

ANÁLISIS DE LABORATORIOS VIRTUALES Y SU COMUNICACIÓN CON PLATAFORMAS EDUCATIVAS APLICADO A UN LABORATORIO DE SIMULACIÓN DE MICROPROCESADORES POR INTERNET

E. LÓPEZ-ALDEA¹, E. SAN CRISTOBAL², G. DÍAZ², S. MARTÍN²
M. CASTRO², J. PEIRE² Y P. LÓPEZ-ALDEA³

¹NIEDAX. España.

²Departamento de Ingeniería Eléctrica, Electrónica y de Control. Escuela Técnica Superior de Ingenieros Industriales. Universidad Nacional de Educación a Distancia. España.

³INDRA. España.

El presente documento describe una de las líneas de investigación del Departamento de Ingeniería Eléctrica, Electrónica y de Control de la Escuela de Ingenieros Industriales de la UNED, orientada a la formación virtual para el apoyo de las prácticas presenciales mediante laboratorios virtuales y remotos integrados en las diferentes plataformas educativas existentes que cumplen estándares o en servidores intermedios que ofrecen servicios a estos laboratorios, mediante la creación de una interfaz de comunicación en conjunto. Se exponen las directrices generales para la creación un laboratorio de simulación de microprocesadores adaptado a este marco de investigación, así como las experiencias obtenidas por este Departamento en materia de educación virtual en este campo.

1. Introducción

En la actualidad, (2008), el Departamento de Ingeniería Eléctrica, Electrónica y de Control de la Escuela Técnica Superior de Ingenieros Industriales de la UNED está estudiando y diseñando laboratorios virtuales y remotos para su utilización como prácticas docentes y a su vez como apoyo a las prácticas presenciales. La Universidad, que apuesta cada vez más por el aprovechamiento de las nuevas tecnologías aplicadas a la Educación a Distancia, propone que las prácticas de laboratorio se puedan complementar utilizando los recursos de Internet como corresponde a este momento de expansión y omnipresencia computacional.

2. Comunicación entre laboratorios virtuales y remotos, plataformas educativas (LMS) y servidores de servicios para laboratorios (*iLabs*)

Con la idea de aprovechar todo el potencial de una plataforma educativa [1] [2] (LMS – *Learning Management System*) o los servicios que ofrece un servidor intermedio de laboratorios virtuales [3] (*iLab Server Broker* mediante una arquitectura compartida *iLab*), se está diseñando un espacio virtual donde la palabra “reutilizable” tome su máximo sentido. A la hora de elaborar un laboratorio virtual que plantea un objetivo docente, los diferentes recursos que disponen las plataformas educativas se hacen necesarios si se desea optimizar las posibilidades educacionales que brinda la tecnología en Internet. Ciertas plataformas educativas que siguen los estándares de aprendizaje (como ejemplo *Scorm*, *Ims* [4] [5]) disponen de unos servicios de autenticación, generación de *test* y evaluación, foros, *chats* y repositorios entre otros que pueden ser aprovechados para utilizarse en un laboratorio virtual sin tener que reprogramarlos en cada uno de ellos permitiendo así la interactividad entre los diversos laboratorios a nivel de gestión de usuarios. De la misma forma, se está investigando en la integración de los servicios de los servidores intermedios de

arquitectura compartida para laboratorios virtuales (*iLabs*) extrayendo los elementos que son comunes a todos ellos. De esta forma se pretende minimizar el desarrollo y esfuerzo de gestión por parte de los usuarios y desarrolladores de laboratorios remotos, permitir a múltiples universidades con diferentes redes compartir acceso utilizando así el material de cualquier laboratorio en cualquier parte del mundo y proveer de un *set* de servicios y herramientas de desarrollo. Se comprende las posibilidades magnánimas de esta forma de trabajar y por ello, el desempeño actual de desarrollo en este área por parte del Departamento.

