhat/
.bash_profile        .gconfd/             install.log.syslog    .tcshrc
.bashrc              .gnome/              .metacity/           .thumbnails/
.config/             .gnome2/             .mozilla/            .Trash/
.cshrc               .gnome2_private/    .nautilus/           .xsession-errors
Desktop/             .gstreamer-0.10/    prueba/
[root@localhost ~]# cd /root/
```

Existen símbolos que ayudan a representar posiciones en el sistema de archivos, el punto (.) y el doble punto (..). El punto, simboliza que se está en el directorio de trabajo actual, y el doble punto, simboliza al directorio padre.

Si se quiere llegar al directorio Centos desde el directorio *home*, lo único que se hace es poner el punto, *slash* y posteriormente el nombre del directorio. Para una mejor comprensión, ver el siguiente ejemplo.

Para regresar al directorio anterior o directorio padre, se teclea *cd* seguido de los dos puntos.

```
[root@localhost ~]# cd /home
[root@localhost home]# cd ./Centos
[root@localhost Centos]# cd ..
[root@localhost home]# cd ..
[root@localhost /]#
```

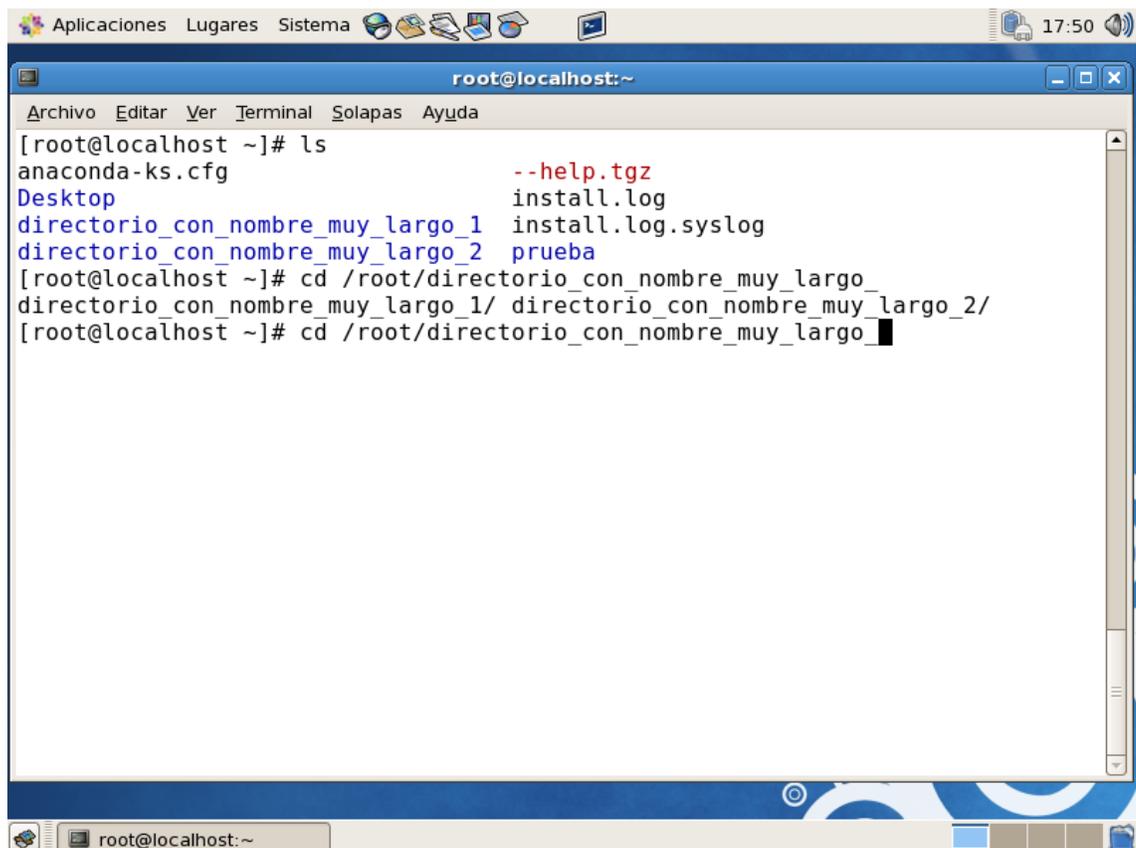
El *slash* significa que se ha llegado al directorio *root*: Dependiendo del usuario en que inició sesión muestra, o el directorio *root* o el directorio *home*, cuando se hace ese tipo de exploraciones por los directorios. Cabe resaltar que el tutorial se hizo con la sesión *root* o administrador. Para regresar al directorio “administrador” de inmediato, así se esté en cualquier directorio, se pulsa el comando de la figura XXXXX.

```
[root@localhost ~]# cd /home/Centos
[root@localhost Centos]# cd
[root@localhost ~]#
```

En la figura 48, se quiere explorar el directorio de nombre *directorio_con_nombre_muy_largo_1*. Una manera rápida de escribir ese nombre tan largo es teclear las primeras letras del nombre del directorio, en este caso, si se teclea *dire*, y posterior a eso se oprime la tecla *tab*, la aplicación sabe que se

quiere llegar a dicho directorio, y por lo tanto llena el nombre del directorio completo. En este caso, hay dos directorios con nombres parecidos, la diferencia es el número, que uno tiene el número 1 y el otro el número 2. Entonces, al oprimir la tecla `tab`, la aplicación llena el nombre hasta antes de llegar a los números y muestra las dos diferencias para que el usuario escoja el directorio con el número 1. Sólo es teclear el número 1 y oprimir `enter` para explorar el directorio destino.

FIGURA 48. EJEMPLO DE ESCRITURA PARA DIRECTORIOS CON NOMBRES LARGOS.

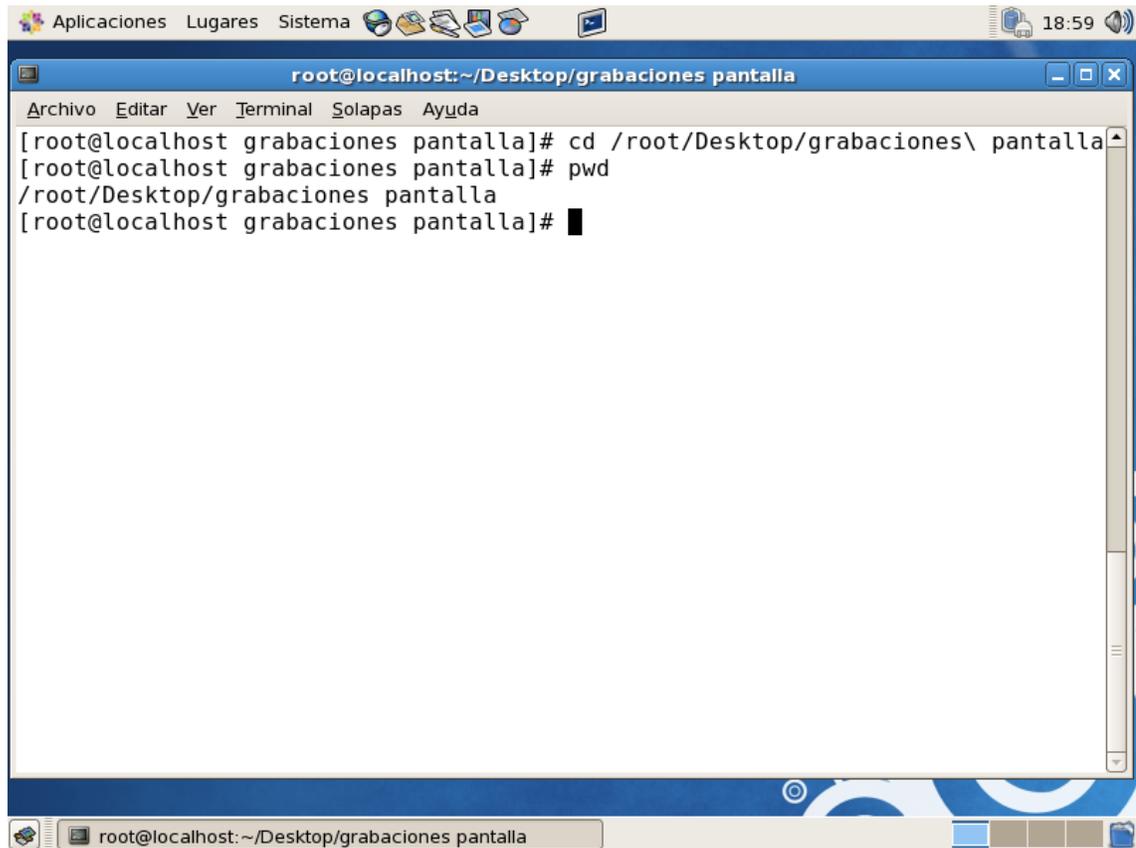


```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@localhost ~]# ls  
anaconda-ks.cfg          --help.tgz  
Desktop                  install.log  
directorio_con_nombre_muy_largo_1  install.log.syslog  
directorio_con_nombre_muy_largo_2  prueba  
[root@localhost ~]# cd /root/directorio_con_nombre_muy_largo_  
directorio_con_nombre_muy_largo_1/ directorio_con_nombre_muy_largo_2/  
[root@localhost ~]# cd /root/directorio_con_nombre_muy_largo_
```

4.2.1.2 Comando `pwd`

Este comando imprime la ruta completa del directorio en el cual se encuentra actualmente el usuario. En la figura 49, se muestra un ejemplo del comando `pwd`.

FIGURA 49. EJEMPLO COMANDO PWD.



4.2.1.3 Comando mkdir

Este comando sirve para crear un directorio o múltiples directorios. El modo de empleo de este comando es:

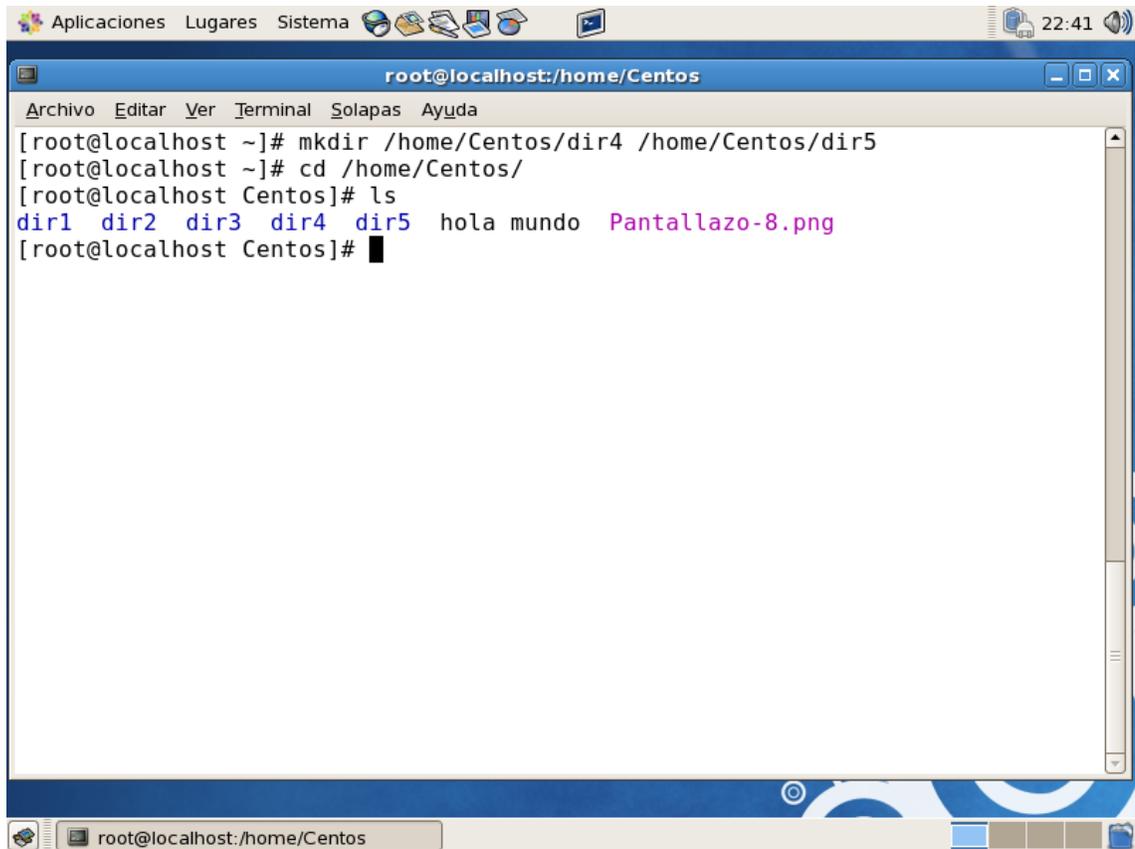
```
mkdir [-opciones] directorios
```

A continuación, se muestra un ejemplo del comando mkdir para crear un directorio, y posteriormente verificar que ya fue creado.

```
[root@localhost Centos]# ls
[root@localhost Centos]# mkdir directorio_de_ejemplo
[root@localhost Centos]# ls
directorio_de_ejemplo
[root@localhost Centos]#
```

Para crear múltiples directorios se procede a escribir de la forma de la figura 50.

FIGURA 50. CREAR MÚLTIPLES DIRECTORIOS CON MKDIR.

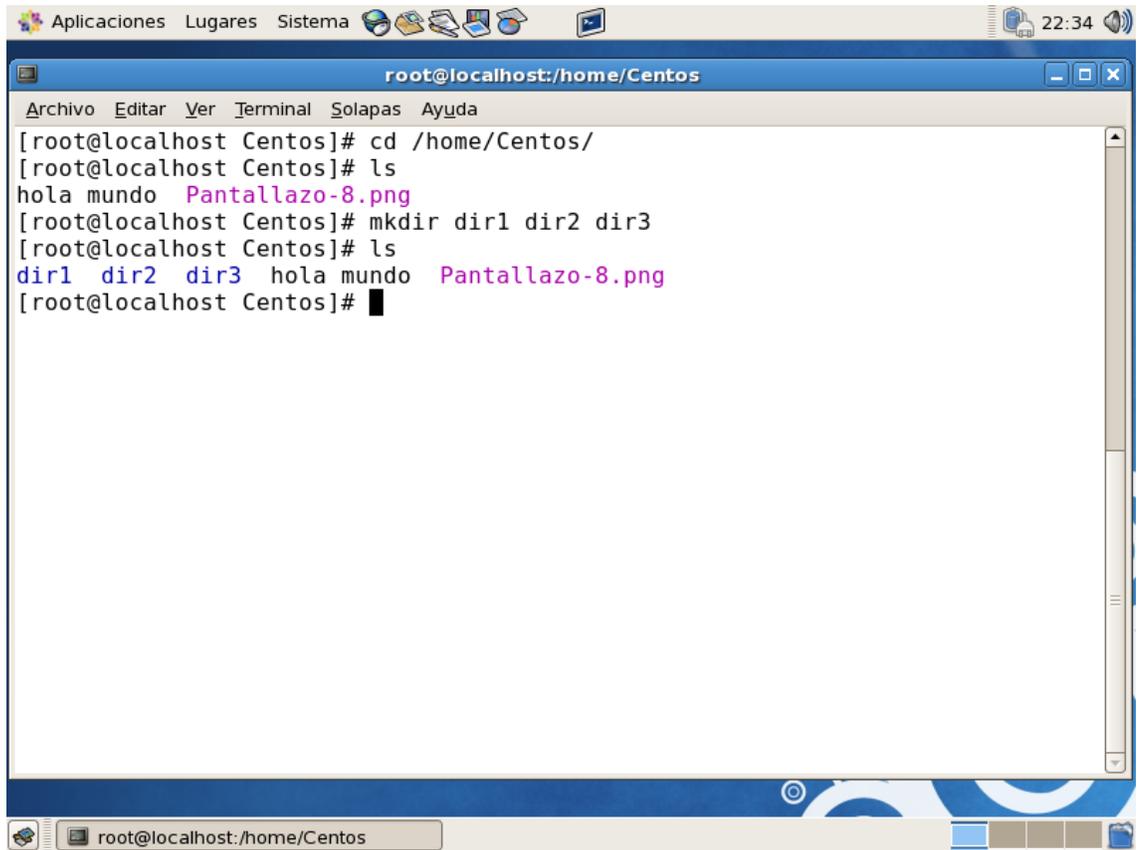


```
root@localhost:/home/Centos
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# mkdir /home/Centos/dir4 /home/Centos/dir5
[root@localhost ~]# cd /home/Centos/
[root@localhost Centos]# ls
dir1 dir2 dir3 dir4 dir5 hola mundo Pantallazo-8.png
[root@localhost Centos]#
```

Como se puede apreciar en la figura anterior, se pueden crear nuevos directorios desde otros directorios, no necesariamente se tiene que estar en el directorio padre para crear directorios hijos.

También se pueden crear nuevos directorios desde el directorio padre para crear directorios hijos como se puede ver en la figura 51.

FIGURA 51. CREAR DIRECTORIOS HIJOS DESDE EL DIRECTORIO PADRE CON MKDIR.



Usualmente, los comandos vienen acompañados de una o más opciones que modifican su comportamiento, agregándole más características a lo que se quiere hacer. Se pueden usar estas opciones de la forma larga o de la corta.

A continuación se muestra la tabla 1, la cual contienen algunas opciones del comando mkdir.

TABLA 1. OPCIONES DEL COMANDO MKDIR

-p, --parents	No hay error si existen, crea los directorios padres en caso necesario.
-v, --verbose	Muestra un mensaje por cada directorio creado.
--help	Muestra la ayuda del comando en pantalla.

4.2.1.4 Comando rmdir

Este comando es usado para borrar uno o varios directorios vacíos. El modo de empleo de este comando es:

```
rmdir [-opciones] directorios
```

Los ejemplos anteriormente mostrados con el comando mkdir son aplicables con el comando rmdir, sólo que cumplen la función contraria. En la figura 52, se muestra un ejemplo del comando rmdir para borrar un directorio.

FIGURA 52. EJEMPLO COMANDO RMDIR.

```
[root@localhost Centos]# ls
[root@localhost Centos]# mkdir directorio_de_ejemplo
[root@localhost Centos]# ls
directorio_de_ejemplo
[root@localhost Centos]# rmdir directorio_de_ejemplo
[root@localhost Centos]# ls
[root@localhost Centos]#
```

Como se dijo anteriormente, este comando es sólo aplicable para directorios vacíos. Para borrar un directorio con información dentro, se utiliza el comando rmdir -r seguido del nombre del directorio a borrar. Se debe tener en cuenta que este comando borrará completamente el directorio a borrar y sus hijos.

A continuación se muestra la tabla 2, la cual, contienen algunas opciones del comando rmdir.

TABLA 2.OPCIONES DEL COMANDO RMDIR

--ignore-fail-on-non-empty	No tiene en cuenta los fallos que se producen únicamente porque un directorio no está vacío.
-p, --parents	Elimina el directorio y sus antecesores.
-v, --verbose	Muestra un diagnóstico para cada directorio procesado.
--help	Muestra la ayuda del comando en pantalla.

4.2.1.5 Comando ls

Este comando se usa para listar el contenido del directorio en el cual se encuentra actualmente el usuario. El modo de empleo de este comando es:

```
ls [-opciones] directorio a listar
```

Se puede ver un ejemplo de este comando en la figura 53.

FIGURA 53. EJEMPLO COMANDO LS.

```
[root@localhost /]# ls
bin  dev  home  lost+found  misc  net  proc  sbin  srv  tmp  var
boot  etc  lib  media  mnt  opt  root  selinux  sys  usr
[root@localhost /]#
```

Si se quiere ver un listado más ordenado e informativo, se usa el parámetro `-l` acompañado anteriormente del comando `ls`, como se puede apreciar en la figura 54.

FIGURA 54. COMANDO LS.

```
[root@localhost ~]# ls -l
total 72
-rw----- 1 root root 1211 sep 6 16:08 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 sep 6 22:54 Desktop
-rw-r--r-- 1 root root 31842 sep 6 16:07 install.log
-rw-r--r-- 1 root root 4575 sep 6 15:55 install.log.syslog
drwxr-xr-x 2 root root 4096 sep 6 22:48 prueba
[root@localhost ~]#
```

La primera columna indica los permisos de acceso de cada archivo o directorio, el signo (-) significa que es un archivo, y la letra (d) significa que es un directorio.

La segunda columna indica el dueño del archivo, en este caso todos son *root* debido a que no se han creado más usuarios. Existen 3 tipos de usuario: administrador o dueño, grupo de usuarios al cual pertenece y resto de usuarios del sistema. Más adelante se explica el significado de cada usuario en la sección permisos de acceso.

La tercera columna indica el nombre del grupo al cual pertenece el archivo.

La cuarta columna indica el tamaño del archivo.

La quinta columna indica la fecha y hora de la última modificación del archivo, y la última columna indica el nombre del archivo o directorio.

En la tabla 3, se muestran algunas opciones del comando ls.

TABLA 3.OPCIONES DEL COMANDO LS.

-a, --all	Muestra todos los archivos, incluidos los nombres que empiecen con punto (.), que normalmente no se muestran sin el uso de esta opción.
-B, --ignore-backups	No muestra las entradas que terminan con ~
-C	Muestra las entradas por columnas
-d, --directory	Muestra el directorio más no su contenido.
-l	Utiliza un formato de listado largo.
-m	Rellena el ancho con una lista de

	entradas separadas por comas
-o	Igual que el -l, sólo que no lista la información del grupo
-Q, --quote name	Encierra los nombres de las entradas entre comillas
-x	Muestra las entradas por líneas en vez de columnas.
-X	Ordena alfabéticamente por la extensión de la entrada
-1	Muestra un fichero por cada línea

4.2.1.6 Comando cp

Este comando sirve para copiar un archivo de un directorio a otro:

```
cp [-opciones] [ruta del archivo a copiar] [ruta del archivo destino]
```

o copiar múltiples archivos en un directorio:

```
cp [archivo1] [archivo2] [archivo3] [directorio a copiar]
```

```
[root@localhost ~]# cp /root/Desktop/Pantallazos/Pantallazo-8.png /home/Centos/
[root@localhost ~]#
```

La tabla 4, muestra algunas opciones del comando cp.

TABLA 4. OPCIONES DEL COMANDO CP.

-f, --force	Forza la copia. Si un archivo de destino existe, se sobrescribe.
-i, --interactive	Pregunta antes de sobrescribir.
-u, --update	Copia sólo cuando el archivo origen es más nuevo que el archivo destino, ó cuando falta el archivo de destino.
-v, --verbose	Detalla sobre lo que se está haciendo.
--help	Muestra la ayuda del comando en pantalla.

4.2.1.7 Comando mv

Este comando sirve para renombrar un archivo:

```
mv [-opciones] [archivo1] [archivo2]
```

O mover archivos y directorios en un directorio destino:

```
mv [archivo1] [archivo2] [directorio3] [directorio4] [directorio destino]
```

La tabla 5, muestra algunas opciones del comando mv.

TABLA 5. OPCIONES DEL COMANDO MV.

-f, --force	No pregunta antes de sobrescribir.
-i, --interactive	Pregunta antes de sobrescribir.
-u, --update	Mueve sólo cuando el archivo origen es más nuevo que el archivo destino, ó cuando falta el archivo de destino
-v, --verbose	Muestra mensajes informativos respecto a lo que se hizo.
--help	Muestra la ayuda del comando en pantalla.

4.2.1.8 Comando rm

Este comando sirve para eliminar archivos. El modelo a seguir este comando es:

```
rm [-opciones] [archivo1] [archivo2] [archivo3]
```

O eliminar directorios no vacíos (con información dentro):

```
rm -r directorio1 directorio2
```

En la figura 55, se muestra un ejemplo del comando rm.

FIGURA 55. EJEMPLO DEL COMANDO RM.

```
[root@localhost Centos]# ls -l
total 16
-rw-r--r-- 1 root root 23 sep  6 23:53 hola mundo
-rw-r--r-- 1 root root 15 sep  6 23:56 prueba_de_texto
[root@localhost Centos]# rm prueba_de_texto
rm: ¿borrar el fichero regular «prueba_de_texto»? (s/n) s
[root@localhost Centos]# ls -l
total 8
-rw-r--r-- 1 root root 23 sep  6 23:53 hola mundo
[root@localhost Centos]#
```

A continuación, la tabla 6, muestra algunas opciones del comando rm.

TABLA 6. OPCIONES DEL COMANDO RM.

-f, --force	Elimina el archivo sin pedir confirmación.
-i, --interactive	Pregunta antes de eliminar cada archivo.
-r, -R, --recursive	Remueve recursivamente directorios y sus contenidos.
-v, --verbose	Detalla sobre lo que se está haciendo.
--help	Muestra la ayuda del comando en pantalla.

4.2.2 Visor de ficheros

4.2.2.1 Comando cat

Este comando es usado para concatenar archivos. La forma de uso de este comando es de la forma:

```
cat [-opciones] [archivos]
```

```
[root@localhost Pantallazos]# cat pantallazos\ para\ copiar\ en\ word
```

En la tabla 7, se muestran algunas opciones del comando cat.

TABLA 7. OPCIONES DEL COMANDO CAT.

-b, --number-nonblank	Enumera líneas de salida que no están en blanco.
-E, --show-ends	Muestra el símbolo \$ al final de cada línea.
-n, --number	Enumera todas las líneas de salida.
-T, --show-tabs	Muestra caracteres de tabulación como ^I.
--help	Muestra la ayuda del comando en pantalla.

4.2.2.2 Comando tail

Este comando muestra las últimas 10 líneas de cada fichero.

```
tail [-opción] [Fichero]
```

En la tabla 8, se muestran algunas opciones del comando tail.

TABLA 8. OPCIONES DEL COMANDO TAIL.

-c, --bytes=N	Muestra los últimos N <i>bytes</i> .
-n, --lines=N	Muestra las últimas N líneas, en vez de las últimas 10.
-s, --sleep-interval=S	Con -f, duerme por aproximadamente S segundos (por defecto 1) entre iteraciones.
-q, --quiet, --silent	Nunca muestra las cabeceras del nombre del fichero dado.
--help	Muestra la ayuda del comando en pantalla.

4.2.3 Propiedades de los ficheros

4.2.3.1 Comando stat

Este comando muestra los atributos de los archivos y directorios. La forma de uso de este comando es de la siguiente forma:

```
stat [-opciones] [archivo]
```

En el ejemplo anterior se muestran características tales como: El nombre del archivo, su peso en *bytes*, su peso en bloques, sus permisos de acceso, etc.

Algunas opciones de este comando se muestran en la tabla 9.

TABLA 9. OPCIONES DEL COMANDO STAT.

-t	Muestra la información en una sola línea.
--help	Muestra la ayuda del comando en pantalla.
-f	Muestra información sobre el sistema de archivos donde los archivos dados están localizados, en vez de información sobre los archivos mismos.

4.2.3.2 Comando wc

Este comando muestra un contador de palabras, líneas y *bytes* en un archivo de texto. El modo de uso es de la siguiente forma:

```
wc [-opciones] [archivos]
```

En la tabla 10, se pueden apreciar algunas opciones de este comando.

TABLA 10. OPCIONES DEL COMANDO WC.

-c	Muestra el número de <i>bytes</i> solamente.
-m	Muestra el número de caracteres solamente.
-w	Muestra el número de palabras solamente.
--help	Muestra la ayuda y finaliza.
--version	Muestra la versión y finaliza.

4.2.3.3 Permisos de acceso

Existen 3 tipos de usuario, el administrador o dueño, el grupo de usuarios al cual pertenece y el resto de usuarios del sistema. Todos estos usuarios a su vez, poseen archivos, los cuales tienen permisos de acceso y no todos los usuarios pueden hacer determinadas tareas con esos ficheros ya que, lo que se busca con tener varios usuarios es mantener la privacidad y seguridad para cada uno. Existen tres permisos de acceso, los cuales son: Ejecución (x), escritura (w) y lectura (r). Se puede ver en la figura 56 un ejemplo de estos permisos.

FIGURA 56. PERMISOS DE ACCESO.

```

Dueño
Grupo de usuarios al cual pertenece
Resto de usuarios del sistema
-rw-r--r-- 1 root root 1211 sep 6 16:08 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 sep 6 22:54 Desktop
-rw-r--r-- 1 root root 31842 sep 6 16:07 install.log
-rw-r--r-- 1 root root 4575 sep 6 15:55 install.log.syslog
drwxr-xr-x 2 root root 4096 sep 6 22:48 prueba

```

Los permisos de acceso están en la primera columna de la figura anterior. El primer caracter significa, si es el signo -, se trata de un archivo y si es la letra d, es un directorio. Las tres letras siguientes significan los permisos que tiene el administrador o dueño. El conjunto de letras siguientes pertenecen a los permisos que tiene el usuario llamado, grupo de usuarios al cual pertenece, y el último conjunto son el resto de los usuarios del sistema.

Por ejemplo, Anaconda-ks.cfg es un archivo, posee permisos de lectura y escritura para el dueño, para el grupo no tiene ningún tipo de permiso, al igual que para el resto de usuarios del sistema.

Otro ejemplo es prueba, el cual, es un directorio, posee permisos de lectura, escritura y ejecución por parte del dueño; para el grupo, posee permisos de lectura y ejecución, al igual que para el resto de usuarios del sistema.

4.2.3.4 Comando chmod

Para modificar los permisos de un archivo o directorio anteriormente mencionados, se utiliza el comando chmod. El modo de empleo de este comando es:

```
chmod [usuario] [+/-] [permiso a dar] [archivo]
```

Cabe resaltar que el único usuario que puede modificar estos permisos es el usuario *root*

El **[usuario]** contiene la letra u si es el administrador del sistema, la letra g para el grupo y la letra o para el resto de usuarios.

Los signos **[+/-]** significan permitir un permiso o no permitirlo respectivamente.

[Permiso a dar], significa si el permiso que se va a dar es de lectura (r), escritura (w) o ejecución (x).

[Archivo] quiere decir, el archivo a dar permiso.

A continuación, se muestran algunos ejemplos del comando chmod.

```
[root@localhost Pantallazos]# chmod g+x Pantallazo-9.png  
[root@localhost Pantallazos]# chmod o-x Pantallazo-10.png  
[root@localhost Pantallazos]# chmod ug-w Pantallazo.png
```

El primer ejemplo, permite la ejecución del archivo Pantallazo-9.png sólo para el dueño o administrador del sistema.

El segundo ejemplo, permite la ejecución del archivo Pantallazo-10.png sólo al grupo al que pertenece.

El tercer ejemplo, permite la escritura del archivo Pantallazo.png tanto al grupo como para el resto de usuarios del sistema.

4.2.3.5 Comando du

El comando du (*disk usage*), mide el espacio del disco ocupado por los ficheros o directorios.

```
du [-opciones] [archivos][directorios]
```

En la tabla 11, se pueden apreciar algunas opciones de este comando.

TABLA 11. OPCIONES DEL COMANDO DU.

-b, --bytes	Muestra el tamaño en <i>bytes</i> .
-k	Muestra el tamaño en <i>kilobytes</i> .
-m	Muestra el tamaño en <i>megabytes</i> .
--help	Muestra la ayuda y finaliza.
--s	Muestra sólo el tamaño total.

4.2.4 Localización de ficheros

4.2.4.1 Comando find

Este comando sirve para buscar ficheros en una jerarquía de directorios.

```
find [directorios] [expresiones]
```

En la tabla 12, se pueden apreciar algunas opciones de este comando.

TABLA 12. OPCIONES DEL COMANDO FIND.

-daystart	Mide los tiempos (para -amin,-atime,-cmin,-ctime,-mmin y -mtime) desde hoy en lugar de 24 horas atrás.
-depth	Procesa cada contenido de los directorios antes que el directorio mismo.
-maxdepth levels	Desciende en la mayoría de los niveles(un entero no negativo) niveles de directorios por debajo de los argumentos de la línea de comando.
--help	Muestra la ayuda y finaliza.

4.2.4.2 Comando locate

Este comando sirve para encontrar ficheros por el nombre. *Locate* no busca realmente desde el disco duro del sistema, sino que lo hace desde un archivo que tiene una lista de todos los ficheros que existen en GNU/Linux.

```
locate [opción] [Patrón]
```

4.3 INSTALACIÓN OPENFLOW

Asumiendo que ya se ha instalado el paquete básico de la NetFPGA, se procede a instalar el paquete de Openflow.

Paso 1: Como primera medida, el usuario debe haber iniciado sesión en *root* o administrador.

Paso 2: Configurar una cuenta de usuario para la instalación.

```
/usr/sbin/adduser openflow
```

Paso 3: Configurar una contraseña para la cuenta.

```
passwd openflow
```

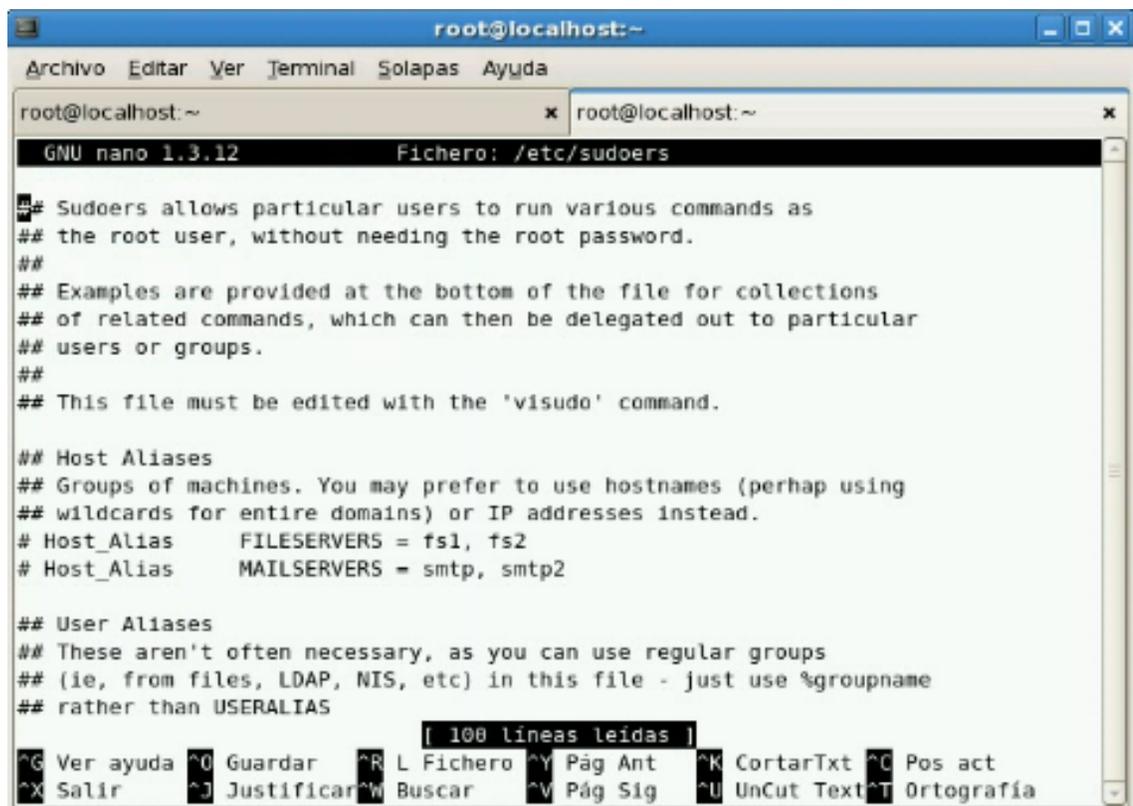
En este paso se agrega una contraseña para la cuenta y se oprime la tecla *enter* para finalizar. Luego, piden reconfirmar la contraseña.

Paso 4: Se le agregan privilegios al super usuario para esta cuenta.

```
nano /etc/sudoers
```

Al momento de digitar el comando anterior, aparece un editor como el de la figura 57.

FIGURA 57. PRIVILEGIOS SUPER USUARIO.



Paso 5: Agregar el siguiente comando al final de la línea del editor.

```
openflow ALL=(ALL) NOPASSWD:ALL
```

Una vez copiada la línea anterior, se guarda el archivo oprimiendo las teclas *control* + *o*, luego se confirma oprimiendo la tecla *enter*. Para salir del editor se oprimen las teclas *control* + *x*.

Paso 6: Instalar el paquete RPMForge con los siguientes comandos. Cabe resaltar que para este paso, toca estar conectado a internet para poder descargar los paquetes.

