

LABORATORIO 1 SISTEMAS DISTRIBUIDOS

“Aplicación Web usando Node.js”

Profesor: Nicolás Hidalgo

Ayudante: Luis Eduardo Celedón (luis.celedon@usach.cl)

Segundo Semestre 2014

OBJETIVOS

Realizar una aplicación distribuida basada en el modelo cliente/servidor, en tiempo real y orientada a eventos utilizando como tecnología principal Node.js.

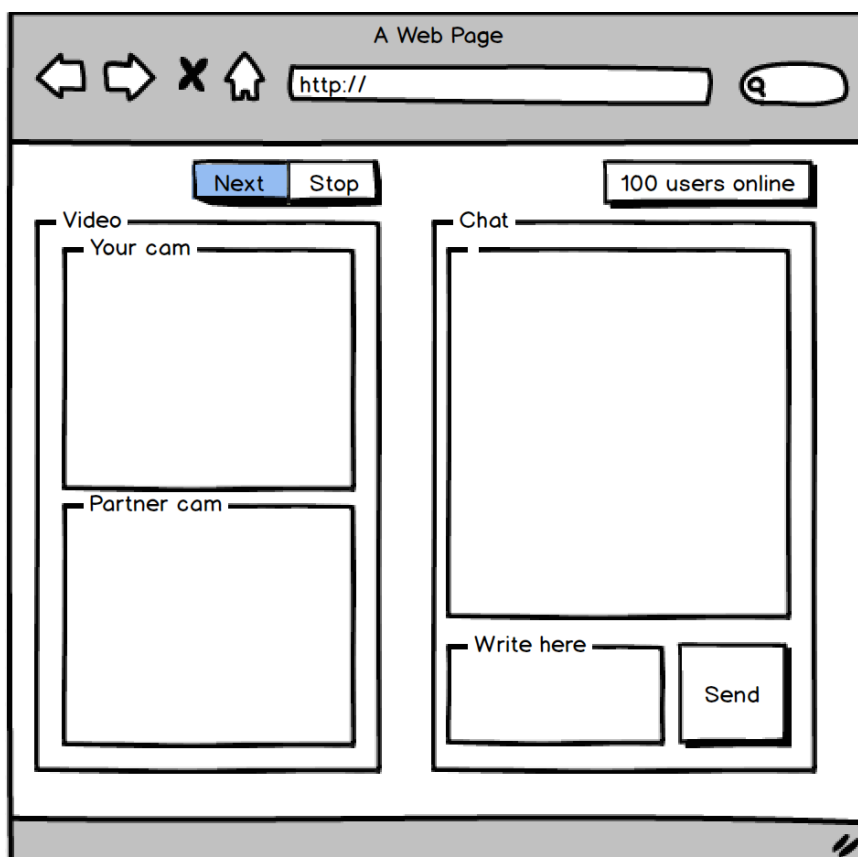
INTRODUCCIÓN

Cada día son más las aplicaciones web que requieren interacciones en tiempo real por parte de los usuarios y del servidor. Con el surgimiento de la Web 2.0, los usuarios ya no son entes consumidores de contenidos, sino que son ellos los principales generadores del mismo, interactuando y colaborando entre sí. Ejemplo de esto son las ya conocidas redes sociales y Wikis, entre muchos otros casos.

El presente laboratorio busca interiorizar al alumno con la tecnología NODEJS, una tecnología basada en JavaScript que permite el procesamiento y manejo de eventos en tiempo real dentro de una aplicación web. De manera opcional (PERO MUY RECOMENDADO) se podrá utilizar la librería Socket.io para establecer de mejor manera las comunicaciones entre clientes y el servidor, además de Express.js, el cual es un framework que ayuda a manejar servidores HTTP sirviendo como middleware.

ENUNCIADO

Se le solicita a los alumnos del curso de sistemas distribuidos que generen una aplicación Web que permita que usuarios aleatorios puedan interactuar por medio de mensajes de texto (chat) y video, utilizando sus cámaras Web. Los usuarios de la aplicación serán capaces de interactuar por medio de estas dos vías de manera simultánea. Adicionalmente, el usuario debe poder cambiar de usuario con el cual interactúa. Dicho cambio se realiza de manera aleatoria utilizando la opción NEXT. El nuevo usuario con el cual se debe interactuar se debe elegir de manera aleatoria desde aquellos usuarios que no tienen compañero de chat asignado o se encuentran en el mismo estado que el usuario que aplicó NEXT.



Al comienzo de la sesión, el usuario debe ser capaz de introducir un *nickname* el cual servirá para identificarse dentro del universo de usuarios del sistema, por lo que debe ser único. Esto debe hacerse antes de ingresar a la ventana de chat.

En la figura anterior se muestra como podría lucir su laboratorio una vez terminado, sin embargo no tiene porque ser exactamente igual, lo importante es que cumpla las siguientes funcionalidades mínimas:

- Realizar una conexión entre dos usuarios de manera aleatoria, utilizando el servidor como conector entre los distintos dispositivos.
- La elección aleatoria debe ser entre usuarios que no estén participando de alguna videoconferencia. El algoritmo de selección queda a su criterio.
- La videoconferencia, aparte de video debe ser capaz de transmitir audio desde el micrófono del equipo.
- El chat debe ser único entre usuarios y no puede haber una conversación entre un usuario y otro que interrumpa la comunicación entre dos usuarios.
- Cada usuario puede mantener solo una conversación a la vez.
- Si no existen suficientes usuarios en línea, se debe informar eso al usuario para que espere.
- El video será entre solo dos, no podrá haber interferencias con otras video conferencias.

El servidor deberá además poder:

- Manejar múltiples conexiones de usuarios de manera paralela.
- Es acá donde se realiza la selección de compañeros de chat, esto no se debe hacer del lado del cliente.
- Recordar que toda comunicación debe pasar por el servidor, los clientes no interactúan directamente entre ellos, es acá donde se manejan todas las conexiones.

Para realizar este laboratorio, debe utilizar JavaScript, Node.js (como servidor para procesar las peticiones de los usuarios) y como se mencionó anteriormente puede utilizar librerías adicionales como socket.io, express.js o cualquier otra que usted estime pertinente y que mejoren la experiencia y usabilidad de su aplicación.

Los aspectos administrativos para esta experiencia son:

- El laboratorio debe realizarse en grupos de máximo 2 personas, esto es INAPELABLE.
- Tiene 2 semanas para completar el laboratorio a partir del día en que el enunciado sea subido al moodle.
- El código fuente debe ser subido a moodle en el apartado que se habilitará, bajo la forma: “Apellido 1 - Apellido 2.zip”
- Se debe entregar un código debidamente comentado e indentado. El no cumplimiento de esto resultará en un descuento a su nota final.
- La revisión se realizará de manera presencial en horario de laboratorio, utilizando la versión enviada a usachvirtual.
- Por cada día de atraso, se descontará 1 punto de la nota final.
- El no uso de GIT, resultará en la calificación mínima para el grupo, por lo que se debe compartir el proyecto con el ayudante.
- Si se basan en código de Internet, este debe estar con su correspondiente referencia o serán evaluado con la calificación mínima.
- Cualquier copia entre compañeros, resultará con la calificación mínima para todos los involucrados.
- Para este laboratorio no se pedirá informe.

RECURSOS

- Git (Obligatorio)
- Trello (Muy recomendado)

REFERENCIAS

- NodeJS: <http://nodejs.org/>
- Socket.io: <http://socket.io/>
- ExpressJS: <http://expressjs.com/>
- JavaScript: <http://www.w3schools.com/js/default.asp>