

# LABORATORIOS VIRTUALES COLABORATIVOS

Carlos Alberto Jara Bravo, Francisco Andrés Candelas Herías y Fernando Torres Medina  
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal  
Universidad de Alicante  
[cajib@dfists.ua.es](mailto:cajib@dfists.ua.es), [Francisco.Candelas@ua.es](mailto:Francisco.Candelas@ua.es), [Fernando.Torres@ua.es](mailto:Fernando.Torres@ua.es)

Sebastián Dormido Bencomo  
Departamento de Informática y Automática  
Universidad Nacional a Distancia  
[sdormido@dia.uned.es](mailto:sdormido@dia.uned.es)

Francisco Esquembre Martínez  
Departamento de Matemáticas  
Universidad de Murcia  
[fem@um.es](mailto:fem@um.es)

## Resumen

*El uso del laboratorio virtual (LV) como herramienta para el aprendizaje y experimentación práctica a distancia, cada vez es mayor en el entorno de educación superior. Este objeto de enseñanza virtual representa actualmente una revolución dentro del ámbito educativo, ya que permite adquirir al alumno habilidades prácticas desde cualquier lugar y en cualquier momento. Por otro lado, el desarrollo de entornos web que permiten la colaboración a distancia entre profesores y alumnos, se ha convertido en una cuestión fundamental para la investigación de muchas universidades del mundo. Por lo tanto, con el objetivo de unir las ventajas de ambos recursos educativos, este artículo presenta la creación de un sistema de comunicación que permite a un grupo de alumnos, coordinados por un profesor, compartir experiencias y experimentos prácticos con laboratorios virtuales (LVs).*

**Palabras Clave:** colaboración a distancia, e-learning, java, laboratorio virtual.

## 1 INTRODUCCIÓN

La irrupción de las Tecnologías de la Información y la Comunicación (TIC) en todos los ámbitos de nuestro entorno, y por supuesto en el educativo, ha sido espectacular durante las últimas décadas. Esta revolución tecnológica ha dado lugar a la aparición de nuevas formas de enseñanza y aprendizaje, tales como los LVs. Este innovador objeto de aprendizaje representa actualmente una herramienta indispensable en la enseñanza universitaria, sobre todo en las carreras y cursos de contenido científico-

técnico [2]. Los LVs cubren desde simples simulaciones de fenómenos físicos [9], hasta aplicaciones que permiten el acceso remoto a los recursos de un sistema real [3], [7]. Son muchas las ventajas que ofrecen los LVs para la comunidad educativa y entre ellas, cabe destacar las siguientes:

- Son herramientas de auto-aprendizaje, donde el alumno puede configurar nuevos experimentos permitiendo obtener una visión más intuitiva de los fenómenos que ocurren en esos experimentos.
- Acercan y facilitan a un mayor número de alumnos la realización de experiencias sin riesgo a dañar los equipos, flexibilizando sus horarios y evitando el solapamiento con otras asignaturas.
- Reducen el coste del montaje y mantenimiento de los laboratorios tradicionales, siendo una alternativa barata y eficiente, donde el estudiante simula los fenómenos a estudiar como si los observase en la realidad.

Sin embargo, la mayoría de los laboratorios virtuales son diseñados para ser usados de forma individual (*laboratorio virtual monolítico*), y no permiten el trabajo en grupo ni la colaboración entre los estudiantes y el profesor.

Muchas de las investigaciones realizadas para determinar el impacto de los laboratorios virtuales en la docencia universitaria, tienen como conclusión que el estudiante valora muy positivamente la docencia virtual, y que los laboratorios virtuales le ayudan a comprender mejor los conceptos. Sin embargo, consideran los laboratorios virtuales como un apoyo a la docencia, no como una alternativa, ya que preferían asistir al laboratorio real, compartir sus problemas con los demás estudiantes y tener el apoyo de un profesor [4].