```
wget http://apt.sw.be/redhat/el5/en/i386/rpmforge/RPMS/rpmforge-  
release-0.3.6-1.el5.rf.i386.rpm  
sudo rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt  
sudo rpm -Uhv rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

Paso 7: Instalar las siguientes librerías para que el paquete de Openflow se pueda instalar de forma correcta.

```
sudo yum -y install git automake pkgconfig libtool gcc
```

Dependiendo del computador se va a pedir que se instalen más o menos librerías. En caso tal que aparezcan más, se escribe `sudo yum -y install [nombre de la librería que pide que se instale]`.

Paso 8: Instalar Autoconf 2.60 o superior. Se debe instalar desde el código fuente para compilar desde el repositorio Git.

```
wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.63.tar.gz  
tar xvzf autoconf-2.63.tar.gz  
cd autoconf-2.63  
./configure --prefix=/usr  
make  
sudo make install  
cd
```

Paso 9: Descargar el repositorio Git con el código e historia del Openflow. La última ejecución de la NetFPGA reside temporalmente en una rama separada, de modo que toca cambiar la rama de la siguiente manera.

```
cd --
cd /root/
git clone git://openflow.org/openflow.git
cd openflow
git checkout -b 1.0.0-netfpga origin/devel/tyabe/1.0.0-netfpga
./boot.sh
```

Paso 10: Descargar el bitfile de la NetFPGA y extraerlo en el directorio de Openflow.

```
cd hw-lib/nf2
wget
http://openflow.org/downloads/netfpga/openflow\_switch.bit.100\_3.tar.gz
tar xfvz openflow_switch.bit.100_3.tar.gz
```

Paso 11: Compilar los *switches* Openflow *user-space* con el soporte NetFPGA en el directorio de Openflow.

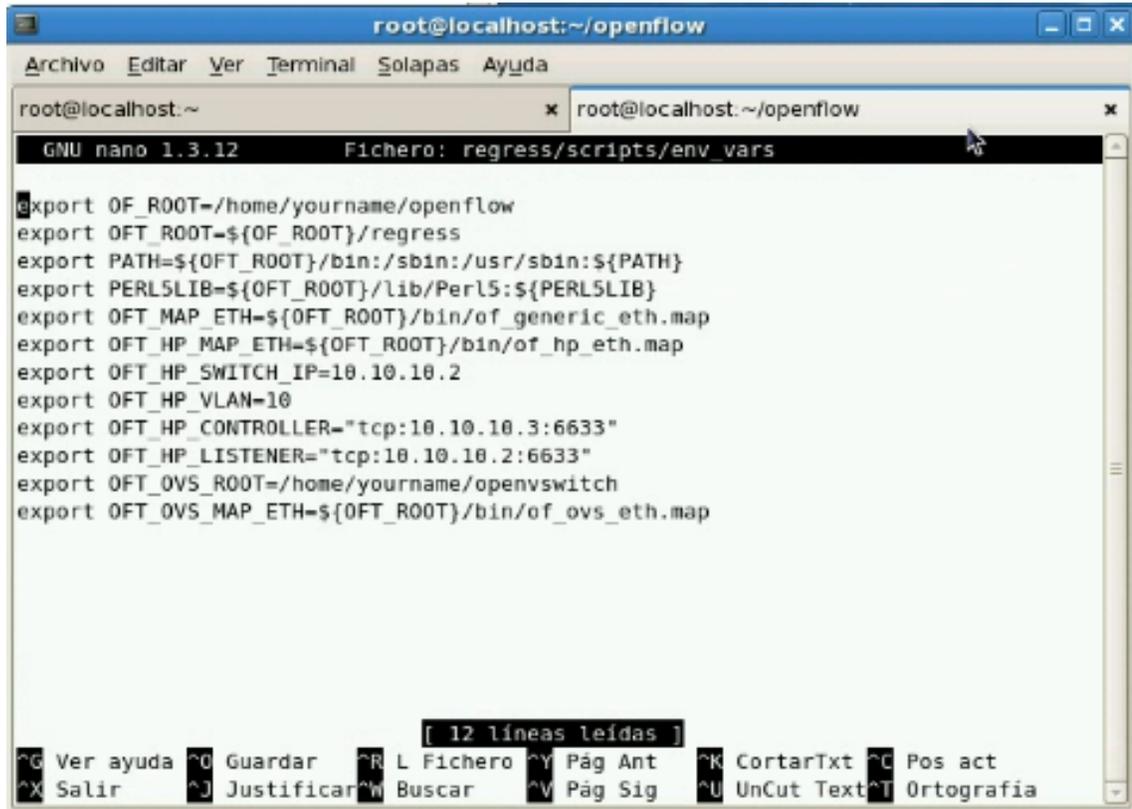
```
cd ..
cd ..
./configure --enable-hw-lib=nf2
make
sudo make install
```

Paso 12: Desde el directorio de Openflow copiar el siguiente comando.

```
nano openflow/regress/scripts/env_vars
```

Luego de escribir el comando anterior, aparecerá un editor de texto como el de la figura 58.

FIGURA 58. FICHERO ENV_VARS.



```
root@localhost:~/openflow
Archivo Editar Ver Terminal Solapas Ayuda
root@localhost:~ x root@localhost:~/openflow
GNU nano 1.3.12 Fichero: regress/scripts/env_vars

export OF_ROOT=/home/yourname/openflow
export OF_ROOT=${OF_ROOT}/regress
export PATH=${OFT_ROOT}/bin:/sbin:/usr/sbin:${PATH}
export PERLSLIB=${OFT_ROOT}/lib/Perl5:${PERLSLIB}
export OFT_MAP_ETH=${OFT_ROOT}/bin/of_generic_eth.map
export OFT_HP_MAP_ETH=${OFT_ROOT}/bin/of_hp_eth.map
export OFT_HP_SWITCH_IP=10.10.10.2
export OFT_HP_VLAN=10
export OFT_HP_CONTROLLER="tcp:10.10.10.3:6633"
export OFT_HP_LISTENER="tcp:10.10.10.2:6633"
export OFT_OVS_ROOT=/home/yourname/openvswitch
export OFT_OVS_MAP_ETH=${OFT_ROOT}/bin/of_ovs_eth.map

[ 12 líneas leídas ]
^G Ver ayuda ^O Guardar ^R L Fichero ^Y Pág Ant ^K CortarTxt ^C Pos act
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U UnCut Text ^T Ortografia
```

Paso 13: Encontrar la siguiente línea.

```
export OF_ROOT=/home/yourname/openflow
```

Modificar la línea anterior por:

```
export OF_ROOT=/root/openflow
```

Paso 14: También, encontrar la siguiente línea.

```
export PERL5LIB=${OFT_ROOT}/lib/Perl5:${PERL5LIB}
```

Modificar la línea anterior por:

```
export PERL5LIB=${OFT_ROOT}/lib/Perl5
```

Una vez copiada la línea anterior, se guarda el archivo oprimiendo las teclas *control* + *o*, luego se confirma oprimiendo la tecla *enter*. Para salir del editor se oprimen las teclas *control* + *x*.

Paso 15: Se apaga el demonio avahi con los siguientes comandos.

```
sudo /sbin/chkconfig avahi-daemon off  
sudo /etc/rc.d/init.d/avahi-daemon stop
```

Paso 16: Se deshabilita IPV6 con el siguiente comando.

```
sudo /sbin/chkconfig ip6tables off
```

Paso 17: Ahora, se crea y edita el archivo disable-ipv6.conf.

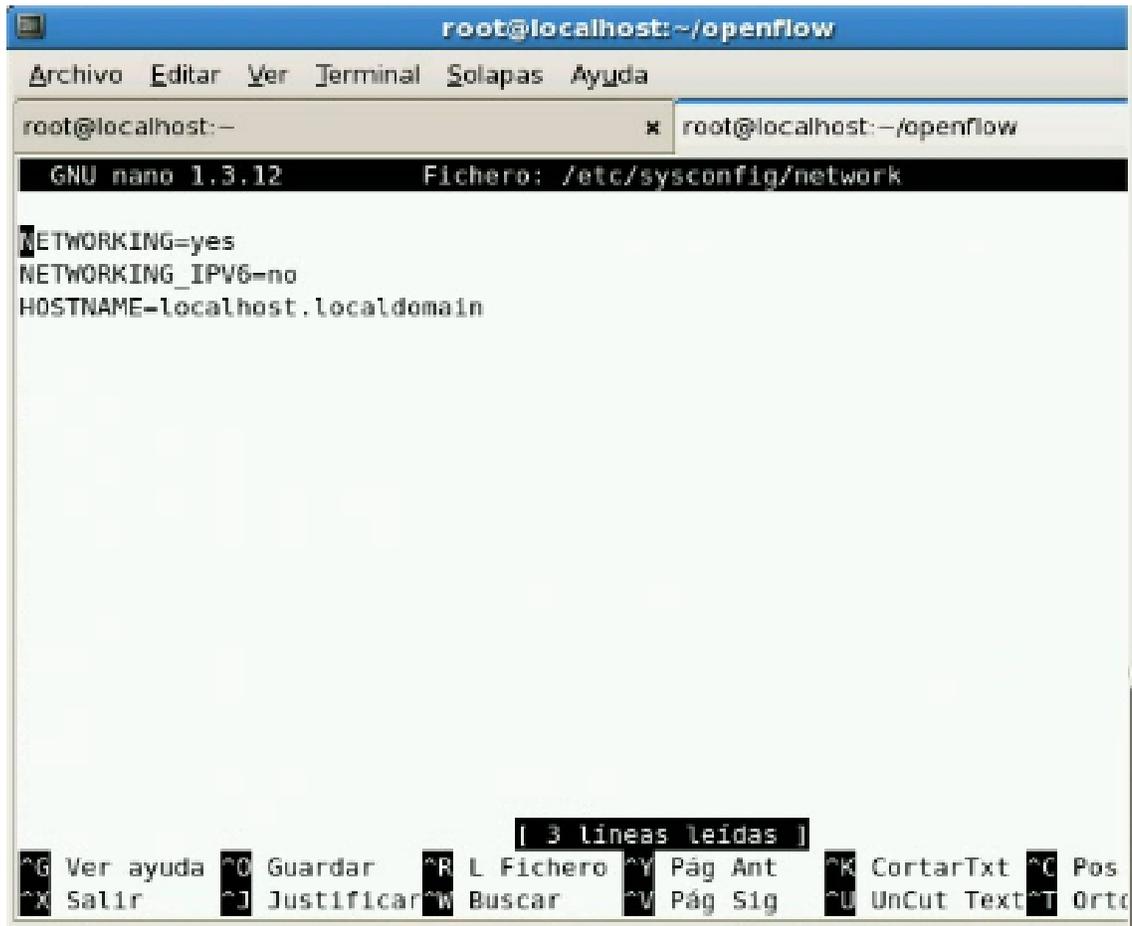
```
echo 'options ipv6 disable=1' > /etc/modprobe.d/disable-ipv6.conf
```

Paso 18: Se edita el fichero de red.

```
nano /etc/sysconfig/network
```

Luego del paso anterior aparece un editor de texto como el de la figura 59.

FIGURA 59. FICHERO DE RED.



```
root@localhost:~/openflow
Archivo Editar Ver Terminal Solapas Ayuda
root@localhost:~ root@localhost:~/openflow
GNU nano 1.3.12 Fichero: /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=localhost.localdomain
[ 3 líneas leídas ]
^G Ver ayuda ^O Guardar ^R L Fichero ^Y Pág Ant ^K CortarTxt ^C Pos
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U UnCut Text ^T Ortc
```

En la parte donde dice NETWORKING IPV6 debe decir no, de lo contrario, se debe cambiar. Luego de lo anterior, se guarda el archivo.

Paso 19: Ahora se deben cargar las variables de entorno con el siguiente comando.

```
source /regress/scripts/env_vars
```

Después del paso anterior, se supone que ya está instalado el Openflow como a su vez, compilado con la NetFPGA. Se aconseja que se reinicie el computador para empezar a hacer las distintas prácticas de laboratorio.

4.4 COMANDOS DEL OPENFLOW

Estos comandos se pueden encontrar en los manuales instalados en los paquetes del Openflow, digitando man seguido del comando Openflow a investigar.

4.4.1 Comando dpctl

Dpctl- Administrador de datapaths Openflow.

SINOPSIS

dpctl [*options*] *command* [*switch*] [*args&...*]

DESCRIPCIÓN

El programa dpctl es una herramienta de línea de comandos para el monitoreo y la administración de los datapaths Openflow. Es capaz de mostrar el estado actual de un datapath, incluyendo características, configuración y tablas de entrada. Cuando se utiliza el módulo del kernel Openflow, dpctl se utiliza para agregar, eliminar, modificar y monitorear los datapaths.

La mayoría de los comandos dpctl toman un argumento que especifican el método para conectarse al switch Openflow. Los siguientes métodos de conexión son compatibles:

nl: *dp_idx*

El datapath local Netlink enumerado *dp_idx*. Esta forma requiere que el host local cuente con el módulo del kernel Openflow para Linux cargado.

ssl: *host[:port]*

El puerto SSL especificado (por defecto: 6633) en el host remoto dado. Las opciones **--private-key**, **--certificate**, y **--ca-cert** son obligatorias cuando esta forma se utiliza.

tcp: *host[:port]*

El puerto TCP especificado (por defecto: 6633) en el host remoto dado.

unix: *file*

El socket del servidor de dominio Unix. El nombre de tal socket es *file*.

COMANDOS

Con el programa `dpctl`, los datapaths que se ejecutan en el núcleo se pueden crear, eliminar y modificar. Una sola máquina puede alojar hasta 32 datapaths (numerados del 0 al 31). En la mayoría de las situaciones, una máquina aloja sólo un datapath.

Un nuevo datapath creado no está asociado con cualquiera de los dispositivos de red del host, por lo tanto no procesa el tráfico entrante. Para interceptar y procesar el tráfico en un dispositivo de red dado, el dispositivo de red debe ser añadido explícitamente a un datapath a través del comando `addif`.

Los siguientes comandos manejan datapaths locales:

adddp nl:*dp_idx*

Crea datapaths enumerados `dp_idx` en el host local. Lo anterior fallará si `dp_idx` no está en el rango de 0 a 31, o si el datapath con ese número ya existe en el host.

deldp nl:*dp_idx*

Borra el datapath `dp_idx` en el host local. `dp_idx` debe ser un datapath existente. Todos los dispositivos de red de un datapath deben ser eliminados explícitamente antes que el datapath pueda ser borrado (Ver el comando `delif`).

addif nl:*dp_idx netdev...*

Agrega cada `netdev` a la lista de monitores del datapath `dp_idx` de los dispositivos de red, donde `dp_idx` es el ID de un camino de datos existente y `netdev` es el nombre de uno de los dispositivos de red del host, por ejemplo, `eth0`. Una vez que un dispositivo de red ha sido añadido a un camino de datos, la trayectoria de datos tiene la propiedad completa de tráfico del dispositivo de red y el dispositivo de red aparece silencioso para el resto del sistema.

delif nl:*dp_idx netdev...*

Elimina cada `netdev` de la lista de los dispositivos de red del datapath `dp_idx`. Removes each `netdev` from the list of network devices datapath `dp_idx` monitors.

get-idx of:*dev*

Imprime el índice del datapath para el dispositivo OpenFlow `of_dev`.

Los siguientes comandos se pueden aplicar a los switches OpenFlow independientemente del método de conexión.

show switch

Muestra información en la consola sobre el switch datapath que incluye información sobre sus tablas de flujo y puertos.

status switch [key]

Imprime en la consola una serie de pares clave-valor (key-value pairs) que informan sobre el estado del switch. Si se especifica la clave *key*, sólo se imprimen los pares de clave y valor cuyos nombres comienzan con la palabra *key*. Si la clave se omite, se imprimirán todos los pares clave-valor.

(En la implementación de referencia OpenFlow el comando **status** es implementado en ofprotocol (8), no en el módulo del kernel, por lo que el método de conexión nl:dp_idx no debe ser utilizado con este comando. Especifique **-l** o **--listen** en la línea de comandos del **ofprotocol** y decir al dpctl utilizar el método de conexión allí especificado).

show-protostat switch

Imprime al protocolo Openflow la información estadística del switch.

(En la implementación de referencia OpenFlow, el comando **show-Protostat** es implementado en ofprotocol (8), no en el módulo del kernel, por lo que, la conexión del método nl:dp_idx no debe ser utilizado con este comando. Especifique **-l** o **--listen** en la línea de comandos del ofprotocol y decir al dpctl utilizar el método de conexión allí especificado).

dump-tables switch

Muestra en la consola estadísticas para cada una de las tablas de flujo utilizados por el switch datapath.

dump-ports switch [port number]

Muestra en la consola estadísticas para cada interfaz controlada por el switch. Si el número del puerto se especifica, muestra las estadísticas sólo para la interfaz correspondiente al número de puerto.

mod-port switch netdev action

Modifica las características de una interfaz supervisada por el switch. Netdev puede ser referido por su número de puerto asignado de OpenFlow o por el nombre del dispositivo, por ejemplo, eth0. La acción puede ser uno cualquiera de los siguientes:

Up

Activa la interfaz. Lo anterior es equivalente a **ifconfig up** en un sistema Unix.

Down

Desactiva la interfaz. Lo anterior es equivalente a **ifconfig down** en un sistema Unix.

Flood

Cuando una inundación o **flood** se especifica, el tráfico se enviará fuera de esta interfaz. Esta es la postura por defecto de los puertos supervisados.

Noflood

Cuando una acción de inundación o noflood se especifica, el tráfico no se enviará fuera de esta interfaz. Esto es especialmente útil para evitar bucles cuando un protocolo spanning tree o árbol de expansión no está en uso.

dump-flows *switch [flows]*

Muestra en la consola todos los flujos de entrada en las tablas de caminos del switch que coinciden con los flujos. Si el flujo se omite, todos los flujos, exceptuando los flujos de emergencia en los flujos del datapath son recuperados. Ver **flow syntax** más abajo para la sintaxis de los flujos.

desc *switch string*

Establece la descripción del switch(como el devuelto en ofp_desc_stats) de cadena(longitud máxima es DESC_STR_LEN).

dump-aggregate *switch [flows]*

Muestra en la consola estadísticas agregadas para los flujos en las tablas de los datapaths del switch que coinciden flujos. Si el flujo es omitido, las estadísticas se agregan a través de todos los flujos en las tablas de flujo del datapath. . Ver **flow syntax** más abajo para la sintaxis de los flujos.

add-flow *switch flow*

Agrega la entrada de flujo como la descrita por el flujo de las tablas de los datapaths del switch. La tabla de flujo está en el formato descrito en **flow syntax**.

add-flows *switch file*

Añade flujos como entradas de la forma como se describe en el archivo *file* a las tablas que contienen los datapaths del switch. Cada línea en el archivo es un flujo de entrada en el formato descrito en **flow syntax**.

mod-flows *switch flow*

Modifica las acciones en las entradas de los datapaths de las tablas del switch que coinciden el flujo. Cuando se invoca con la opción **-strict option**, los comodines no son tratados como activos para propósitos de coincidencia. Ver **flow syntax** más abajo para la sintaxis de los flujos.

del-flows *switch [flow]*

Borra las entradas de los datapaths de las tablas del switch que coinciden el flujo. Cuando se invoca con la opción **-strict option**, los comodines no son tratados como activos para propósitos de coincidencia. Si un flujo se omite, y la opción **-strict** no es usada, todos los flujos de las tablas del datapath son eliminadas. Ver **flow syntax** más abajo para la sintaxis de los flujos.

monitor *switch*

Se conecta al switch y muestra en la consola todos los mensajes Openflow recibidos. Usualmente, el switch puede especificar una conexión nombrada en la opción de línea de comando **ofprotocol(8) -m** o **-monitor**, en la cual, los mensajes mostrados serán todos aquellos enviados o recibidos por **ofprotocol** hacia o desde del módulo datapath del kernel de linux. Un switch de la forma `nl:dp_idx` mostrará todos los mensajes generados en Openflow de manera asíncrona (tales como los mensajes **packet-in**), pero no mostrarán los mensajes enviados kernel por **ofprotocol** y otros procesos, ni va a mostrar respuestas enviadas por el kernel en respuesta a dichos mensajes.

Los siguientes comandos supervisan y controlan la configuración de cola de la salida para un switch Openflow si el switch soporta esas operaciones. Después se crea una cola con la operación de añadir o modificar, la acción de poner en cola (enqueue) del Openflow puede ser especificada para dirigir paquetes a una cola en particular. Las colas se asocian con puertos específicos (por lo que el mismo id de la cola se puede utilizar en diferentes puertos y hará referencia a diferentes colas). La única característica que puede ser configurada para colas es la garantía de ancho de banda mínimo. Este parámetro se especifica en décimas de por ciento (osea que el ancho de banda máximo es 1000).

add-queue *switch port q-id [bandwidth]*

Conecta al switch y agrega una cola de salida identificada como **q-id** para el puerto. Si se especifica, el ancho de banda indica la garantía de ancho de banda mínimo para la cola y se especifica en décimas de un por ciento. Esta es la única característica de la cola que se puede configurar.

mod-queue *switch port q-id bandwidth*

Conecta al switch y modifica la configuración de ancho de banda para una cola de salida identificada como **q-id** para el puerto. La cola no tiene que haber sido creada con **add-queue** previamente. El parámetro ancho de banda indica el ancho de banda mínimo garantizado para la cola y se especifica en décimas de por ciento. Esta es la única característica de la cola que se puede configurar.

del-queue *switch port q-id*

Elimina una cola de salida identificada como **q-id** para el puerto que había sido creada por **add-queue** o **mod-queue**.

dump-queue *switch [port [q-id]]*

Vuelca la configuración actual de la cola. Un puerto puede ser especificado. Si lo es, un **queue-id** puede también ser especificado.

Los siguientes comandos se pueden utilizar independientemente del método de conexión. Se aplican a switches y controladores Openflow.

probe *vconn*

Se conecta a *vconn* y envía una única solicitud de eco (**echo-request**) Openflow y espera la respuesta. Con la opción **-t** o **-timeout**, este comando puede comprobar si un switch o controlador Openflow está en funcionamiento.

ping *vconn [n]*

Envía una serie de 10 paquetes de solicitud de eco a *vconn* y los tiempos de cada respuesta. Los paquetes de la solicitud de eco consisten en una cabecera Openflow más *n* bytes (por defecto 64) de carga útil generada aleatoriamente. Esto mide la latencia de las solicitudes individuales.

benchmark *vconn n count*

Envía contar los paquetes de solicitud de eco, que consisten cada uno de una cabecera Openflow más *n* bytes de carga útil y espera para cada respuesta. Reporta el tiempo total requerido. Esta es una medida del ancho de banda máximo a *vconn* para viajes de ida de mensajes de *n* bytes.

FLOW SYNTAX (sintaxis del flujo)

Algunos comandos *dpctl* aceptan un argumento que describe un flujo o flujos. Tales descripciones de flujo constituyen un campo de serie= asignaciones de valores, separados por comas o espacios en blanco.

Las siguientes asignaciones de campo describen cómo un flujo coincide con un paquete. Si cualquiera de estas asignaciones se omiten en la sintaxis del flujo, el campo se trata como si fuera un comodín; por lo que, si todas de ellas son omitidas, el flujo resultante coincide con todos los paquetes. La cadena * o ANY puede especificar un valor para señalar explícitamente cualquiera de estos campos como un comodín.

in_port=*port_no*

Enlaza puertos físicos **port-no**. Los puertos del switch se numeran como los que se muestran por **dpctl show**.

dl_vlan=vlan

Enlaza LAN virtuales basadas en el estándar IEEE 802.1q con la etiqueta *vlan*. . Especifica 0xffff como *vlan* para enlazar paquetes que no están etiquetados con una LAN virtual, de lo contrario, especifica un número entre 0 y 4095, como el 12-bit VLAN ID para enlazar.

dl_src=mac

Enlaza la dirección Ethernet *mac* de origen, la cual puede ser especificada como 6 pares de dígitos hexadecimales delimitados por dos puntos, como por ejemplo: AA:BB:CC:DD:EE:FF.

dl_dst=mac

Enlaza la dirección Ethernet *mac* de destino.

dl_type=ethertype

Enlaza con el protocolo Ethernet tipo *ethertype*, que debe ser especificado entre 0 y 65535, inclusive, ya sea en formato decimal o como un número hexadecimal con el prefijo 0x, por ejemplo, 0x0806 para comparar paquetes ARP.

nw_src=ip[/netmask]

Enlaza la dirección IP de origen IPv4, la cual debe ser especificada como una dirección IP o host name, por ejemplo, 192.168.1.1 o www.ejemplo.com. La máscara de red opcional permite que se enlace sólo en un prefijo de dirección IPv4. Puede ser especificado por ejemplo, como 192.168.1.0/255.255.255.0 o como un conteo de bits, por ejemplo, 192.168.1.0/24.

nw_dst=ip[/netmask]

Enlaza la dirección IP IPv4 de destino.

nw_proto=proto

Enlaza protocolos IP tipo *proto*, los cuales pueden ser especificados como un número decimal entre 0 y 255, por ejemplo, 6 para que enlace con paquetes TCP.

nw_tos=tos/dscp

Enlaza campos ToS/DSCP(sólo 6 bits, no modifican 2 bits reservados para uso futuro) de cabeceras IPv4 *tos/dscp*, los cuales pueden ser especificados como un número decimal entre 0 y 255.

tp_src=port

Enlaza puertos de origen UDP o TCP, los cuales pueden ser especificados como un número decimal entre 0 y 65535, por ejemplo, 80 para que enlace con paquetes procedentes de un servidor HTTP.

tp_dst=port

Enlaza puertos de destino UDP o TCP.

icmp_type=type

Enlaza mensajes ICMP con *type*, los cuales pueden ser especificados como un número decimal entre 0 y 255.

icmp_code=code

Enlaza mensajes ICMP con *code*.

Las siguientes notaciones abreviadas también están disponibles:

ip

Lo mismo que **dl_type=0x0800**.

icmp

Lo mismo que **dl_type=0x0800,nw_proto=1**.

tcp

Lo mismo que **dl_type=0x0800,nw_proto=6**.

udp

Lo mismo que **dl_type=0x0800,nw_proto=17**.

arp

Lo mismo que **dl_type=0x0806**.

Los comandos **add-flow** y **add-flows** requieren un campo adicional:

actions=target[,target...]

Especifica una lista separada por comas de acciones a tomar en un paquete cuando el flujo de entrada se enlaza. El *target* puede ser un número de puerto decimal designando el puerto físico en el cual se envía el paquete, o una de las siguientes palabras clave.

output:port

Envía el paquete en el puerto especificado por el *puerto*.

enqueue:port:q-id

Pone en cola el paquete a la cola especificada por **q-id** en el puerto especificado por *port*.

controller:max_len

Envía el paquete al controlador Openflow como un paquete “*packet in*”. Si *max_len* es un número, especifica el máximo número de bytes que deben ser enviados. Si *max_len* es *ALL* u *omitted*, el paquete entero es enviado.

Local

Envía el paquete al puerto ``local'', que corresponde al dispositivo de red.

mod_vlan_vid:*vlan_vid*

Modifica el ID de la VLAN en un paquete. La etiqueta VLAN se agrega o modifica según sea necesario para que coincida con el valor especificado. Si la etiqueta VLAN se añade, una prioridad de cero es usada (ver la acción **mod_vlan_pcp** para configurarla).

mod_vlan_pcp:*vlan_pcp*

Modifica la prioridad VLAN en un paquete. La etiqueta VLAN se agrega o modifica según sea necesario para que se enlace con el valor especificado. Los Valores válidos están entre 0(más bajo) y 7(más alto). Si la etiqueta VLAN se añade, una vid de cero se utiliza(ver la acción **mod_vlan_vid** para configurarla).

mod_dl_dst:*dst_mac*

Modifica la dirección MAC de destino de un paquete, por ejemplo, **actions=mod_dl_dst:12:34:56:78:9a:bc.**

mod_dl_src:*src_mac*

Modifica la dirección MAC de origen de un paquete, por ejemplo, **actions=mod_dl_src:12:34:56:78:9a:bc.**

mod_nw_tos:*tos/dscp*

Modifica el campo ToS/DSCP(sólo 6-bits, no modifican dos bits reservados para uso futuro) de la cabecera IPv4 de un paquete.

strip_vlan

Quita la etiqueta VLAN de un paquete si está presente.

Los comandos **add-flow**, **add-flows**, **del-flows** y **del-emerg-flows** soportan un campo adicional opcional:

priority=*value*

Establece la prioridad del flujo para ser añadidos o suprimidos a *value*, que debe ser un número entre 0 y 65535. Si este campo no se especifica, el valor predeterminado es 32768.

Los comandos **add-flow** y **add-flows** soportan campos adicionales opcionales:

idle_timeout=*seconds*

Hace que el flujo expire después de un número determinado de segundos de inactividad. El valor 0 impide que un flujo expire debido a inactividad. Por defecto son 60 segundos.

hard_timeout=seconds

Hace que el flujo expire después de un número determinado de segundos, independientemente de la actividad.

Los comandos **dump-flows**, **dump-aggregate** y **del-flows** un soportan campo adicional de opcional:

out_port=port

Si se establece, un flujo enlazado debe incluir una acción de salida al puerto.

Los comandos **add-flow**, **del-flows**, **dump-flows** y **dump-aggregate** soportan el campo opcional adicional:

table=number

Si se especifica, limita los flujos sobre los cuales las estadísticas se recopilan en la tabla con el número dado. Las tablas normales (no de emergencia) son nombradas como se muestra con el comando **dump-tables**.

Si este campo no se especifica, o si el número se dá como 255, se recopilan estadísticas sobre los flujos de todas las tablas normales (no de emergencia) y la manipulación de los flujos son aplicadas a las tablas normales.

Si este campo se dá como 254, se recopilan las estadísticas sobre los flujos de la tabla de emergencia y las manipulaciones de flujo son aplicadas a la tabla de emergencia.

OPCIONES

--strict

Usar el enlace *strict* cuando se ejecutan comandos de modificaciones de flujo.

-t, --timeout=secs

Limita a dpctl para que se ejecute aproximadamente en segundos. Si el tiempo de espera expira, dpctl terminará con una señal SIGALRM.

-p, --private-key=privkey.pem

Especifica un archivo PEM que contiene la clave privada, que se utiliza como la identidad para conexiones SSL a un switch.

-c, --certificate=cert.pem

Especifica un archivo PEM que contiene un certificado, firmado por la autoridad del certificado del controlador (CA), que certifica la clave privada para identificar un controlador confiable.

-C, --ca-cert=cacert.pem

Especifica un archivo PEM que contiene el certificado (CA) usado para verificar que un switch es confiable.

--log-file[=file]

Habilita el registro de un archivo. Si el archivo es especificado, se utiliza como el nombre exacto del archivo de registro. El registro del nombre del archivo predeterminado se utiliza si se omite el archivo var/log/openflow/dpctl.log.

-h, --help

Muestra en la consola un mensaje de ayuda.

-V, --version

Muestra información de la versión a la consola.

4.5 PRÁCTICAS DE LABORATORIO

4.5.1 Tarjeta NetFPGA como switch Openflow

Introducción

Esta práctica de laboratorio consiste en crear un switch Openflow por medio de la tarjeta NetFPGA con el fin de comunicar varios computadores y poder enviarse paquetes entre sí.

Objetivos

- Crear un switch Openflow por medio de la tarjeta NetFPGA.
- Enviar paquetes desde un computador a otro.

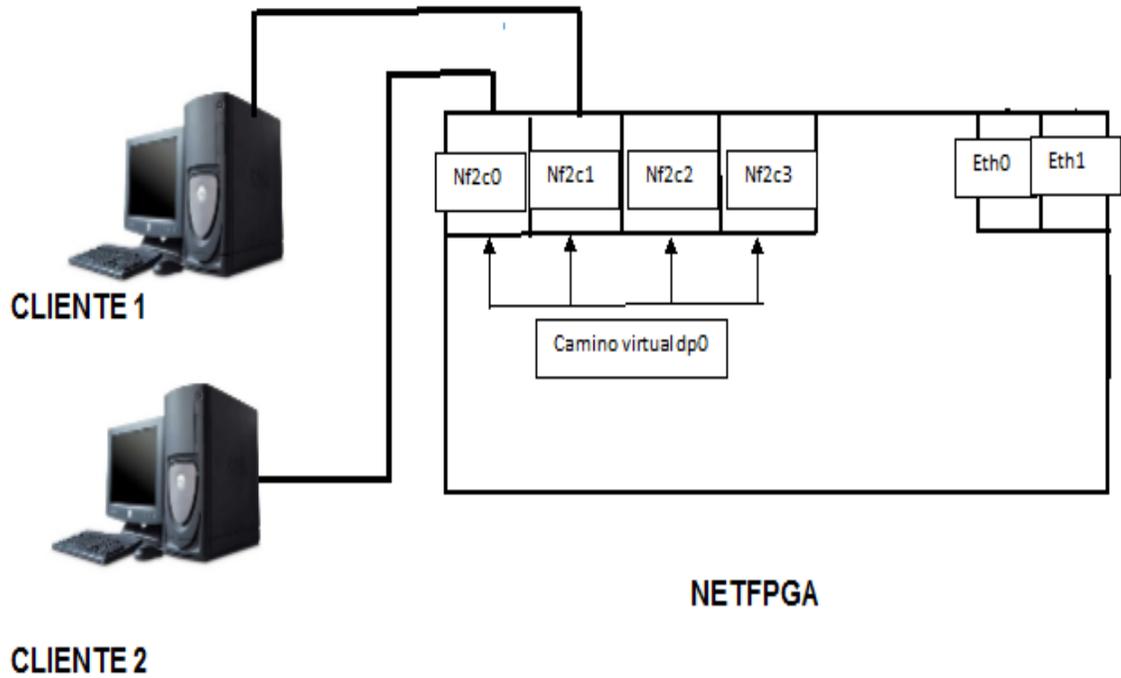
Materiales y equipos a utilizar

- 1 NetFPGA con GNU/Linux distribución CentOS y con el paquete de Openflow instalado.
- 2 computadores con GNU/Linux distribución CentOS para ser conectados como clientes.
- 2 patchcords.

NOTA:

- Tener en cuenta que los puertos de los dos clientes deben ser de 1Gbit Ethernet.
- El terminal de Linux diferencia entre letras mayúsculas y minúsculas, no es lo mismo teclear hola a Hola.

Diagrama del laboratorio



Desde la solapa 1 al 7 los comandos deben ser ejecutados en solapas diferentes.

Procedimiento:

Solapa 1: Limpiar la memoria de la NetFPGA.

`cpci_reprogram.pl`

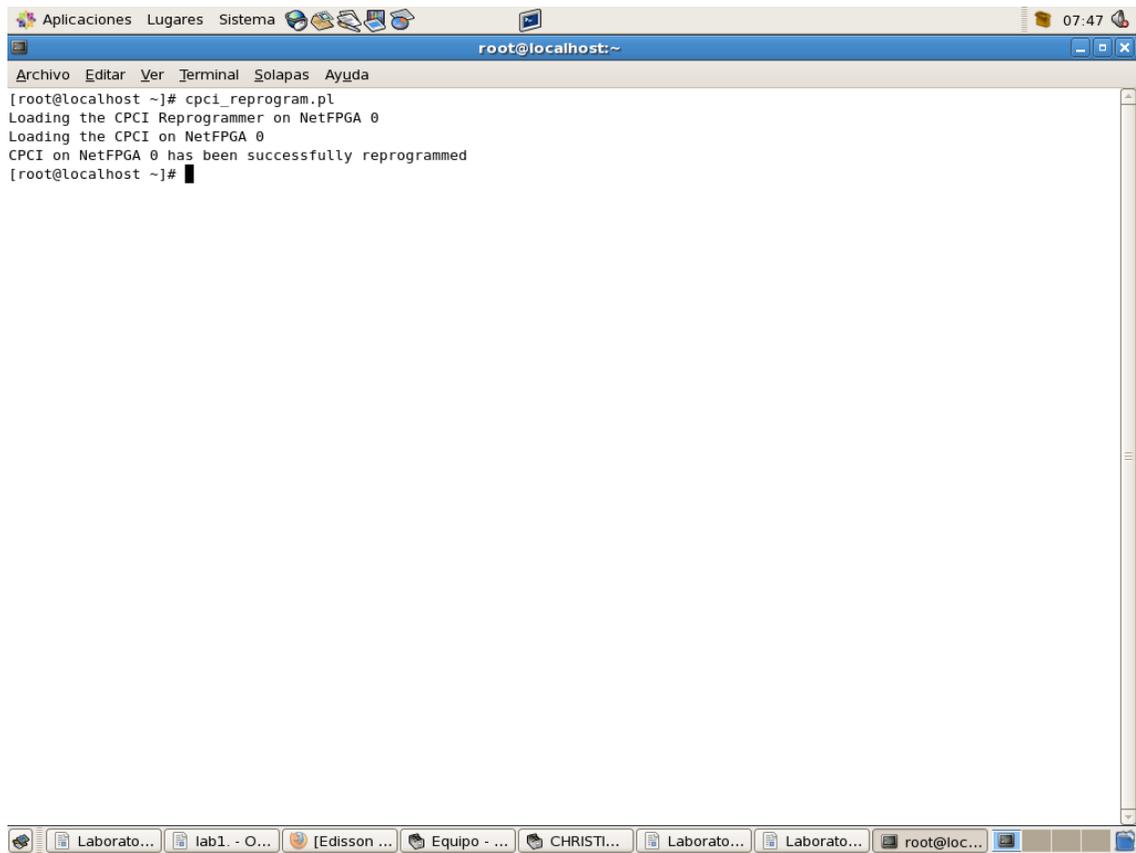


Imagen 1. Limpiar la memoria de la NetFPGA.

