Prácticas TSR
Sesión 2

Juan Sánchez (jsanchez@dsic.upv.es)

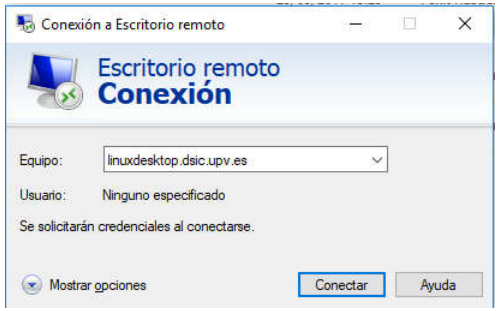
Despacho 2D08

Copiar archivos a la máquina virtual

- Podemos abrir una sesión de escritorio remoto en Linux y luego conectarnos a nuestra máquina virtual.
- Si suponemos que nuestra máquina es

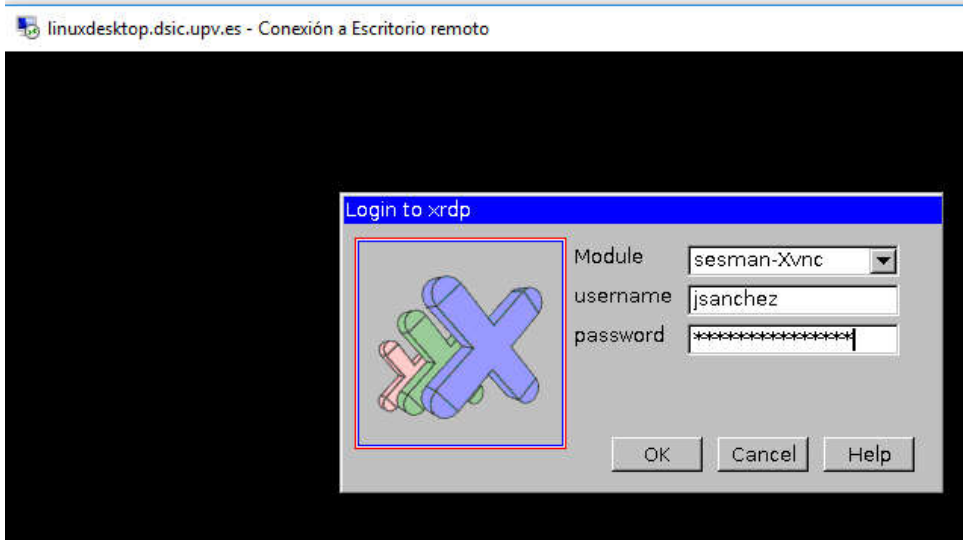
  TSR-474-1718.dsic.cloud

- Editaremos sobre el escritorio remoto y copiaremos los archivos a la máquina virtual
- Nos conectamos a linuxdesktop.dsic.upv.es
- En el laboratorio arrancamos Linux, desde fuera de la Universidad y desde un sistema Windows



Copiar archivos a la máquina virtual

- Utilizamos las credenciales del DSIC



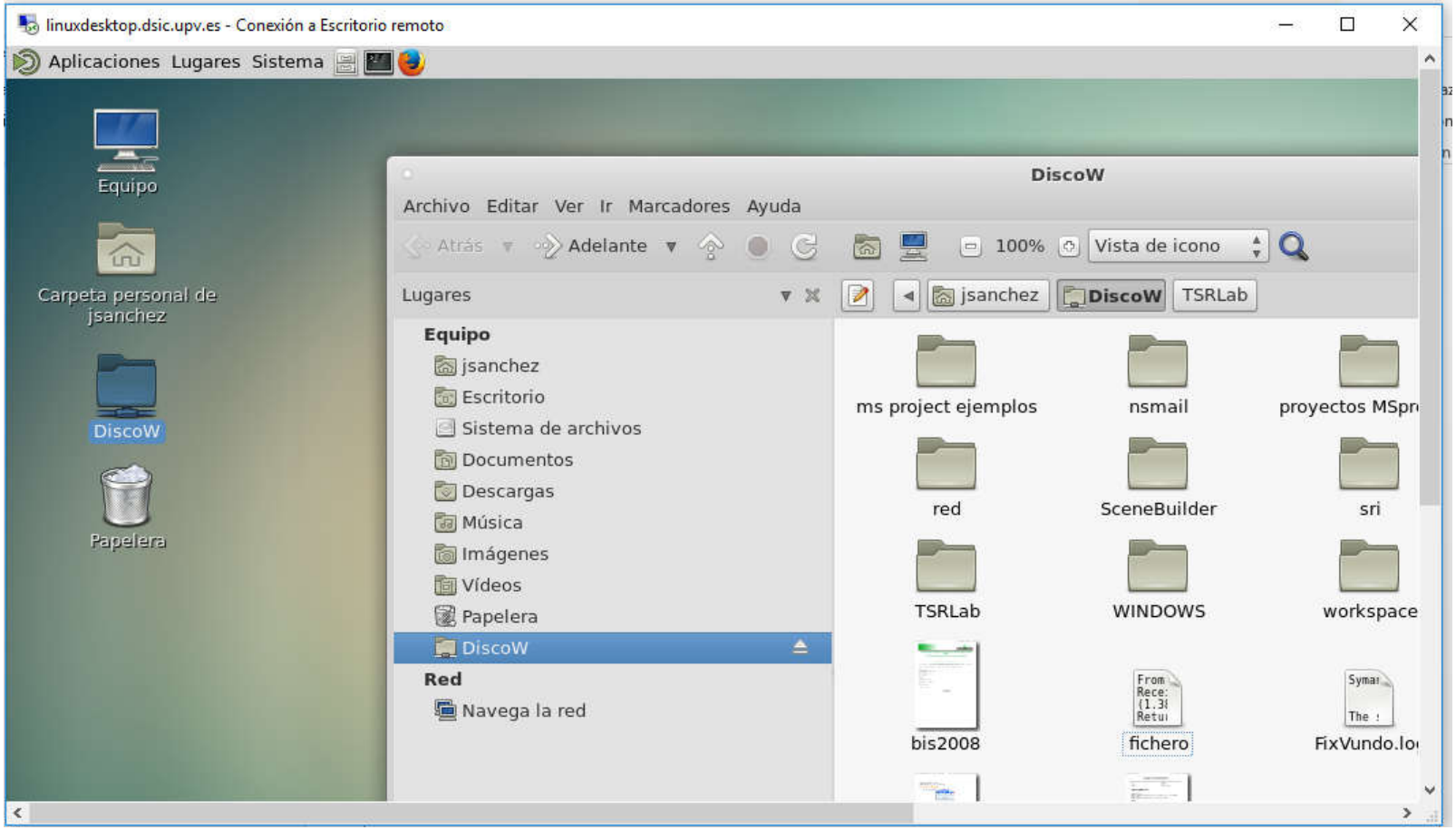
- Una vez dentro, arrancamos el navegador apuntando al portal: portal-ng.dsic.cloud y arrancamos nuestra máquina virtual si estuviese detenida