Con la idea de realizar esta múltiple comunicación que permita tener toda la flexibilidad a la hora de desarrollar y trabajar con un laboratorio virtual, se define una Interfaz de comunicación (*middleware*) para que el desarrollador de laboratorios no tenga que preocuparse del uso de las diferentes plataformas para su servicio. Se crea así una capa de software intermedia entre los recursos que dispone la plataforma y el laboratorio virtual. Así mismo, otro objetivo del *middleware* es integrar los servicios que ya ofrece un servidor intermedio de servicios en los laboratorios (*iLabs*). De esta forma se podrá utilizar el laboratorio con los servicios que se le desee integrar de forma directa o utilizar la capa de *software* intermedia o *middleware* para utilizar cualquier entidad (los servicios del servidor intermedio o de los propios de las plataformas educativas por Internet) (Fig. 1). Todas las configuraciones son posibles para la creación de un entorno educativo por Internet donde el desarrollador de laboratorios virtuales y remotos pueda utilizar para su conveniencia.

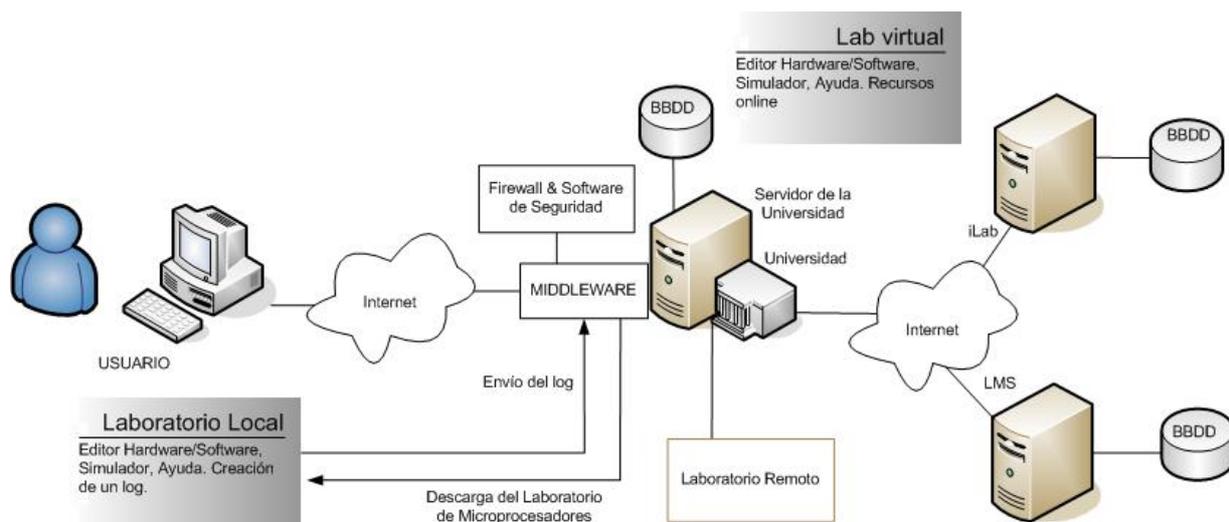


Figura 1. Relación entre laboratorio virtual, remoto, plataformas educativas e *iLab* mediante una interfaz de adaptación o *middleware* orientado a un Laboratorio Virtual y Remoto de Microprocesadores.

El laboratorio virtual se comunicará con la Interfaz creada a tal efecto para el aprovechamiento de los recursos completos de una plataforma educativa dejando en la aplicación del laboratorio virtual las tareas que sólo le competen a éste. Todas las tareas de gestión de usuarios y perfiles, autenticación, generación de *test* y evaluación, así como, todos los aspectos relacionados con el análisis de evolución del alumnado, foros, *chats*, repositorios, etc. pueden quedar delegados a la plataforma educativa que utilice los estándares de comunicación y que disponga de esos servicios. Esta forma de trabajar también supone poder evaluar desde la plataforma aspectos más globales que implique relacionar un contexto con otras áreas de estudio e incluso relación entre los trabajos realizados en los diferentes laboratorios. Éste, por tanto, sólo tendrá que ocuparse de presentar los contenidos técnicos y elaborar las simulaciones pertinentes haciendo hincapié en el aprendizaje de las diferentes materias.