Durante los últimos años, se han desarrollado muchos entornos web para el aprendizaje a distancia donde se ha incluido el concepto de colaboración. El término anglosajón que define este tipo de entornos es *Computer-Supported Collaborative Learning Environments* (CSCLE), que puede traducirse como “Entornos de Aprendizaje Colaborativo” (EAC). Actualmente, podemos distinguir dos tipos de EAC dependiendo del tipo de interacción profesor-alumno: los *on-line* o síncronos y los *off-line* o asíncronos. Los síncronos son los entornos que más se aproximan a la enseñanza tradicional, ya que permiten a los estudiantes y profesores compartir experiencias en tiempo real durante la clase virtual. En cuanto a los asíncronos, la comunicación profesor-alumno no es de manera constante, y no tienen la necesidad de estar conectados al mismo tiempo al entorno colaborativo.

La mayoría de los EAC de carácter síncrono existentes en la actualidad, utilizan herramientas clásicas para la comunicación tales como chats, video, pizarras compartidas [8] o incluso con paneles de dibujo interactivos [10]. Sin embargo, dichas herramientas por sí solas carecen de la potencia educativa que puede aportar el LV a la hora de explicar conceptos de carácter técnico. Con respecto a los EAC de carácter asíncrono, cabe destacar *eMersion* [6]. Este sistema permite el acceso individual a LVs y remotos a través de la web, permitiendo compartir resultados de experimentos (*eJournal*), pero de una manera asíncrona.

Considerando lo expuesto, se extrae la conclusión de que existe una carencia de EAC que incluyen LVs como herramienta educativa. Los existentes se basan en combinar los LVs y las herramientas clásicas de comunicación [1]. Sin embargo, esa combinación ha dado como resultado entornos complejos y la necesidad de utilizar multitud de software para implementar la colaboración, lo que además conlleva a interfaces de usuario poco homogéneas. Por todo ello, nuestro grupo de investigación ha creado un sistema de comunicación síncrono a través de los protocolos TCP y UDP entre applets Java desarrollados con *Easy Java Simulations* (Ejs) [5], y de esta manera, poder integrar los LVs en entornos colaborativos *on-line* que permitan compartir experiencias prácticas en tiempo real a través de Internet entre alumnos y profesores.

El artículo se ha organizado de la forma siguiente. A continuación, se explican las características generales de las simulaciones creadas con Ejs. Posteriormente, se describen los aspectos más importantes del sistema de comunicación. En la sección 4, se muestran varias experiencias realizadas entre distintos LVs con el método de sincronización presentado. Finalmente, se comentan algunas conclusiones.

## 2 SIMULACIONES EJS

Ejs es una herramienta de software *open-source* diseñada para la creación de simulaciones dinámicas interactivas en lenguaje Java [5]. Ejs es parte del proyecto *Open Source Physics*, que se estableció con el objetivo de crear y distribuir simulaciones para su uso pedagógico en la enseñanza de la física.

Las simulaciones Ejs constan de dos partes: el modelo y la vista. El modelo está constituido por las variables de la simulación y por las ecuaciones matemáticas que rigen su evolución. La vista es la interfaz de usuario donde se muestra la representación gráfica de los diferentes estados del sistema. Cualquier cambio en el estado del modelo es visualizado en la vista, y si el usuario interacciona con ella modifica el valor de alguna variable del modelo.

Durante la ejecución de la simulación, Ejs crea las variables y ejecuta las ecuaciones del modelo cada cierto diferencial de tiempo. Esta última fase es conocida como paso de simulación o *step*. Posteriormente, Ejs actualiza la vista y vuelve a ejecutar las ecuaciones para continuar la evolución del sistema.

La interactividad de Ejs se encarga de gestionar los eventos que se producen cada vez que el usuario interacciona con algún objeto de la vista. El usuario es el encargado de darle funcionalidad a estos eventos. Para esta tarea, Ejs proporciona una serie de métodos predefinidos entre los que hay que destacar los siguientes:

- `_play()`: ejecuta la simulación.
- `_pause()`: detiene la evolución de la simulación.
- `_reset()`: detiene la simulación y la reinicia.
- `_update()`: actualiza las variables del modelo.