Copiar archivos a la máquina virtual

- Nos ha montado nuestra unidad W, podemos utilizar DiscoW para editar ficheros y luego moverlos a la máquina virtual, o bien podemos editar los archivos en la máquina virtual usando una conexión al sistema de archivos (explicado después)

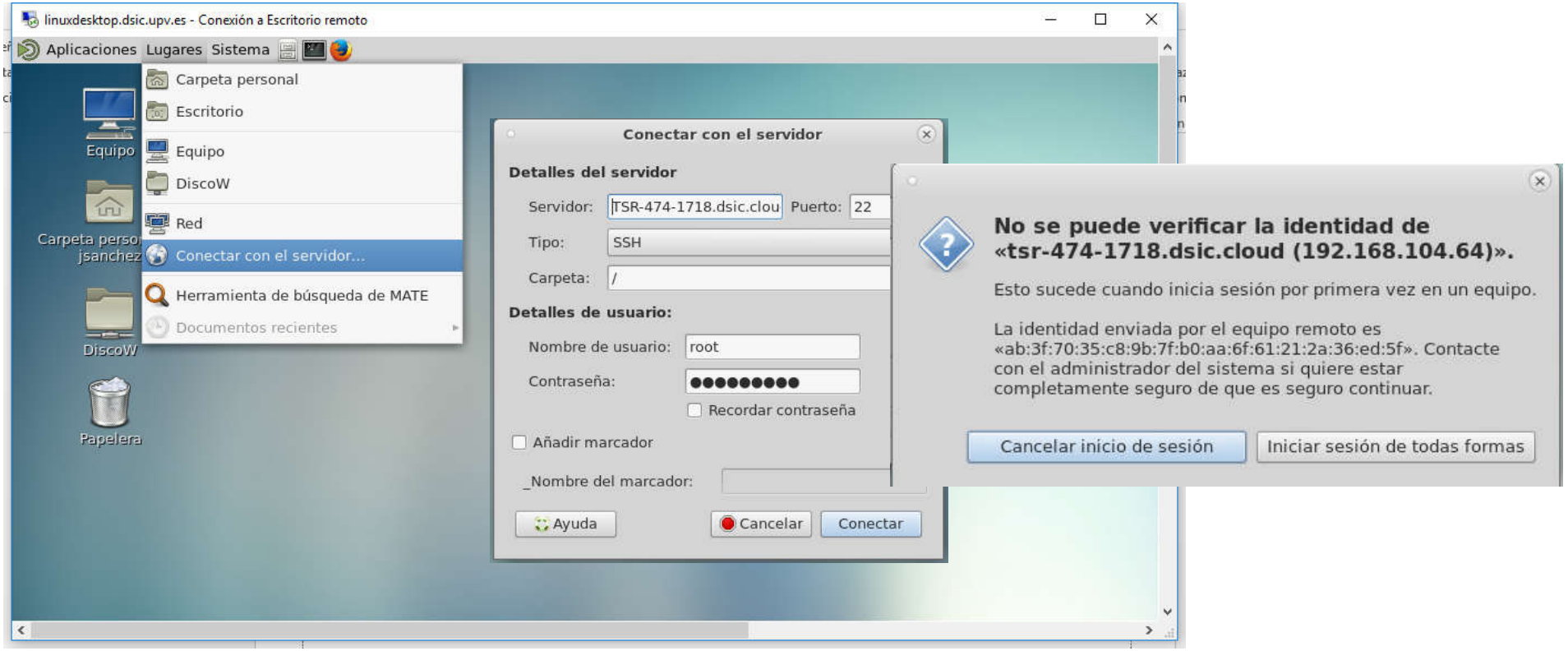
Copiar archivos a la máquina virtual