Solapa 2: Cargar el bitfile del switch. Ir a la ruta raíz del Openflow (donde está instalado). Para esto, seguir la siguiente ruta.

```
cd /root/openflow/hw-lib/nf2
```

```
nf_download openflow_switch.bit
```

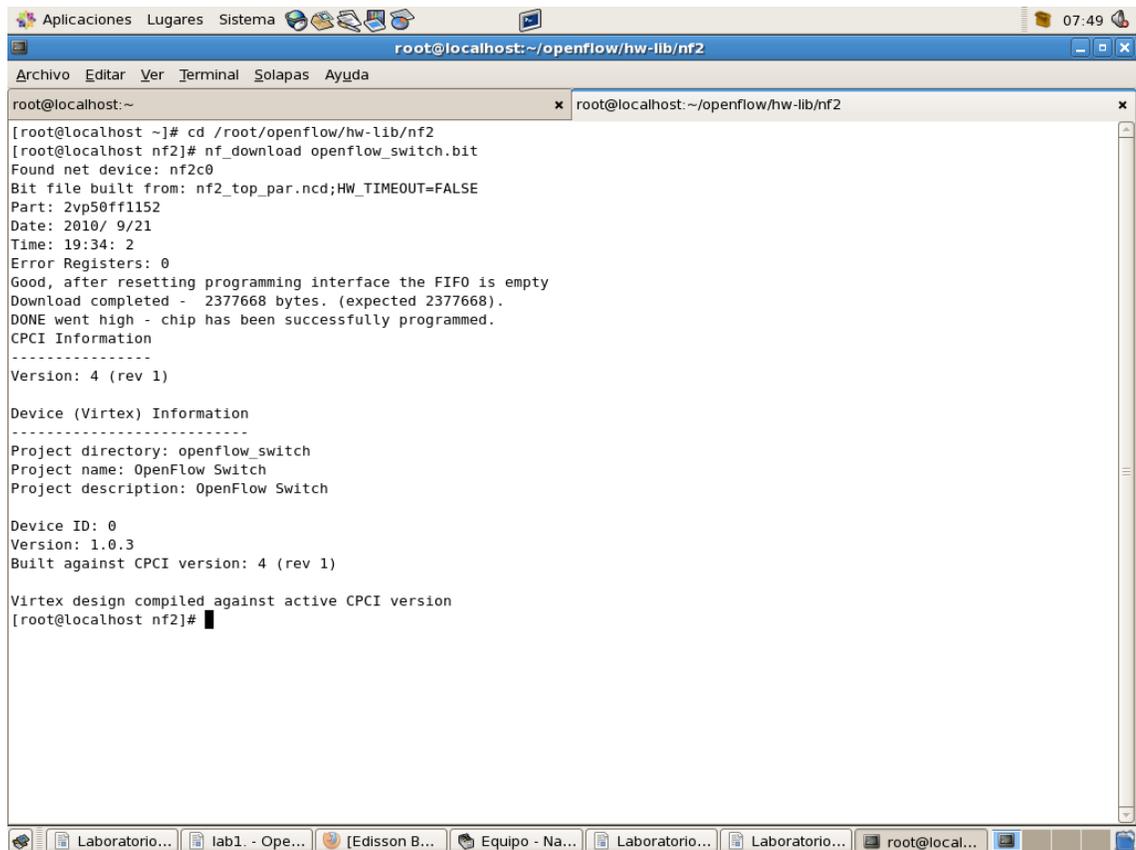


Imagen 2. Cargar el bitfile del switch.

En la figura 2, se debe comprobar que en la opción Error Registers aparezca un cero (0), de contrario, reprogramar la NetFPGA.

Solapa 3: Cargar el controlador y enlazarlo a un puerto tcp. Para esto, buscar el ejecutable controller en la carpeta controller cuya ruta es la siguiente.

```
cd /root/openflow/controller
```

```
./controller ptcp:6633
```

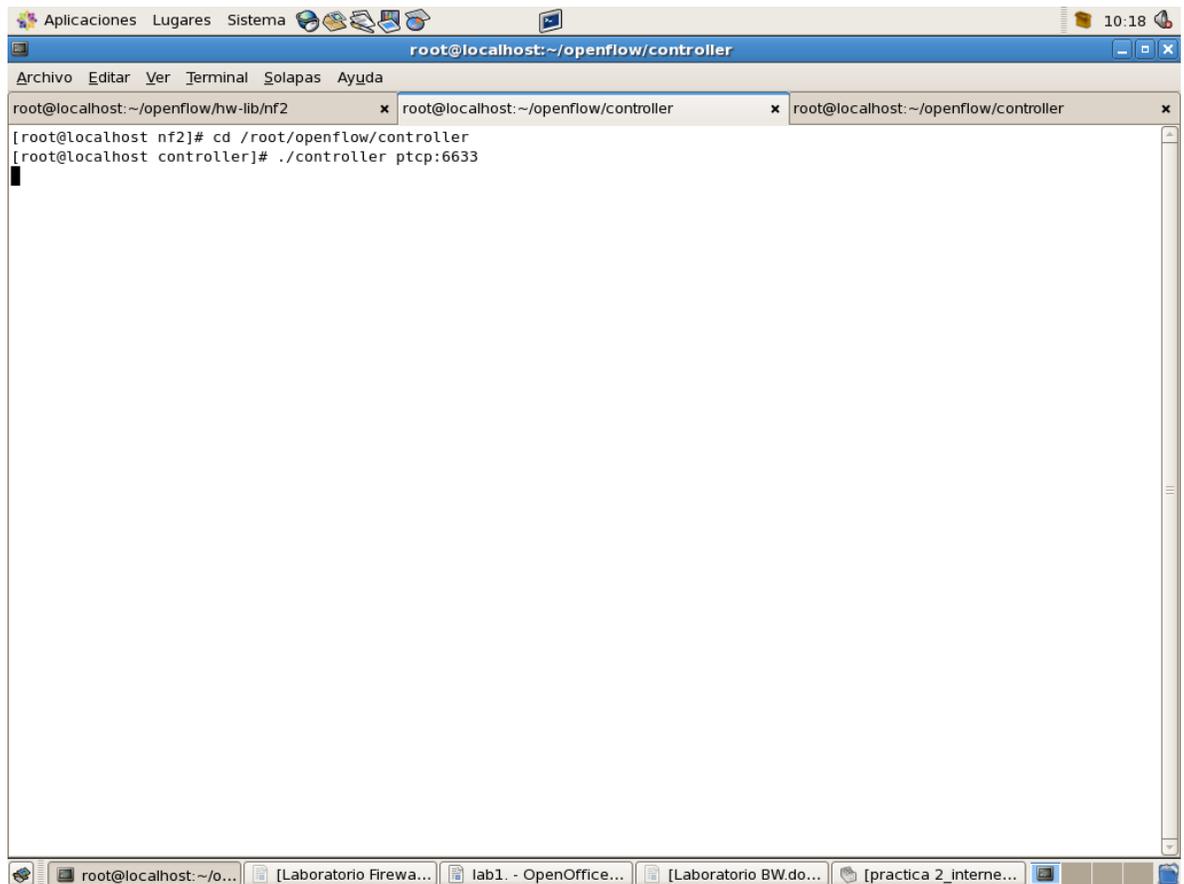


Imagen 3. Cargar el controlador.

Solapa 4: Comprobar que el controlador está encendido.

`netstat -atunp | grep 6633`

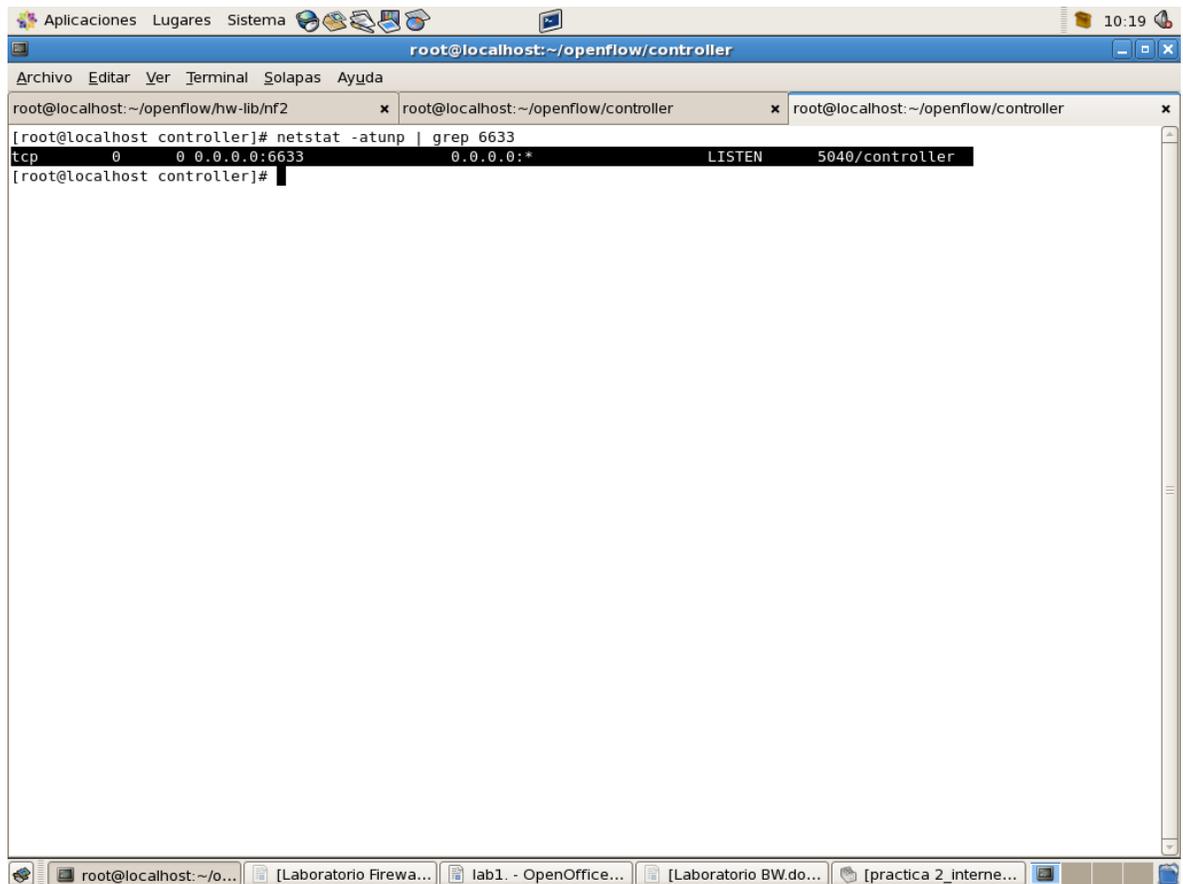


Imagen 4. Verificar que el controlador esté en escucha.

Solapa 5: Crear el camino virtual del switch con todos los puertos de la NETFPGA y verificar la creación de una interfaz virtual donde se puede ver el flujo (tapX).

```
ofdatapath --detach punix:/var/run/dp0 -d aabbccddeeff -i nf2c0,nf2c1,nf2c2,nf2c3  
&& ifconfig | grep tap
```

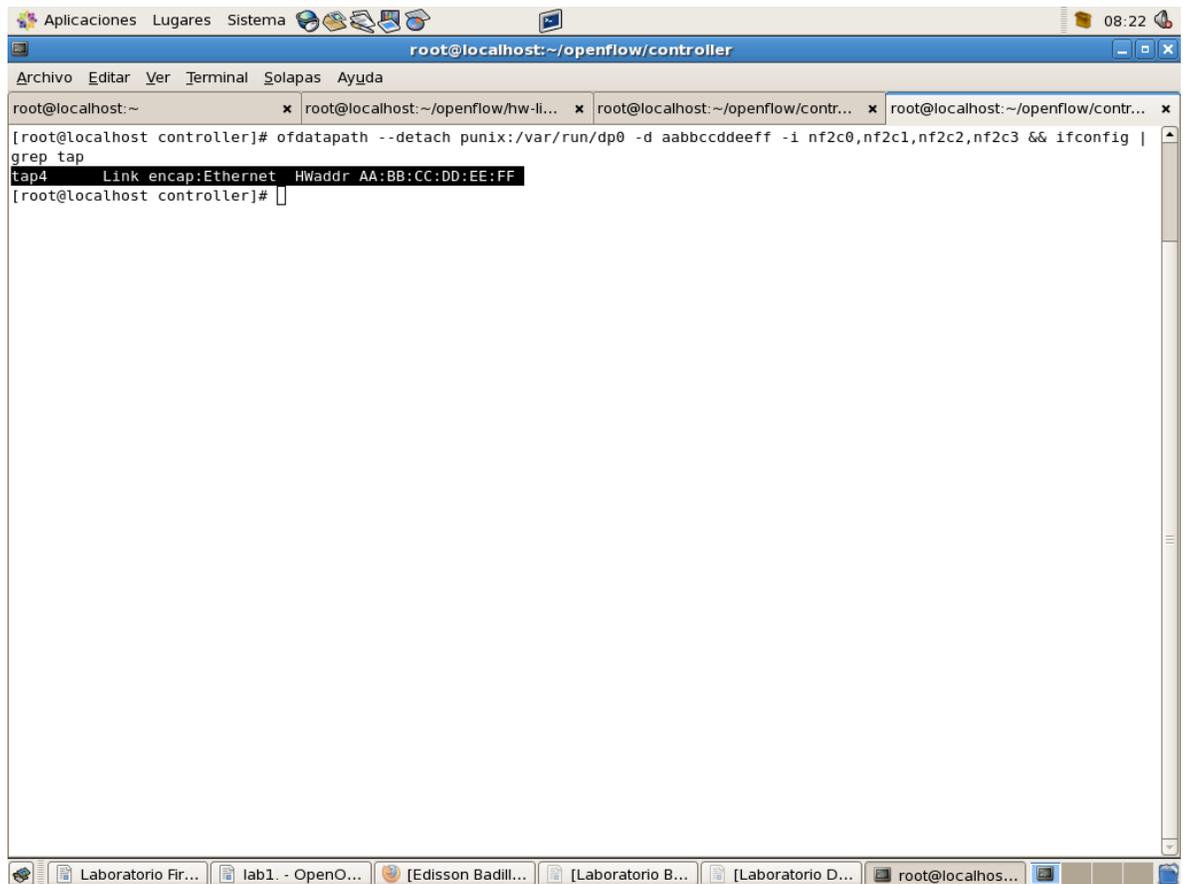


Imagen 5. Camino virtual dp0.

Solapa 6: Comunicación directa con el controlador usando dirección IP y puerto (socket).

ofprotocol unix:/var/run/dp0 tcp:127.0.0.1:6633

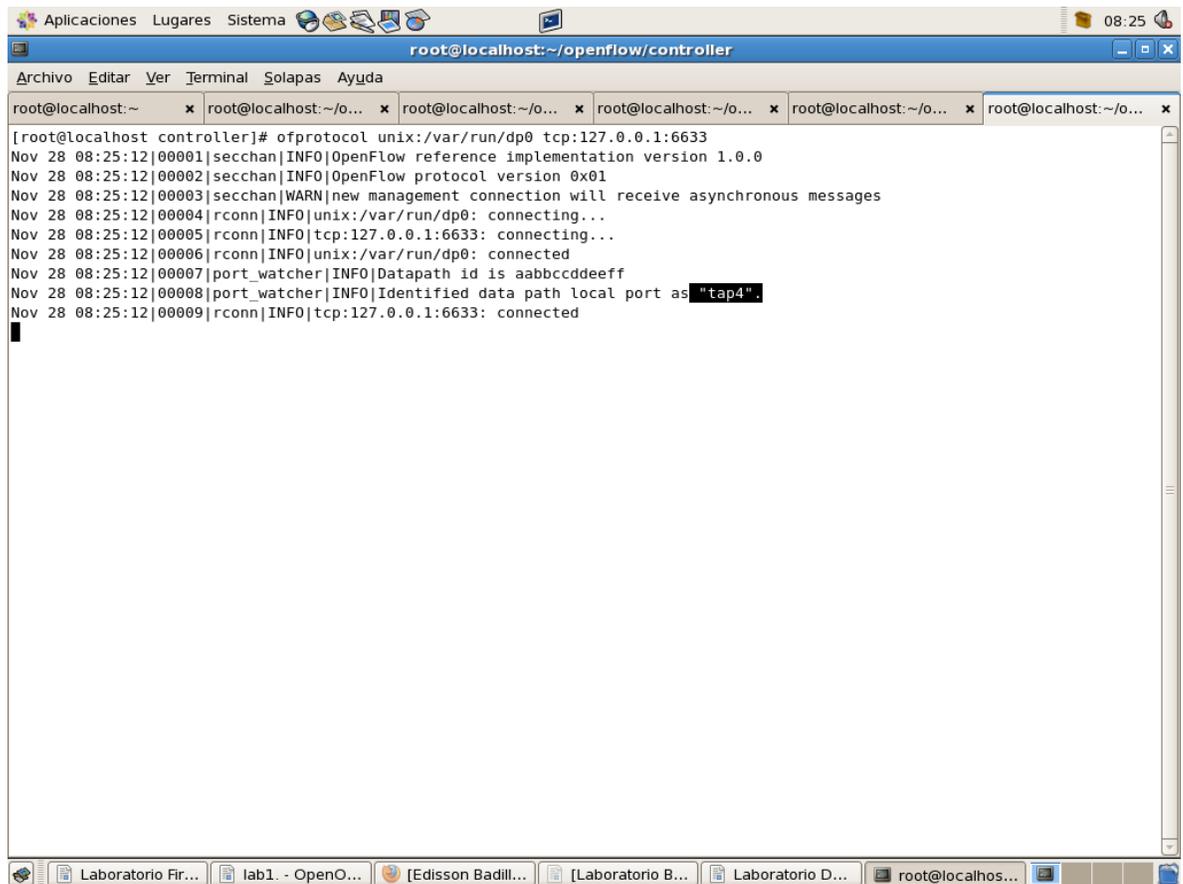


Imagen 6. Camino dp0.

Solapa 7: Ver el flujo del camino virtual dp0.

dpctl dump-flows unix:/var/run/dp0

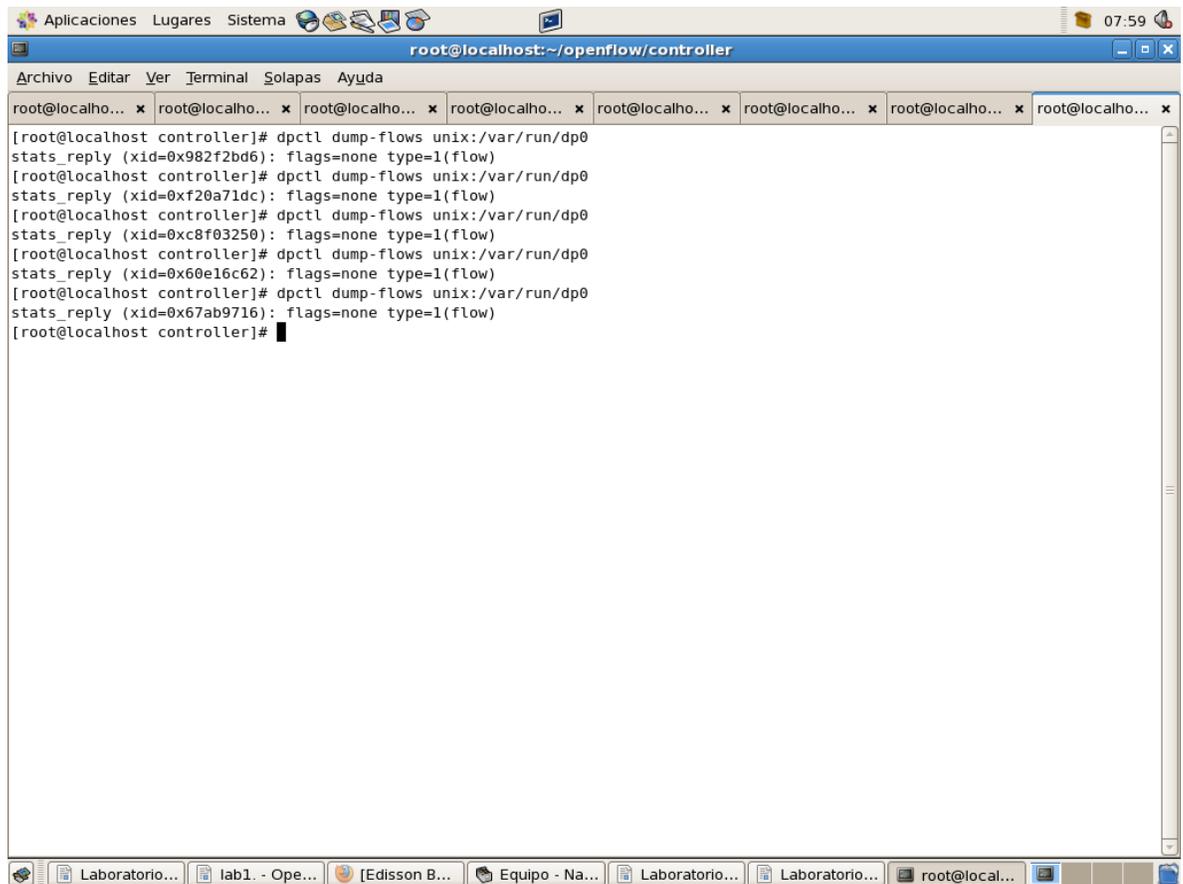


Imagen 7. Flujo dp0.

Solapa 8: Hacer ping entre computadores.

Desde PC cliente 1: Conectar un patchcord desde el puerto eth1 del cliente 1 hacia el puerto nf2c3 de la NetFPGA. Configurar el puerto del cliente 1 con el siguiente comando.

```
ifconfig eth1 192.168.1.10
```

Desde PC cliente 2: Conectar un patchcord desde el puerto eth1 del cliente 2 hacia el puerto nf2c2 de la NetFPGA. Configurar el puerto del cliente 2 con el siguiente comando.

```
ifconfig eth1 102.168.1.20
```

Después de haber configurado los puertos, desde el cliente 1 teclear el siguiente comando.

ping -I eth1 192.168.1.20

siendo:

eth1 puerto de origen.

192.168.1.20 IP de destino.

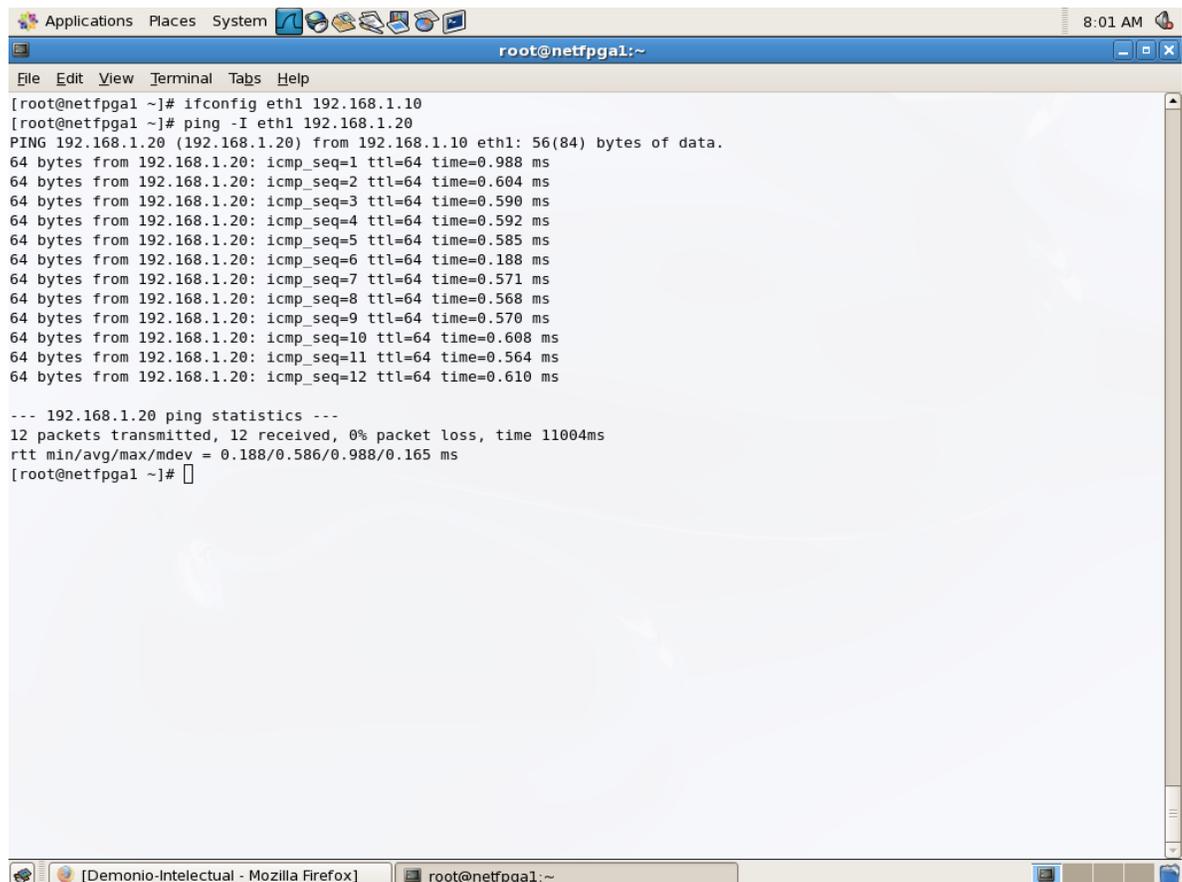


Imagen 8. Ping satisfactorio.

Como se puede apreciar en la figura 11, el ping fue satisfactorio, lo que quiere decir que ya se pueden comunicar los dos clientes. Para dejar de enviar ping oprimir las teclas control + c. En este ejemplo, se enviaron 12 paquetes y los mismos fueron recibidos por el cliente 2 sin ninguna pérdida.

ACTIVIDAD.

Comunicar el resto puertos de la NetFPGA y verificar que realmente se esté comportando como un switch.

4.5.2 Creación de VLAN mediante la tarjeta NetFPGA como switch Openflow

Introducción

Una VLAN es una agrupación lógica de los puertos de un switch, de manera que se comportan como si fueran *switches* independientes. Esta práctica de laboratorio consiste en crear dos VLAN`s en los puertos Gbit Ethernet que tiene la NetFPGA. El objetivo de esta práctica lograr comunicar los puertos nf2c0 y nf2c1 mediante la VLAN 1 y los puertos nf2c2 y nf2c3 mediante la VLAN 2, pero que no se puedan comunicar los puertos de la VLAN 1 y 2 respectivamente.

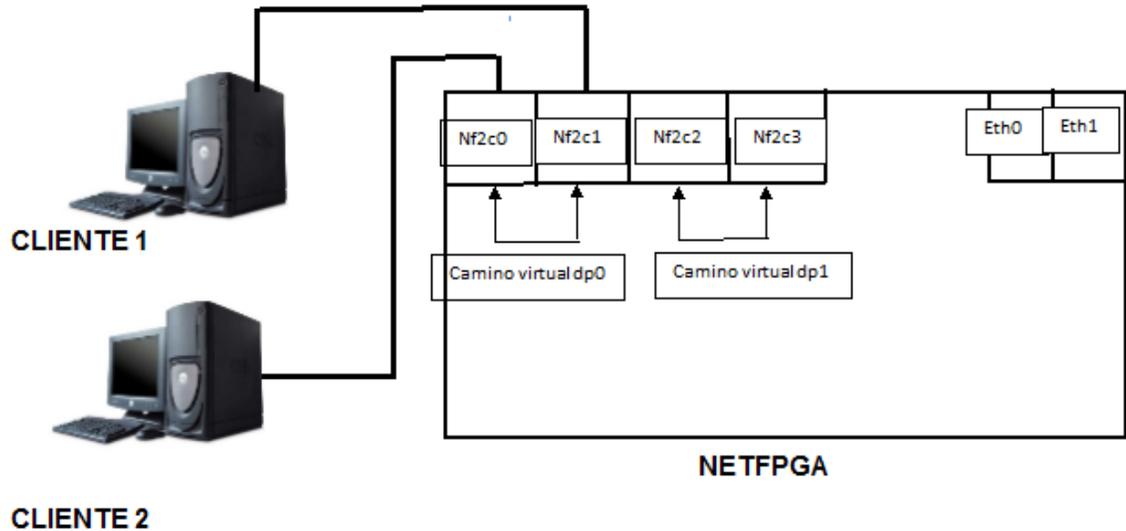
Objetivos

- Crear dos VLAN`s en los puertos de la NetFPGA.
- Verificar que la VLAN 1 no se comuniquen con la VLAN 2.
- Verificar que en cualquier VLAN se comuniquen los puertos unidos virtualmente.

Materiales y equipos a utilizar

- 1 NetFPGA con GNU/Linux distribución CentOS y con el paquete de Openflow instalado.
- 2 computadores con GNU/Linux distribución CentOS para ser conectados como clientes.
- 2 patchcords.

Diagrama del laboratorio



NOTA:

- Tener en cuenta que los puertos de los dos clientes deben ser de 1Gbit Ethernet.
- El terminal de Linux diferencia entre letras mayúsculas y minúsculas, no es lo mismo teclear hola a Hola.

Procedimiento:

Desde la solapa 1 al 10 los comandos deben ser ejecutados en terminales diferentes.

Solapa 1: Limpiar la memoria de la NetFPGA.

`cpci_reprogram.pl`

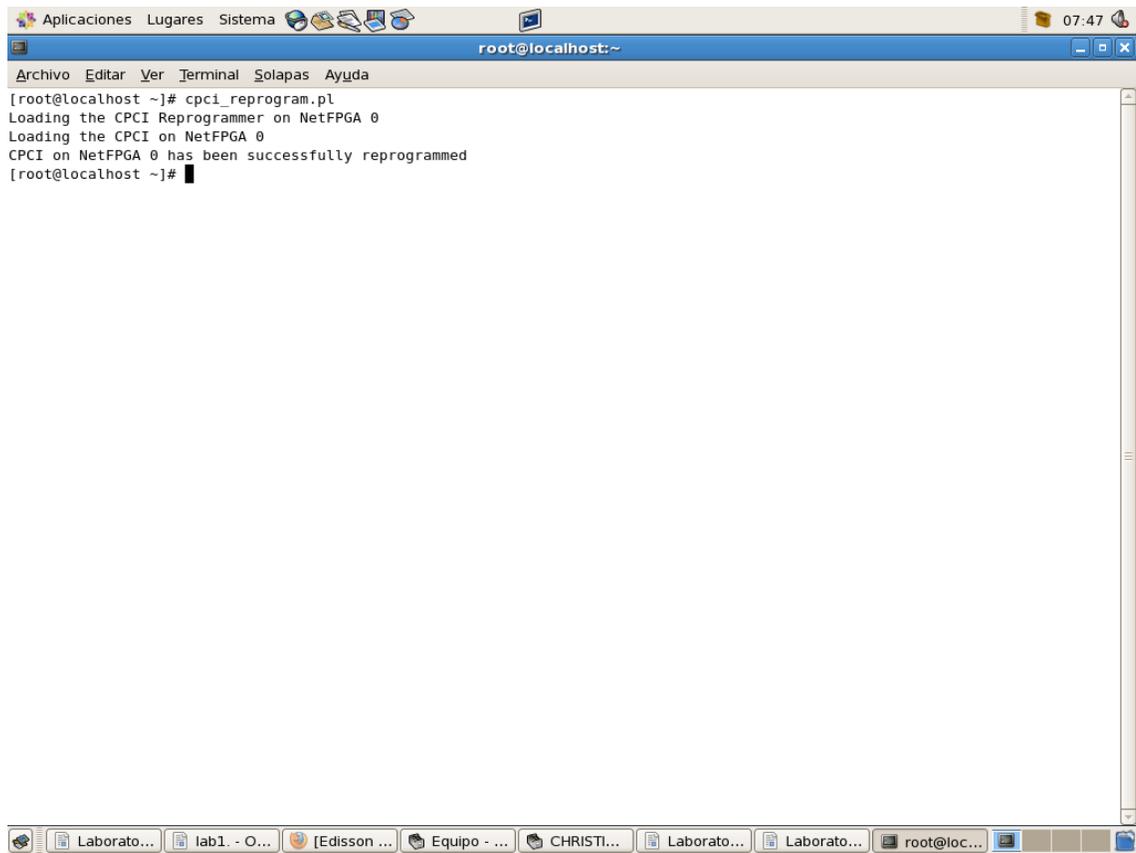


Imagen 1. Limpiar la memoria de la NetFPGA.

Solapa 2: Cargar el bitfile del switch. Ir a la ruta raíz del Openflow (donde está instalado). Para esto, seguir la siguiente ruta.

```
cd /root/openflow/hw-lib/nf2
```

```
nf_download openflow_switch.bit
```

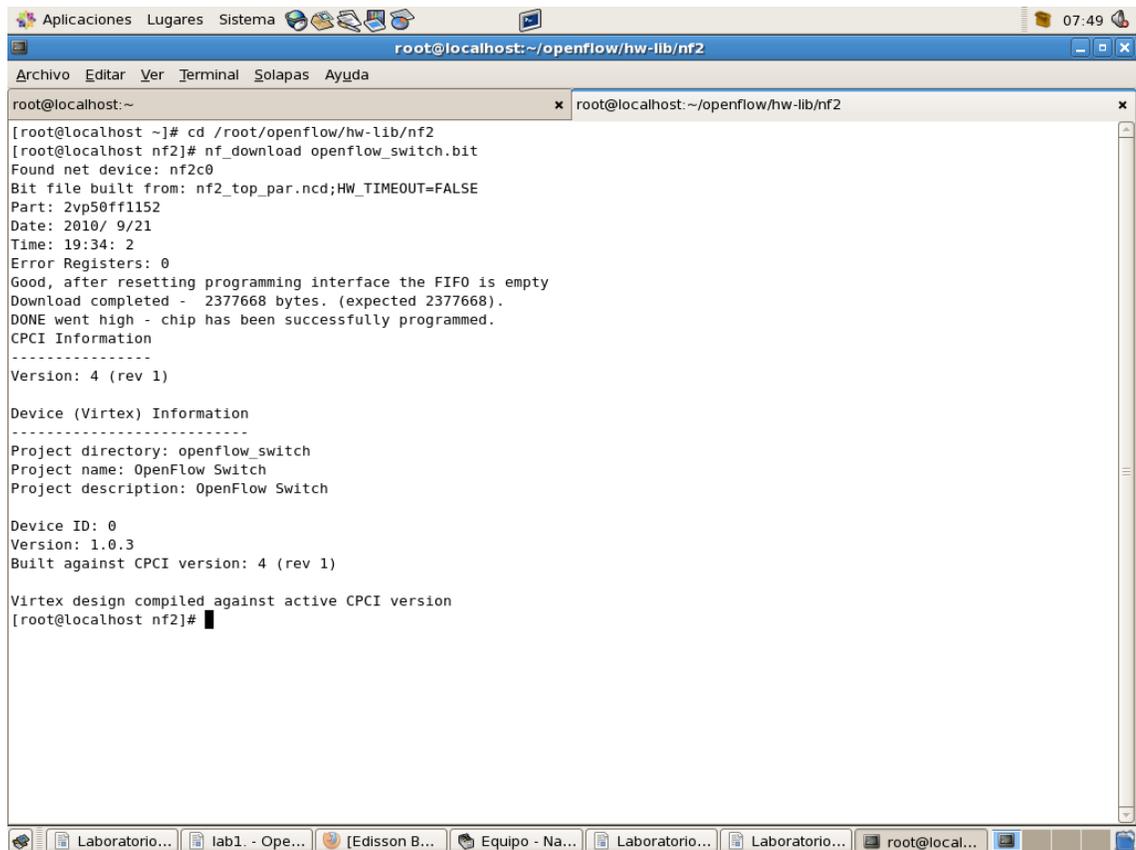


Imagen 2. Cargar el bitfile del switch.