## 3 SISTEMA DE COMUNICACIÓN

### 3.1 COMPONENTES DEL SISTEMA

La clase colaborativa virtual se compone del *applet profesor* y de los *applets alumno*. El *applet profesor* es el encargado de gestionar la sesión y la evolución de la simulación. Posee una lista de los alumnos conectados a los que en cualquier momento puede desconectar. Además, es el único que puede interactuar sobre el LV y comunicar a los alumnos conectados el estado del mismo.

Otra opción importante que posee el sistema es la *asignación de la tiza*. Esta funcionalidad permite al profesor dar la posibilidad a un alumno de controlar la clase colaborativa virtual. Tan sólo son permisos

para ejercer el control sobre el LV, ya que la gestión de la sesión siempre la realiza el profesor.

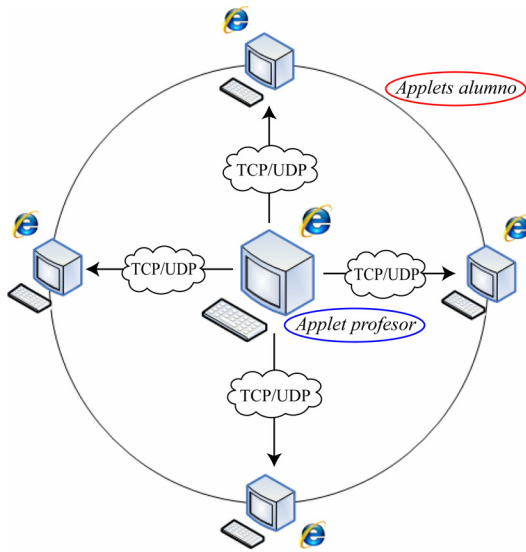


Figura 1: Componentes del sistema

### 3.2 ARQUITECTURA SOFTWARE DEL SISTEMA

Aunque con el sistema propuesto sólo se utiliza un applet para un determinado laboratorio virtual, éste tendrá un comportamiento diferente según actúe como applet del profesor o de un alumno. Así, dependiendo de la funcionalidad requerida, el applet empleará una arquitectura software diferente. En esta sección se muestra cómo se componen y cómo funcionan ambas partes del sistema.

El inicio de la comunicación se realiza el applet profesor cuando activa el modo colaborativo. A partir de este momento un *SocketServer* Java se abre un puerto determinado y comienza la escucha de peticiones de los alumnos (Figura 2). Este proceso de escucha es el responsable de la conexión y desconexión de las simulaciones de los alumnos en la clase virtual.

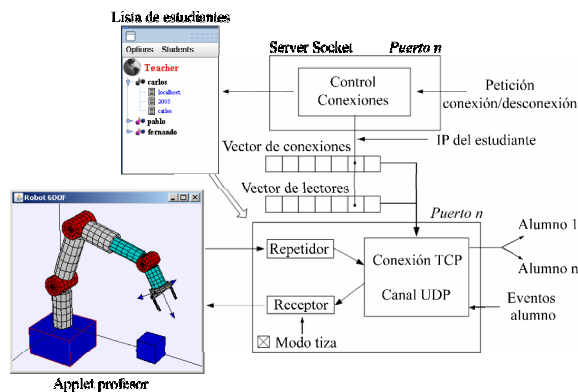


Figura 2: Arquitectura del applet profesor

Por cada petición aceptada, se crea un elemento más en un vector de conexiones, además de actualizar la lista de alumnos. Dicho elemento, contiene información referente a la procedencia del alumno (dirección IP pública) conectado a la clase virtual. El objeto *Repetidor* es el encargado de enviar los datos del estado actual de la simulación alumnos conectados a través de los protocolos TCP y UDP. Cada cambio o evento en el LV del profesor es comunicado al *Repetidor*, que se encarga de ponerlo en la red y hacerlo llegar a los clientes IP.