- Creamos un directorio en W, TSRLab



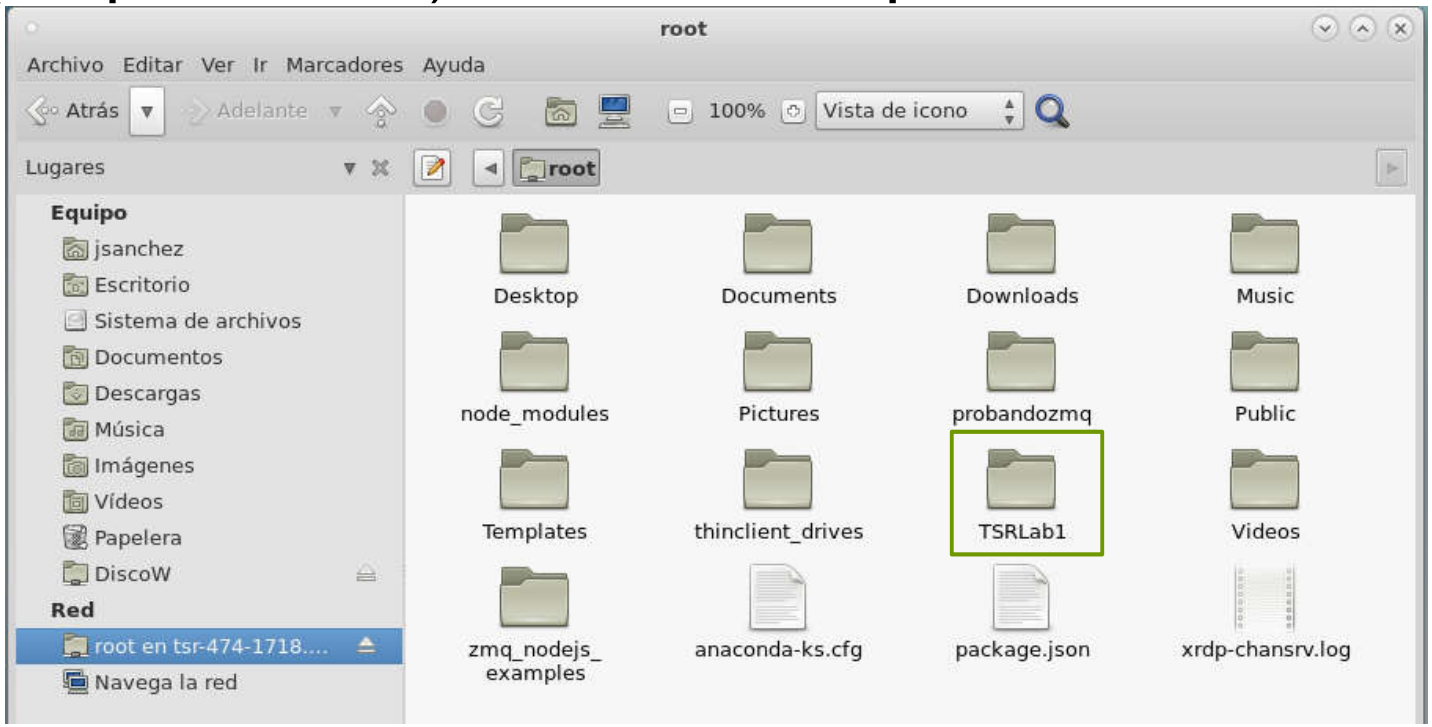
Copiar archivos a la máquina virtual

- Usaremos ahora una conexión al sistema de archivos de la máquina virtual



Copiar archivos a la máquina virtual

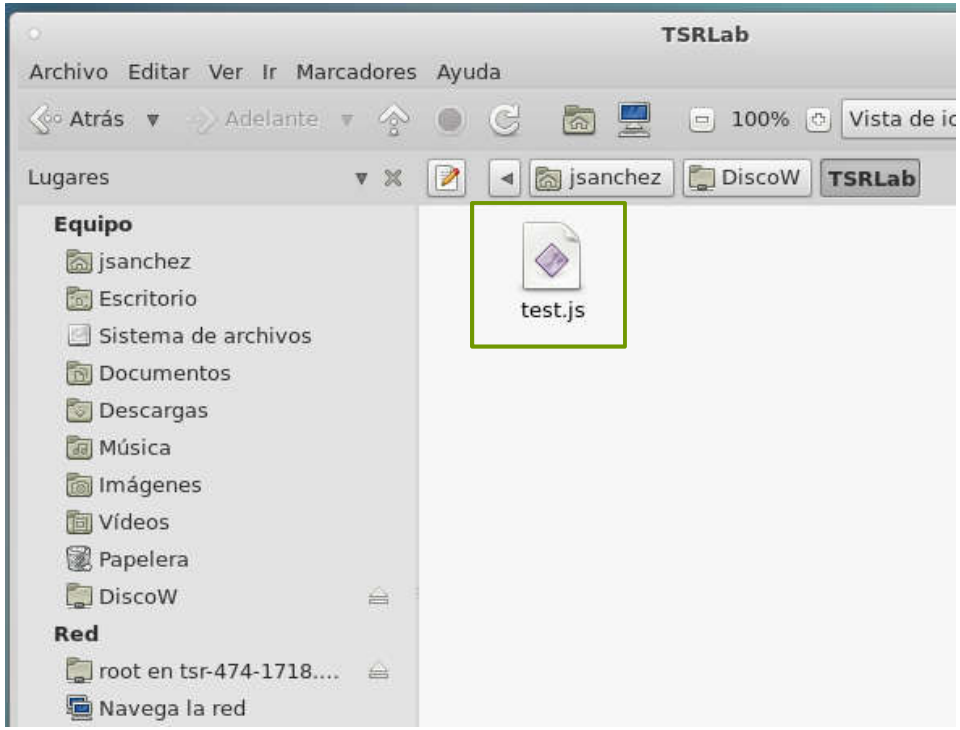
- Ahora podemos trabajar con el sistema de archivos remoto (máquina virtual) mediante el explorador