En la figura 2, se debe comprobar que en la opción Error Registers aparezca un cero (0), de contrario, reprogramar la NetFPGA.

Solapa 3: Cargar el controlador y enlazarlo a un puerto tcp. Para esto, buscar el ejecutable controller en la carpeta controller cuya ruta es la siguiente.

```
cd /root/openflow/controller
```

```
./controller ptcp:6633
```

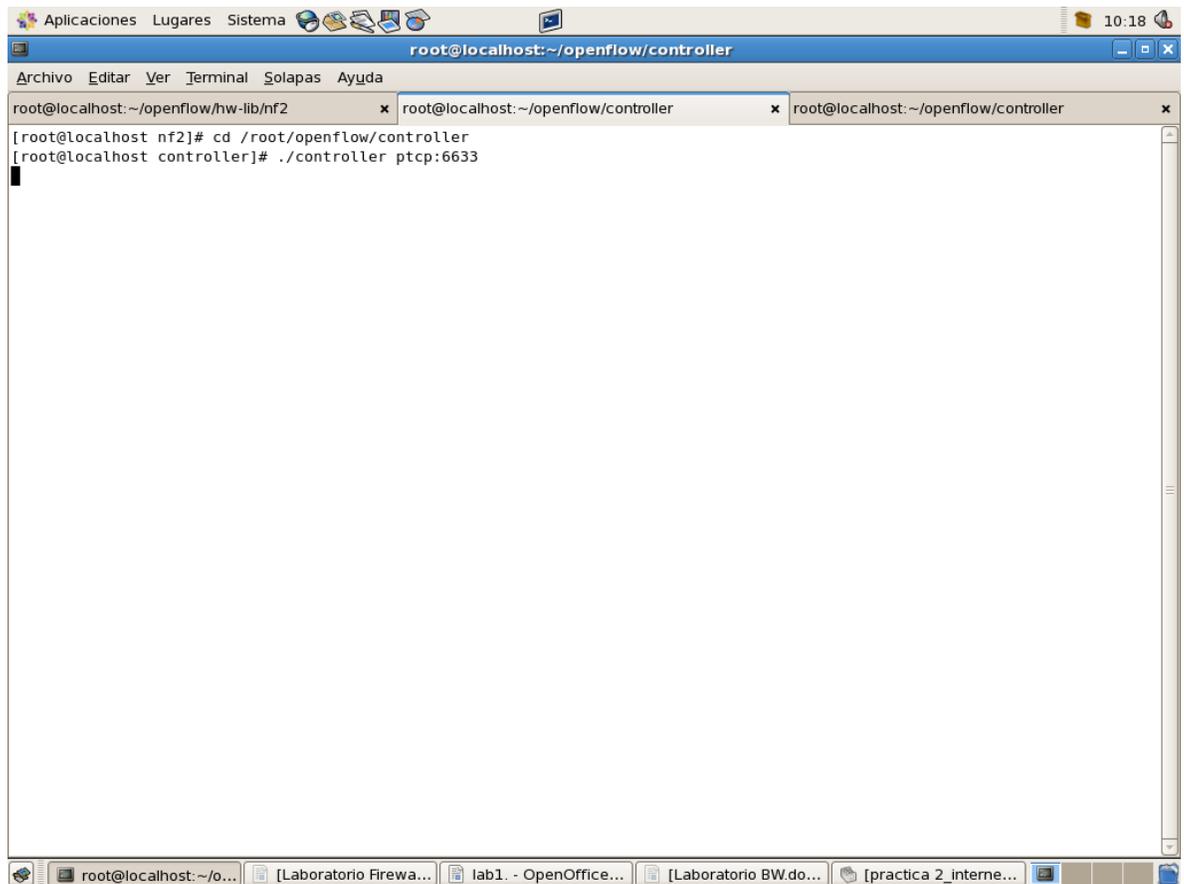


Imagen 3. Cargar el controlador.

Solapa 4: Comprobar que el controlador está encendido.

`netstat -atunp | grep 6633`

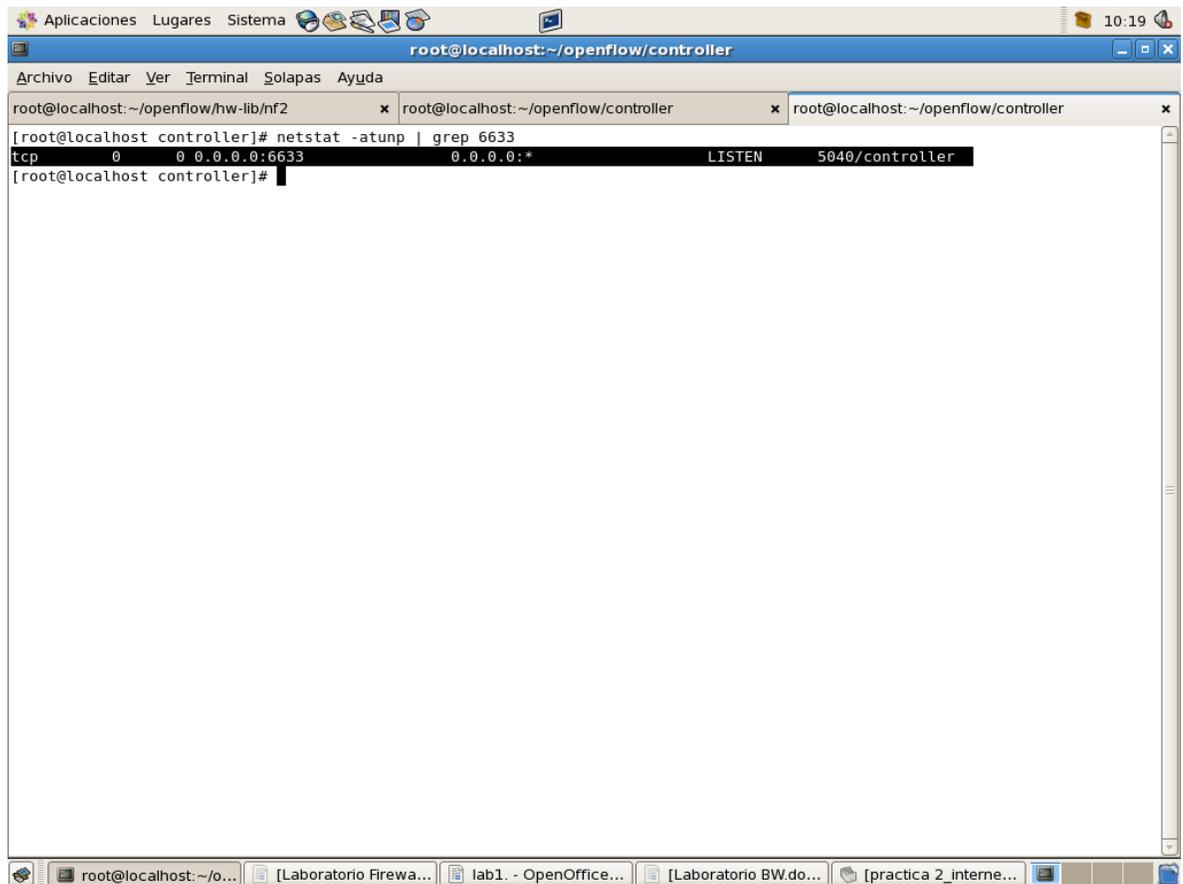


Imagen 4. Verificar que el controlador esté en escucha.

Solapa 5: Creación del camino virtual dp0 en tap2 con ID del datapath AA:BB:CC:DD:EE y puertos nf2c0, nf2c1 de la NetfPGA.

```
ofdatapath --detach punix:/var/run/dp0 -d aabbccddeeff -i nf2c0,nf2c1 && ifconfig |  
grep tap
```

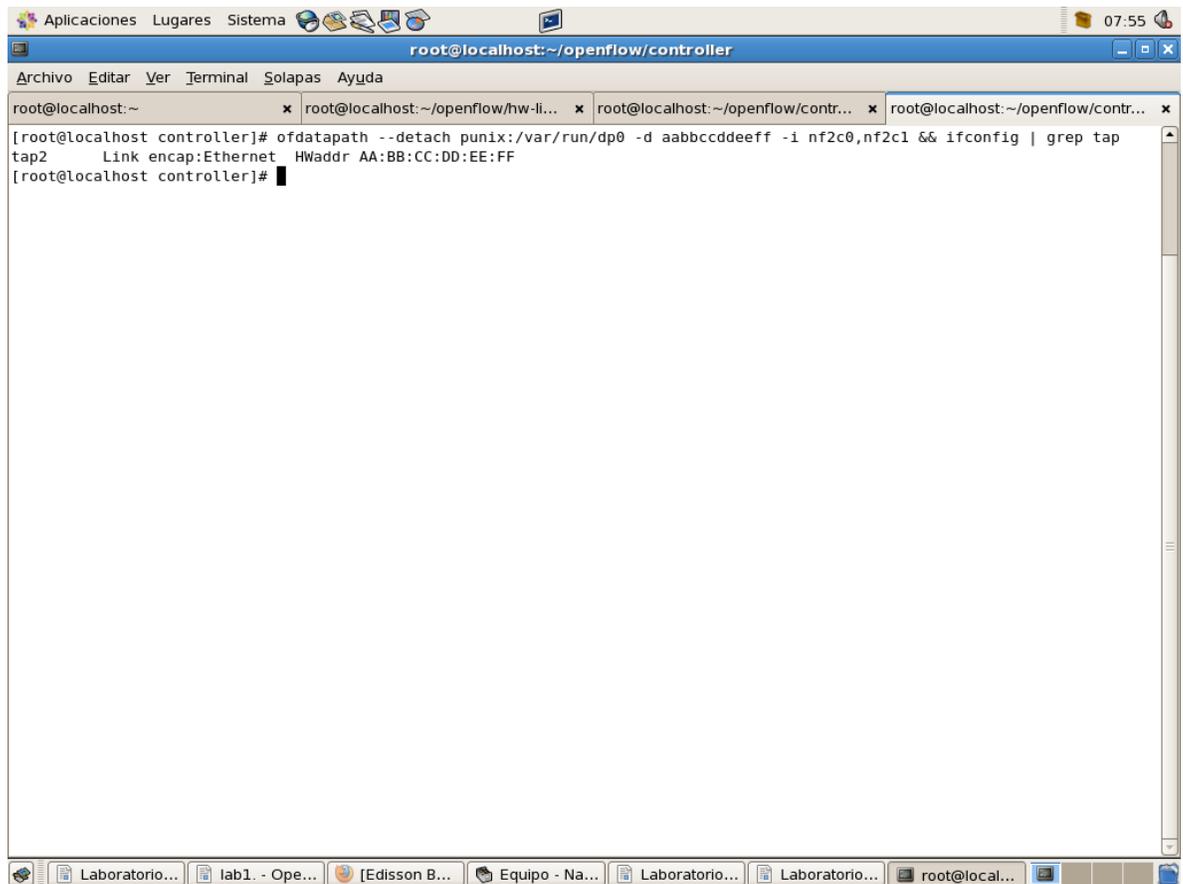


Imagen 5. Camino virtual dp0.

Solapa 6: Creación del camino virtual dp1 en tap3 con ID del datapath 00:11:22:33:44:55 y puertos nf2c2, nf2c3 de la NetfPGA.

```
ofdatapath --detach punix:/var/run/dp1 -d 001122334455 -i nf2c2,nf2c3 && ifconfig | grep tap
```

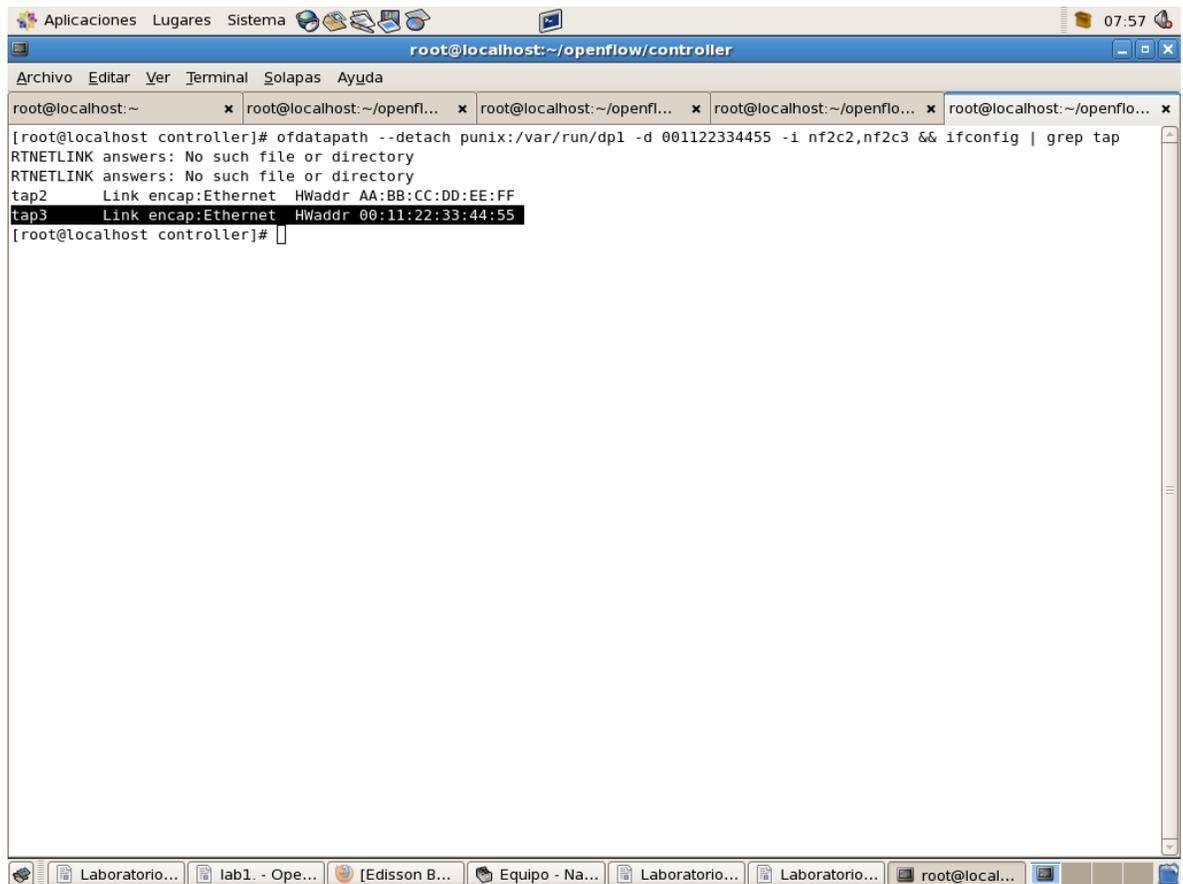


Imagen 6. Camino virtual dp1.

Solapa 7: Comunicación directa con el controlador usando dirección IP y puerto (socket). Camino dp0.

ofprotocol unix:/var/run/dp0 tcp:127.0.0.1:6633

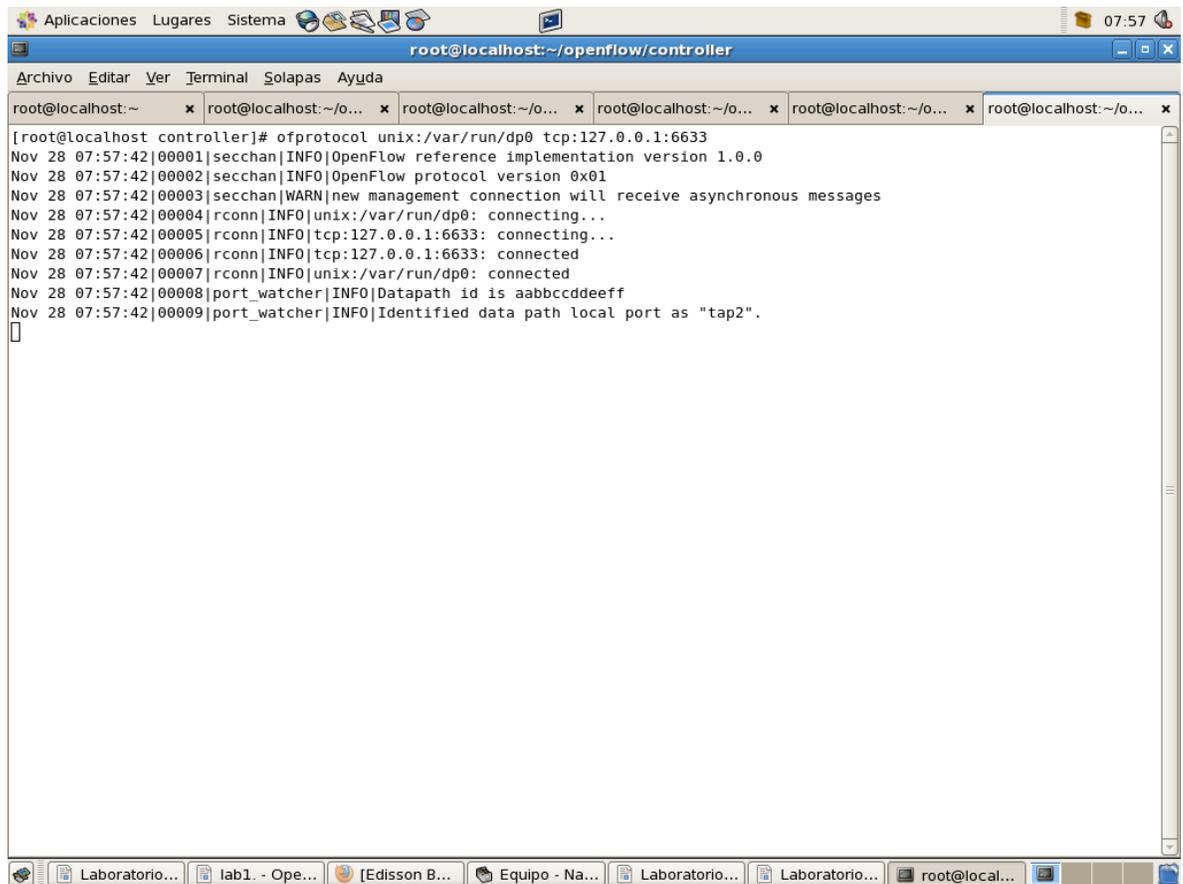


Imagen 7. Camino dp0.

Solapa 8: Comunicación directa con el controlador usando dirección IP y puerto (socket). Camino dp1.

```
ofprotocol unix:/var/run/dp1 tcp:127.0.0.1:6633
```

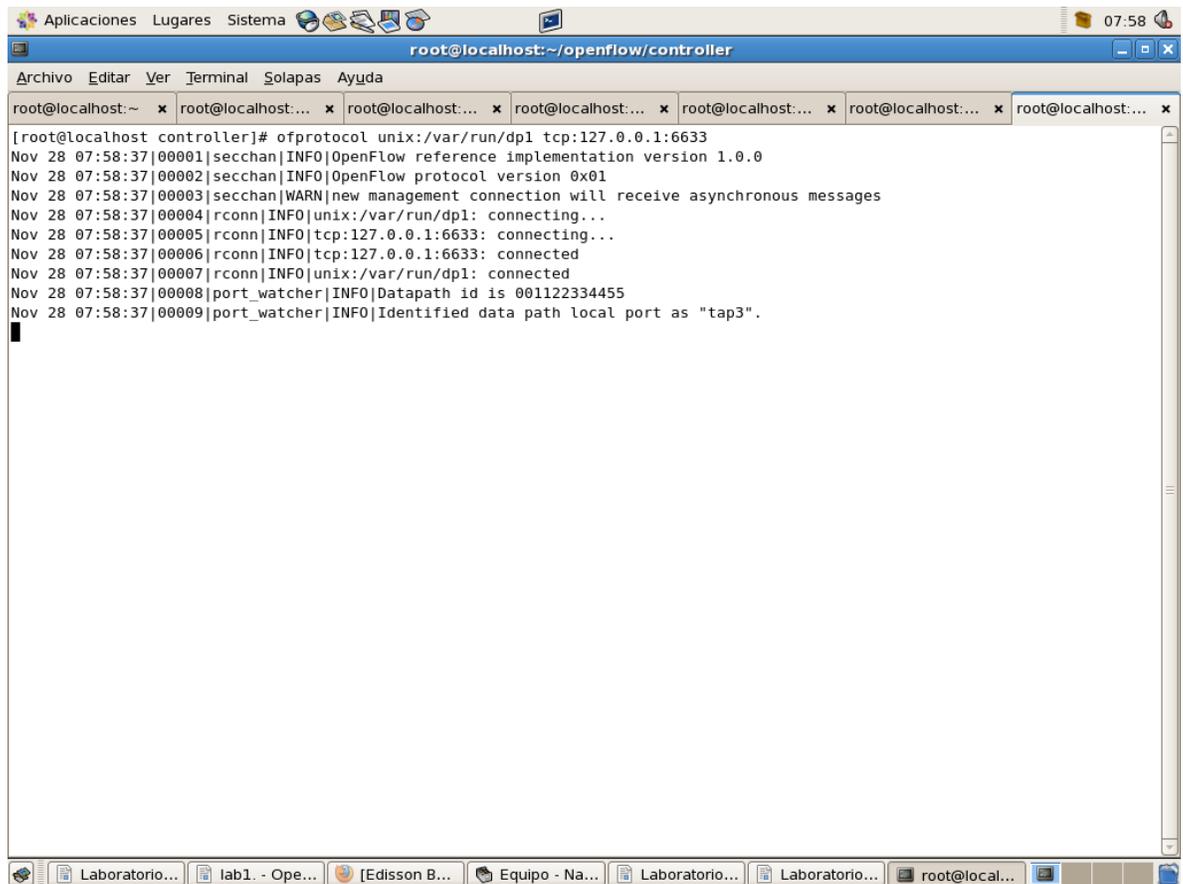


Imagen 8. Camino dp1.

Solapa 9: Ver el flujo del camino virtual dp0.

```
dpctl dump-flows unix:/var/run/dp0
```

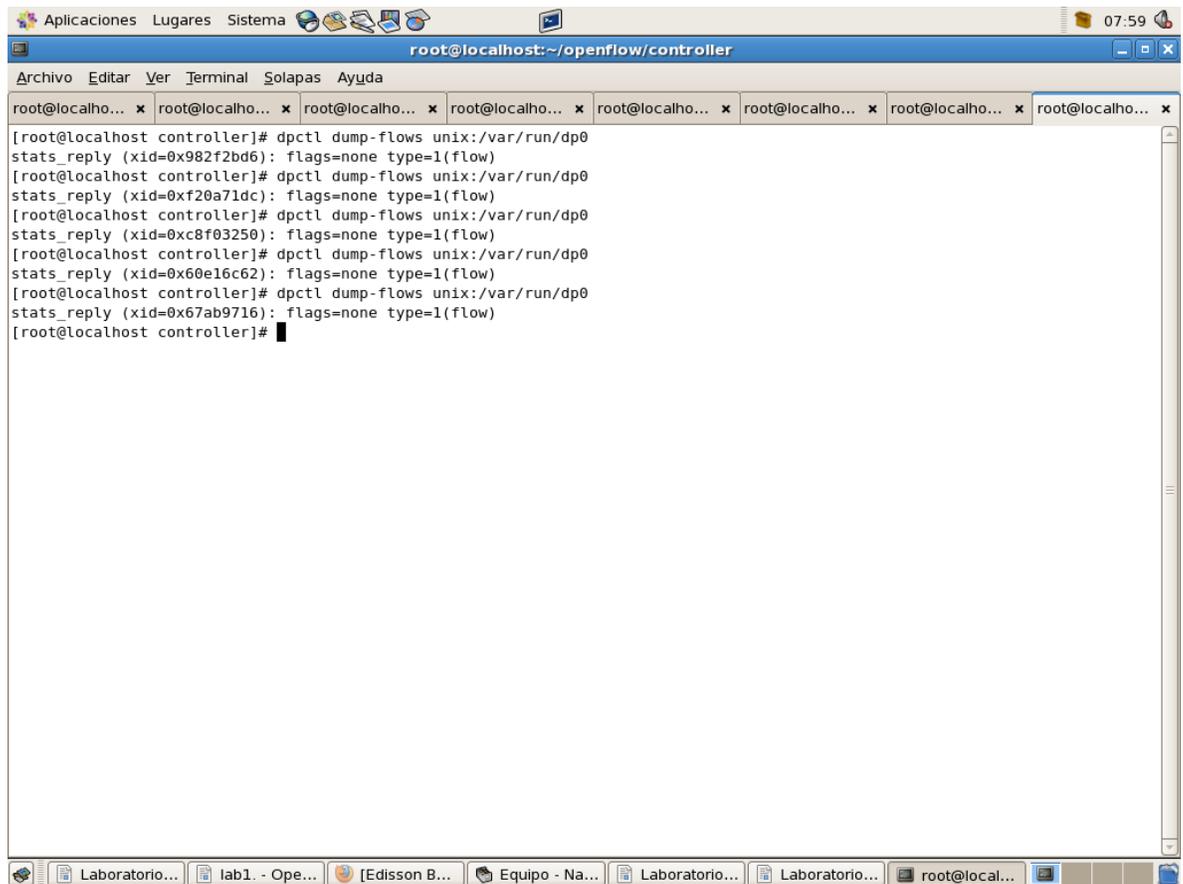


Imagen 9. Flujo dp0.

Solapa 10: Ver el flujo del camino virtual dp1.

```
dpctl dump-flows unix:/var/run/dp1
```

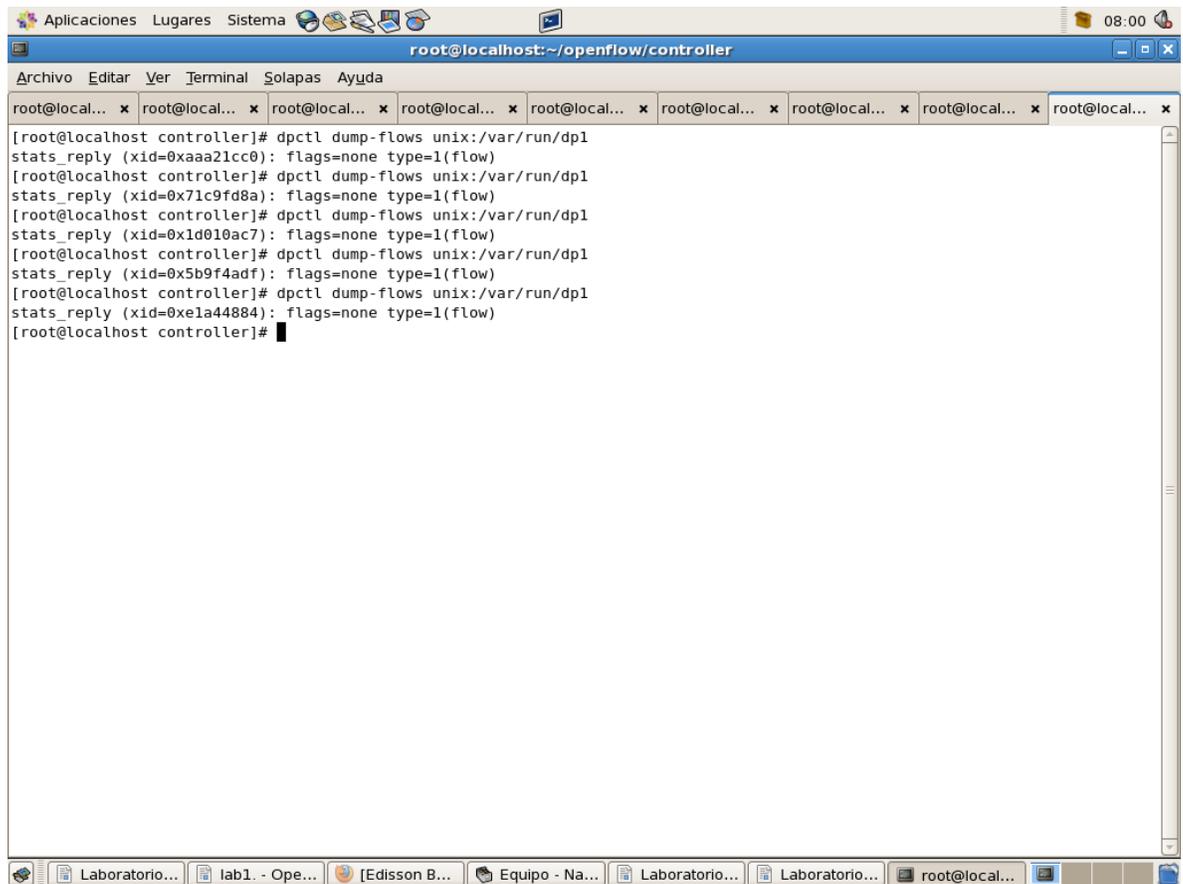


Imagen 10. Flujo dp1.

Desde PC cliente 1: Ping satisfactorio desde cliente 1 hacia cliente 2.

Conectar un patchcord desde el puerto eth1 del cliente 1 hacia el puerto nf2c3 de la NetFPGA. Configurar el puerto del cliente 1 con el siguiente comando.

```
ifconfig eth1 192.168.1.10
```

Conectar un patchcord desde el puerto eth1 del cliente 2 hacia el puerto nf2c2 de la NetFPGA. Configurar el puerto del cliente 2 con el siguiente comando.

```
ifconfig eth1 102.168.1.20
```

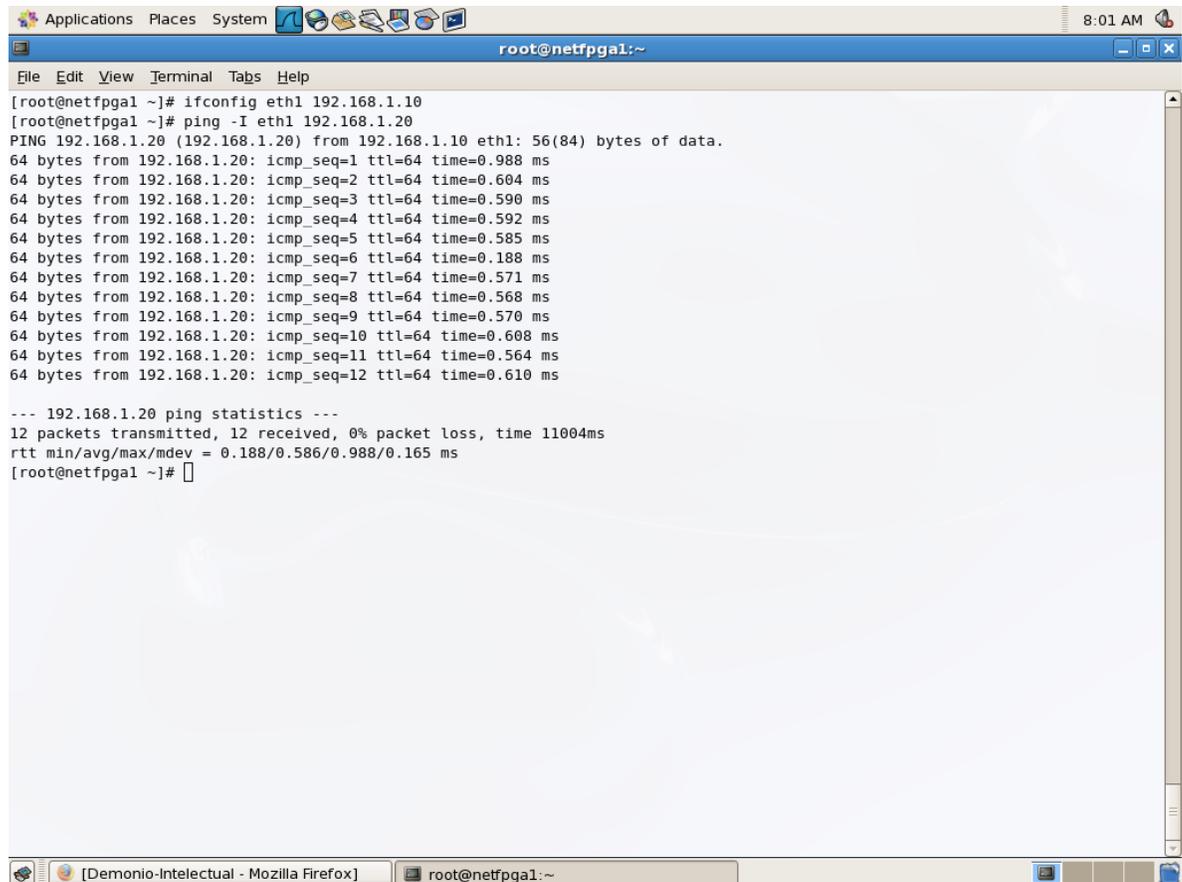
Después de haber configurado los puertos, desde el cliente 1 teclear el siguiente comando.

ping -I eth1 192.168.1.20

siendo:

eth1 puerto de origen.

192.168.1.20 IP de destino.



```
[root@netfpgal ~]# ifconfig eth1 192.168.1.10
[root@netfpgal ~]# ping -I eth1 192.168.1.20
PING 192.168.1.20 (192.168.1.20) from 192.168.1.10 eth1: 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=64 time=0.988 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=64 time=0.604 ms
64 bytes from 192.168.1.20: icmp_seq=3 ttl=64 time=0.590 ms
64 bytes from 192.168.1.20: icmp_seq=4 ttl=64 time=0.592 ms
64 bytes from 192.168.1.20: icmp_seq=5 ttl=64 time=0.585 ms
64 bytes from 192.168.1.20: icmp_seq=6 ttl=64 time=0.188 ms
64 bytes from 192.168.1.20: icmp_seq=7 ttl=64 time=0.571 ms
64 bytes from 192.168.1.20: icmp_seq=8 ttl=64 time=0.568 ms
64 bytes from 192.168.1.20: icmp_seq=9 ttl=64 time=0.570 ms
64 bytes from 192.168.1.20: icmp_seq=10 ttl=64 time=0.608 ms
64 bytes from 192.168.1.20: icmp_seq=11 ttl=64 time=0.564 ms
64 bytes from 192.168.1.20: icmp_seq=12 ttl=64 time=0.610 ms

--- 192.168.1.20 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11004ms
rtt min/avg/max/mdev = 0.188/0.586/0.988/0.165 ms
[root@netfpgal ~]#
```

Imagen 11. Ping satisfactorio.

Como se puede apreciar en la figura 11, el ping fue satisfactorio, lo que quiere decir que ya se pueden comunicar los dos clientes. Para dejar de enviar ping oprimir las teclas control + c. En este ejemplo, se enviaron 12 paquetes y los mismos fueron recibidos por el cliente 2 sin ninguna pérdida.

Desde PC cliente 1: Ping no satisfactorio desde el cliente 1 hacia el cliente 2.

Ahora, desconectar el puerto de la NetFPGA nf2c2 y conectarlo al puerto nf2c0 de la misma.