Además, para poder conocer el estado de cada una de applets alumno, se establece un *Lector* por cada conexión aceptada. Dicho objeto se encarga de comunicar al profesor si el alumno ha realizado correctamente el paso de simulación que le envió el *Repetidor*. Una vez que se han recibido todas las confirmaciones de los alumnos, el profesor puede continuar con la evolución del sistema. Así, se consigue la sincronización deseada entre los LVs.

En el caso de existir un alumno con la tiza asignada, el applet profesor actúa como si fuera un alumno, además de comunicarle al resto las órdenes que recibe de dicha simulación.

La arquitectura software del applet del alumno es mucho más sencilla que la del applet del profesor (Figura 3). Una vez establecido el canal de comunicación entre ambos, sólo existe un intercambio de datos profesor-alumno. Un proceso *Receptor* se ocupa de recibir las órdenes del profesor para controlar y actualizar la simulación. Y un *Repetidor*, comunica al *Lector* correspondiente del vector de lectores del profesor (ver Figura 2), el estado del applet alumno.

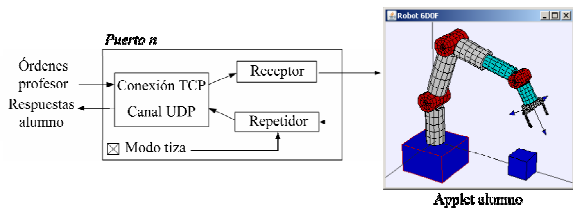


Figura 3: Arquitectura del applet alumno

Si el alumno tiene la tiza asignada, sus controles se habilitan para que pueda interactuar sobre el LV compartido y comportarse como si fuera el profesor de la clase virtual.

### 3.3 SINCRONIZACIÓN ON-LINE

El sistema de comunicación y sincronización entre LVs se ha incluido como una opción más del software Ejs en una versión experimental, de forma que se puede aplicar a todas las simulaciones que se desarrollen con este programa. Por lo tanto, después de desarrollar un LV con Ejs, y de instalar el applet

generado en un servidor web, es posible realizar una clase virtual a distancia.

Dado que, para un mismo grupo de trabajo colaborativo, se está trabajando con LVs idénticos, el modelo del sistema es el mismo para todos los componentes de la sesión virtual, facilitando la tarea de actualización de variables y refresco de la vista.

Por el canal de comunicación que se establece entre el profesor y el alumno mediante los protocolos TCP y UDP, se envía objetos Java que normalmente contienen una cadena de caracteres (*String*). Aquí, es necesario distinguir entre dos tipos de mensajes:

1. Los correspondientes al paso de simulación.
2. Los correspondientes a eventos como: `_pause()`, `_reset()`, `_update()`, etc.

El paso de simulación o *step* es la resolución del modelo del sistema en un diferencial de tiempo. Por tanto, para sincronizar las simulaciones, es necesario que todas se encuentren en el mismo *step*. Para ello, cuando el applet profesor comienza la ejecución de la simulación y da el primer paso de simulación, envía a través del repetidor la orden “step” para el resto de simulaciones. En este momento, para la ejecución y se pone en espera hasta que todas las simulaciones de los alumnos hayan realizado el *step* para continuar con la evolución del sistema. Dada la multitud de pasos de simulación que se generan cuando se está ejecutando una simulación, la sincronización del *step* se realiza a través de datagramas UDP. Esto es debido a que este protocolo, aunque sea menos fiable que el TCP, posee más velocidad de transmisión y así, la sincronización se realiza casi en tiempo real.

En el caso de los eventos, la comunicación se realiza a través del protocolo TCP, más fiable que el UDP. Ejs capta los eventos y el sistema de comunicación los envía a través de la red. En este caso no es necesario esperar a que los clientes verifiquen que han ejecutado la orden, ya que el protocolo TCP nos asegura su llegada.