- TSRLab1 es un directorio de nuestra máquina virtual

Copiar archivos a la máquina virtual

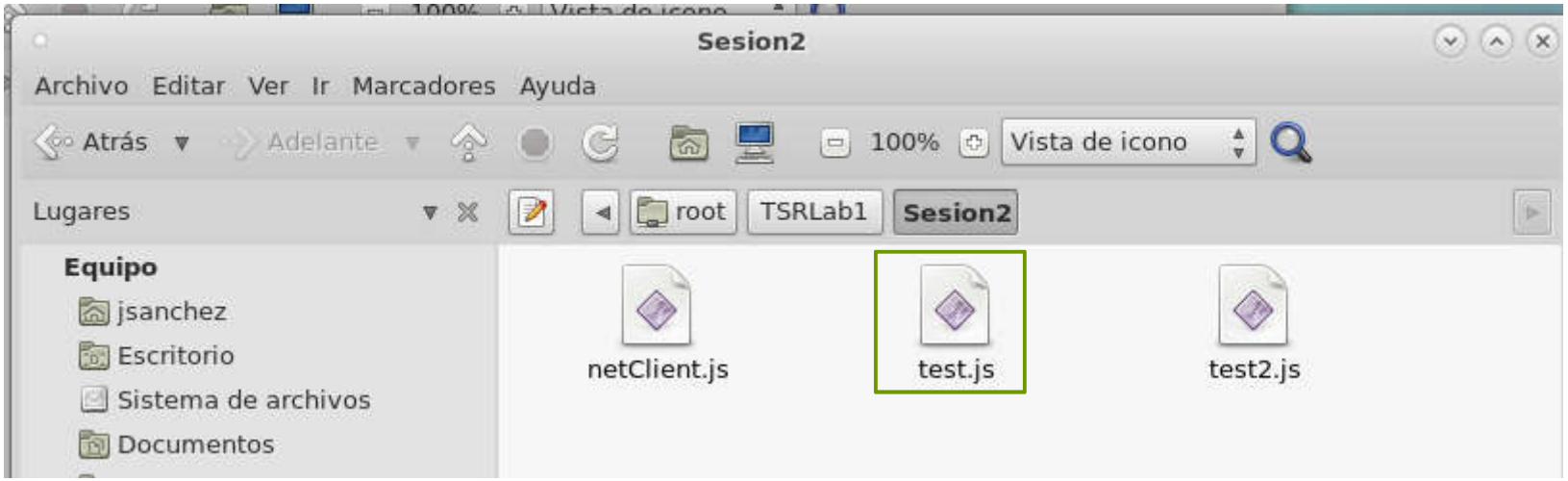
- En DiscoW hemos creado un directorio TSRLab con un archivo test.js



- Lo seleccionamos y copiamos (botón derecho del ratón) y navegamos a la máquina virtual para pegarlo.