Hacer ping desde el cliente 1 nuevamente hacia el cliente 2 y mirar qué pasa.

ping -I eth1 192.168.1.20

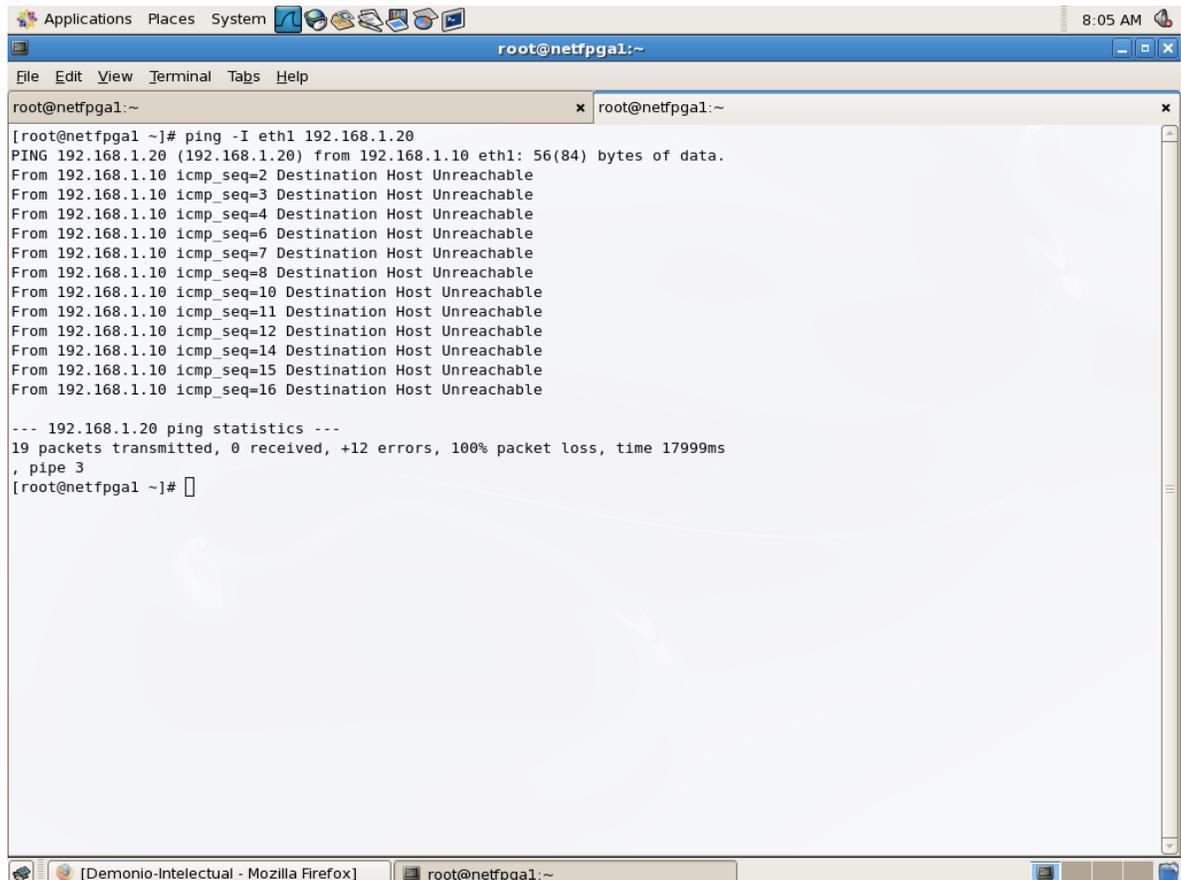
A screenshot of a Linux terminal window. The window title is 'root@netfpga1:~'. The terminal shows the execution of the command 'ping -I eth1 192.168.1.20'. The output consists of 19 lines of 'Destination Host Unreachable' messages, each with a different ICMP sequence number from 2 to 16. Below this, the ping statistics are displayed: '19 packets transmitted, 0 received, +12 errors, 100% packet loss, time 17999ms, pipe 3'. The prompt '[root@netfpga1 ~]#' is visible at the end of the terminal output. The terminal window is part of a desktop environment with a taskbar at the bottom showing a Firefox browser window and the terminal window itself.

Imagen 12. Ping no satisfactorio.

Como se puede apreciar en la figura 12, no se pudieron conectar los dos clientes; esto es debido a que al momento de la creación de los caminos, se estipuló que se iban a conectar virtualmente los puertos nf2c0, nf2c1 y los puertos nf2c2, nf2c3 de la NetFPGA. Por lo tanto, se creó una VLAN para que se pudieran conectar dichos puertos entre sí. Un ejemplo de esta aplicación puede ser en una empresa, en la que se crea una red para que todos los empleados se puedan conectar a ella, pero el dueño de la empresa quiere que en esa misma red exista una manera

que se comunique el área de presidencia con el área administrativa sin que los subordinados tengan acceso a la información de sus superiores.

ACTIVIDAD.

Comunicar los puertos de la NetFPGA nf2c0 (cliente 1) con el nf2c1 (cliente 2) y el puerto nf2c1 (cliente 2) con el nf2c2 (cliente 1) y observar el resultado de la comunicación.

4.5.3 Compartir a internet por medio del switch Openflow

Introducción

Esta práctica de laboratorio consiste en crear dos *datapaths* en los cuales, en uno, se conecte a internet por medio de la red de la Universidad, mientras que en el otro se crea un *switch* Openflow para que se conecten los distintos clientes, y así unir los dos *datapaths* por medio de un *patchcord* y ofrecer internet al switch.

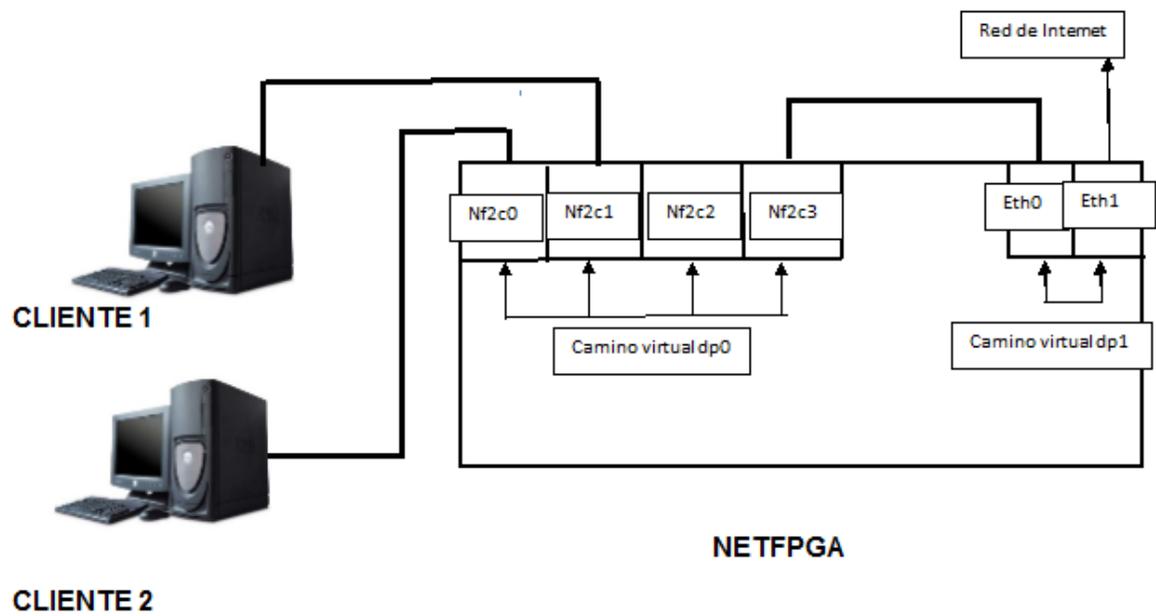
Objetivos

- Hacer la conexión entre los dos *datapaths* para ofrecer internet a los puertos de la NetFGPA.

Materiales y equipos a utilizar

- 1 NetFPGA con GNU/Linux distribución CentOS y con el paquete de Openflow instalado para que funcione como switch.
- 3 computadores con GNU/Linux distribución CentOS para ser conectados como clientes.
- 4 patchcords.

Diagrama de laboratorio



Procedimiento:

Solapa 1: Limpiar la memoria de la NetFPGA.

`cpci_reprogram.pl`

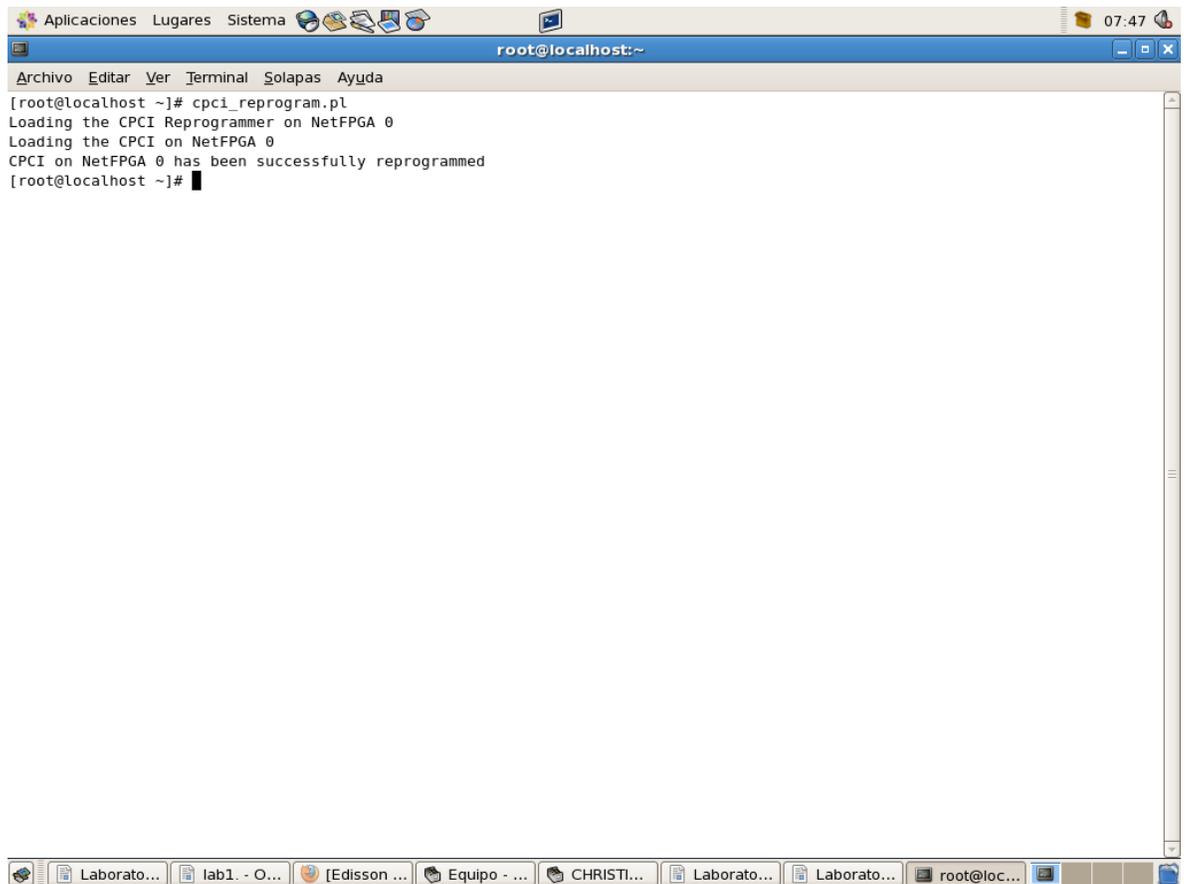


Imagen 1. Limpiar la memoria de la NetFPGA.

Solapa 2: Cargar el bitfile del switch. Ir a la ruta raíz del Openflow (donde está instalado). Para esto, seguir la siguiente ruta.

```
cd /root/openflow/hw-lib/nf2
```

```
nf_download openflow_switch.bit
```

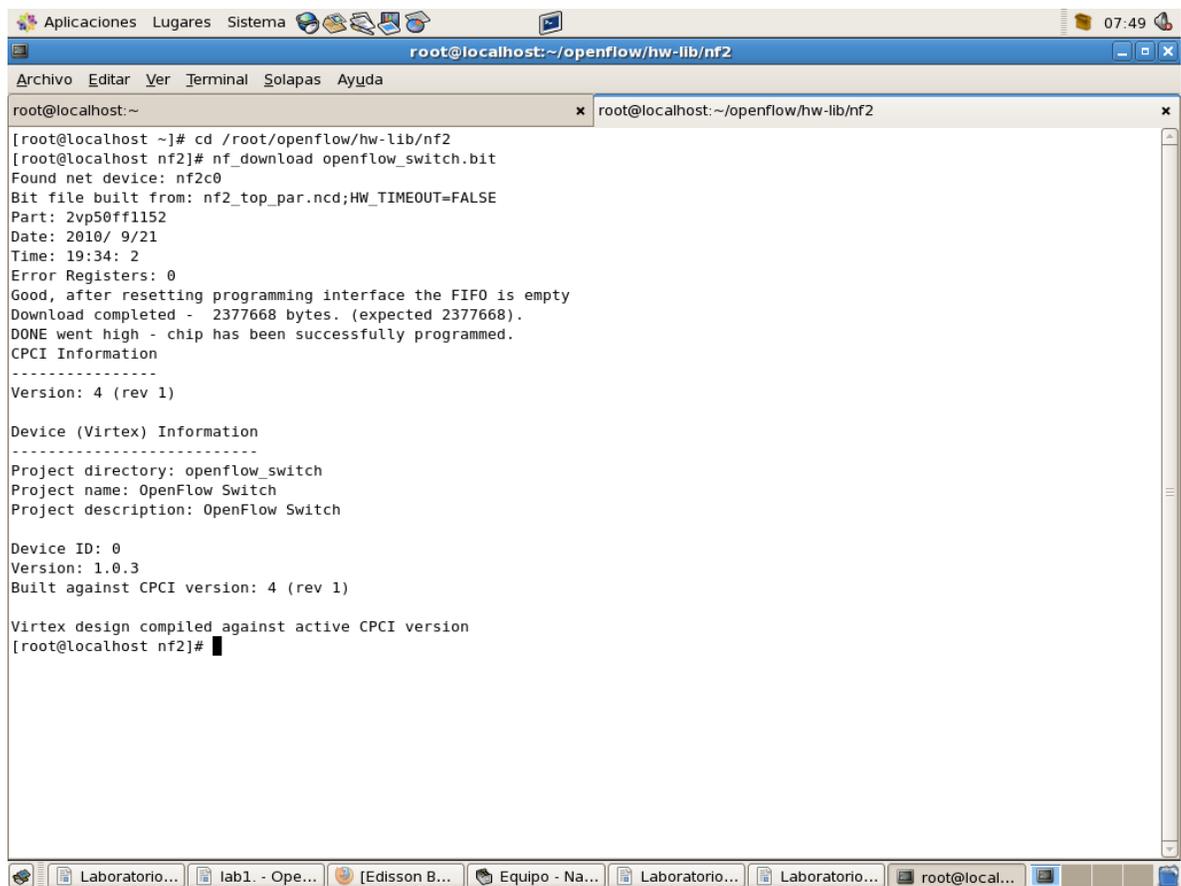


Imagen 2. Cargar el bitfile del switch.

En la figura 2, se debe comprobar que en la opción Error Registers aparezca un cero (0), de contrario, reprogramar la NetFPGA.

Solapa 3: Cargar el controlador y enlazarlo a un puerto tcp. Para esto, buscar el ejecutable controller en la carpeta controller cuya ruta es la siguiente.

```
cd /root/openflow/controller
```

```
./controller ptcp:6633
```

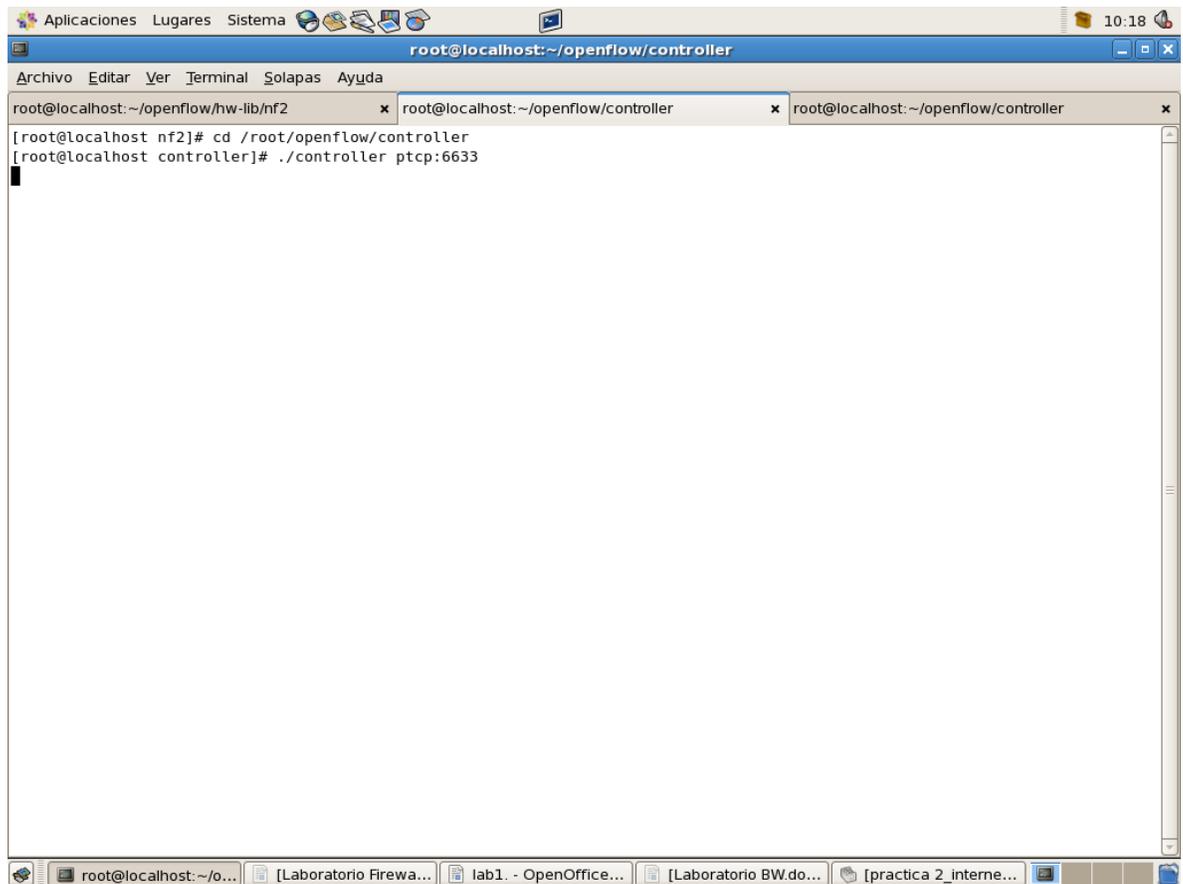


Imagen 3. Cargar el controlador.

Solapa 4: Creación del camino virtual dp0 en tap4 con ID del datapath AA:BB:CC:DD:EE y puertos nf2c0, nf2c1, nf2c2, nf2c3 de la NetfPGA.

```
ofdatapath --detach punix:/var/run/dp0 -d aabbccddeeff -i nf2c0,nf2c1,nf2c2,nf2c3  
&& ifconfig | grep tap
```

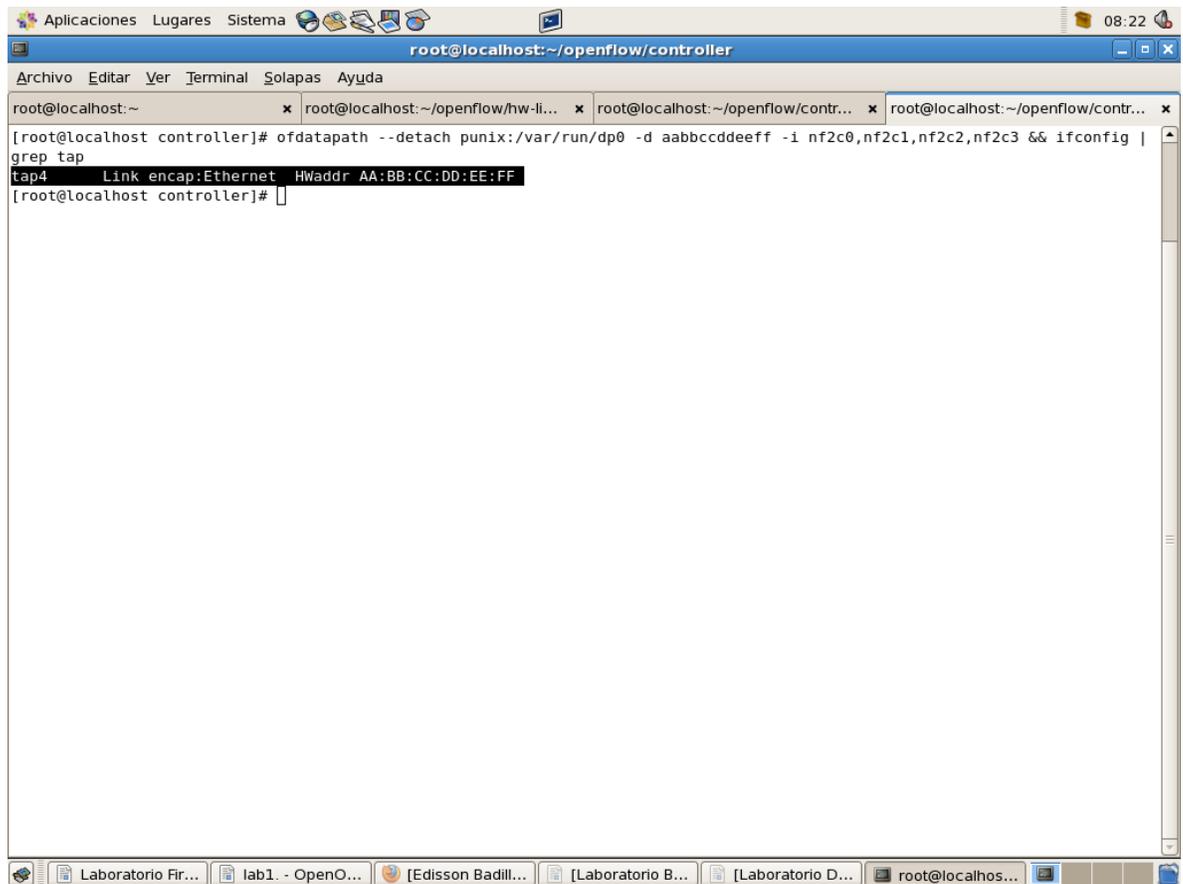


Imagen 4. Camino virtual dp0.

Solapa 5: Creación del camino virtual dp1 en tap5 con ID del datapath 11:22:33:44:55 y puertos de red eth1, eth2 de la NetfPGA.

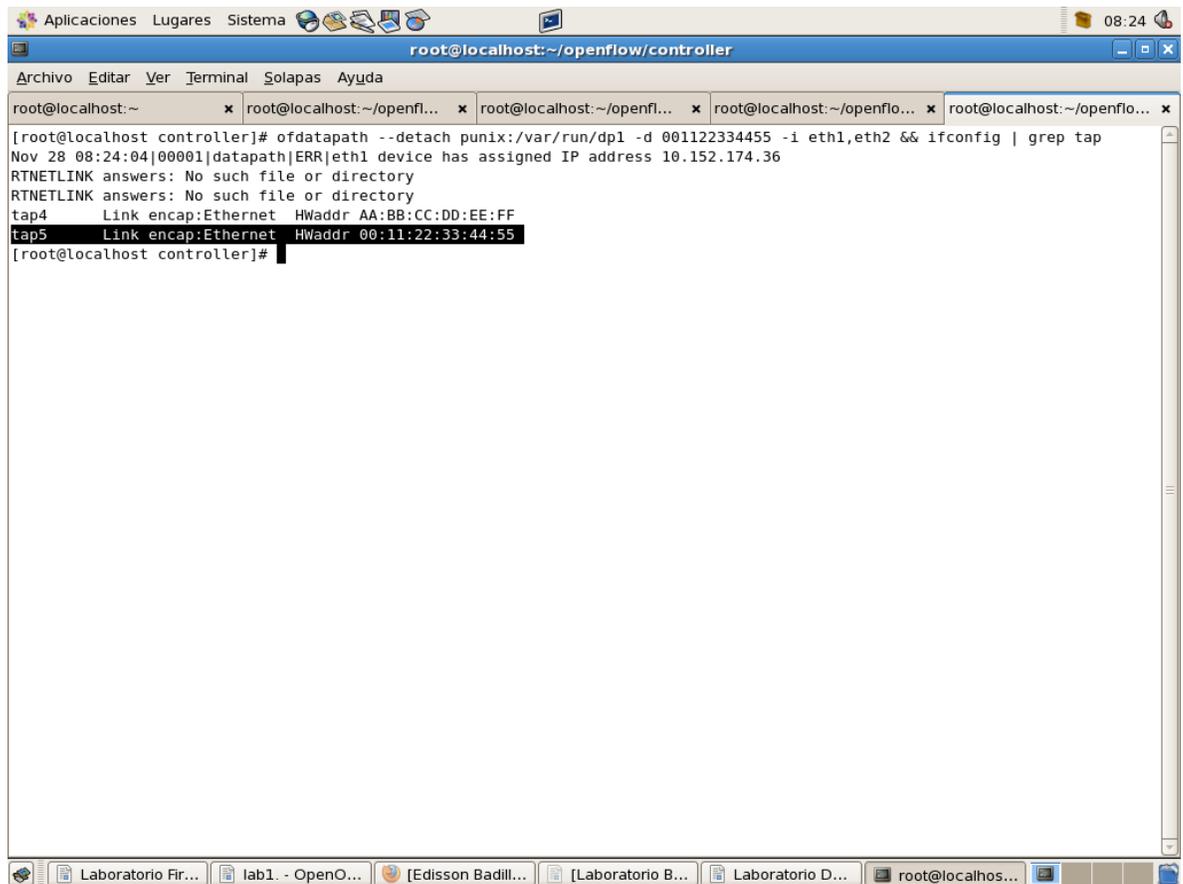


Imagen 5. Camino virtual dp1.

Solapa 6: Comunicación directa con el controlador usando dirección IP y puerto (socket). Camino dp0.

ofprotocol unix:/var/run/dp0 tcp:127.0.0.1:6633

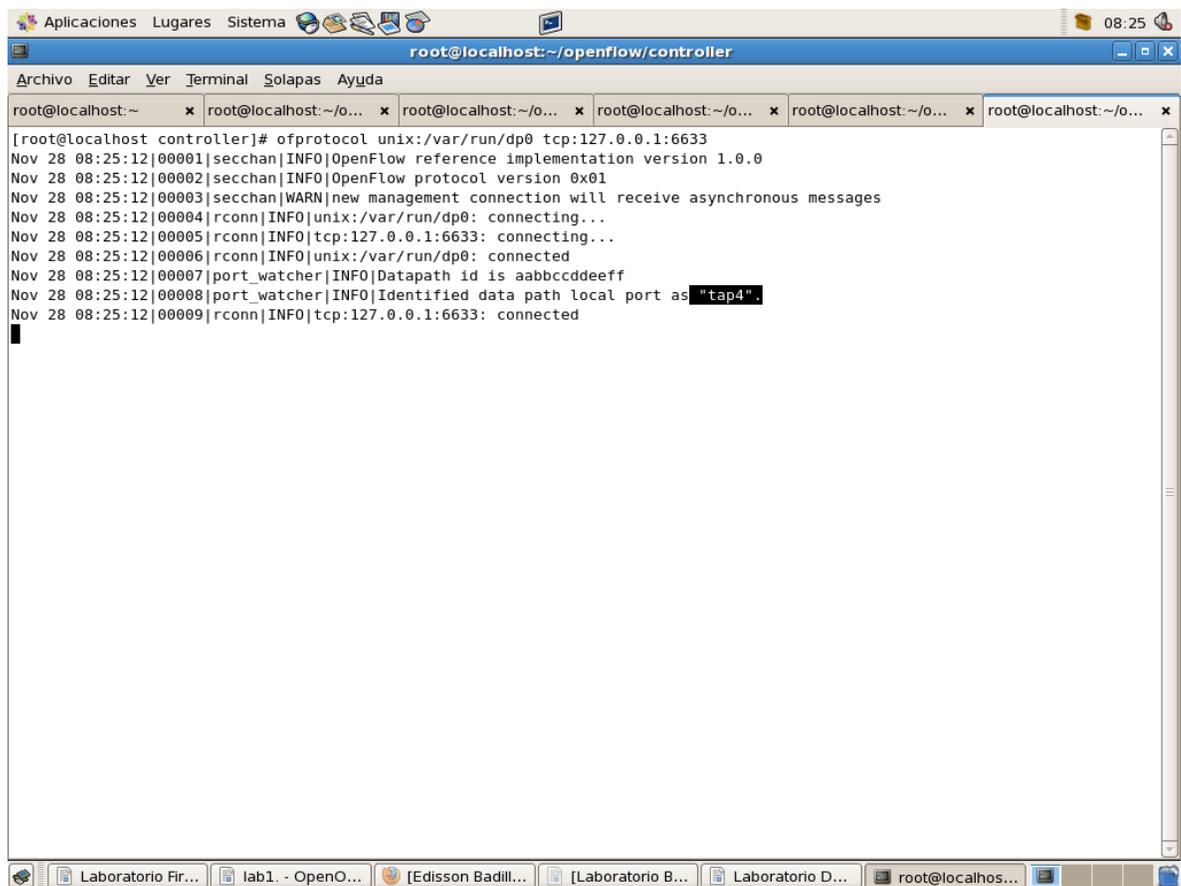


Imagen 6. Camino dp0.

Solapa 7: Comunicación directa con el controlador usando dirección IP y puerto (socket). Camino dp1.

```
ofprotocol unix:/var/run/dp1 tcp:127.0.0.1:6633
```

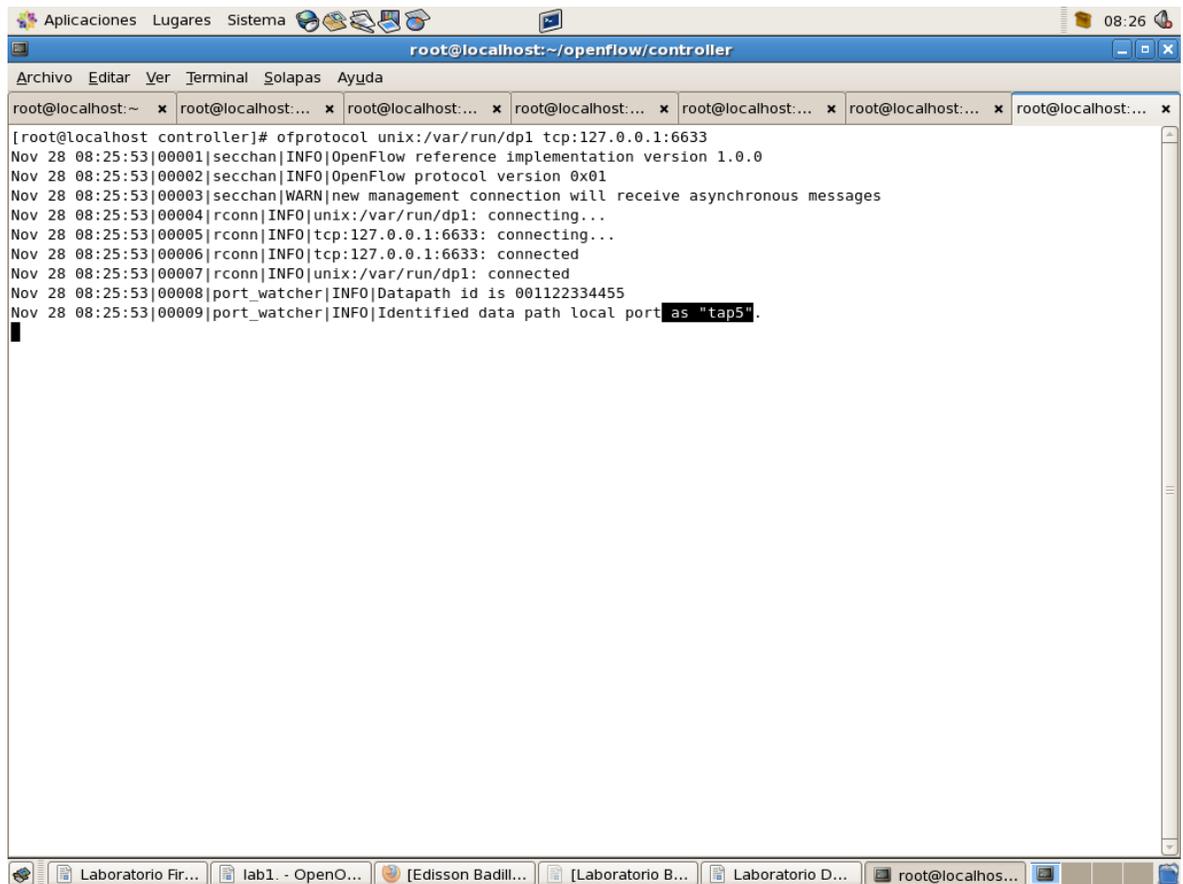


Imagen 7. Camino dp1.

Solapa 8: Ver el flujo del camino virtual dp0.

```
dpctl dump-flows unix:/var/run/dp0
```

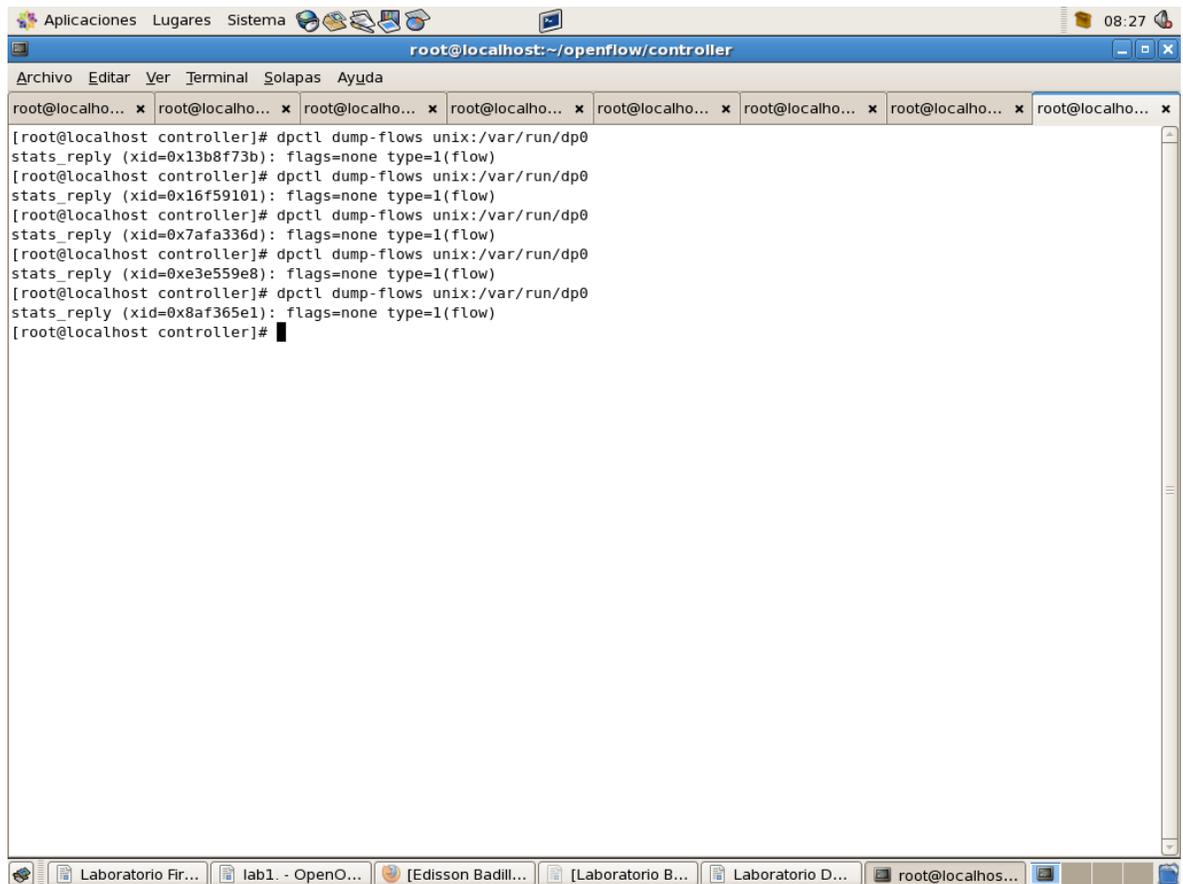


Imagen 8. Flujo dp0.