## 4 EXPERIMENTACIÓN

En esta sección vamos a mostrar cómo se utiliza el sistema de comunicación síncrono. Para ello, utilizaremos LVs que simulan el control del depósito de un tanque mediante un controlador PID [11].

### 4.1 USO DEL SISTEMA

El proceso para poder crear una clase colaborativa virtual con LVs mediante el sistema que se presenta en este artículo, el profesor sólo debe seguir estos dos pasos:

1. Creación del applet profesor y su colocación en el servidor.
2. Generación de la dirección web, que contiene el applet alumno, para que los estudiantes puedan conectarse a la clase virtual.

El primer paso (1) se realiza desde el software Ejs. Una vez desarrollado el LV, tan sólo es necesario completar dos campos (Figura 4): dirección web del servidor donde están instalados los applets (*URL Server*) y el puerto que se usará para la comunicación (*Port*).

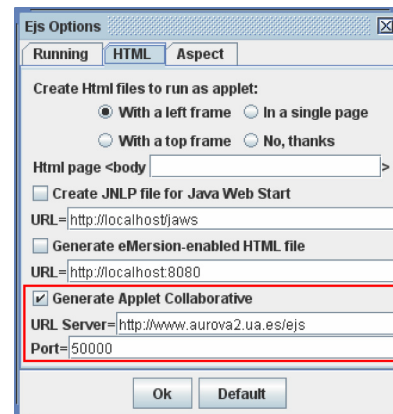


Figura 4: Opciones para generar el LV colaborativo

El applet generado por Ejs está listo para ser publicado en un servidor web desde donde será accesible por profesores y alumnos.

Seguidamente el profesor puede acceder al servidor web y ejecutar el applet. La primera acción que debe realizar es generar la URL para los estudiantes (2). Esta acción se realiza directamente desde el applet profesor, completando la interfaz que se muestra en la Figura 5 (Diálogo de generación). Una petición HTTP ordena la generación de la URL al servidor público donde están instalados los applets. Una página PHP, crea la dirección web que se devuelve como respuesta a dicha petición.

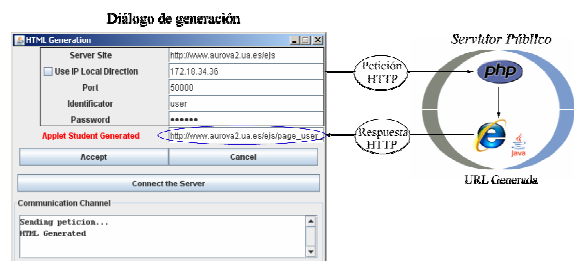


Figura 5: Generación de la URL para los alumnos

Finalmente, el profesor puede difundir la dirección web generada a los alumnos del grupo, a través de un chat o email, para que se puedan a dicha dirección y

obtener el applet de simulación con el comportamiento colaborativo de un applet de alumno.

Un alumno puede utilizar su applet individualmente sin restricciones, hasta que activa la opción de “Clase Virtual”, la cual conecta con el applet del profesor e inicia el modo colaborativo. Al finalizar el modo colaborativo, el applet vuelve a tener un funcionamiento independiente.

## 4.2 RESULTADOS

En la Figuras 6 y 7 podemos observar la sincronización realizada entre LVs. En las imágenes puede verse cómo ambas simulaciones (applet profesor y applet alumno) se encuentra en el mismo estado de simulación y poseen los mismos valores en el sistema físico.