3. Estudio inicial de un Laboratorio virtual de Microprocesadores

Uno de los laboratorios que se están diseñando en el Departamento consiste en un laboratorio virtual y remoto de Microprocesadores para la realización de prácticas docentes y la mejora de la calidad de la enseñanza en este área. Tal y como se ha indicado, se utilizará a través de una Interfaz diseñada debidamente, los servicios de una plataforma educativa o servidor intermedio de servicios que se definan para su reutilización. No obstante, el laboratorio podrá tener existencia propia sin integrar en ninguna plataforma dotándole de los elementos necesarios para la gestión de los estudiantes. Por tanto y para facilitar el desarrollo, el análisis se va a dividir en dos partes: por un lado los servicios de las plataformas y por otro lo que entendemos como taller de simulación de microprocesadores propiamente dicho. La utilización correcta y bien relacionada de ambos, creará un laboratorio virtual donde poder utilizar todos los recursos de estudio, prácticas, simulación y evaluación de forma automática y prácticamente sin la presencia de un tutor constante.

Dadas las sugerencias de los estudiantes, que se comentan en detalle más adelante, se pretende que el usuario pueda trabajar tanto conectado a Internet como sin conexión para facilitar el estudio en cualquier sitio y obviando los inconvenientes que pudiera tener la necesidad de trabajar con la dependencia de una conexión. El laboratorio, cuando sea utilizado por el estudiante, creará en local un archivo de propiedades (*properties*) con toda la información necesaria para su identificación y su evaluación, así como un *log* errores de la aplicación. Posteriormente, si se desea, se descargará en la plataforma o gestor del laboratorio cuando se realice una conexión. Por tanto, se puede trabajar de forma asíncrona con todo el potencial de la conexión a Internet y las ventajas de no depender de ésta.

A continuación se detallan los puntos principales que debe disponer un laboratorio de simulación de microprocesadores extraído de los conocimientos de los profesionales docentes y de las necesidades que los estudiantes han expuesto en diversos trabajos y cuestionarios que se han elaborado con ellos.

En primer lugar se definen los servicios que debe ofrecer el laboratorio tanto en modo local como en modo conexión.

Servicios del laboratorio en modo local

En caso de desear trabajar con el laboratorio a modo local, éste dispondrá de los siguientes servicios:

- Editor/Depurador
- Compilador
- Simulador
- Configuración Hardware
- Interfaz de usuario
- Generación de informes (hoja de cálculo, *xml*, pdf o archivo de texto plano)
- Generación de archivo de propiedades y *logs* de errores
- Ayuda *offline*

La generación de un archivo de propiedades y un *log* de errores consiste en un fichero *xml* donde documentar toda la actividad del usuario y los errores que se hayan podido producir durante la ejecución de la aplicación, permitiendo tener un control de posibles *bugs*, respectivamente.

El archivo de propiedades incluirá la siguiente información:

- El usuario que ha trabajado en el laboratorio
- El tiempo que ha dedicado
- Las actividades que ha realizado (en qué pantallas ha trabajado y cuantas veces ha utilizado cada recurso, por ejemplo, la ayuda).
- El nombre de los programas que ha creado o editado.

Servicios del laboratorio en modo conexión

En el caso de trabajar con el laboratorio conectado a Internet, los servicios que debe tener el usuario son los mismos que los que se indican en modo local añadiendo:

- Ayuda dinámica
- Herramientas colaborativas
- Gestión de la educación (integrando la gestión automática de resultados)

La parte que realmente se diferencia con respecto a la parte *offline* está en las herramientas colaborativas, en la ayuda más dinámica y en los elementos de la gestión educativa, como son los servicios de autenticación y evaluación incluso a través de *iLabs* o LMS.

Los servicios que dispondrá el laboratorio y que serán utilizados fundamentalmente desde el *middleware* son:

- Autenticación
- Chat/Clases dinámicas
- Foro de Microprocesadores
- Gestión de calendario
- Repositorio de archivos
- Generación y realización de Test

Los esquemas funcionales del laboratorio tanto en modo local como en modo conexión se muestran en la Figura 2.