Copiar archivos a la máquina virtual

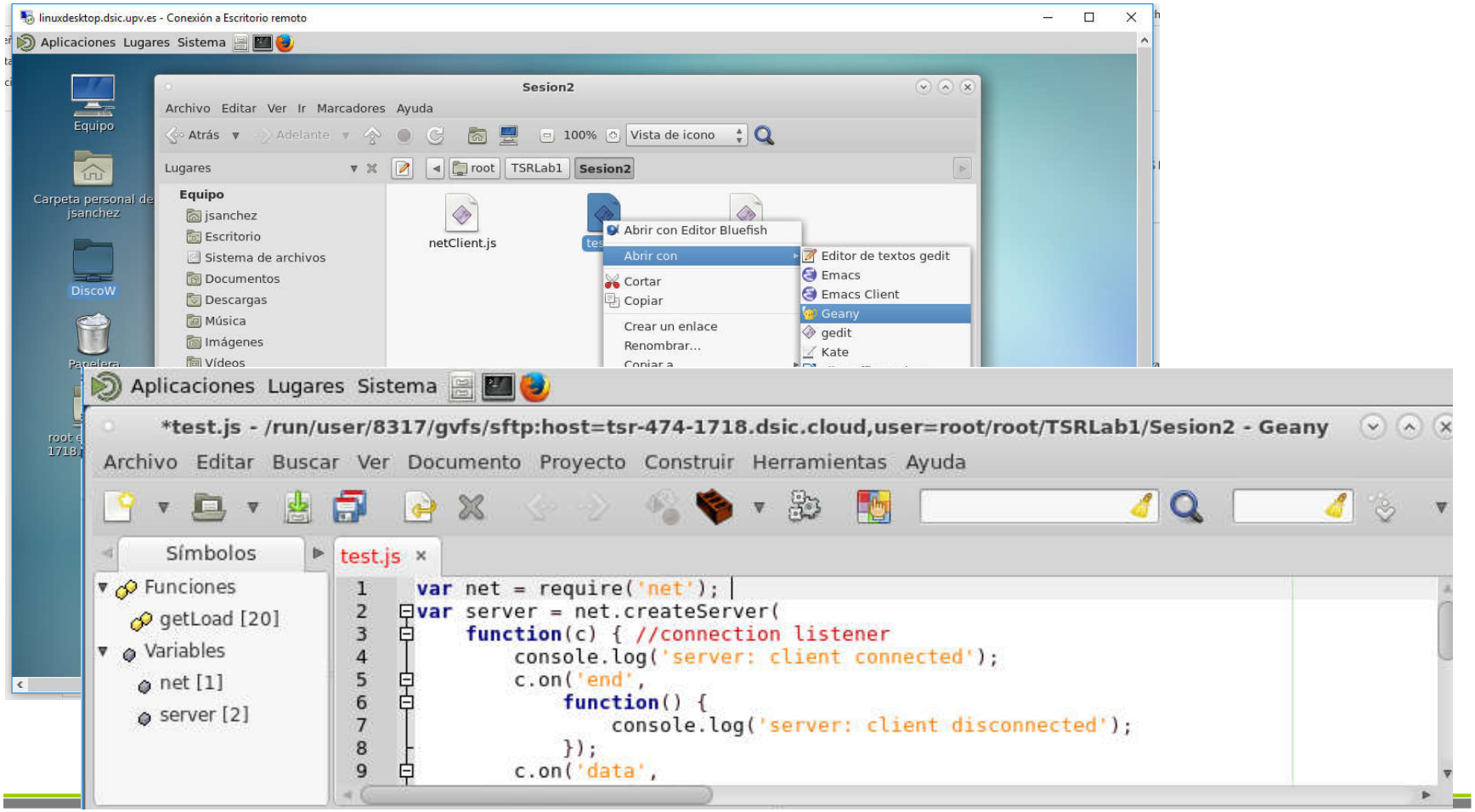
- Lo pegamos en el subdirectorio de TSRLab1 Sesion2



- Puede aquí abrirse con cualquier editor funcionando las teclas {}[]