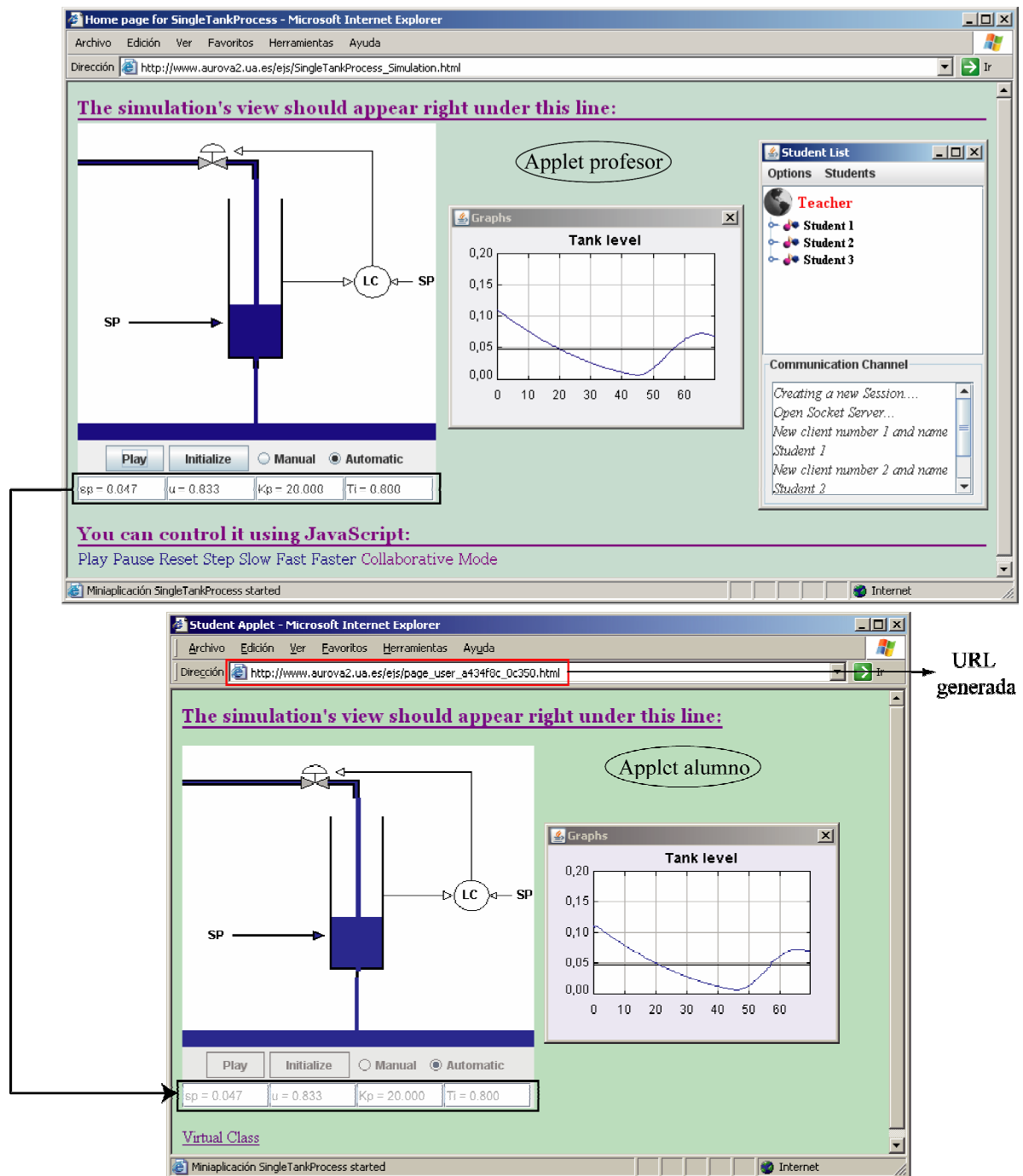


Figura 6. Sincronización de applets (I)

El sistema colaborativo se combinó con *Skype*, un software que permite la comunicación en tiempo real de audio y video (Figura 7). El resultado de esta combinación resultó ser muy satisfactorio, con lo que el sistema de colaboración presentado es una herramienta muy útil para los CSCLE

## 5 CONCLUSIONES

La combinación de los LVs y los EAC síncronos es, hoy en día, una alternativa a considerar para la educación a distancia. Los LVs son herramientas de auto-aprendizaje que ayudan a comprender mejor los

conceptos a los alumnos. Y los EAC síncronos permiten compartir ideas y experiencias entre el profesor y los alumnos en tiempo real.

