



# Prácticas TSR

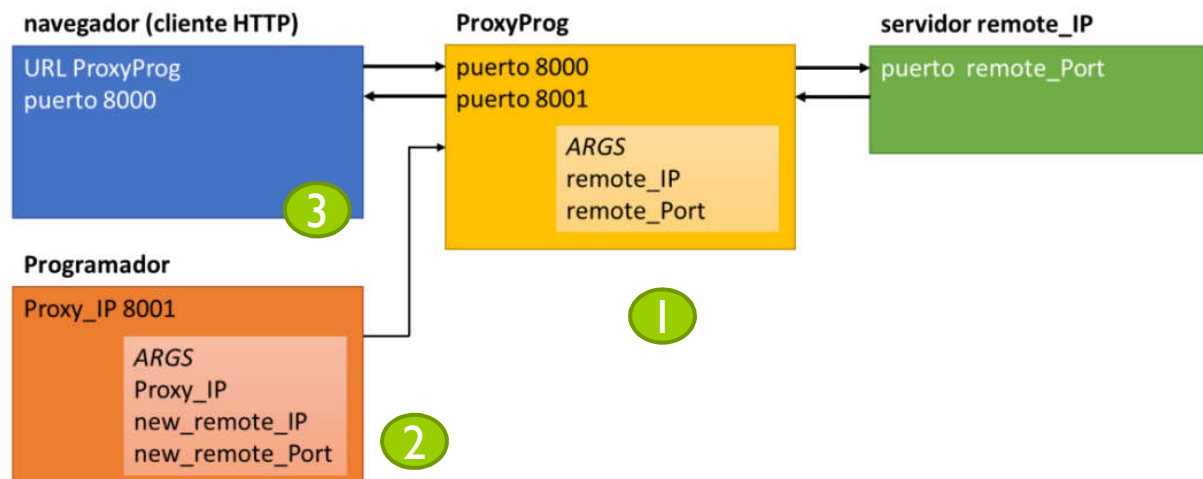
## Sesión 3

Juan Sánchez ([jsanchez@dsic.upv.es](mailto:jsanchez@dsic.upv.es))