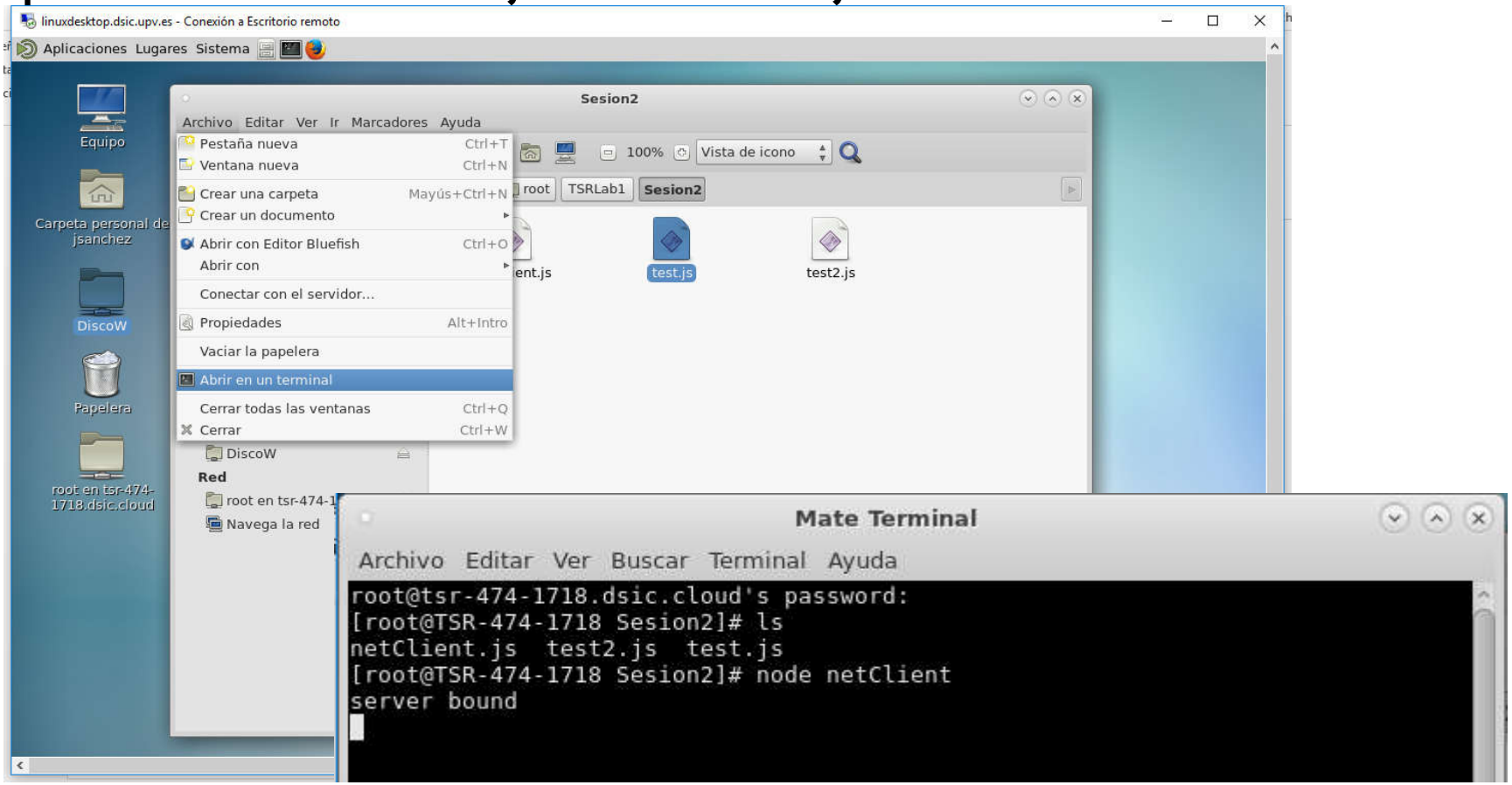
Copiar archivos a la máquina virtual

- Una vez seleccionado y abierto podemos modificar el contenido



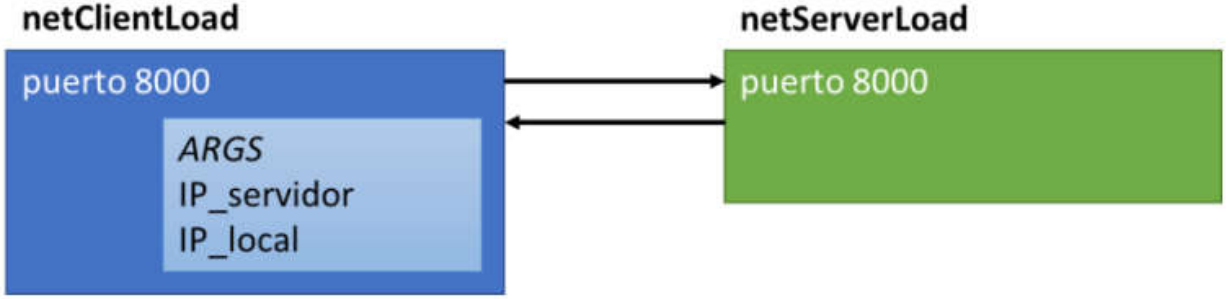
Ejecutar Node en la máquina virtual

- Dentro de la conexión al sistema de archivos remoto podemos también ejecutar node.js abriendo un terminal



Sesión 2: 2.1 Consulta de la carga del equipo

- La solución es similar al ejemplo de la página 8, hay que tener en cuenta que la conexión del cliente no es en localhost
- Utilizaremos la estructura del cliente y servidor de la página 8 del boletín



- Deshabilite en el cortafuegos los puertos 8000-8100 para permitir el acceso a la máquina servidora

```
$ firewall-cmd --permanent --add-port=8000-8100/tcp
$ firewall-cmd --reload
```

Sesión 2: 2.1 Consulta de la carga del equipo

Estructura general del cliente

```
var net = require('net');
if (process.argv.length == 4) {
    IP_remota = process.argv[2].toString();
    IP_local = process.argv[3].toString();
    console.log('IP remota '+ IP_remota);
}
else {
    console.log('número de argumentos incorrecto');
    process.exit ();
}

var client = net.connect({port:8000, host:……},
    function() { //connect listener
        console.log('client connected');
        Peticion = … //construir la petición al server
        client.write(Peticion);
    });
client.on('data',
    function(data) {
        console.log(data.toString());
        client.end(); // finaliza el cliente
    });
client.on('end',
    function() {
        console.log('client disconnected');
    });
```

Captura los argumentos de la línea de comandos

Si se omite host: la conexión es a localhost, Debe poner la IP del servidor

Petición al servidor

Respuesta del servidor

Eco de cliente desconectado

Sesión 2: 2.1 Consulta de la carga del equipo

Estructura general del servidor

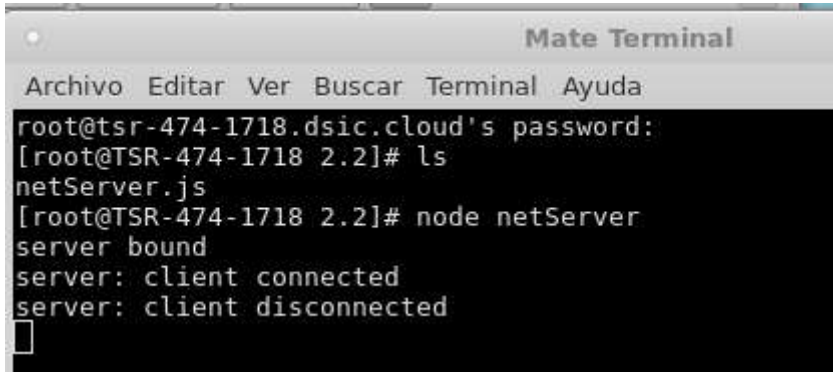
```
var net = require('net');
var server = net.createServer(
    function(c) { //connection listener
        console.log('server: client connected');
        c.on('end',
            function() {
                console.log('server: client disconnected');
            });
        c.on('data',
            function(data) {
                Respueta = f(data); // construye la respuesta
                c.write(Respuesta); // envía la respuesta
                c.end(); // close socket
            });
    });
server.listen(8000,
    function() { //listening listener
        console.log('server bound');
    });
…
```