Muchos entornos colaborativos existentes en la actualidad incluyen LVs. Sin embargo, muy pocos de ellos son de carácter síncrono.

Teniendo en cuenta lo anterior, y que la colaboración puede acortar el tiempo necesario para el aprendizaje sobre experiencias prácticas con VL con respecto a herramientas asíncronas, los autores han desarrollado el sistema de comunicación *on-line* entre VLs desarrollados con Ejs que se presenta en este artículo.

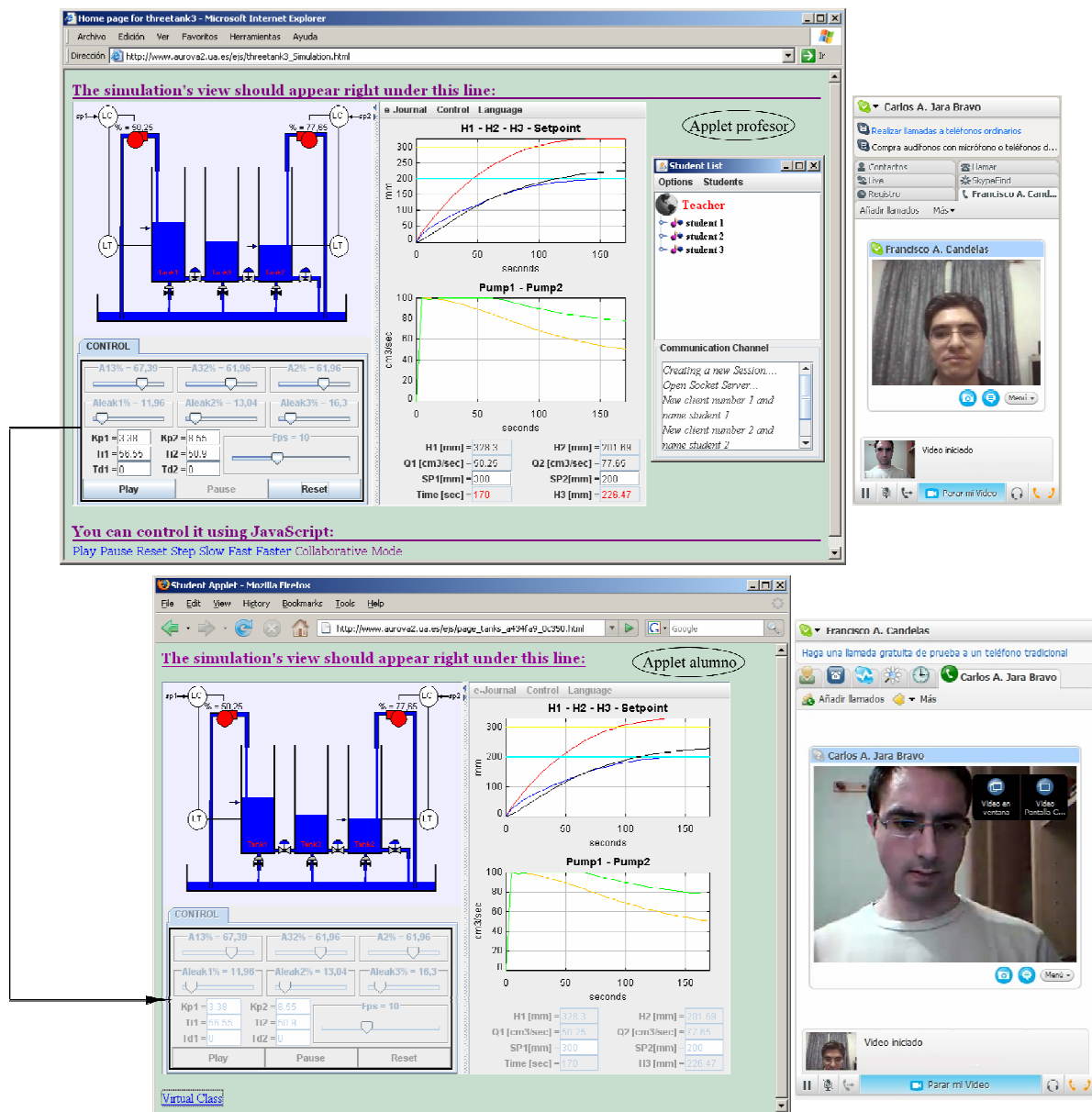


Figura 7: Sincronización de applets (II)

