

# Prácticas de Laboratorio

## Fases 1.2 y 1.3: Desarrollo de un foro

### Servidor de datos y cliente

## Servicios y Aplicaciones Distribuidas

### Introducción

---

En esta fase del trabajo, se proporciona el foro completamente desarrollado, en su versión cliente web y servidor web. Tal y como se proporciona para esta fase del trabajo, cuenta con una funcionalidad que permite:

- Como usuario anónimo, permite ver los mensajes contenidos en los diferentes temas y crear usuarios
- Conectarse al foro introduciendo las correspondientes credenciales. Para todos los usuarios ya creados, la contraseña es '1234';
- Como usuario conectado, se pueden crear temas, enviar mensajes a los diferentes temas y usuarios, navegar entre los temas y mensajes privados y desconectar.

El foro consiste en una parte cliente desarrollada para la Web, y una parte servidora desarrollada en Node.js

La parte cliente está realizada usando las tecnologías: HTML5, Javascript, jQuery, y socket.io, y no debería ser necesario modificarlo, si bien un breve estudio, sería conveniente. Para ello, **resulta conveniente repasar las actividades del "seminario 4"**, en el que se presenta la versión más simplificada posible del foro.

La parte servidora está desarrollada en Javascript sobre Node.js, usando los módulos express y socket.io. **En esta práctica nos centraremos en la parte servidora, desarrollando un servidor adicional de datos y un cliente** que nos permitirá probar su funcionalidad y estudiar

diferentes casos presentes en las prácticas siguientes.

## Instalación del software

---

Descargar el código fuente completo para esta fase del trabajo, disponible en Poliformat  
→ SAD → recursos → laba → lab1.3 → forum.tgz

Descomprimir el archivo que hemos descargado:

```
tar xzvf forum.tgz
```

Para completar la instalación, debemos situarnos en el directorio 'forum' e instalar las dependencias que tiene el software mediante la orden 'npm'. Esta orden nos descargará los módulos que nuestro servidor necesita.

```
cd forum  
npm install
```

Ponemos en marcha el servidor utilizando la orden 'node':

```
node forum.js
```

Una vez realizado esto, el servidor para el foro estará funcionando en el puerto 10000 de la máquina local. Abrimos un navegador Chrome que actuará como cliente. En la url, ponemos <http://localhost:10000>. Podemos tener varios navegadores o pestañas abiertas al mismo tiempo para observar el dinamismo del foro, una vez vamos creando temas, usuarios y mensajes.

También se proporciona un servidor de datos, completado en su mayoría, a falta de incluirle algunas funciones. El servidor es funcional para servir listas de mensajes y listas de temas, faltando el resto de funciones propias del foro.

Este servidor, que por defecto escucha en el puerto 9000, puede ejecutarse mediante la orden 'node':

```
node dmserver.js
```

De igual forma, se proporciona un cliente mínimo que accede al servidor "dmserver" y le pide la lista de temas actuales. Para ejecutarlo:

```
node dmclient.js
```

## Descripción de los ficheros de código fuente proporcionados.

---

**views/index.html:** Es la interfaz gráfica del foro. No necesita modificarlo.

**forum.js:** El servidor web. Puede observarse que está completamente desarrollado para la funcionalidad que espera el cliente web. No debe modificarse más que para cambiar el módulo de datos, entre local y remoto.

**dm.js:** Módulo básico de datos, que mantiene toda la información del foro en memoria, mediante simples vectores y tablas hash. No debería modificarse.

**dm\_local.js:** Módulo de acceso local a 'dm'. Este módulo simplemente añade funcionalidad de callbacks al módulo "dm". Por tanto, como puede observarse al ver su código no tiene más que la reescritura de las funciones de "dm", para que usen callbacks al terminar. No debe modificarse.

**dm\_remote.js:** Módulo que tiene la misma signatura que "dm\_local", cuya función consiste en enviar las peticiones al servidor "dmserver". Este fichero es el más interesante para esta práctica. Está desarrollado al completo en su parte más delicada, de gestión de invocaciones y la gestión de callbacks, mientras que las diferentes funciones del foro están parcialmente completadas. Debe implementar las funciones que faltan.

**dmclient.js:** Sencillo cliente, que tan solo pide la lista de temas al servidor. Puede observarse la simplicidad de este programa, contando con la ayuda de "dm\_remote". Es programa lo deberá ampliar.

**dmserver.js:** Programa servidor TCP que sirve los datos del foro. Debe ampliar ligeramente su funcionalidad para dotarle de todas las funciones del foro (10 funciones).

## Trabajo a realizar

---

Esta práctica viene a suceder a la anterior y a ampliarla, dando más ayuda de la que se proporcionaba en la fase anterior. Adicionalmente se pide el desarrollo de un cliente compacto con el que interrogar al servidor.

El desglose de las tareas a realizar es el siguiente:

1.- Completar "dm\_remote", para que cuente con todas las funciones presentes en "dm\_local". Para completarlas, debe tomar como base las funciones que ya están implementadas.

2.- Completar "dmserver", para que:

- Escuche en el puerto que indiquemos como argumento
- Sirva todas las peticiones que le puedan enviar los clientes del foro. Observe el

código e implemente las funciones que faltan.

3.- Ampliar el programa "dmclient", para que:

- Conecte al servidor:puerto que pasemos como argumento
- Los siguientes argumentos en línea de órdenes serán las órdenes a enviar al servidor "dmserver". Por ejemplo, debería ser posible ejecutar la orden:

```
node dmclient.js 127.0.0.1:9000 'get private message list' troll mudito
```

4.- Conectar "forum.js" a "dm\_remote". Una vez realizados y probados tando "dmclient" como "dmserver", conectar el servidor web a "dm\_remote", es una tarea trivial, que tan sólo exige cambiar la sentencia "require" y comenzar el módulo remoto con "Start(host, port)".

## Recursos y bibliografía

---

- JavaScript:  
<http://www.w3schools.com/js/default.asp>  
[http://www.hunlock.com/blogs/Essential\\_Javascript\\_--\\_A\\_Javascript\\_Tutorial](http://www.hunlock.com/blogs/Essential_Javascript_--_A_Javascript_Tutorial)
- Node.js  
<http://nodejs.org/documentation/api/>
- Apuntes de la asignatura: Seminario 3: JavaScript y Node.js
- Actividades del Seminario 4.