Despacho 2D08

## De la sesión anterior: Escenario 3

- ▶ Programador: puede ejecutarlo en la misma máquina

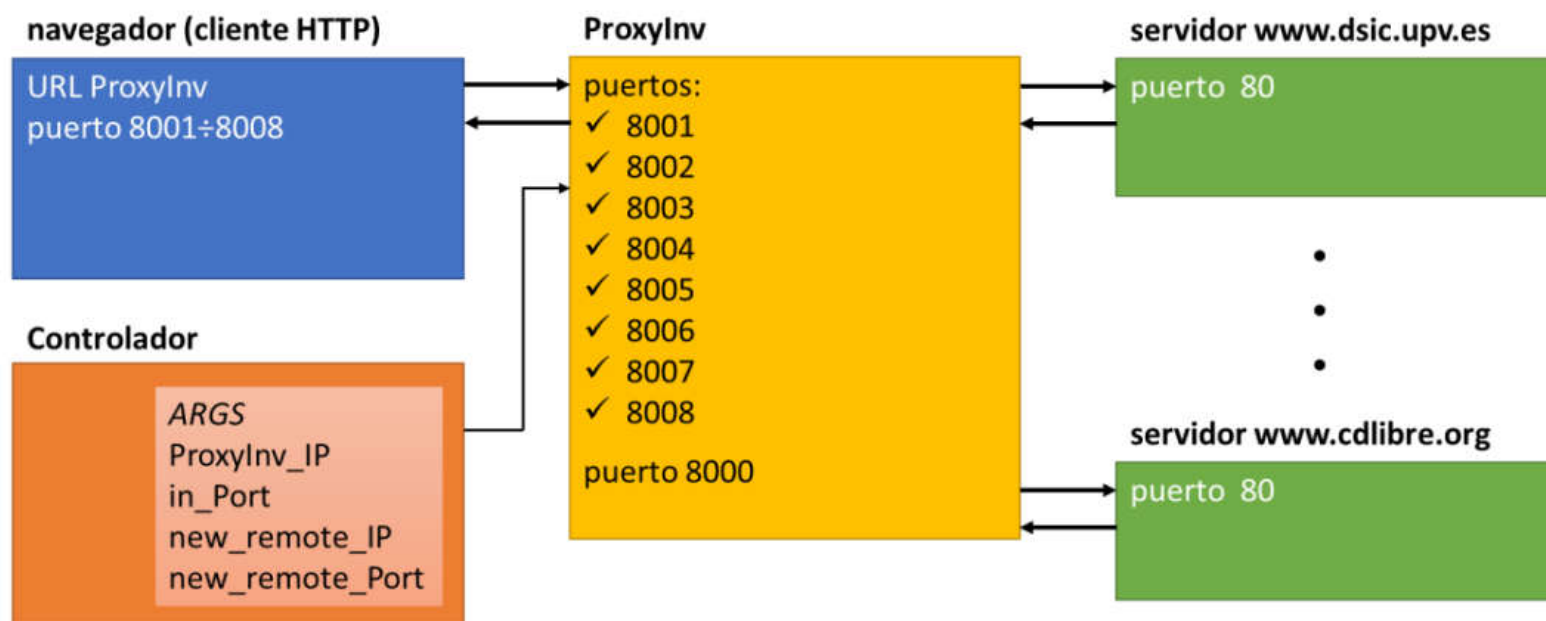


- ▶ Secuencia de arranque 1,2 y 3 con el navegador apuntando a localhost:8000
- ▶ Al ejecutar aparecerá la página del DSIC



## Sesión 3

- El problema es similar al anterior, ahora tenemos





## Sesión 3

- Para manejar las ips y los puertos podemos definir un objeto

```
// sesión 3 proxy inverso
// acepta peticiones desde el puerto 8001 al puerto 8008
// en el puerto 8000 se pueden modificar tanto la ip_remota como el puerto remoto,
// indicando qué se quiere cambiar

// Valores de prueba para ip y puerto de conexión http desde el navegador
var puertoYDirecciones;
puertoYDirecciones = {
    // los valores son ficticios, los transforma posteriormente el controlador
    8001: {IP: '1.2.3', Puerto: 800},
    8002: {IP: '1.2.3', Puerto: 800},
    8003: {IP: '1.2.3', Puerto: 800},
    8004: {IP: '1.2.3', Puerto: 800},
    8005: {IP: '1.2.3', Puerto: 800},
    8006: {IP: '1.2.3', Puerto: 800},
    8007: {IP: '1.2.3', Puerto: 800},
    8008: {IP: '1.2.3', Puerto: 800},
    8009: {IP: '1.2.3', Puerto: 800}
};
```



## Sesión 3

- Con respecto al ProxyInv, escucha al controlador en el puerto 8000

```
// espera a conexiones del controlador en el puerto 8000, Los datos vienen en formato JSON
var server = net.createServer(
  function(c) { //connection listener
    console.log('server: Controlador conectado');
    c.on('end',
      function() {
        console.log('server: Controlador desconectado');
      });
    c.on('data',
      function(data) {
        var nuevosValores = JSON.parse(data);
        // actualiza el objeto puertosYDirecciones
        puertoYDirecciones[nuevosValores.in_port]['IP'] = nuevosValores.remote.ip;
        puertoYDirecciones[nuevosValores.in_port]['Puerto'] = nuevosValores.remote.port;
        c.end(); // close socket
      });
  });
server.listen(8000, // escuchando en localhost
  function() { //listening listener
    console.log('server escuchando puerto 8000 para cambios de ip remota y puertos remotos');
  });
```



## Sesión 3

- También el ProxyInv tenemos que escuchar a peticiones que llegan desde el puerto 8001 al puerto 8009

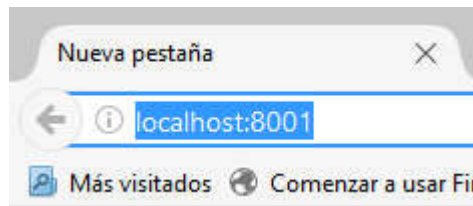
```
// se muestra únicamente el que escucha al puerto 8001
var server = net.createServer(function (socket) {
  var serviceSocket = new net.Socket();
  serviceSocket.connect( parseInt(puertoYDirecciones[8001]['Puerto']),
    puertoYDirecciones[8001]['IP'], function () {
      socket.on('data', function (msg) {
        console.log('Enviando petición al puerto remoto e ip remota ' +
          puertoYDirecciones[8001]['Puerto'] + ' ' + puertoYDirecciones[8001]['IP']);
        serviceSocket.write(msg);
      });
      serviceSocket.on('data', function (data) {
        socket.write(data);
      });
    });
}).listen(8001);
```



## Sesión 3

### ► Secuencia de ejecución:

- Node ProxyInv
- Node Controlador 127.0.0.1 8001 158.42.184.5 80
- Navegador





## Sesión 3: Programador

---

- ▶ El programador envía el siguiente dato

```
// pasamos los datos en formato JSON
var msg = JSON.stringify (
  {   "op": "set",    // no es necesario este parámetro
      "in_port": IN_PORT,
      "remote": {
        "ip": NEW_REMOTE_IP,
        "port": NEW_REMOTE_PORT
      }
  }
})
```