El hilo normal de actuación y el uso más frecuente por el usuario en cualquier simulador de microprocesadores es la creación o edición de un programa en ensamblador, compilación, simulación y visualización por pantalla de los resultados en los registros, memoria o periféricos para su evaluación final de los resultados obtenidos. Este será el eje central del laboratorio independientemente del modo de trabajo o de la forma de acceso (Fig. 3).

Etapas generales del Simulador

Centrándonos en las etapas básicas que conforman el simulador de microprocesadores, éstas son:

- Editor
- Compilador
- Simulador
- Interfaz de usuario

El Editor generará un archivo *xml* según unos datos de entrada. El editor puede recibir datos de un repositorio local que disponga el usuario con sus propios programas, descargarse ejemplos del servidor en caso de que esté conectado al laboratorio a través de Internet o escribir el programa en ensamblador. Para el último caso, el editor dispondrá de una herramienta de ayuda a la escritura a través del Explorador de

Internet (aunque no esté conectado). Dispondrá de las opciones para guardar en el Laboratorio (en el repositorio del servidor), guardar en local y las ayudas a la escritura del código en ensamblador. Antes de generar el *xml*, el editor realizará el análisis sintáctico para la depuración de posibles errores donde el este deberá indicar qué tipo de error y su localización en el programa.

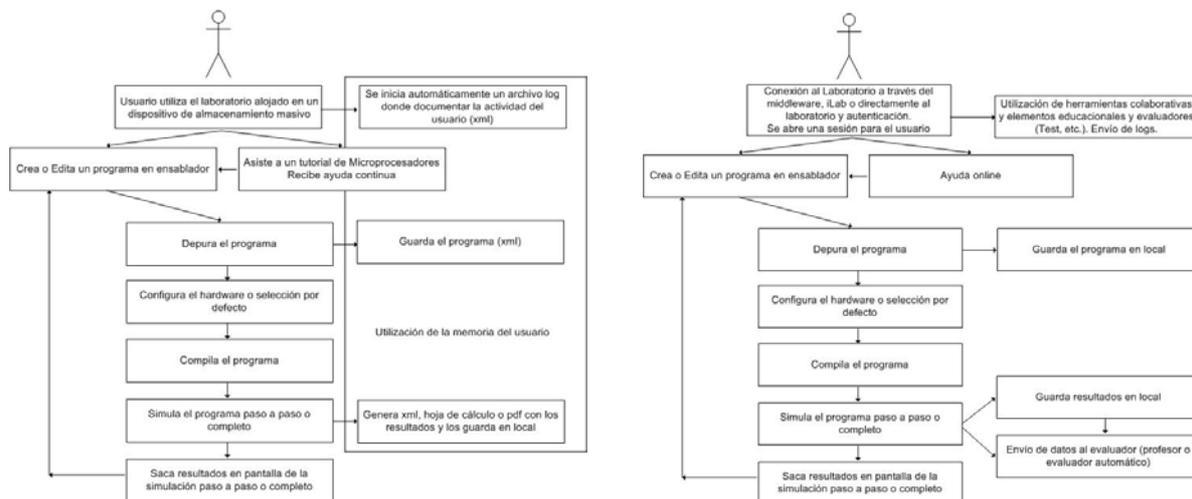


Figura 2. Esquemas funcionales del laboratorio virtual en modo local y en modo conexión respectivamente.