Solapa 9: Ver el flujo del camino virtual dp1.

```
dpctl dump-flows unix:/var/run/dp1
```

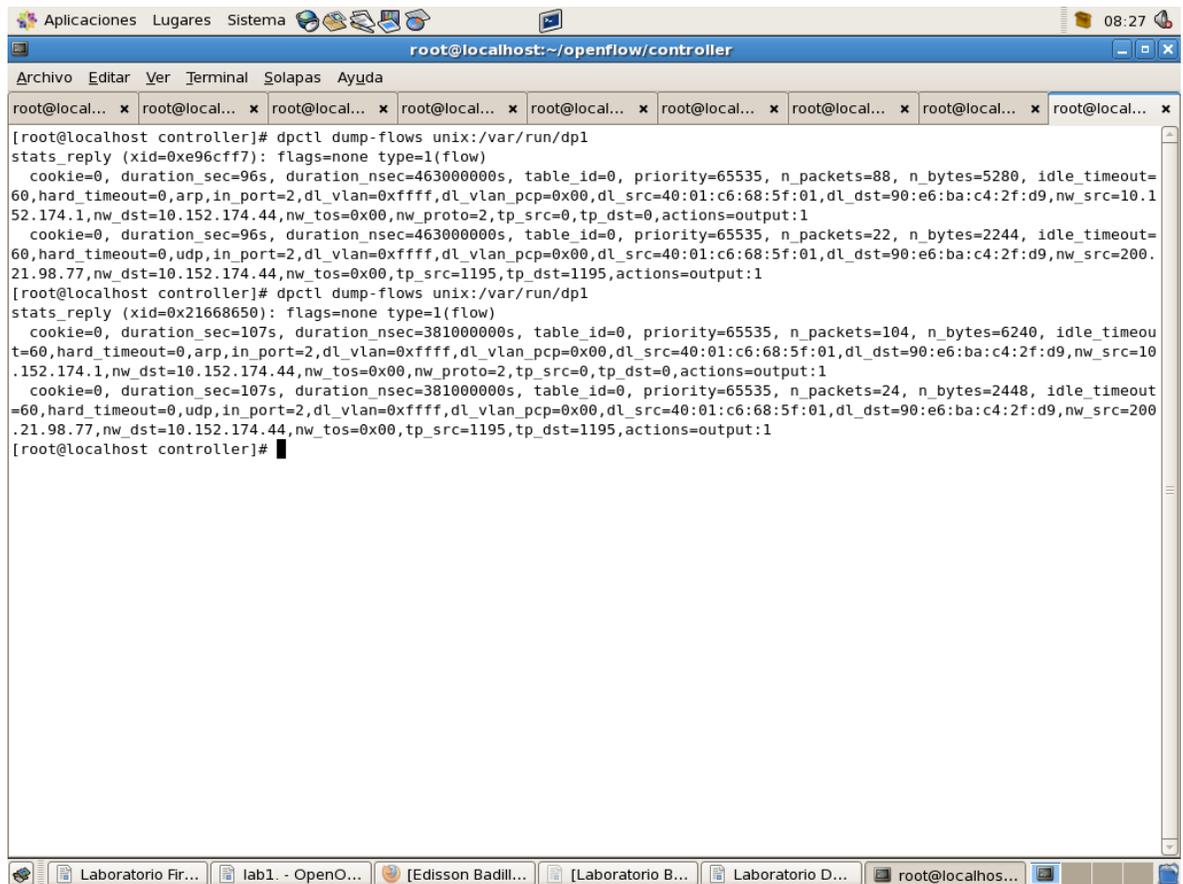


Imagen 9. Flujo dp1.

Antes de seguir con el próximo paso, comprobar que no hay conexión de red DNS en ninguno de los clientes a utilizar, tecleando ping 4.2.2.2. Esto es una herramienta de red de computadora utilizado para comprobar si un host en particular es accesible a través de una red IP.

Para cliente 1:

Ping 4.2.2.2

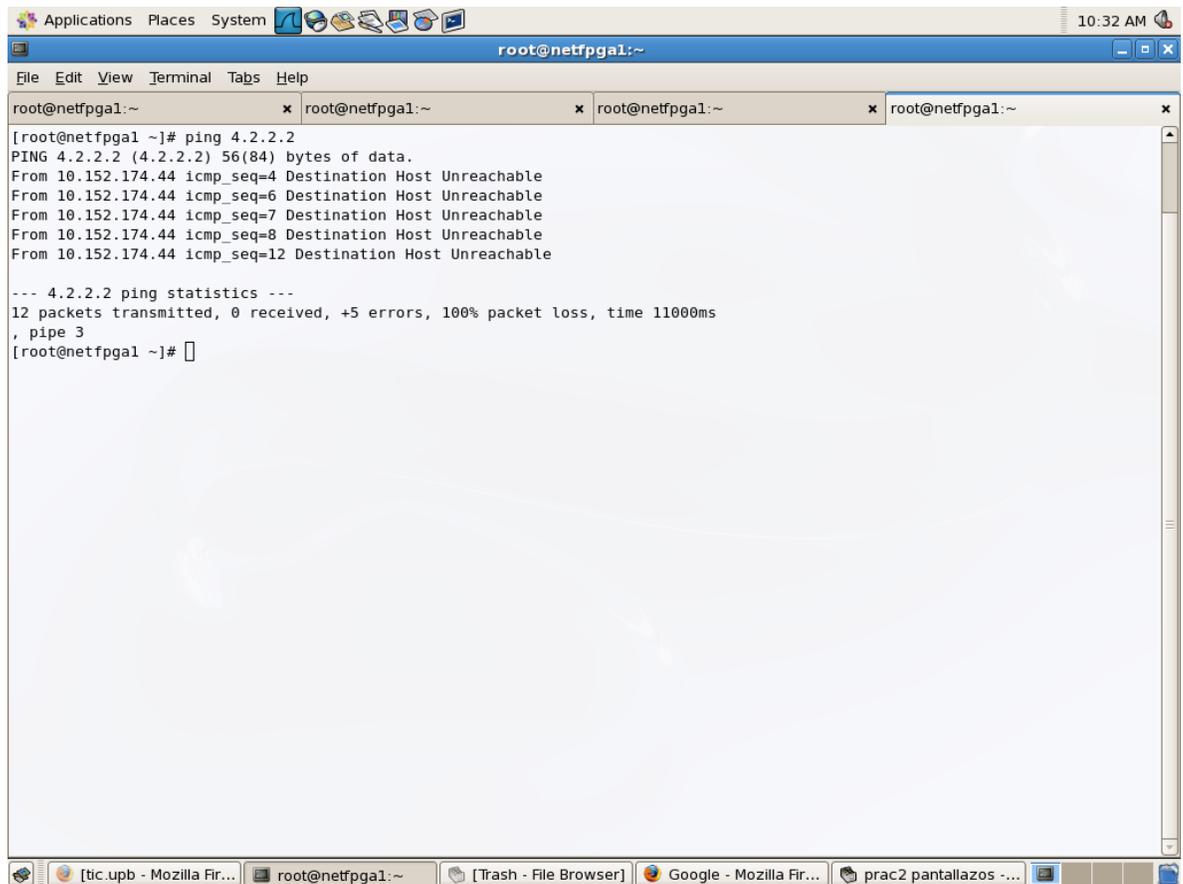


Imagen 10. Ping cliente 1.

Hacer el mismo proceso para el cliente 2.

Solapa 10: Poner las IP de los puertos de red del servidor en cero, para que funcione como switch.

```
ifconfig eth1 0.0.0.0
ifconfig eth2 0.0.0.0
```

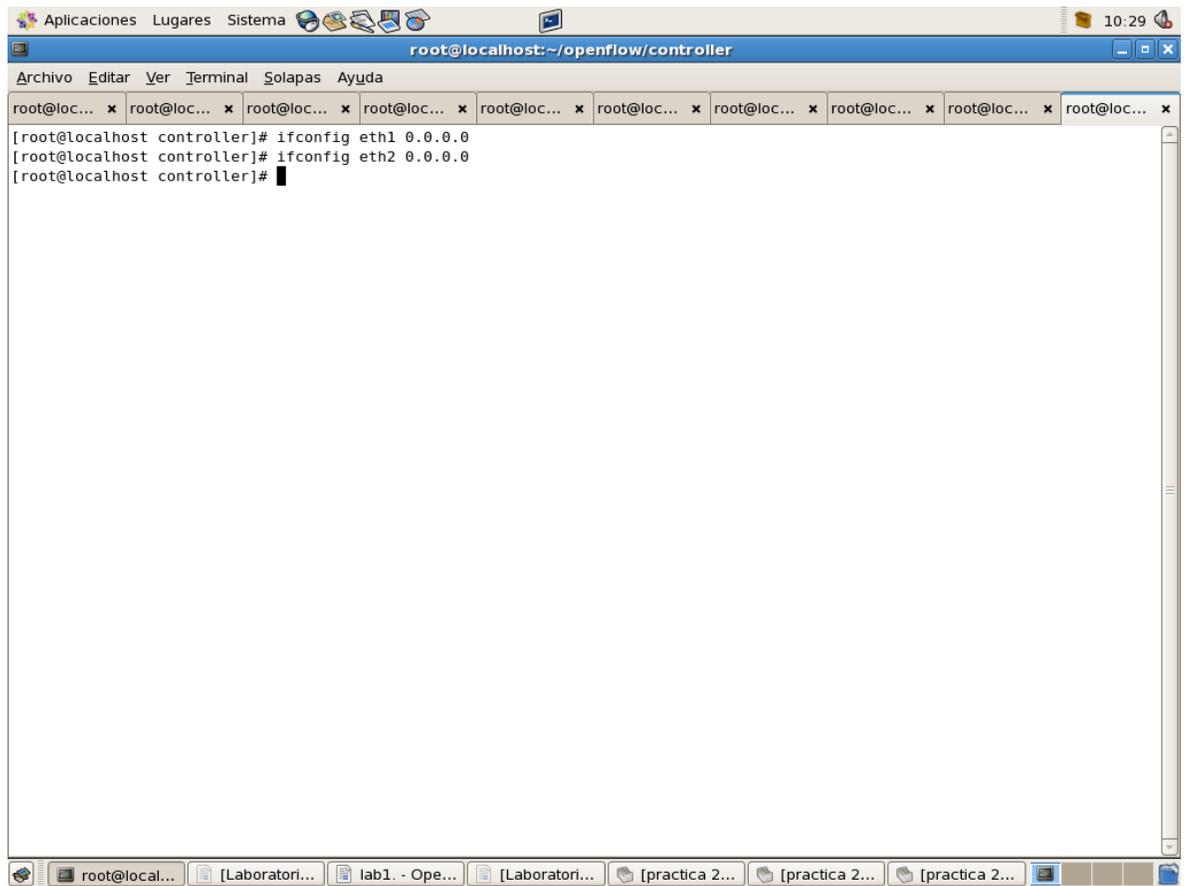


Imagen 11. Ping cliente 2.

Para cliente 1: Renovar la dirección IP para la interfaz eth0.

```
killall dhclient
dhclient eth0
```

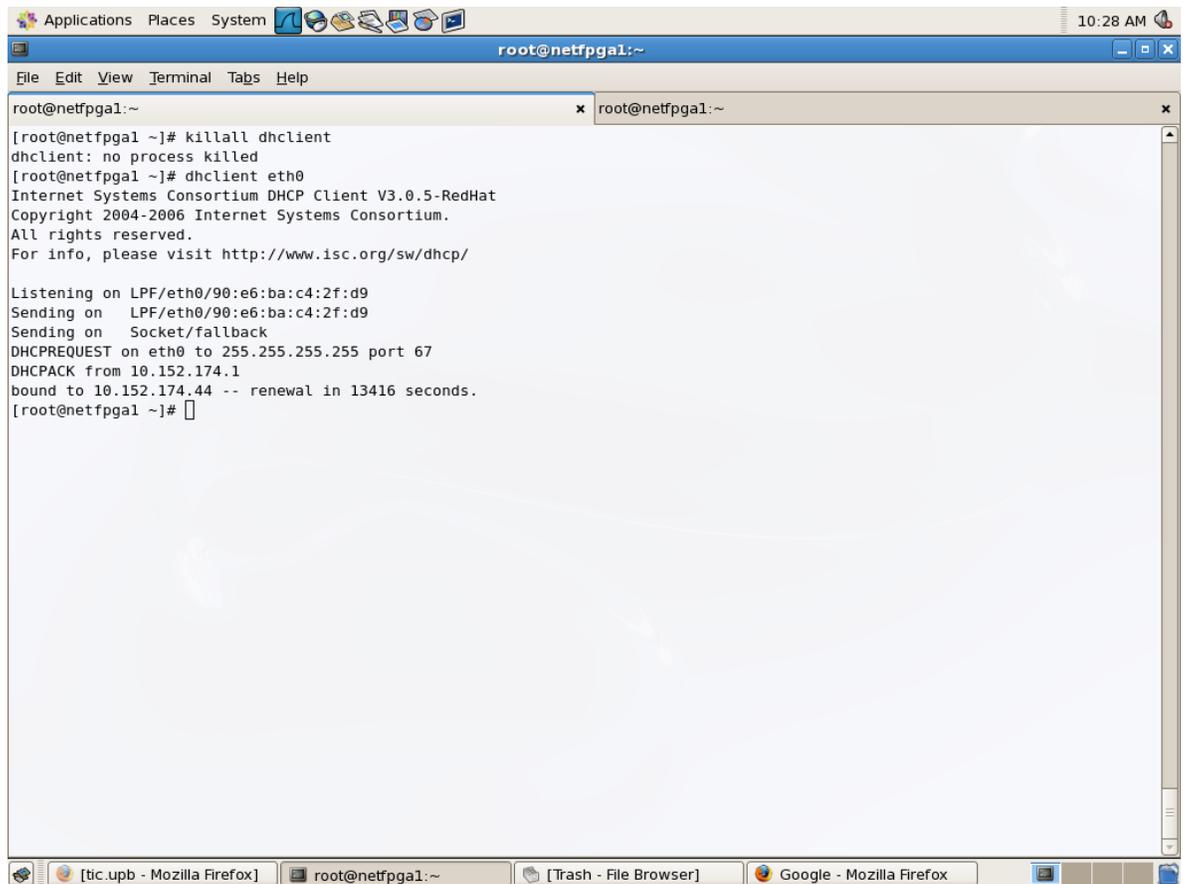


Imagen 12. Renovar IP.

Hacer prueba de ping nuevamente para comprobar que ahora si hay conexión.

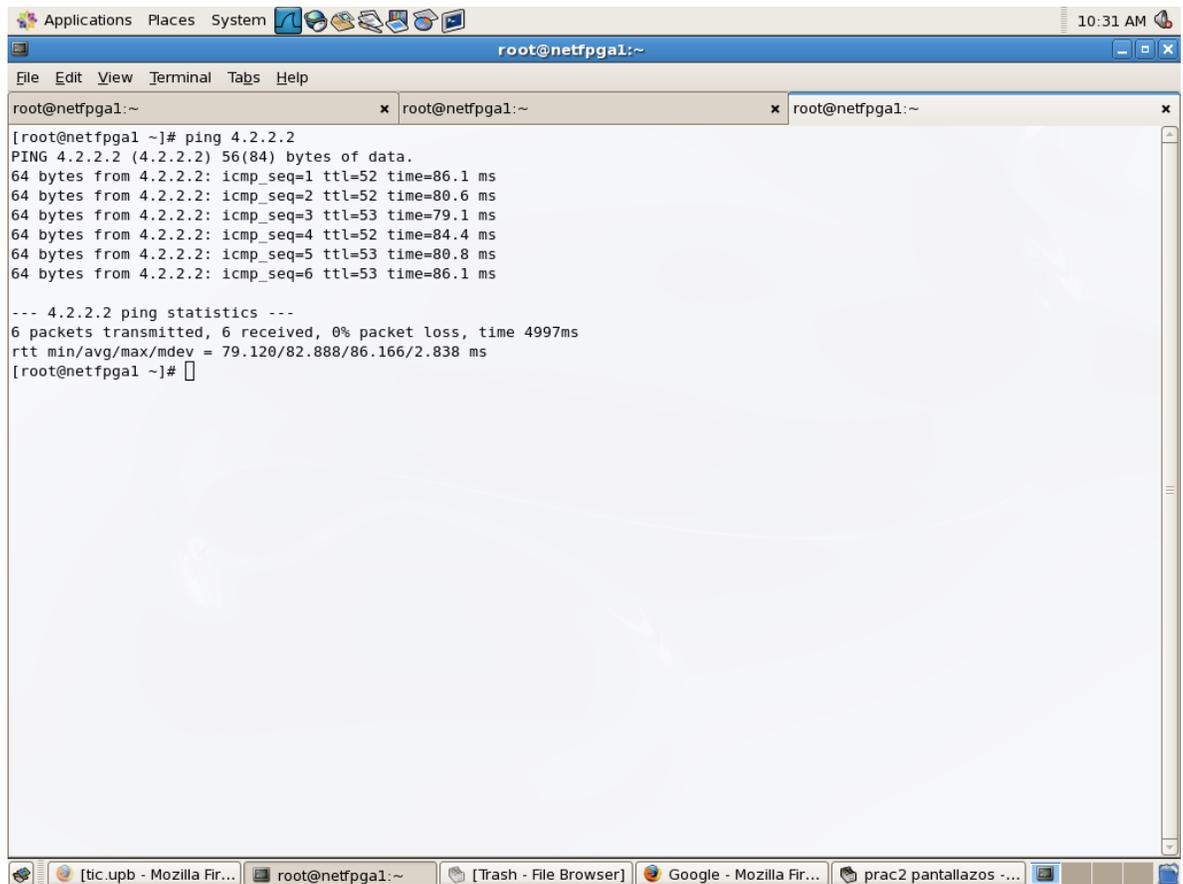


Imagen 13. Prueba de ping.

Para cliente 2: Renovar la dirección IP para la interfaz eth1.

```
killall dhclient
dhclient eth1
```

ACTIVIDAD.

Introducir un cliente más al switch Openflow mediante la asignación del comando dhclient.

4.5.4 Servidor DHCP dinámico con dos puertos

Introducción

Este laboratorio especifica cada uno de los pasos necesarios para instalar y configurar un servidor DHCP en GNU/Linux para interconectar dos clientes por medio de la NETFPGA usando ésta como un switch por medio de Openflow. Como servidor DHCP se va a usar ISC, el cual se ha convertido en el estándar de facto para DHCP y es utilizado por numerosas organizaciones para gestionar y asignar direcciones IP de forma dinámica.

Objetivos

- Configurar un servidor DHCP dinámico en GNU/LINUX.
- Configurar la NetFPGA como switch Openflow.
- Aprender a gestionar y asignar direcciones IP dinámicas.

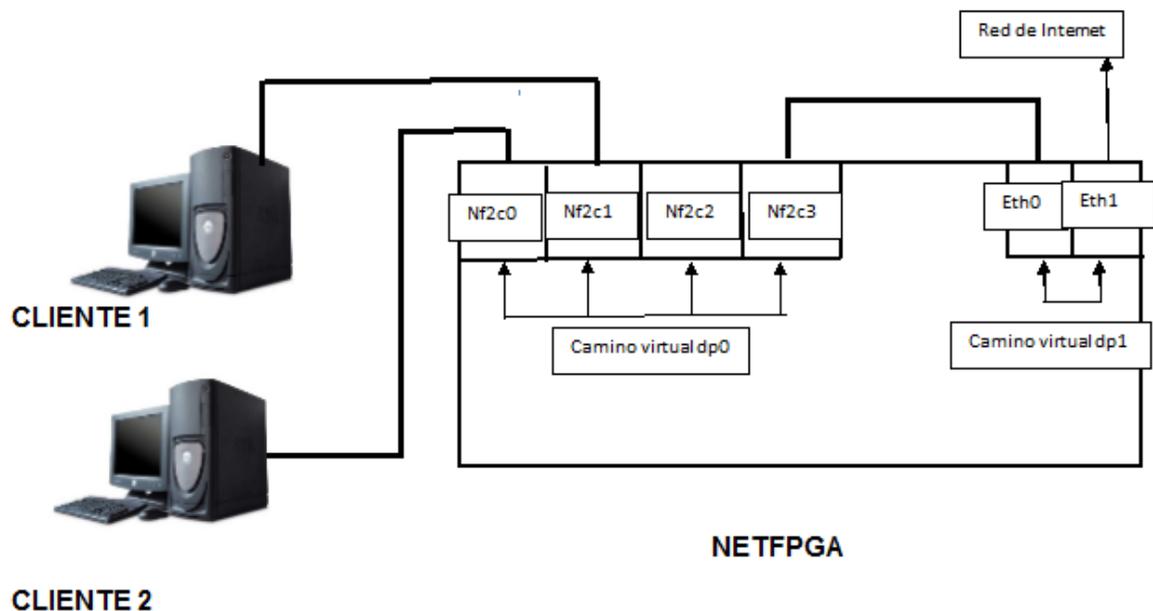
Materiales y equipos a utilizar

- 1 NetFPGA con GNU/Linux distribución CentOS y con el paquete de Openflow instalado para que funcione como servidor.
- 3 computadores con GNU/Linux distribución CentOS para ser conectados como clientes.
- 4 patchcords.

Marco teórico

DHCP es un protocolo de red que permite a los clientes conectarse a un servidor por medio de una red IP de forma automática. Es decir, se trata de un protocolo de tipo cliente-servidor en el que el servidor tiene pre configuradas una lista de direcciones IP dinámicas y a medida que los clientes se van conectando, el servidor va asignando dichas direcciones a los diferentes clientes, obteniendo en una base de datos información acerca de quién tiene cada dirección IP, el tiempo en el cual se conectó, etc. Como esta práctica se trata de asignar las direcciones IP de forma dinámica, el servidor determina un rango de direcciones IP para que los clientes se conecten por un periodo de tiempo, por lo que, en la base de datos aparecerá también el tiempo de finalización de la conexión.

Diagrama de laboratorio



Procedimiento:

Solapa 1: Limpiar la memoria de la NetFPGA.

cpci_reprogram.pl

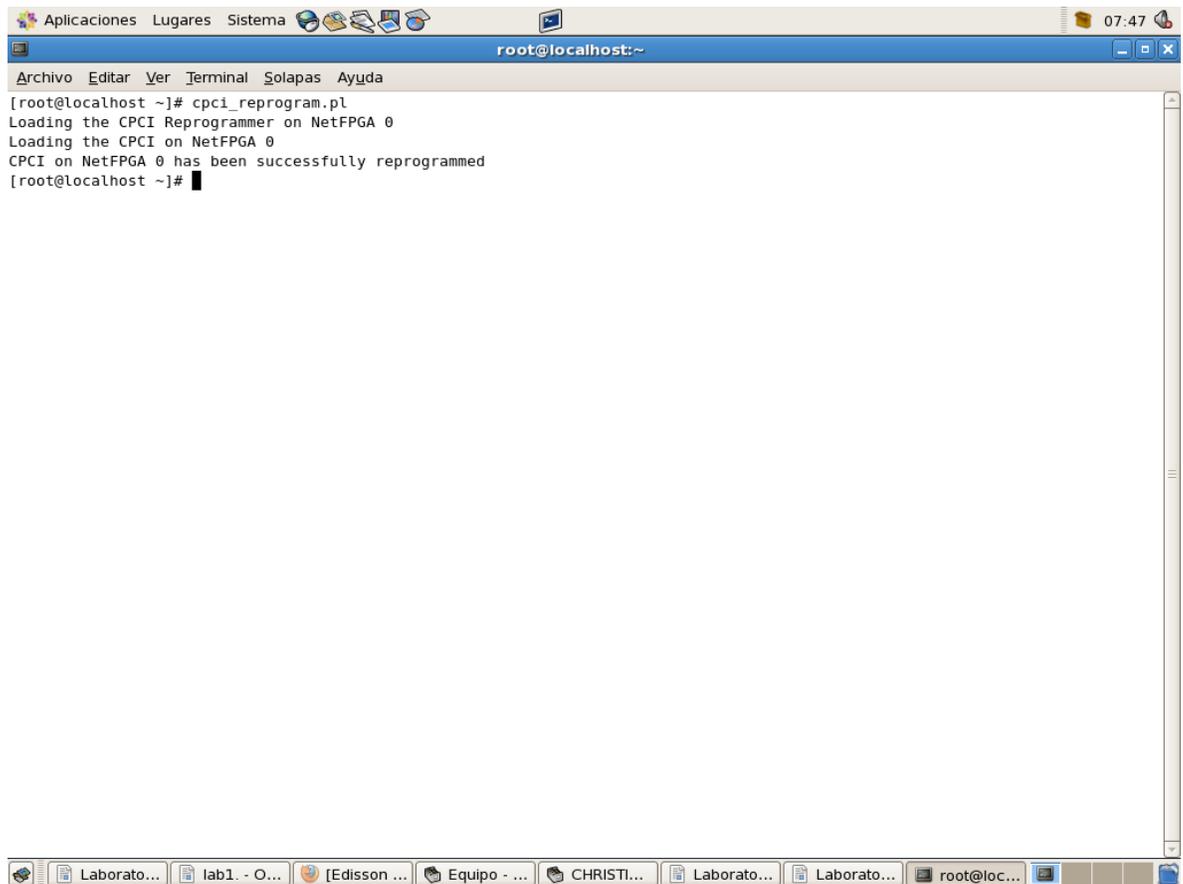
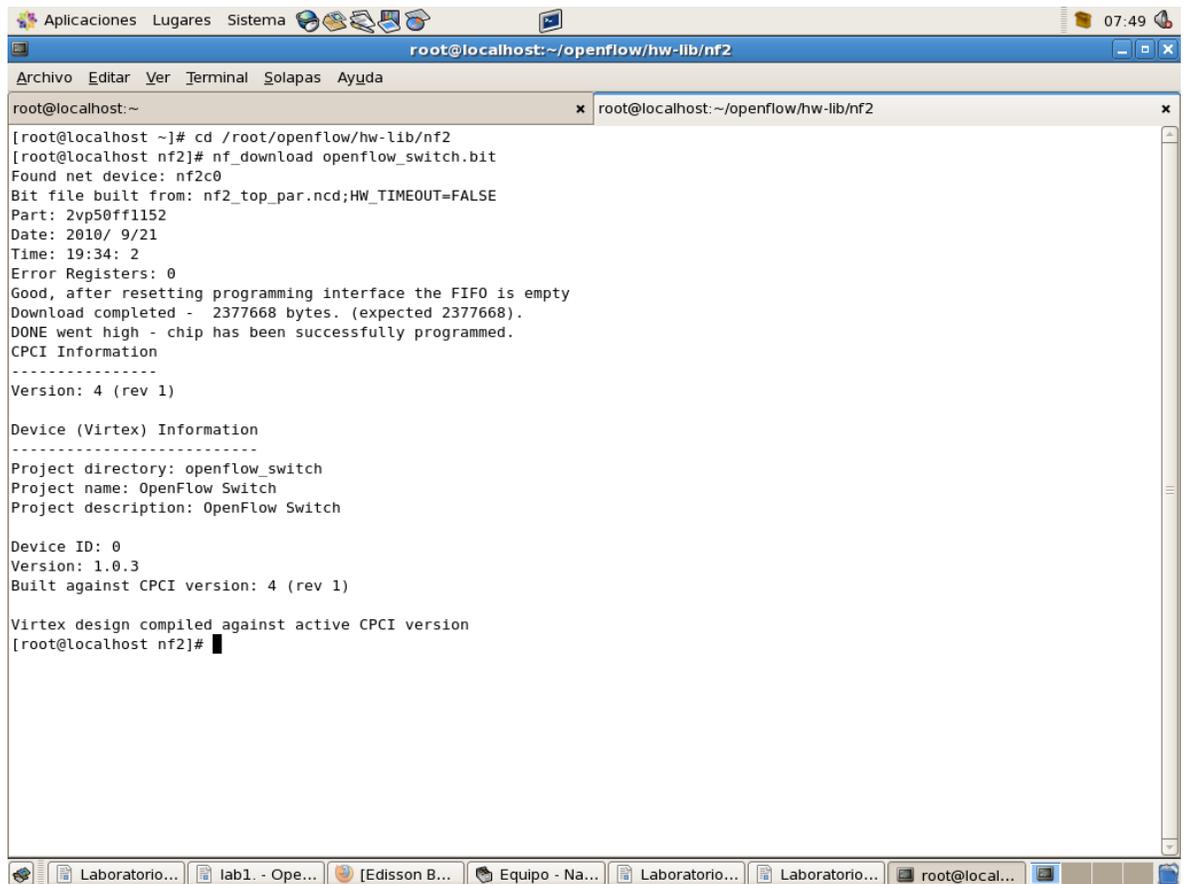


Imagen 1. Limpiar la memoria de la NetFPGA.

Solapa 2: Cargar el bitfile del switch. Ir a la ruta raíz del Openflow (donde está instalado). Para esto, seguir la siguiente ruta.

```
cd /root/openflow/hw-lib/nf2
```

```
nf_download openflow_switch.bit
```



```
root@localhost:~/openflow/hw-lib/nf2
[root@localhost ~]# cd /root/openflow/hw-lib/nf2
[root@localhost nf2]# nf_download openflow_switch.bit
Found net device: nf2c0
Bit file built from: nf2_top_par.ncd;HW_TIMEOUT=FALSE
Part: 2vp50ff1152
Date: 2010/ 9/21
Time: 19:34: 2
Error Registers: 0
Good, after resetting programming interface the FIFO is empty
Download completed - 2377668 bytes. (expected 2377668).
DONE went high - chip has been successfully programmed.
CPCI Information
-----
Version: 4 (rev 1)

Device (Virtex) Information
-----
Project directory: openflow_switch
Project name: OpenFlow Switch
Project description: OpenFlow Switch

Device ID: 0
Version: 1.0.3
Built against CPCI version: 4 (rev 1)

Virtex design compiled against active CPCI version
[root@localhost nf2]#
```

Imagen 2. Cargar el bitfile del switch.

En la figura 2, se debe comprobar que en la opción Error Registers aparezca un cero (0), de contrario, reprogramar la NetFPGA.

Solapa 3: Cargar el controlador y enlazarlo a un puerto tcp. Para esto, buscar el ejecutable controller en la carpeta controller cuya ruta es la siguiente.

```
cd /root/openflow/controller
```

```
./controller ptcp:6633
```

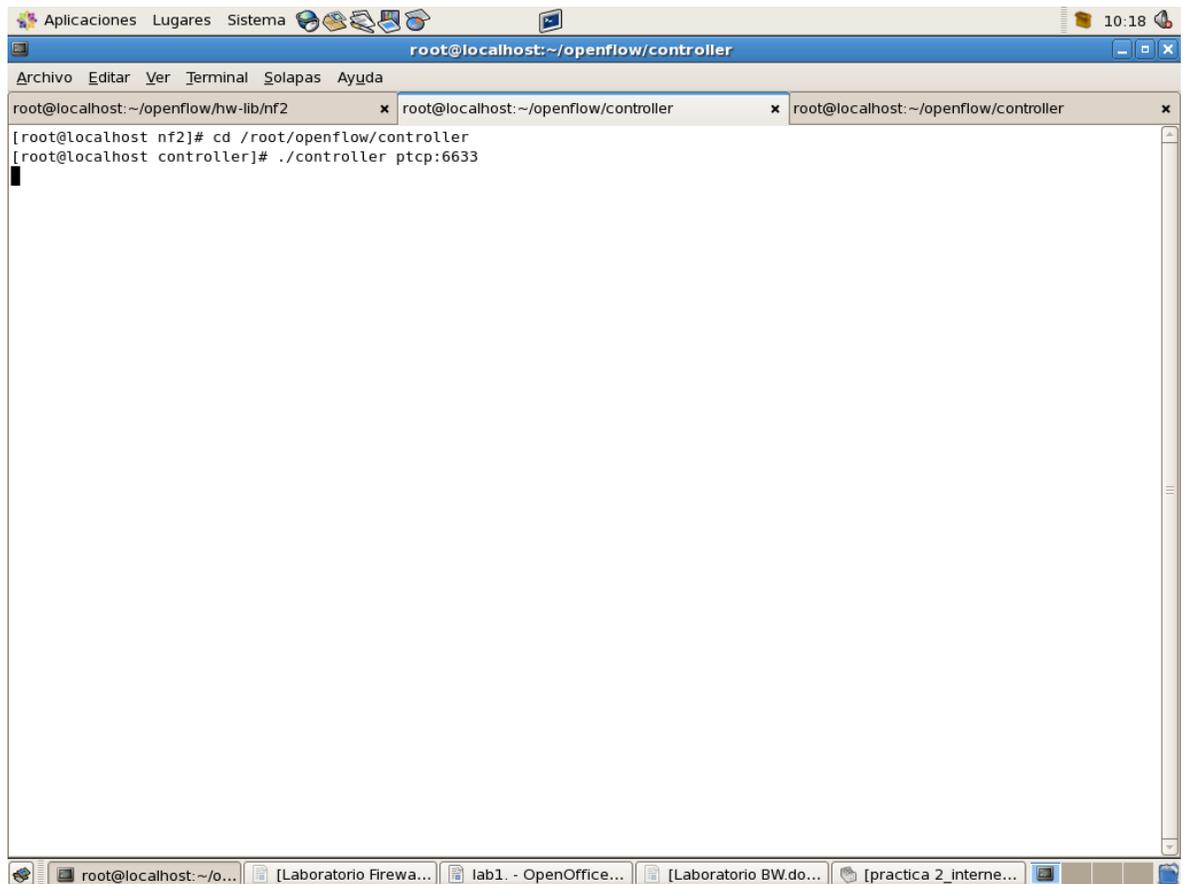


Imagen 3. Cargar el controlador.

Solapa 4: Creación del camino virtual dp0 en tap4 con ID del datapath AA:BB:CC:DD:EE y puertos nf2c0, nf2c1, nf2c2, nf2c3 de la NetfPGA.

```
ofdatapath --detach punix:/var/run/dp0 -d aabbccddeeff -i nf2c0,nf2c1,nf2c2,nf2c3
&& ifconfig | grep tap
```

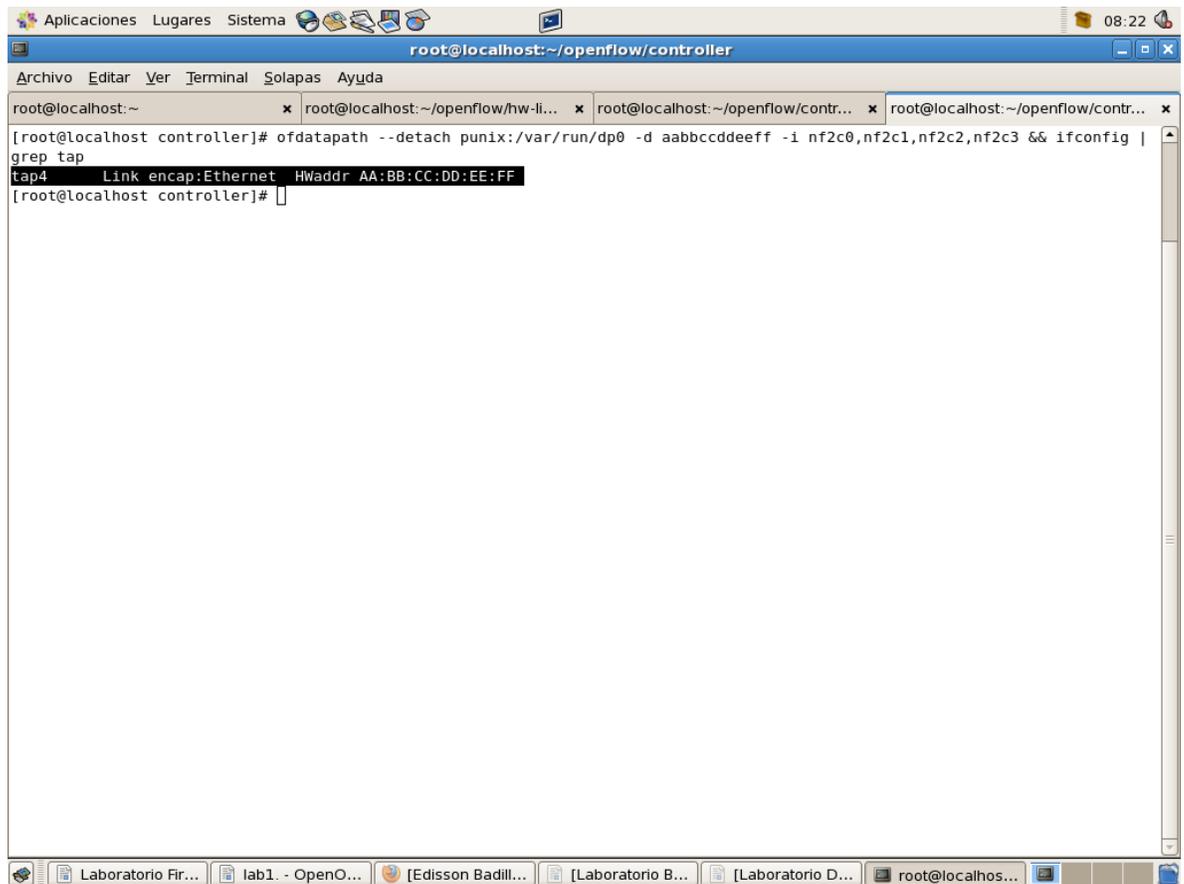


Imagen 4. Camino virtual dp0.