El sistema se ha incluido como una opción más del software Ejs, lo que permite que cualquier LV creado con esta herramienta, pueda utilizarse para realizar una clase virtual a distancia, de forma que los alumnos y el profesor puedan colaborar a través de Internet en tiempo real para resolver el experimento planteado. En cualquier caso, los applets mantienen su funcionamiento independiente cuando no se ha iniciado el modo colaborativo.

### Agradecimientos

El trabajo presentado en este artículo está financiado por el Ministerio de Educación y Ciencia a través del programa de becas FPI (BES-2006-12856, Carlos A. Jara Bravo) y el proyecto de investigación DPI2005-06222. Los autores quieren agradecer esta financiación.

### Referencias

- [1] Abler, R. T. y Wells, I. G., (2005). Distributed Engineering Education: Evolution of the Telecollaboration Stations for Individualized Distance Learning. *Education, IEEE Transactions on*, vol. 48, pp 490-496.
- [2] Candelas, F. A. y Moreno, J. S., (2005). Recursos didácticos basados en Internet para el apoyo a la enseñanza de materias del área de Ingeniería de Sistemas y Automática. *Revista Iberoamericana de Automática e Informática Industrial*, vol. 2, pp 93-99.
- [3] Candelas, F. A., Puente, S. T., Torres, F., Ortiz, F. G., Gil, P. y Pomares, J., (2003). A virtual laboratory for teaching robotics. *International Journal of Engineering Education*, vol. 19, pp 363-370.
- [4] Candelas, F. A., Torres, F., Ortiz, F. G., Gil, P., Pomares, J. y Puente, S. T., (2003). Teaching and Learning Robotics with Internet Teleoperation. *Second International Conference on Multimedia and Information & - Communication Technologies in Education (m-ICTE)*. Badajoz.
- [5] Esquembre, F., (2004). Easy Java Simulations: a software tool to create scientific simulations in Java. *Computer Physics Communications*, vol. 156, pp 199-204.
- [6] Gillet, D., Anh Vu Nguyen, N. y Rekik, Y., (2005). Collaborative web-based experimentation in flexible engineering education. *Education, IEEE Transactions on*, vol. 48, pp 696-704.
- [7] Jiménez, L. M., Puerto, R., Reinoso, O., Fernández, C. y Ñeco, R., (2005). Recolab: Laboratorio remoto de control utilizando Matlab y Simulink. *Revista Iberoamericana de Automática e Informática Industrial*, vol. 2, pp 64-72.
- [8] Kreutz, R., (2000). NetChat: Communication and Collaboration via WWW. *Journal of Educational Technology and Society*, vol. 3.
- [9] Martín, C., Dormido, S., Pastor, R. y Sánchez, J., (2003). Sistema de levitación magnética: un laboratorio virtual en Easy Java Simulations. *XXIV Jornadas de Automática*.
- [10] Snow, C., Pullen, J. M. y McAndrews, P., (2005). Network EducationWare: an open-source web-based system for synchronous distance education. *Education, IEEE Transactions on*, vol. 48, pp 705-712.
- [11] Vargas, H., Sanchez, J., Farias, G., Dormido, S., Dormido, R., Canto, S. y Esquembre, F., (2006). Web-based learning resources for automation technicians vocational training: illustrated with a heat-flow and liquid level laboratories, *IFAC Symposium on Advances Control Education*, pp 110-116.