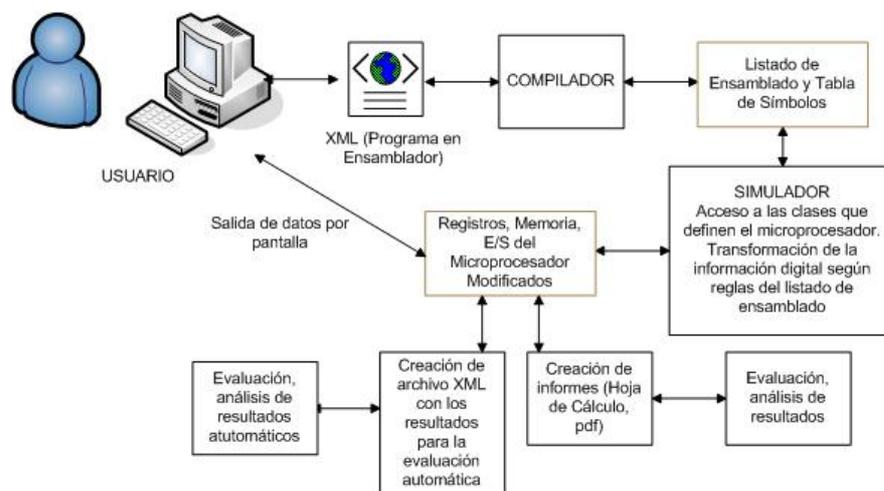


Figura 3. Etapas del programa en ensamblador en el laboratorio virtual dentro del marco de investigación de integración en plataformas educativas.

El Compilador tendrá como entrada el archivo *xml* generado en el editor y creará un listado de ensamblado y su tabla de símbolos a la salida, que a su vez será otro archivo *xml*. Éste contiene toda la información que un microprocesador necesita para ejecutar un programa.

El Simulador leerá los datos de la tabla que contiene el listado de ensamblado y la tabla de símbolos para generar los cambios de datos en los registros y memoria del microprocesador seleccionado tras su configuración hardware. Por tanto, debe:

- Leer el listado de ensamblado y la tabla de símbolos creadas por el compilador y generar los cambios de los datos en registros, memoria y E/S.
- Editar la configuración del Hardware en caso de desear otra configuración diferente a la establecida por defecto (esta operación deberá ser realizada antes de su simulación).
- Generar reportes en una hoja de cálculo o en formato *pdf*.
- Generar un fichero *xml* con los resultados de la simulación para su posible corrección automática a través de la comparación con modelos preestablecidos.

La Interfaz de usuario sacará por pantalla todos los elementos que sean seleccionados por el usuario: registros, memoria, ALU, UCP, camino de datos, a diferentes niveles de abstracción.

En la Figura 4 se resume toda la actividad del laboratorio integrado o sin integrar en los servicios que ofrecen las plataformas.

Por tanto, una vez definidos los elementos que entran en juego, se puede esquematizar las relaciones y las entidades que participarán en la elaboración del laboratorio virtual.

Por otro lado, se está haciendo el estudio en paralelo de la forma de ofrecer en pantalla un Laboratorio de Microprocesadores, es decir, la Interfaz Gráfica de Usuario (IGU). Para ello, se está definiendo una librería de clases en java [6] [7] [8] [9] que permitan organizar la interfaz gráfica de cualquier microprocesador a diferentes niveles *hardware* (Fig. 5). De esta forma, la creación de un simulador de un microprocesador orientado a Internet para la inclusión futura en el laboratorio implicará poder utilizar esta librería para no tener que reprogramar todos los aspectos de imagen que conlleva la Interfaz Gráfica. Incluso, si se desea mejorar el aspecto en un futuro de los simuladores, sólo habría que retocar la librería. De nuevo se trata de reutilizar todos los aspectos de programación que resultan repetitivos en una y otra aplicación de necesidades comunes. Sin embargo, ello implica inicialmente definir bien cuales van a ser los parámetros constantes que van a disponer todos los simuladores orientados a la docencia para llevar a cabo una verdadera creación versátil de este recurso.

Para realizar este esfuerzo con fundamento y tratar de mejorar realmente la calidad de la enseñanza de microprocesadores mediante el estudio de la forma de enseñanza actual y proponiendo mejoras en la presentación de la información, se está realizando el estudio de cursos orientados a tareas y basados en Web que ya se han llevado a cabo y que en la actualidad se siguen realizando y mejorando. El departamento trabaja en varios proyectos de investigación, entre ellos el proyecto DIPSEIL [10] dentro del programa Sócrates. DIPSEIL, actualmente, es una plataforma educativa en Internet que trabaja en la gestión de cursos orientados a tareas (Fig 6). Se han desarrollado varios cursos siguiendo este modelo y entre ellos el curso Arquitectura y Simulación de Microprocesadores con la idea de introducir formación complementaria a las prácticas presenciales.