Solapa 5: Creación del camino virtual dp1 en tap5 con ID del datapath 11:22:33:44:55 y puertos de red eth1, eth2 de la NetfPGA.

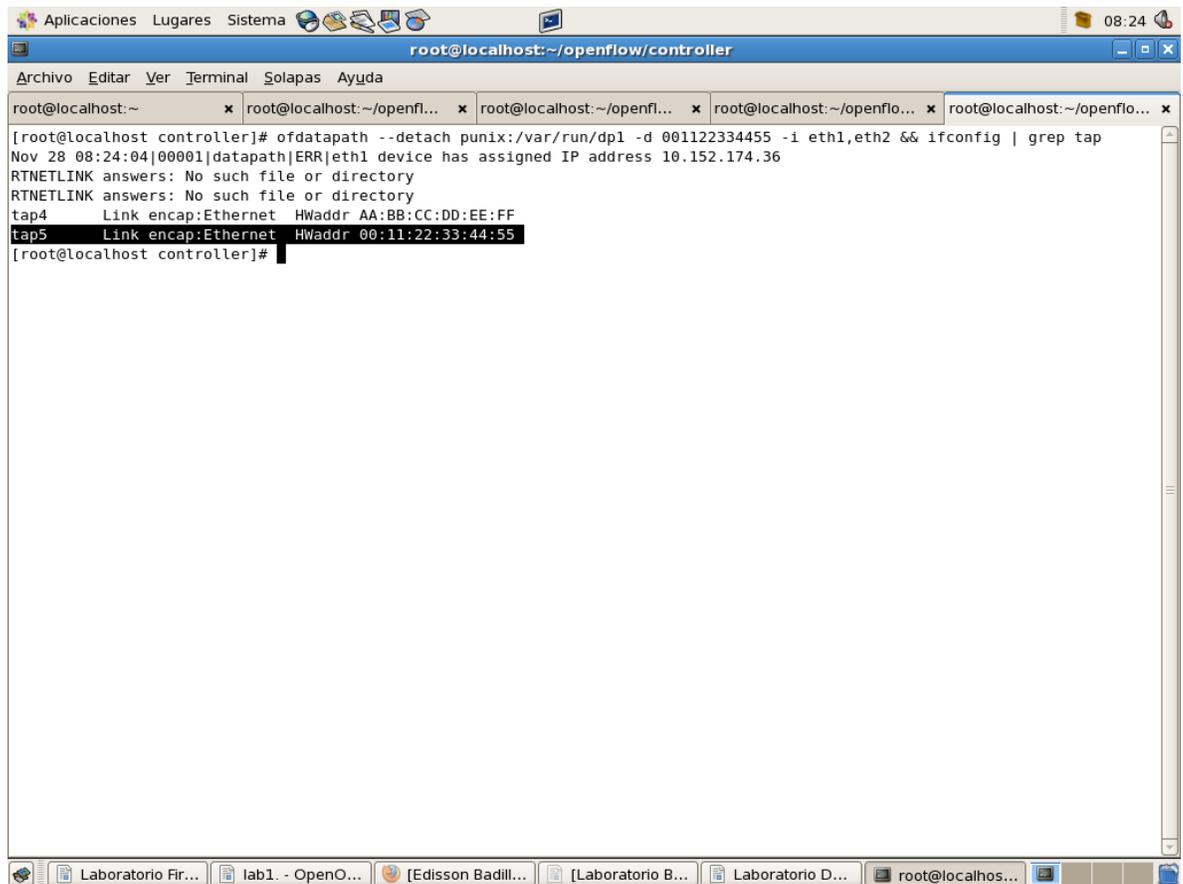


Imagen 5. Camino virtual dp1.

Solapa 6: Comunicación directa con el controlador usando dirección IP y puerto (socket). Camino dp0.

`ofprotocol unix:/var/run/dp0 tcp:127.0.0.1:6633`

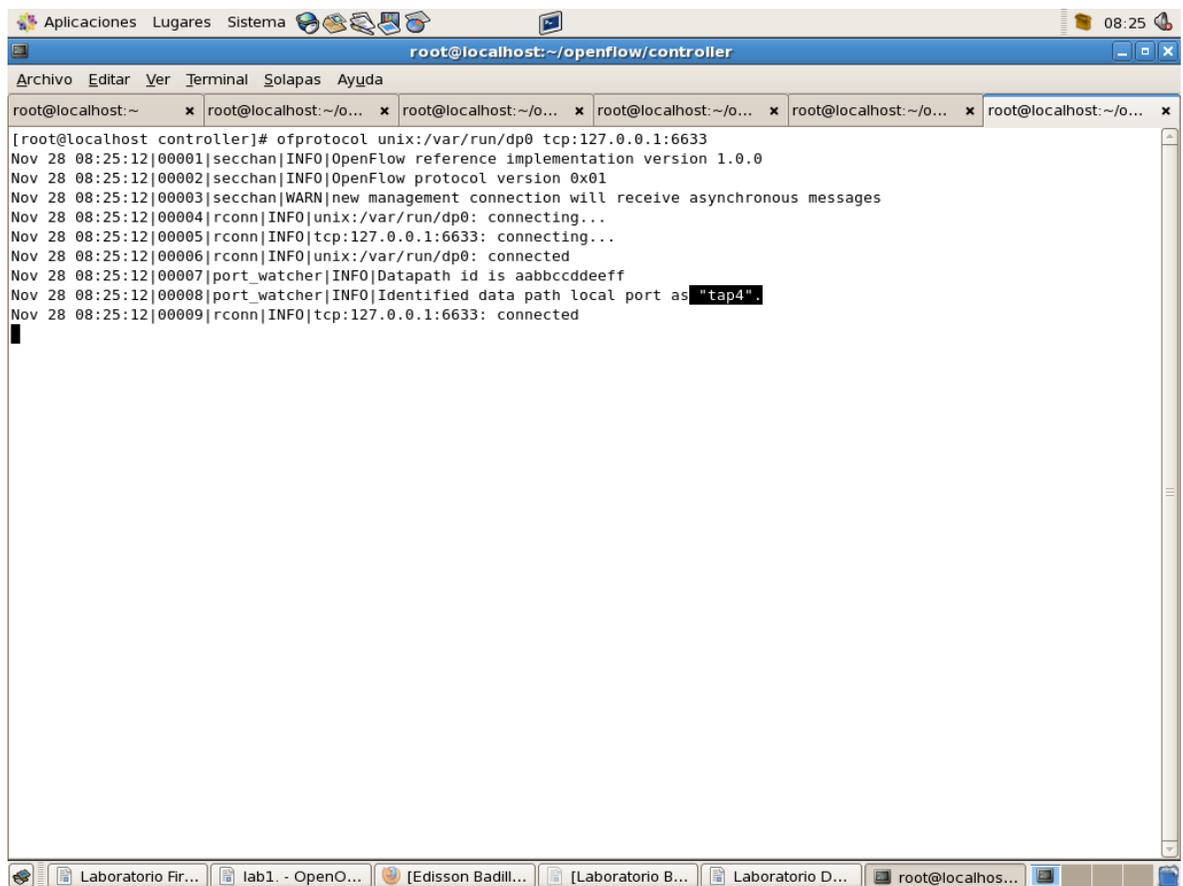


Imagen 6. Camino dp0.

Solapa 7: Comunicación directa con el controlador usando dirección IP y puerto (socket). Camino dp1.

```
ofprotocol unix:/var/run/dp1 tcp:127.0.0.1:6633
```

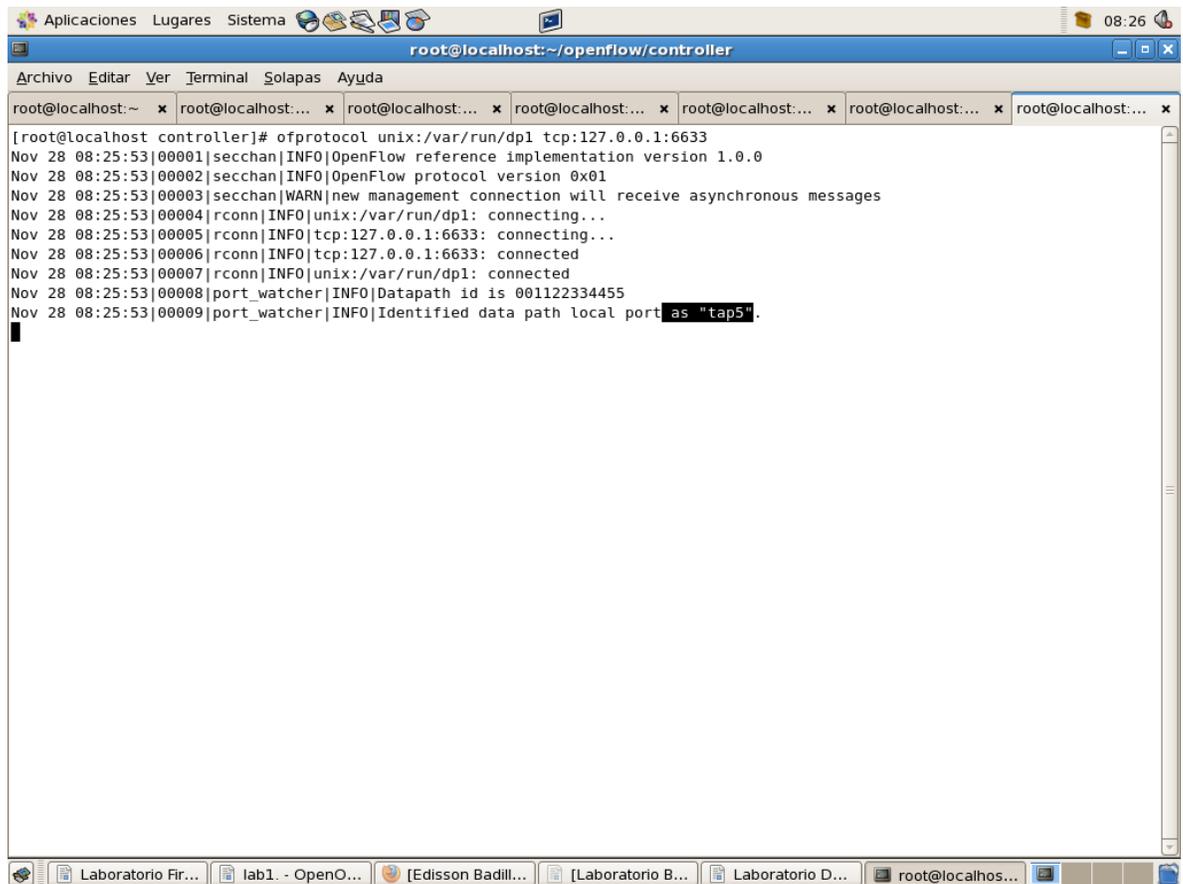


Imagen 7. Camino dp1.

Solapa 8: Ver el flujo del camino virtual dp0.

```
dpctl dump-flows unix:/var/run/dp0
```

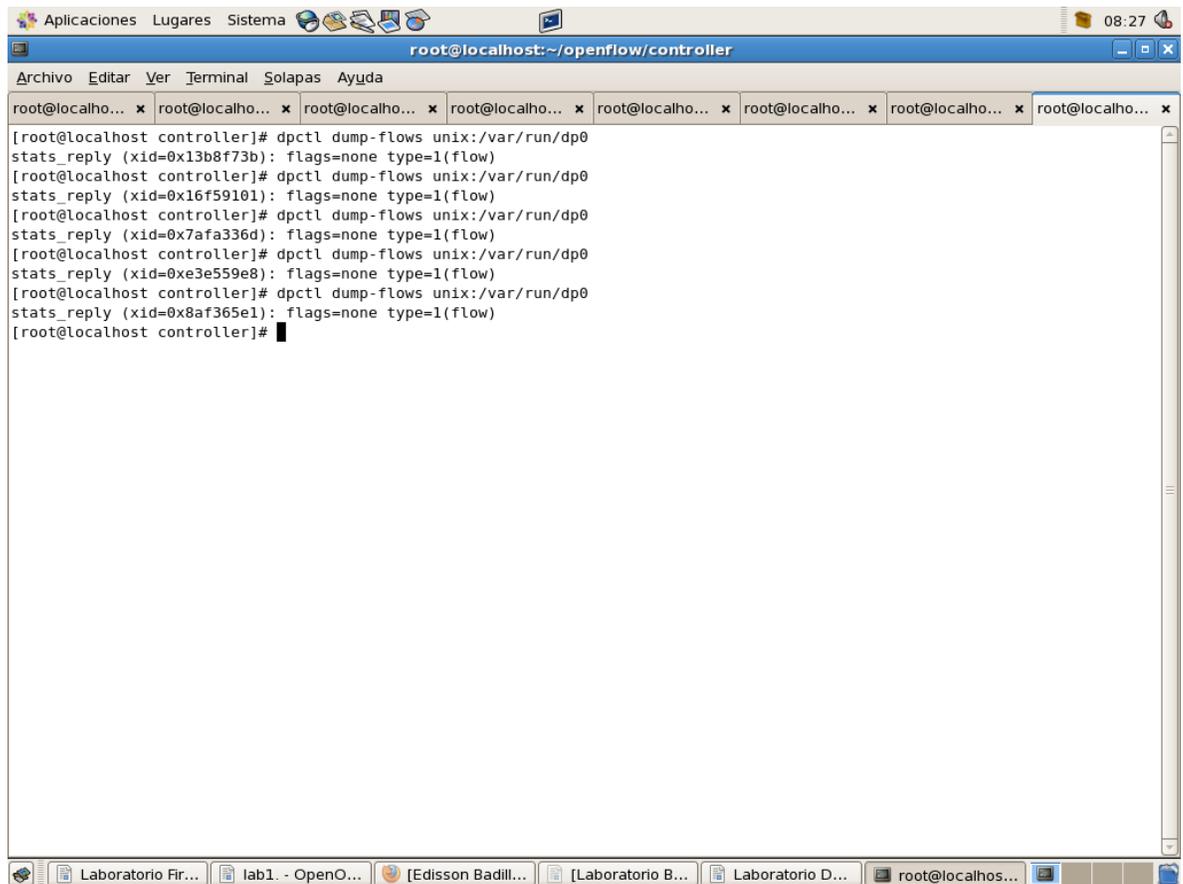


Imagen 8. Flujo dp0.

Solapa 9: Ver el flujo del camino virtual dp1.

```
dpctl dump-flows unix:/var/run/dp1
```

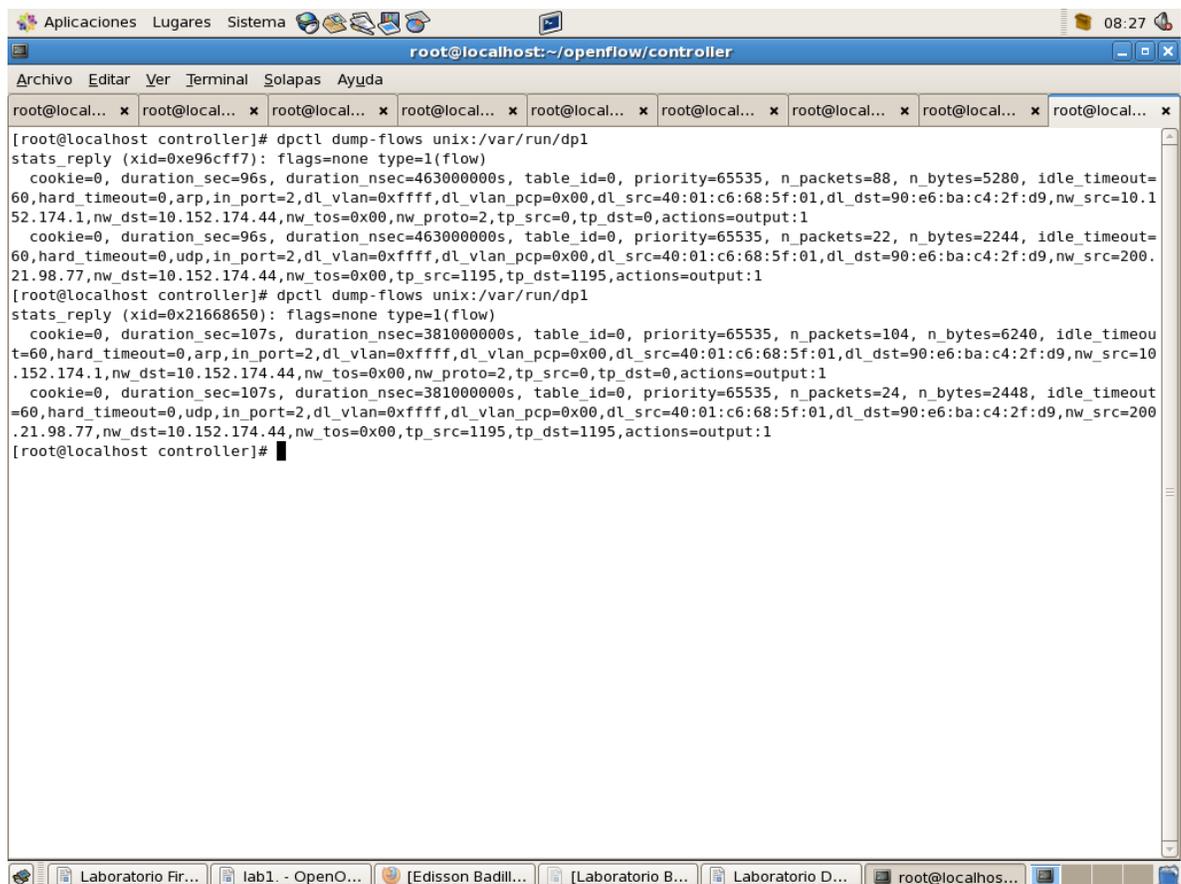


Imagen 9. Flujo dp1.

Antes de seguir con el próximo paso, comprobar que no hay conexión de red DNS en ninguno de los clientes a utilizar, tecleando ping 4.2.2.2. Esto es una herramienta de red de computadora utilizado para comprobar si un host en particular es accesible a través de una red IP.

Solapa 10: Descargar el código fuente de ISC DHCP SERVER.

```
wget http://mirror.sit.wisc.edu/pub/net/dhcp/dhcp-3.1.3.tar.gz
```

Luego, descomprimir e ingresar a la carpeta:

```
tar xvzf dhcp-3.1.3.tar.gz
```

```
cd dhcp-3.1.3
```

Compilar el código fuente para nuestra plataforma e instalarlo en la ruta /usr/local:

```
./configure
```

```
make
```

```
make install
```

Solapa 11: Crear un archivo en la ruta etc con el nombre dhcpd.conf y agregar las siguiente líneas.

```
nano /etc/dhcpd.conf
```

```
default-lease-time 600; #Tiempo de arriendo normal
max-lease-time 1200;      #Tiempo máximo de arriendo
ddns-update-style none; #Desactivar actualizaciones de DNS dinamicos
one-lease-per-client true; #Un arriendo por cliente
get-lease-hostnames true; #Obtener hostnames de clientes
option domain-name "upb.net"; #Asociar los dominos a dns.net
option domain-name-servers 8.8.8.8, 9.9.9.9; #Asignar DNS primario y secundario
ping-check true; #hacer una prueba de ping (ICMP 0 y 8 )
```

En el mismo archivo “/etc/dhcpd.conf” adicionar al final las siguientes líneas para crear la subred y el pool de direcciones (Red, IP, Gw, Broadcast)

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.20;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
}
```

Para guardar el archivo, oprimir las teclas control + o. Luego se confirma con la tecla *enter*, y para salir del editor se oprimen las teclas control + x.

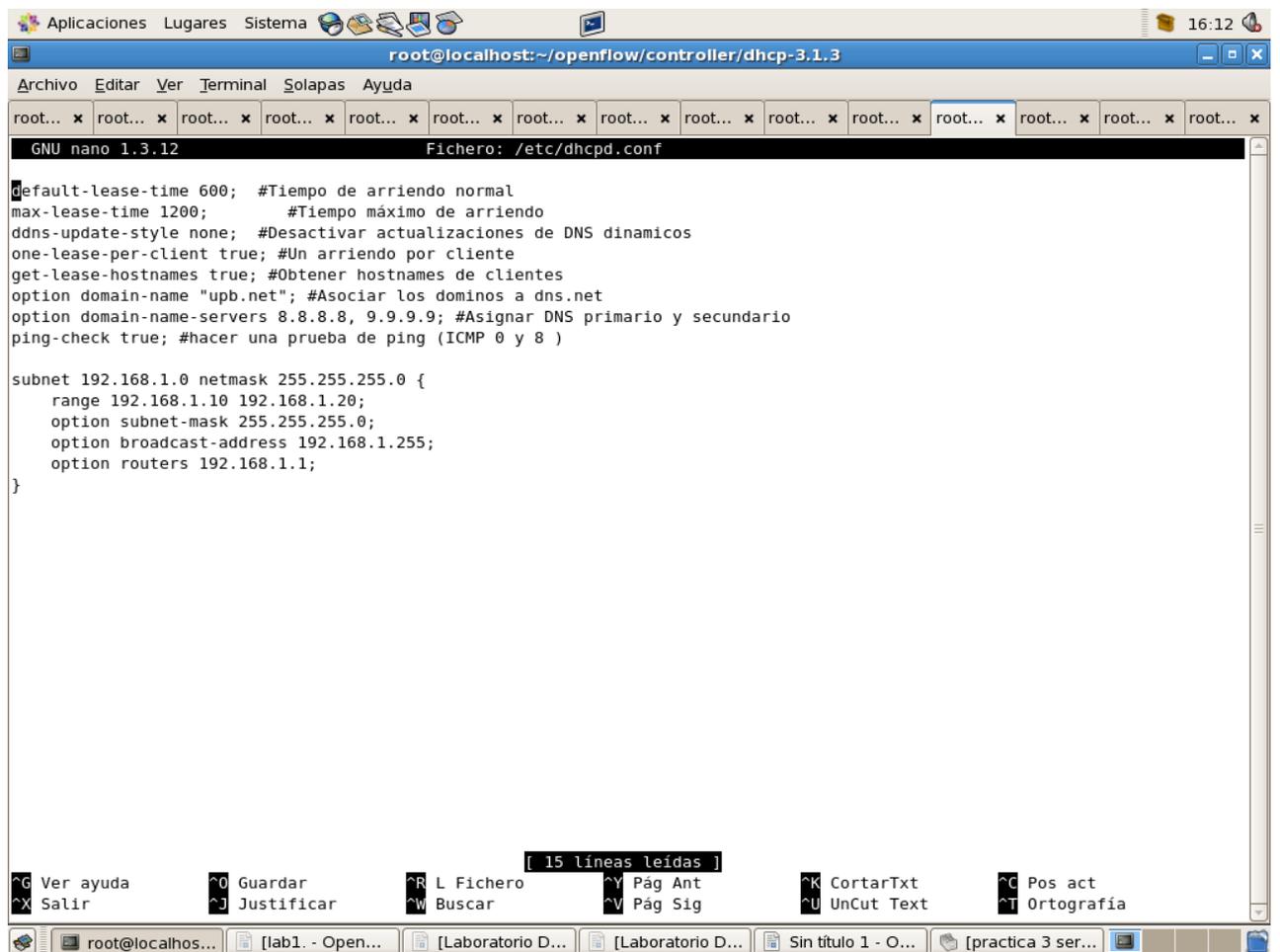


Imagen 10. Modificar código fuente.

Solapa 12: Crear una base de datos con los arrendamientos hechos.

```
touch /var/db/dhcp.leases
```

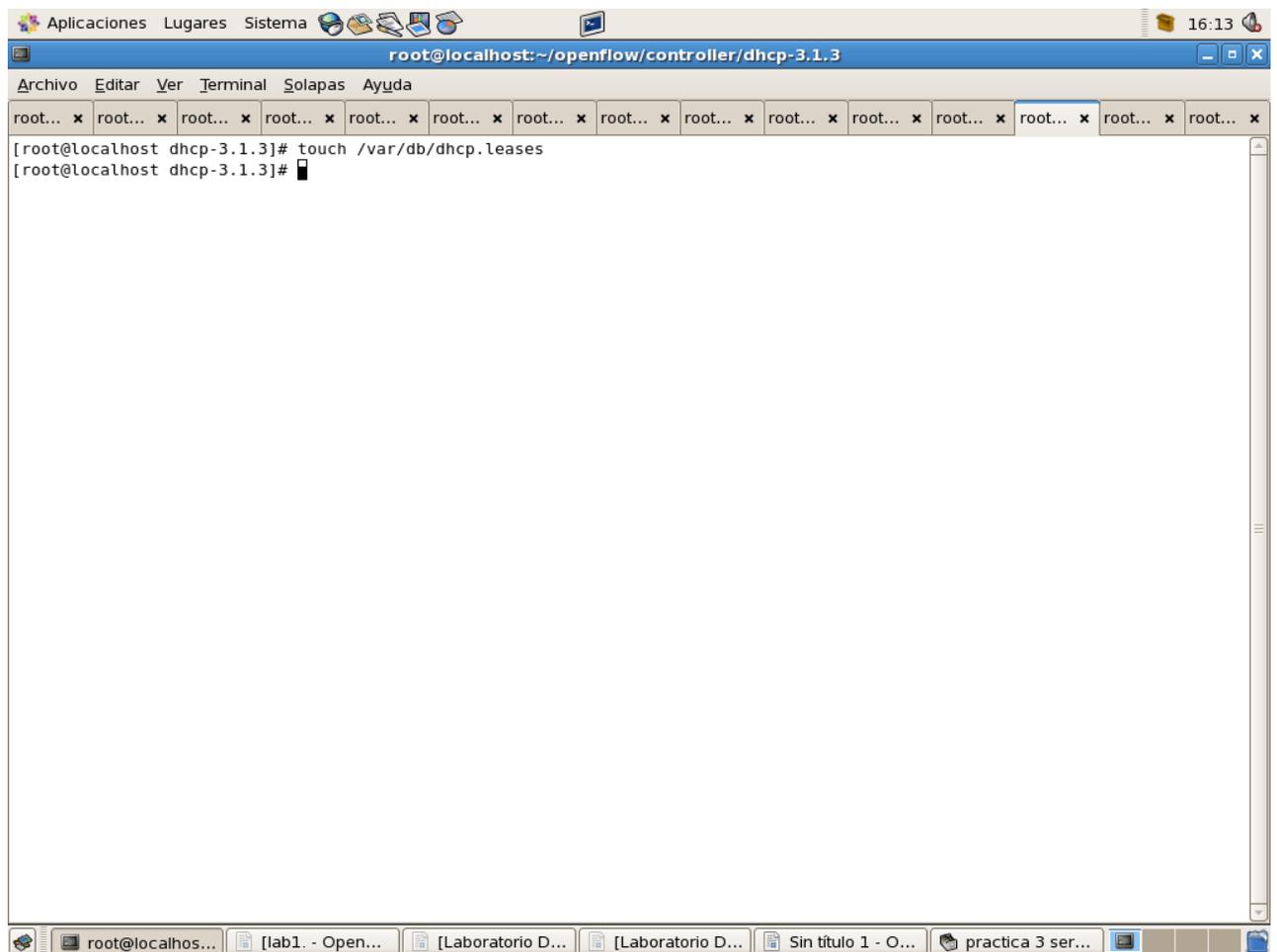


Imagen 11. Creación base de datos.

Solapa 13: Iniciar el servidor DHCP usando el archivo de configuración y de arriendos emitiendo en la interfaz eth1.

```
dhcpd -q -lf /var/db/dhcp.leases -cf /etc/dhcpd.conf eth1
```

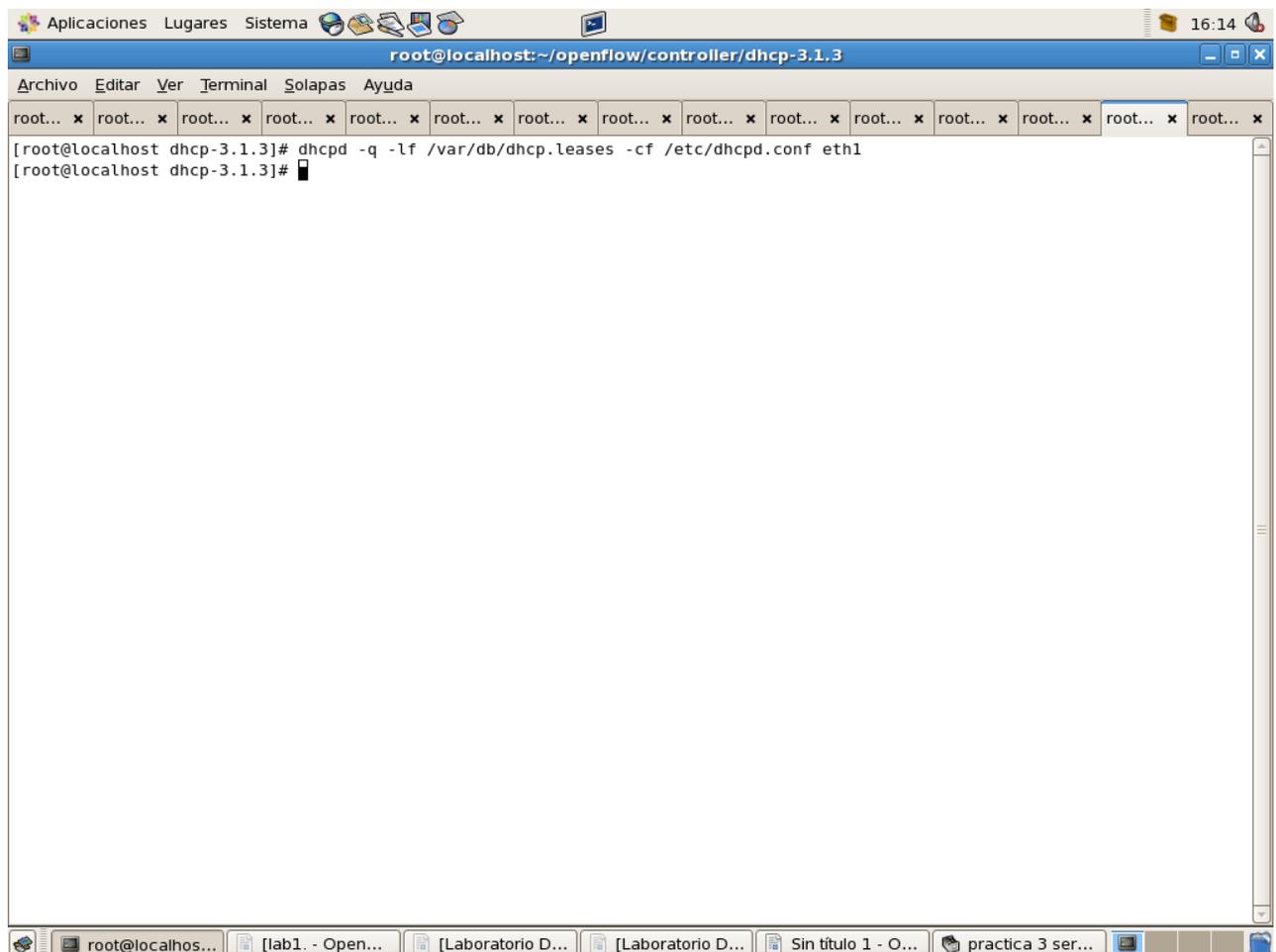


Imagen 12. Inicio del servidor DHCP.

NOTA: Conectar el cable de la interfaz eth1 del servidor a un puerto de la NETFPGA que actúa como openflow-switch.

Conectar un cable directo de un puerto disponible de la NETFPGA a la interfaz de red del cliente 1

Paso 1:

Conectar el cable de la interfaz eth1 del servidor a un puerto de la NETFPGA que actúa como openflow-switch

Si no está configura, configurarla

Paso 2:

Conectar un cable directo de un puerto disponible de la NETFPGA a la interfaz de red del cliente 1

Paso 3:

Conectar un cable directo de un puerto disponible de la NETFPGA a la interfaz de red del cliente 2

Paso 4:

Actualizar la información de red en las interfaces de cada computador cliente y constatar que las Direcciones IP, puerta de enlace y DNS hayan sido asignadas de forma automática.

Paso 5:

Listar las solicitudes de arriendo pasadas y vigentes en el servidor

