

Desarrollo de un entorno  
de virtualización de redes con fines  
docentes



UNIVERSIDAD  
**COMPLUTENSE**  
MADRID

Víctor Fernández Duque  
Director - Juan Carlos Fabero Jiménez

Trabajo de fin de grado del Grado en  
Ingeniería del Software

<b>Reunión anterior 06/02/2018</b>	<b>3</b>
<b>Reunión 06/02/2018</b>	<b>4</b>
<b>Reunión 23/02/2018</b>	<b>5</b>
<b>Reunión 06/03/2018</b>	<b>6</b>
<b>Reunión 03/04/2018</b>	<b>7</b>
<b>Reunión 07/04/2018</b>	<b>8</b>
<b>Reunión 17/04/2018</b>	<b>9</b>
<b>Reunión 24/04/2018</b>	<b>10</b>
<b>Reunión 17/05/2018</b>	<b>11</b>

## Reunión anterior 06/02/2018

Decidimos juntar todas las reuniones antes de navidad en un solo documento, debido a que el alumno no tenía tanto tiempo para dedicarlo al proyecto por las demás asignaturas, por lo que se centró en el estudio de todo lo necesario para poder crear la solución posteriormente.

Lo primero que hicimos fue estudiar las redes SDN, todo lo relacionado con creación de switches, interfaces, puertos, cables virtuales, mirroring, reglas QoS, etc.. Para aprender todo se necesitaron los meses de octubre y noviembre.

El mes de diciembre y enero fue un mes más bien preparatorio para poder desarrollar después de navidades la aplicación, por lo que se realizó la especificación de requisitos para entender qué era lo que queríamos realizar en nuestro proyecto, no se pudo hacer de otra forma debido a que había que prepararse para los exámenes de enero y era totalmente necesario, por lo que el proyecto estuvo parado casi dos meses.

En la siguiente reunión que fue en febrero ya comenzamos a estudiar el lenguaje en el que íbamos a desarrollar nuestra aplicación y a programarla.

# Reunión 06/02/2018

Quedamos de acuerdo en empezar a desarrollar el software tal como hemos estudiado en la especificación de requisitos software y estamos abiertos a nuevos cambios.

Quedamos de acuerdo en generar un diagrama de gantt para especificar la (planificación) temporización del proyecto para de esta forma cumplir con los hitos del proyecto, además de realizar el modelo de la aplicación. Hablaremos en la memoria de los parones que hemos tenido debido a épocas de examen.

TCL-TK: empezaremos a realizar funciones simples, como puede ser el crear un switch, eliminarlo, además buscaremos en la biblioteca documentación sobre este lenguaje para familiarizarnos con el.

En la memoria sería útil añadir las especificación de cada función, la parte de requisitos funcionales de la srs.

# Reunión 23/02/2018

Se comentó una serie de comandos que podríamos ir implementando a la aplicación para ir dándole forma:

- `ovs-vsctl list bridge <punto>`
- `ovs-vsctl list-ifaces <punto>`
- `ovs-vsctl get interface <interfaz> error`
- `ovs-vsctl get interface <interfaz> ofport // -1 no existe cualquier otro existe`
- `ovs-ofctl dump-flows br0`
- `ovs-ofctl add-flows switch file`

Se habló de la interfaz gráfica, el dao(*data access object*, capa de acceso a datos), como podría ser la aplicación a nivel estructural. La forma de guardar y cargar ficheros en Tcl-Tk y limpiar el estado de la aplicación.

Tenemos dos tipos de guardado:

- El guardado general el cual guarda la topología y el control de flujo.
- El guardado únicamente del control de flujo.

# Reunión 06/03/2018

Analizamos los distintos puertos que puede tener un switch, tipo tuntap o veth para crear nuestro formato.

Si es una interfaz física no hace falta crearla solo agregarla a la interfaz a la hora de cargar un estado guardado.

Valorar hacer veth por Open vSwitch con patch para unir dos switches.

Estudiar modificar los módulos de guardar y cargar para hacerlo con formato json en vez de uno propio, se conservan las dos funciones.

Eliminar interfaces valorar que puedes dar ya una lista hecha, interesante para el apartado de mejoras futuras.

## Reunión 03/04/2018

Esta reunión comenzó con un problema en el módulo guardar y cargar que ocasionó que se saturara el sistema virtualizado. Lo estuvimos observando y llegamos a la conclusión que no era un error del código, a la hora de cargar la topología se escogió un ejemplo que producía bucles por lo que saturaba la máquina. Estos fueron los comandos usados para observar el problema:

- nmcli connection
- nmcli connection delete br0
- ovs-vsctl add-port br0 uml1.0
- ovs-ofctl mod-port br0 uml1.0 up
- systemctl disable NetworkManager

Mostramos el funcionamiento de la función guardar y cargar con json funcionando correctamente.

# Reunión 07/04/2018

Esta reunión trató sobre cómo redactar la memoria.

Lo primero de lo que debemos hablar es sobre el problema que nos encontramos y porque decidimos investigarlo

Hablamos de las redes SDN, Openflow, la importancia de la virtualización y la posibilidad de aplicar a entornos físicos.

Hablar del porqué hemos realizado este proyecto, la docencia y el aprender un tema interesante.

Alternativas a nuestro proyecto neutrón de openstack, vMWare y Proxmox.

En el apartado de arquitectura hablar de metodología, porque y las ventajas, las dificultades que nos hemos encontrado.

presentar solución de arquitectura, hablar de todo lo relacionado con nuestra aplicación desde la estructura, a las carpetas, la metodología, riesgos, etc..

Explicar el manual de usuario completamente, todas las funciones de nuestra aplicación con imágenes. Comentar porque es mantenible nuestra aplicación.

Posibles nuevos desarrollos como la aplicación de sockets.



# Reunión 17/04/2018

Contenido para realizar los requisitos de nuestra memoria:

- SDN:
  - Definición como la tengo, añadir imágenes. Las aplicaciones de usuario se comunican mediante la api con la capa de control y está mediante el protocolo openflow define la infraestructura o capa de datos, puede utilizarse tanto en entornos virtuales como físicos, imagen de página 6.
  - Las SDN no SDNS, error típico español.
  - Ventajas: en un principio las SDN se desarrollaron para un entorno virtualizado pero ya que los fabricantes lo han llevado a su hardware se amplía muchos más el ámbito de aplicación.
- VLAN:
  - Serían los antecedentes de las SDN, las cuales permiten todavía este sistema pero a día de hoy se ha evolucionado y mejorado.
- Reglas QoS
  - Estudiar que es linux htb (Cubo).

# Reunión 24/04/2018

Seguimos realizando la memoria.

Hablamos de nuevo de las opciones que solucionas el mismo problema que nosotros como dijimos neutron de OpenStack, vMWare y Proxmox.

Tenemos que hablar de nuestra solución:

- Un pequeño resumen.
- Tcl-Tk independencia de binarios y de sistemas operativos, ya que está disponible en múltiples sistemas operativos.
- OpenVswitch hablar de los comandos usados.
- Arquitectura, como hemos distribuido la aplicación, carpetas, bocetos, metodología, etc..

Comentar todas las posibles mejoras a futuro que podríamos realizar para mejorar la solución.

## Reunión 17/05/2018

Esta reunión se dedica totalmente a la memoria, hablamos de la finalización de la misma y que la entrega será el siguiente día, hablamos de los apartados que faltan como el resumen y abstract, como formatear el texto, y recolocar una figura que sobraba y revisar todo en general.