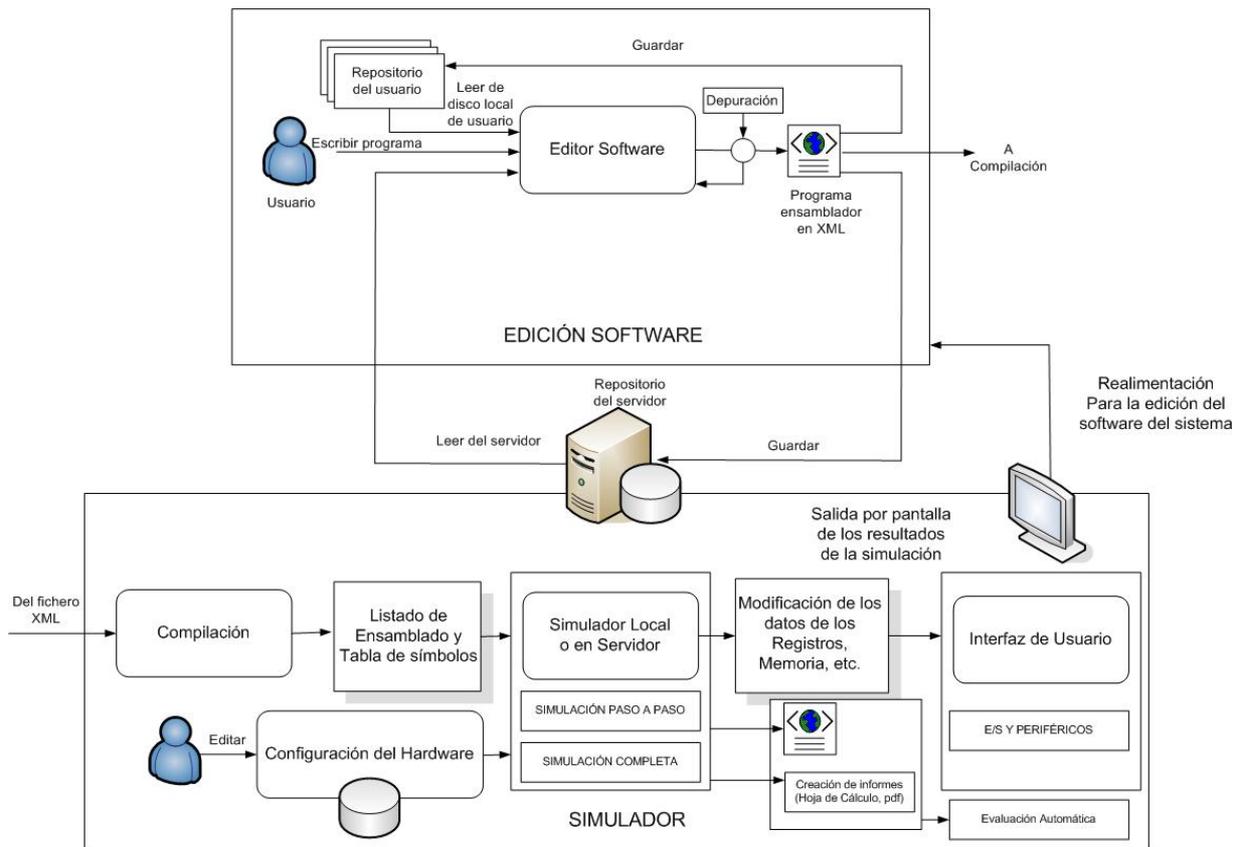


Figura 4. Esquema funcional global del Laboratorio de Microprocesadores.