```
cat /var/db/dhcp.leases
```

Verificar que esté corriendo el servidor

```
ps -aux | grep dhcpd
```

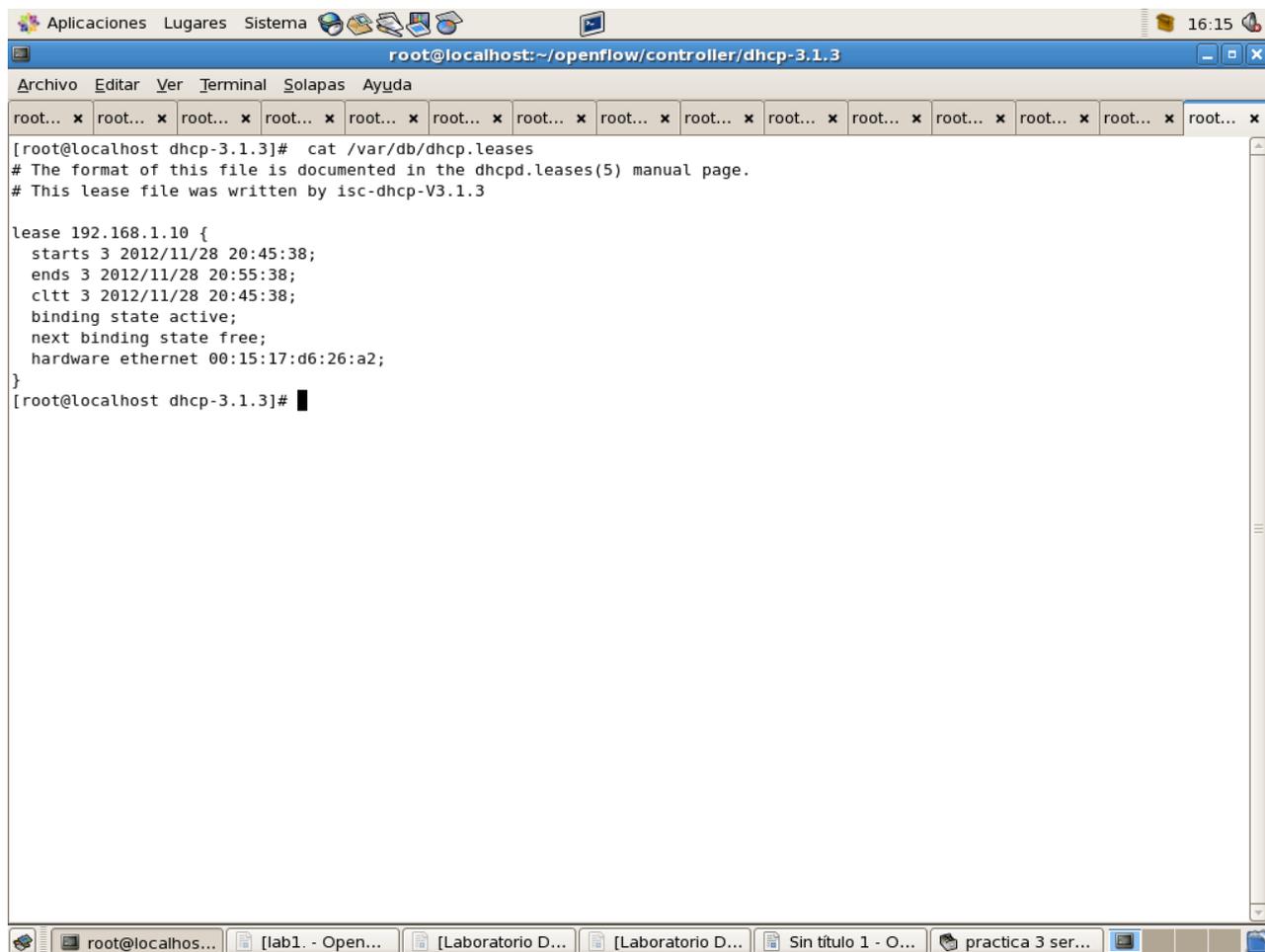


Imagen 13. Listar solicitudes de arriendo.

ACTIVIDAD.

Introducir un cliente más al servidor DHCP y verificar qué dirección IP se le asignó.

4.5.5 Laboratorio firewall por medio del switch Openflow

Introducción

Esta práctica de laboratorio consiste en crear un switch Openflow por medio de la tarjeta NetFPGA, con el fin de restringir el acceso a los paquetes que el switch está enviando a los clientes conectados al mismo, hasta que el administrador de la red decida quitarle esa restricción.

Objetivos

- Crear un switch Openflow por medio de la tarjeta NetFPGA.
- Restringir el acceso de los paquetes a los clientes.

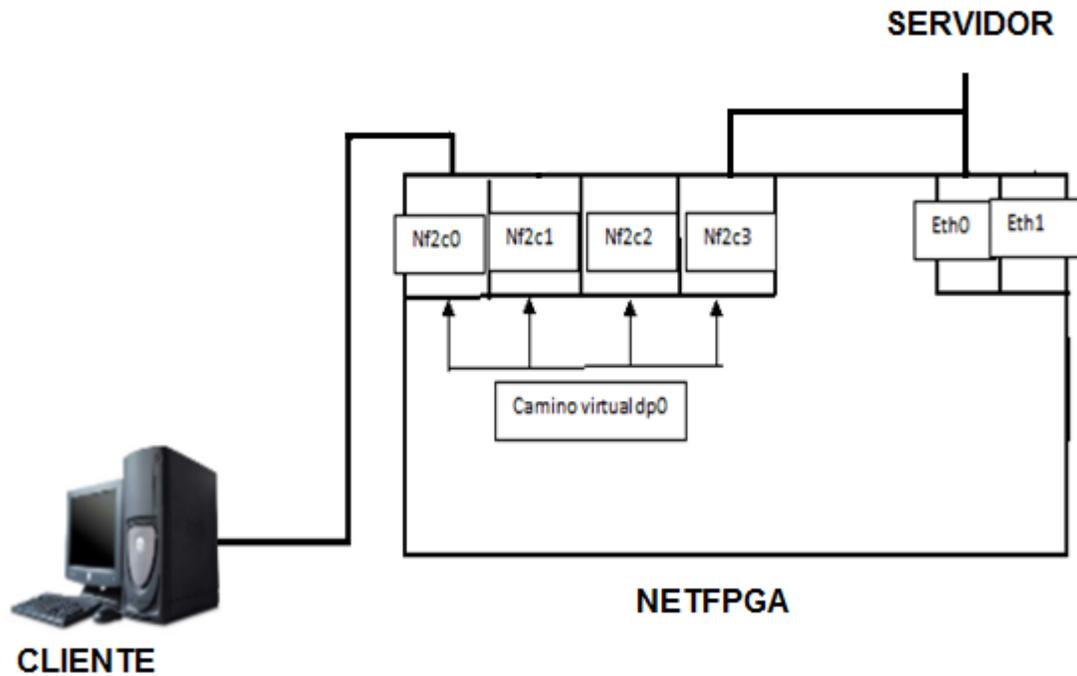
Materiales y equipos a utilizar

- 1 NetFPGA con GNU/Linux distribución CentOS y con el paquete de Openflow instalado.
- 2 computadores con GNU/Linux distribución CentOS para ser conectados como clientes.
- 4 patchcords.

NOTA:

- Tener en cuenta que los puertos de los dos clientes deben ser de 1Gbit Ethernet.
- El terminal de Linux diferencia entre letras mayúsculas y minúsculas, no es lo mismo teclear hola a Hola.

Diagrama del laboratorio



Desde la solapa 1 al 7 los comandos deben ser ejecutados en solapas diferentes.

Procedimiento Servidor:

Solapa 1: Limpiar la memoria de la NetFPGA.

cpci_reprogram.pl

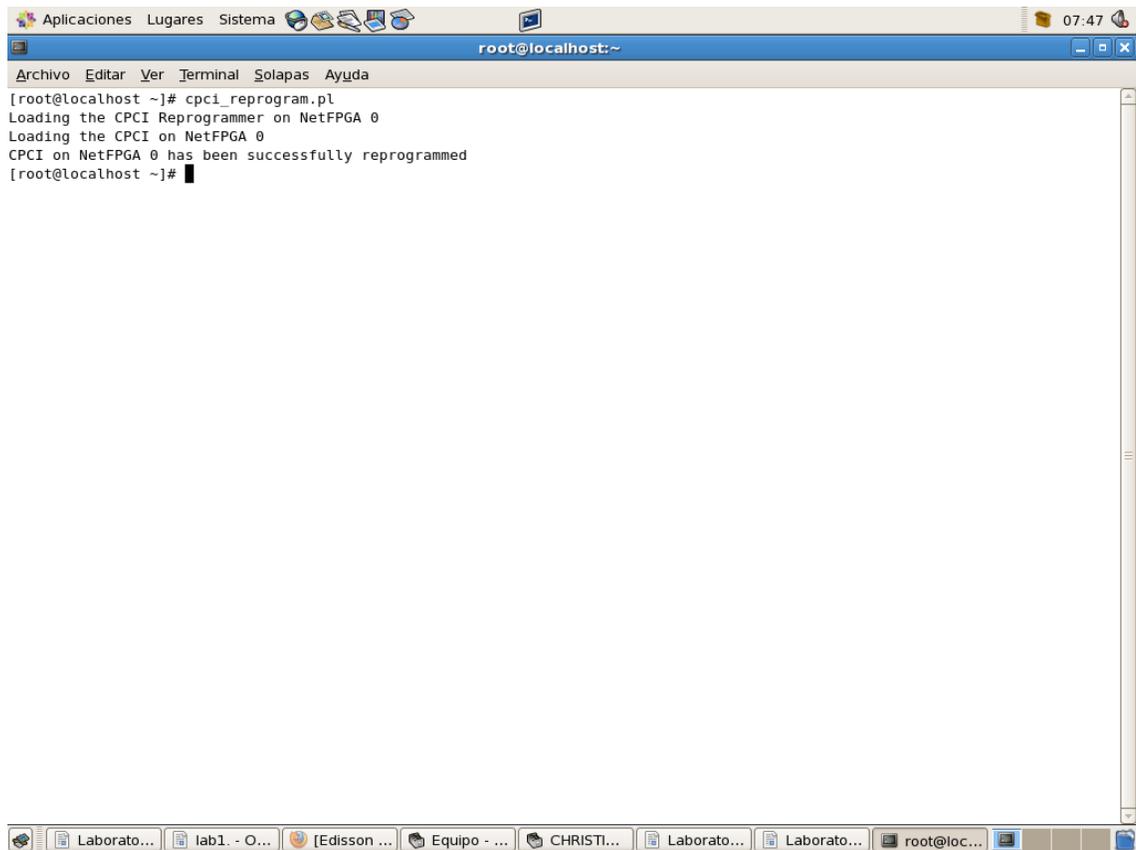


Imagen 1. Limpiar la memoria de la NetFPGA.

Solapa 2: Cargar el bitfile del switch. Ir a la ruta raíz del Openflow (donde está instalado). Para esto, seguir la siguiente ruta.

```
cd /root/openflow/hw-lib/nf2
```

```
nf_download openflow_switch.bit
```

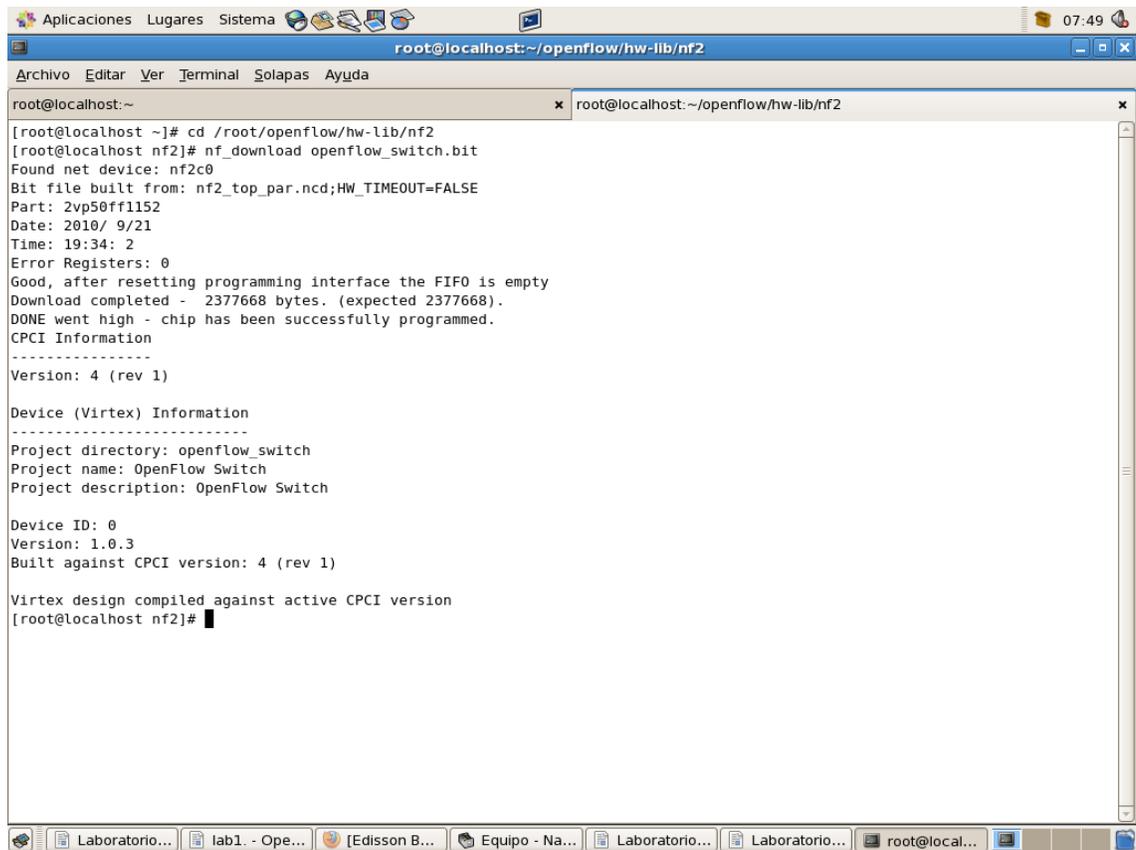


Imagen 2. Cargar el bitfile del switch.

En la figura 2, se debe comprobar que en la opción Error Registers aparezca un cero (0), de contrario, reprogramar la NetFPGA.

Solapa 3: Cargar el controlador y enlazarlo a un puerto tcp. Para esto, buscar el ejecutable controller en la carpeta controller cuya ruta es la siguiente.

```
cd /root/openflow/controller
```

```
./controller ptcp:6633
```

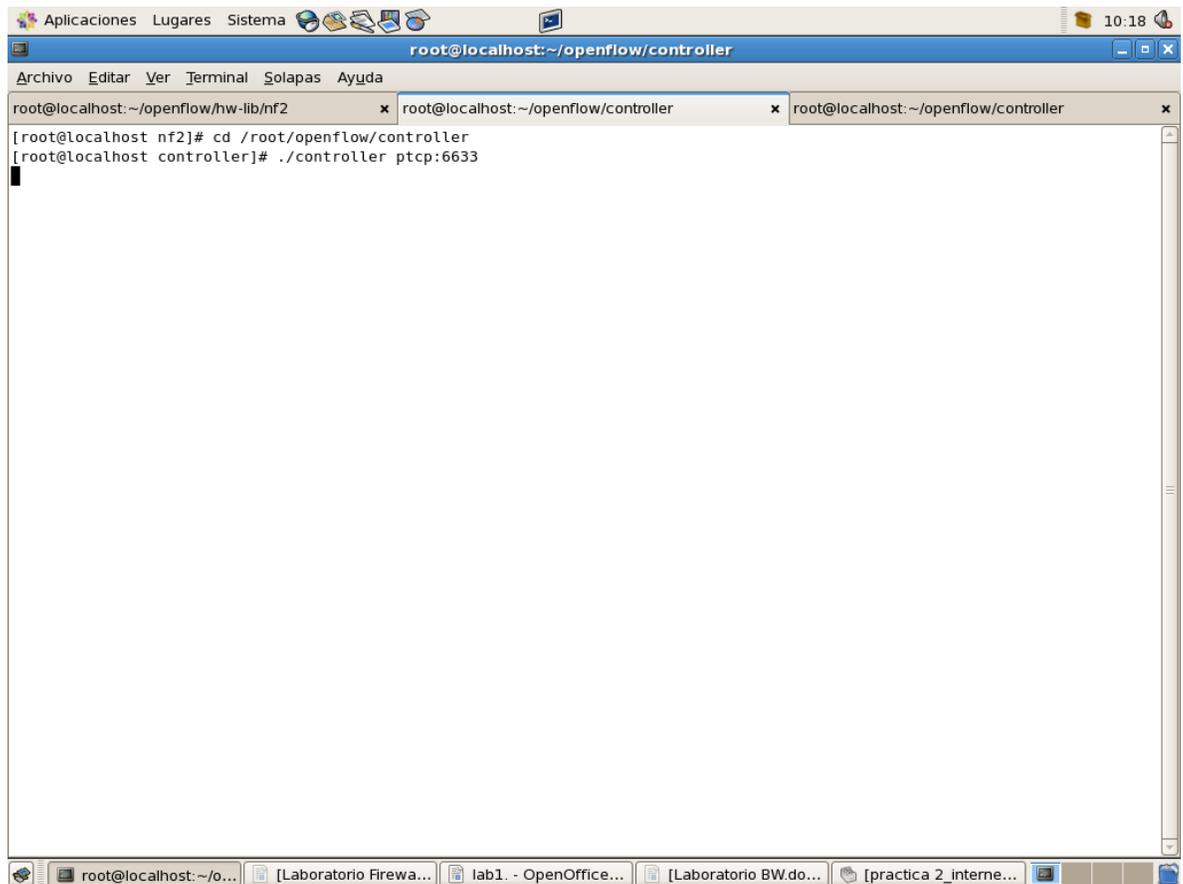


Imagen 3. Cargar el controlador.

Solapa 4: Crear el camino virtual del switch con todos los puertos de la NETFPGA y verificar la creación de una interfaz virtual donde se puede ver el flujo (tapX).

```
ofdatapath --detach punix:/var/run/dp0 -d aabbccddeeff -i nf2c0,nf2c1,nf2c2,nf2c3  
&& ifconfig | grep tap
```

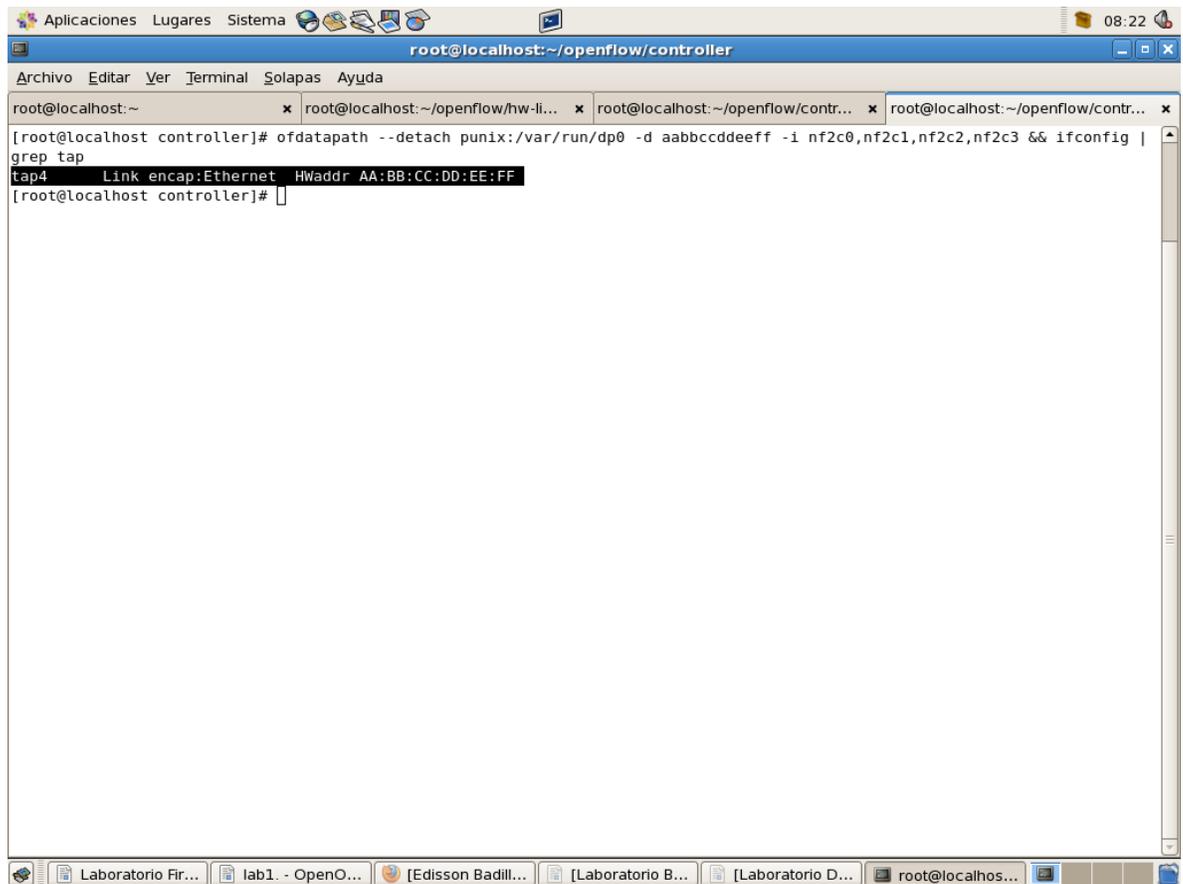


Imagen 5. Camino virtual dp0.

Solapa 6: Comunicación directa con el controlador usando dirección IP y puerto (socket).

ofprotocol unix:/var/run/dp0 tcp:127.0.0.1:6633

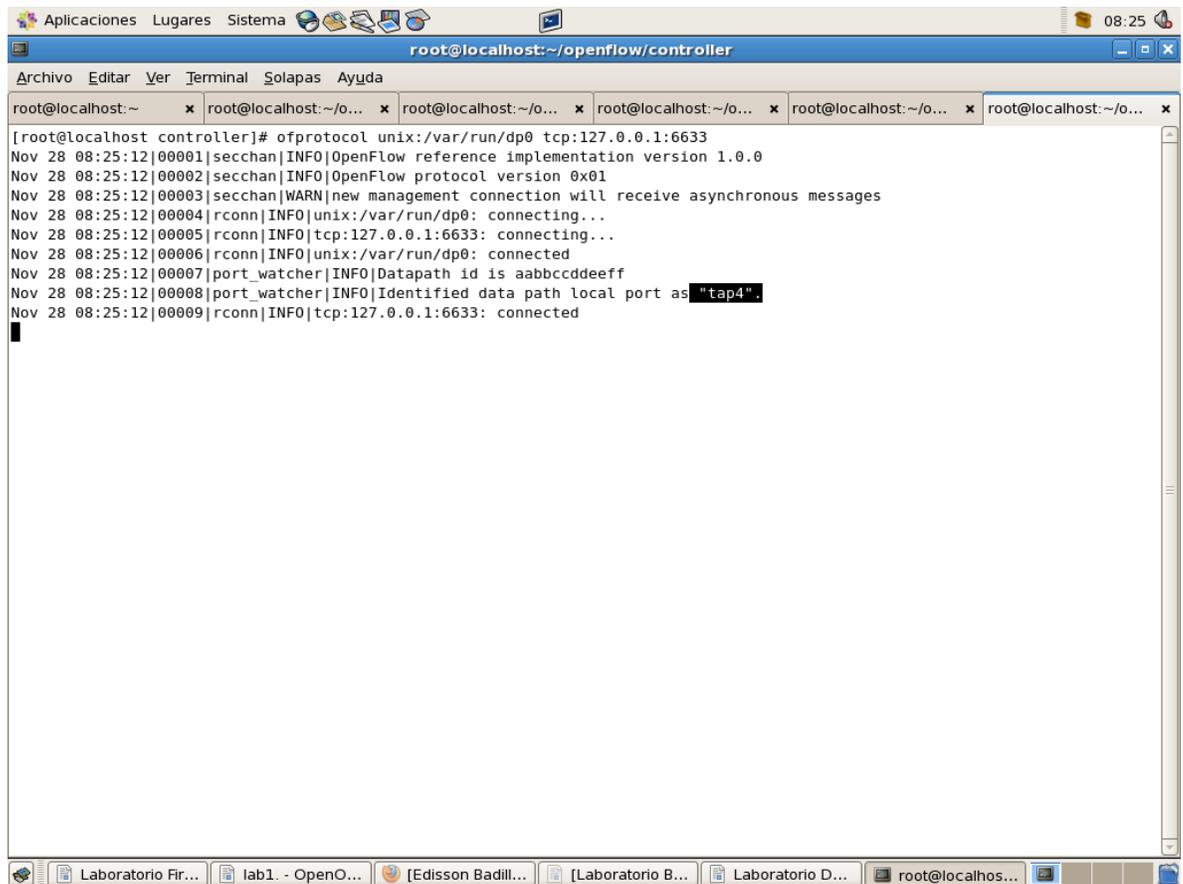


Imagen 6. Camino dp0.

Solapa 7: Ver el flujo del camino virtual dp0.

dpctl dump-flows unix:/var/run/dp0

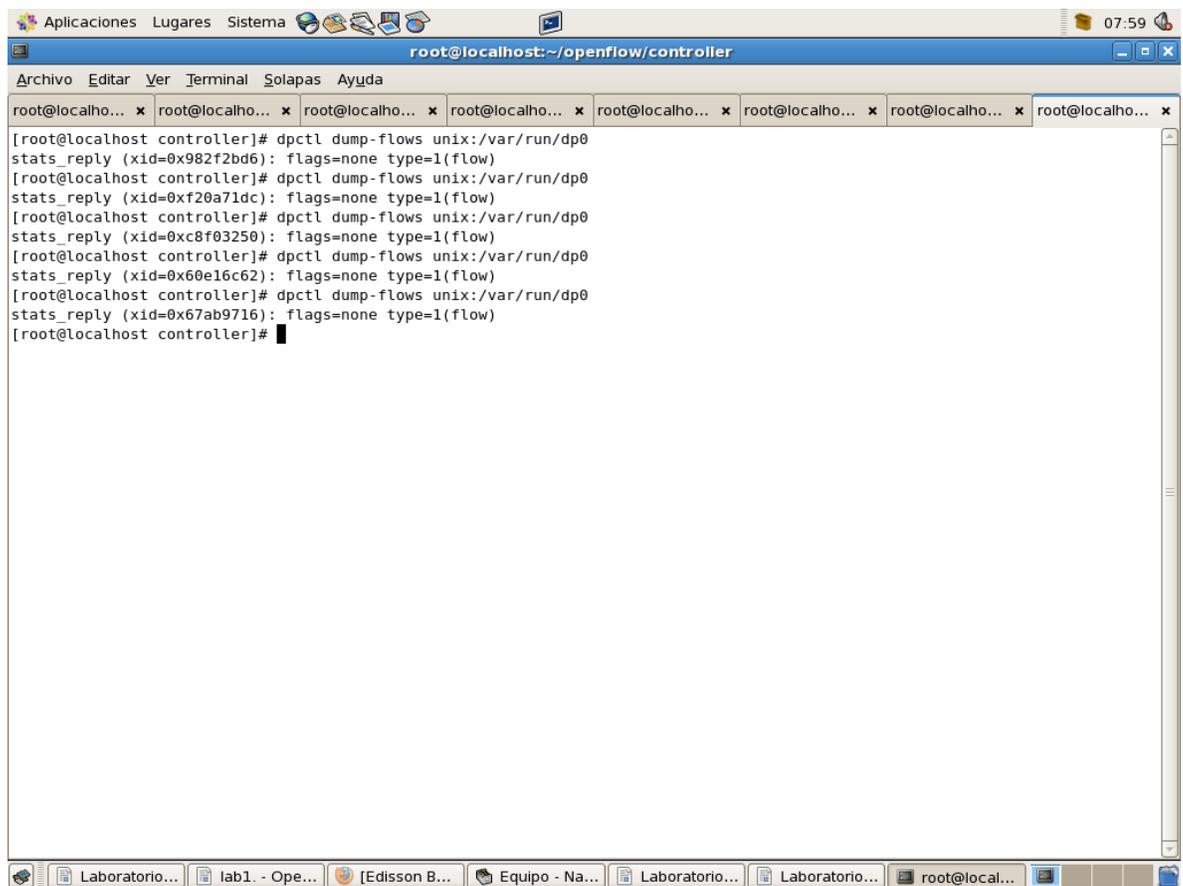


Imagen 7. Flujo dp0.

Hacer ping entre servidor- cliente y verificar que se estén conectando. Para dejar de hacer ping en cualquier momento oprimir las teclas control+c al mismo tiempo.

DESDE CLIENTE:

```
ifconfig eth1 192.168.1.20
```

DESDE SERVIDOR:

```
ifconfig eth1 192.168.1.10
```

```
ping -l eth1192.168.1.20
```

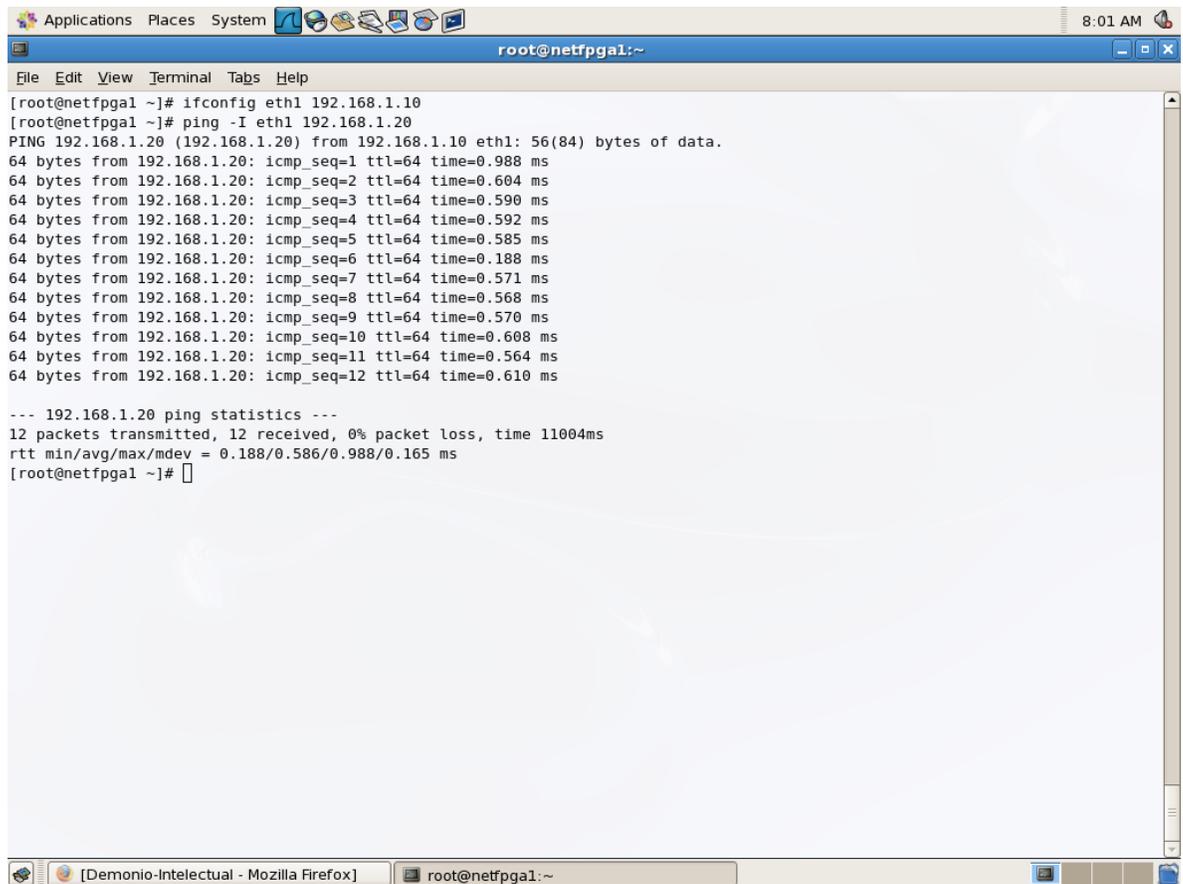


Imagen 8. Ping entre servidor y cliente.

DESDE SRVIDOR:

dpctl show unix:/var/run/dp0

Nota: Debe aparecer algo parecido a lo siguiente.

```
features_reply (xid=0x2b692ce9): ver:0x1, dpid:1122334455
n_tables:3,
n_buffers:256
features: capabilities:0xc7,
actions:0xeff
```

```
1(nf2c0): addr:00:4e:46:32:43:00, config: 0x1, state:0
current: 1GB-FD
advertised: COPPER
supported: 1GB-FD
```

```
2(nf2c1): addr:00:4e:46:32:43:01, config: 0x1, state:0
current: 1GB-FD
advertised: COPPER
supported: 1GB-FD
```

```
3(nf2c2): addr:00:4e:46:32:43:02, config: 0x1, state:0
current: 1GB-FD
advertised: COPPER
```

```
supported: 1GB-FD
4(nf2c3): addr:00:4e:46:32:43:03, config: 0, state:0
current: 1GB-FD
advertised: COPPER
supported: 1GB-FD
LOCAL(tap0): addr:00:11:22:33:44:55, config: 0, state:0
current: 10MB-FD COPPER
get_config_reply (xid=0x2479fb44): miss_send_len=128
```

Para restringir la comunicación entre servidor y cliente, desde el terminal del servidor se teclea el siguiente comando. Verificar mediante un ping que no se estén comunicando

```
dpctl mod-port unix:/var/run/dp0 1 down
ping -I eth1 192.168.1.20
```

Para volver a comunicar entre servidor y cliente, desde el terminal del servidor se teclea el siguiente comando.

```
dpctl mod-port unix:/var/run/dp0 1 up
```

ACTIVIDAD

Conectar dos clientes más al Switch Openflow y restringir el acceso sólo al puerto nf2c3 (mirar a qué puerto Ethernet corresponde en la conexión realizada) de la NetFPGA.

5. GLOSARIO

CentOS: Es una clase de distribución de Linux.

Linux: Sistema operativo que posee un núcleo del mismo nombre. El código fuente es abierto.

NetFPGA: Es una tarjeta especializada en la construcción de redes, que permite la construcción de prototipos de alta velocidad, acelerados por hardware a bajo costo.

Openflow: es una herramienta que permite a un servidor decirle a los conmutadores de red adonde enviar paquetes.

Redes Definidas por Software: Son una manera de abordar la creación de redes.

Switch: Es un dispositivo de red que permite interconectar por lo menos 2 segmentos de red o diferentes redes entre sí.

Bootear: Modificar el orden de arranque de los dispositivos de almacenamiento.

Librería: Es un conjunto de subprogramas usados para desarrollar software.

Open Networking Foundation: Fundación encargada de definir los estándares de la herramienta Openflow.

Kernel: Su función principal es lograr que el software y el hardware del computador trabajen juntos.

Virtualbox: Es una aplicación encargada de instalar sistemas operativos de forma virtual.

Desktop-Gnome: Es la aplicación que permite ver el escritorio mediante gráficos de la distribución Centos 5.6.

Cortafuegos: Permite el acceso a servicios específicos desde otros computadores hacia el computador que se quiere permitir ese acceso.

Yum: Es un paquete administrador de software. Es utilizado para instalar, actualizar y eliminar paquetes con sus dependencias.

Terminal: Es un intérprete de comandos que realiza el usuario para que el sistema operativo entienda lo que se quiere hacer.

Solapa: Es una pestaña nueva en la misma ventana del terminal de Linux.

Ruta: Es la dirección en la cual se encuentra un directorio o fichero en Linux.

Ping: Es una utilidad que comprueba el estado de la conexión del host local con uno o varios equipos remotos de una red.

6. CONCLUSIONES

Pese a la poca información que hay sobre este proyecto, ya que es relativamente nuevo, se logró utilizar la herramienta Openflow para realizar las distintas prácticas de laboratorio que se habían planeado al inicio del proyecto de grado.

A su vez se realizaron los distintos manuales tanto de instalación como de compilación de los software requeridos, para que los estudiantes de las facultades de ingeniería electrónica e ingeniería informática se interesen por investigar más a fondo este proyecto que es innovador para el futuro de las telecomunicaciones mediante la aplicación de las Redes Definidas por Software o SDN`s.

Se elaboró un manual acerca de los comandos básicos de Linux distribución CentOS, para que el estudiante se familiarice con este tema, debido a que el proyecto Openflow requería de ciertos conocimientos de este sistema operativo muy utilizado actualmente por las empresas.

7. REFERENCIAS BIBLIOGRÁFICAS

PADILLA AGUILAR, Jhon Jairo; BECERRA SÁNCHEZ, Line Yasmin y MORALES GONZÁLEZ, Alvaro Ignacio. *U2-route: university universal router*. Guía práctica. Primera edición. Bucaramanga., Sic Editorial Ltda. Diciembre de 2011. 117p.

FACUNDO ARENA, Héctor. La biblia del Linux. Ciudad de Buenos Aires: MP Ediciones. Enero de 2003. 264 p.

SHOTTS JR, William E. The Linux Command Line. San Francisco.: Creative Commons. Julio 18 de 2009. 489 p.

BARRET, Daniel J. Linux Pocket Guide. O`reilly Media. Marzo de 2012. 232 p.

REFERENCIAS ELECTRÓNICAS:

BALDONADO, Omar. Openflow desmitificado. [en línea]. 15 de noviembre de 2011. [Consultado 20 de marzo de 2012]. Disponible en: <<http://cioperu.pe/articulo/8642/openflow-desmitificado>>

Elvalle. Google mejora su propia red interna a través de la tecnología de código abierto Openflow. [en línea]. 22 de abril de 2012. [Consultado 10 de julio de 2012]. Disponible en: <<http://www.vvv.mx/2012/04/22/portada/tecnologia>>

Channel News. OpenFlow/SDN. La innovación llega a la red. [En línea]. Mayo de 2012. [Consultado Junio 10 de 2012]. Disponible en: <<http://www.emb.cl/channelnews/articulo.mvc?xid=2048&tip=11>>

Página oficial CentOS. [Consultado 1 abril de 2012]. Disponible en: <<https://www.centos.org>>

Página oficial del proyecto Openflow. [Consultado 12 de marzo de 2012].
Disponibile en: <www.openflow.org>

GIBILISCO, Stan. *Software-Defined networking* (SDN). [en línea]. Junio de 2012.
[Consultado 20 de Julio de 2012]. Disponible en: <<http://whatistechtarget.com/definition/software-defined-networking-SDN>>

Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. ONF White Paper. Abril 13 de 2012. [en línea]. Consultado julio 14 de 2012. Disponible en: <<https://www.opennetworking.org/images/stories/downloads/White-papers/wp-sdn-newnorm.pdf>>