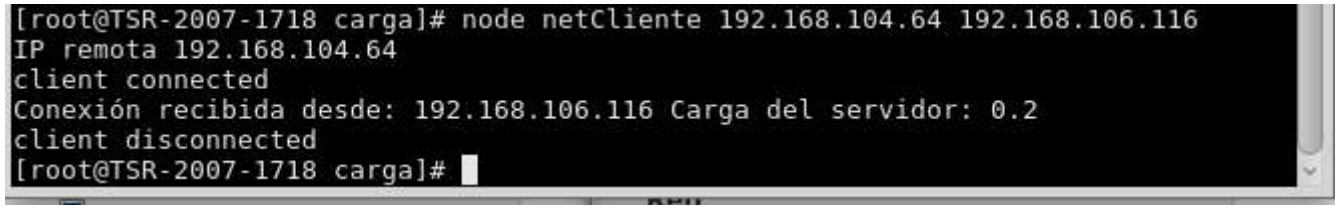
Cuando recibe una petición del cliente construye la respuesta. f(data) tienen que implementarla

Escucha peticiones en el puerto 8000

Sesión 2: 2.1 Consulta de la carga del equipo

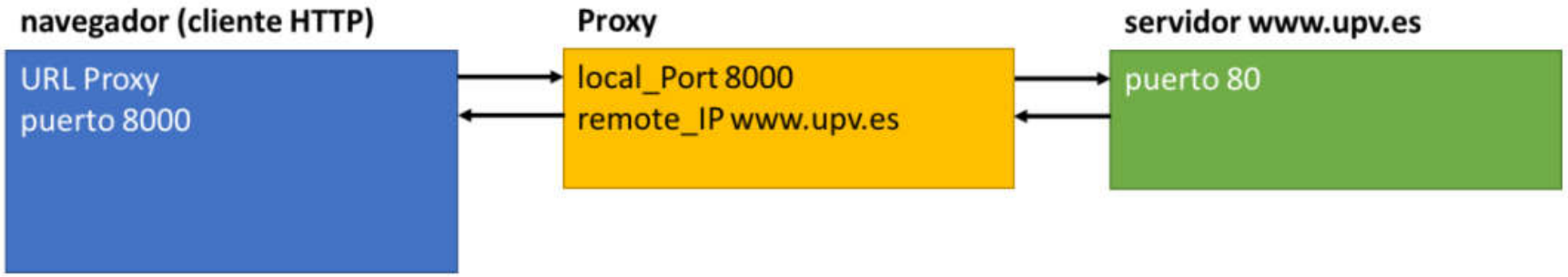
- Desde Linuxdesktop podemos acceder a la máquina servidora y cliente.
- Al ejecutar debe producir algo parecido a:



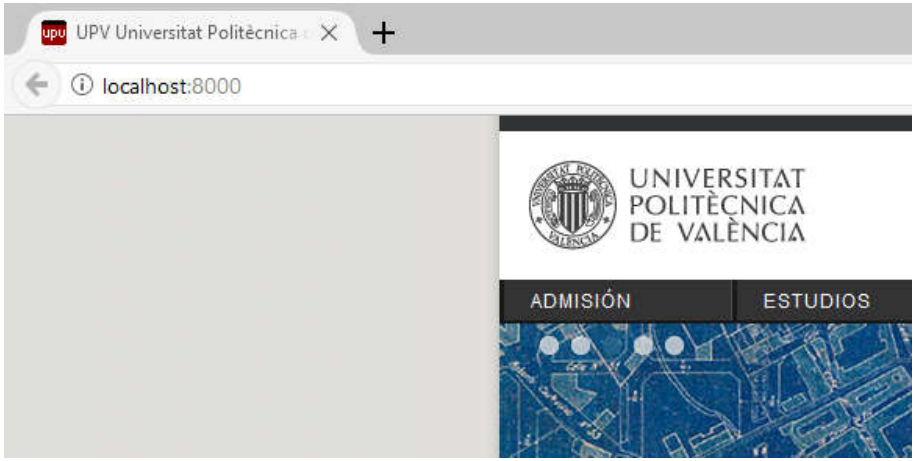


Sesión 2: Escenario 1

- Proxy básico

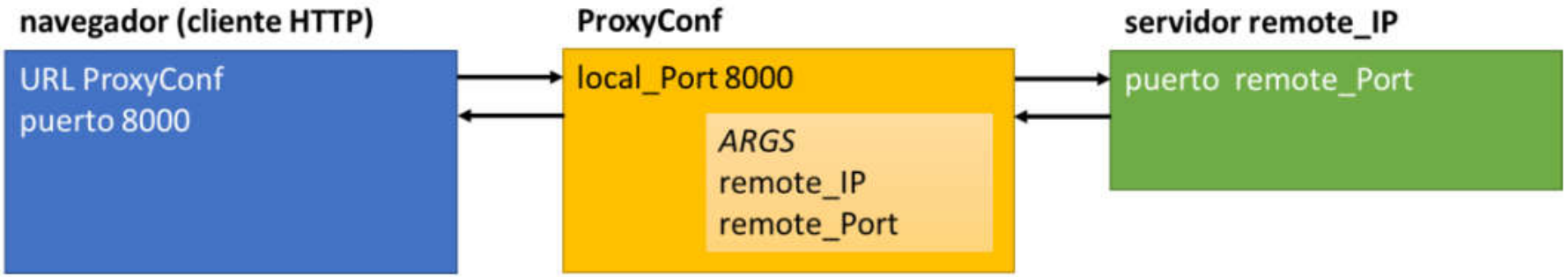


- Ejecutar el programa de la página 12 y arrancar el navegador



Sesión 2: Escenario 2

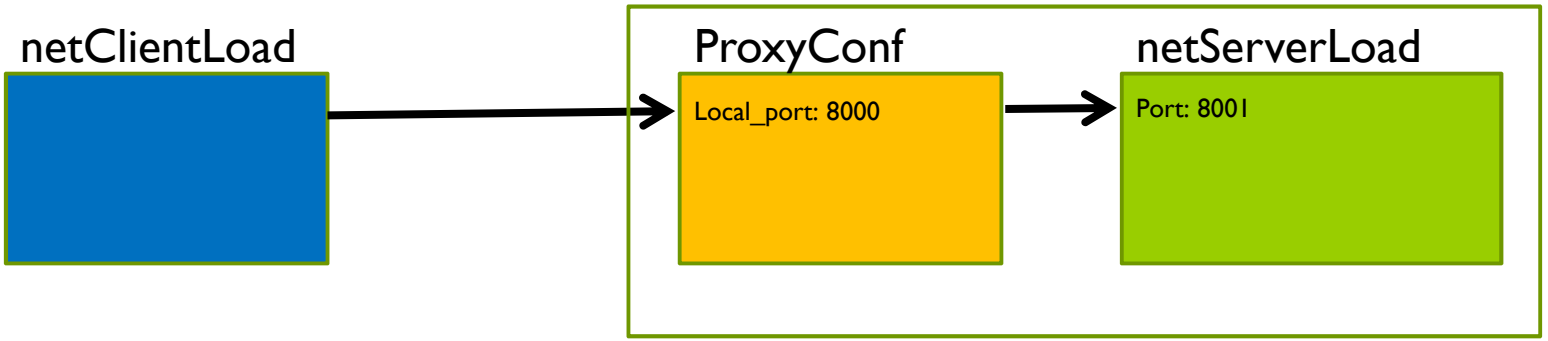
- Proxy configurable



- Probarlo con el navegador Web accediendo a localhost:8000, recuerde que el puerto y la IP del servidor remoto (Upv en este caso) se pasan desde la línea de comandos a ProxyConf
- ProxyConf como intermediario entre netClientLoad y netServerLoad

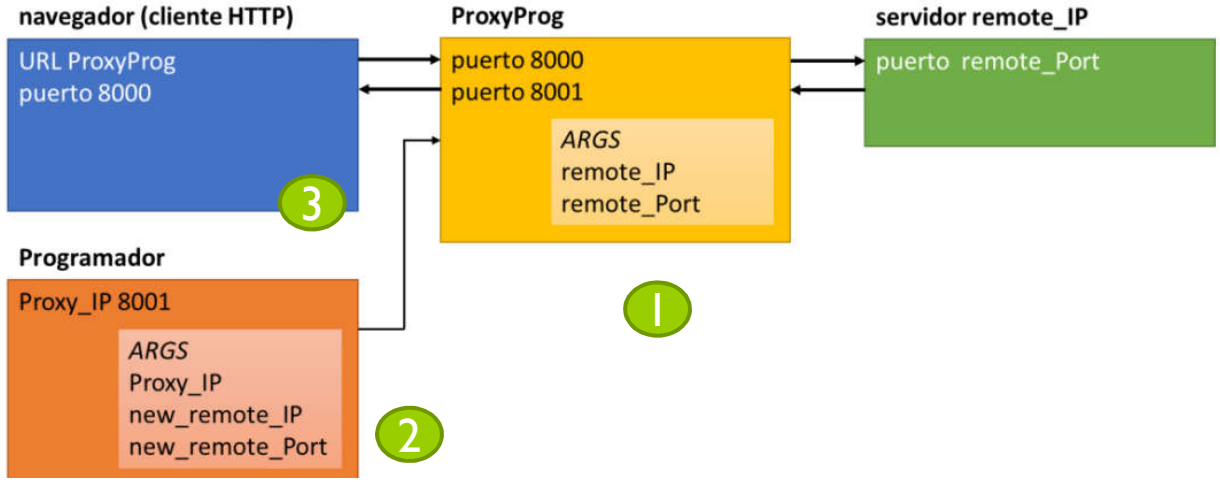
Sesión 2: Escenario 2

- Situar ProyConf y NetServerLoad en la misma máquina, debe cambiar los puertos para no tener colisiones



Sesión 2: Escenario 3

- Programador: puede ejecutarlo en la misma máquina



- Secuencia de arranque 1,2 y 3 con el navegador apuntando a localhost:8000
- Al ejecutar aparecerá la página del DSIC