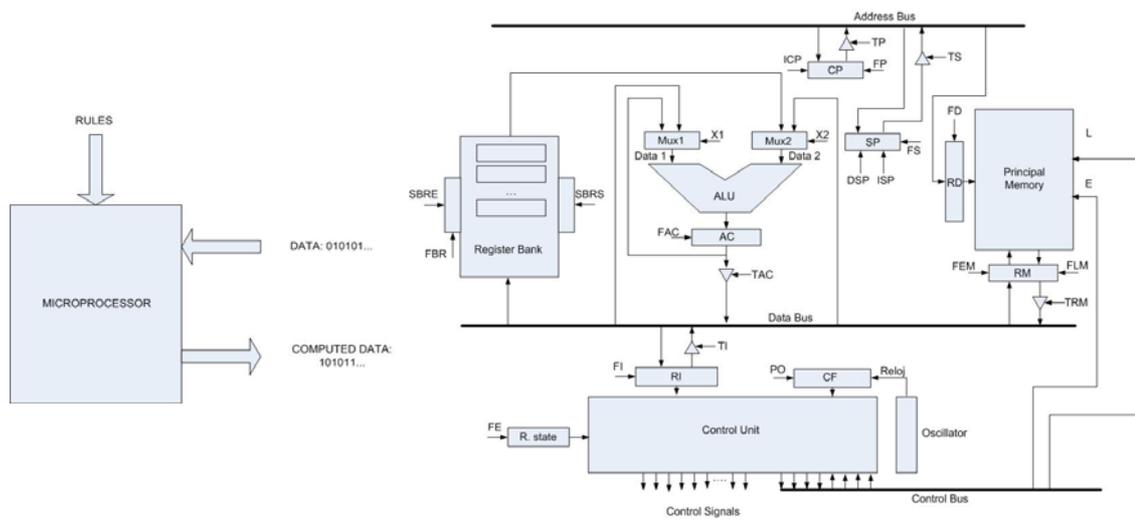


Figura 5. Ejemplo de la Interfaz de un microprocesador a diferentes niveles hardware.

4. Análisis de la elaboración del laboratorio virtual de Microprocesadores

Dentro de los planes para la estipulación de los requisitos necesarios para el establecimiento del laboratorio virtual de Microprocesadores que plantea el departamento, se trata de analizar las necesidades de los alumnos desde su propia perspectiva. Para ello, en la actualidad, el Departamento que dispone de un curso *online* de Microprocesadores desarrollado en la plataforma DIPSEIL con doce tareas organizadas cubriendo todas las partes de un curso de arquitectura y organización de microprocesadores integra un cuestionario general sobre la utilización del simulador. Este se ha diseñado para tener conocimiento de la satisfacción de la utilización de un simulador ya creado (Easy68k [11] Fig 7) y el análisis de las necesidades docentes en la creación del futuro laboratorio virtual de microprocesadores. En la Tabla 1 se expone los datos generales de los alumnos que realizaron este cuestionario y a continuación se relata algunas de las conclusiones obtenidas tras realizar el curso.



Figura 6: Página principal de DIPSEIL.

Algunas de las conclusiones sacadas de los cuestionarios del curso de microprocesadores se resumen a continuación (Fig 8). El taller de Simulación deberá disponer de los siguientes servicios:

- *Editor de software*, donde el usuario pueda editar los programas en ensamblador que utilizará para simular en el microprocesador correspondiente, así como cargar un programa en la memoria.
- *Editor de hardware*, donde se podrá realizar el diseño del mapa de memoria y recursos hardware.
- *Área de Simulación*, donde se expondrán los diferentes estados de los registros y memoria. También se podrá visualizar diferentes periféricos de salida como pueden ser *Leds*, teclados, *Displays*, etc. En esta área se podrá representar tanto los cronogramas que se generen como el microprocesador en bloques que se interrelacionan y comunican según el programa cargado en la memoria.
- *Área de evaluación*, en este espacio se creará los elementos necesarios para la evaluación del usuario. Se pretende que poder disponer de *test* de evaluación y autoevaluación, así como de la evaluación

automática de resultados del simulador por comparación con patrones. Esta área depende de los servicios que puedan usarse en la plataforma educativa.

- *Gestor de ayuda*, donde por un lado se diseñarán unos materiales básicos de enseñanza de microprocesadores y por otro lado se diseñará una herramienta de ayuda que el tutor pueda manejar para exponer los recursos necesarios para que el usuario reciba la ayuda adecuada en cada práctica, microprocesador o tarea.

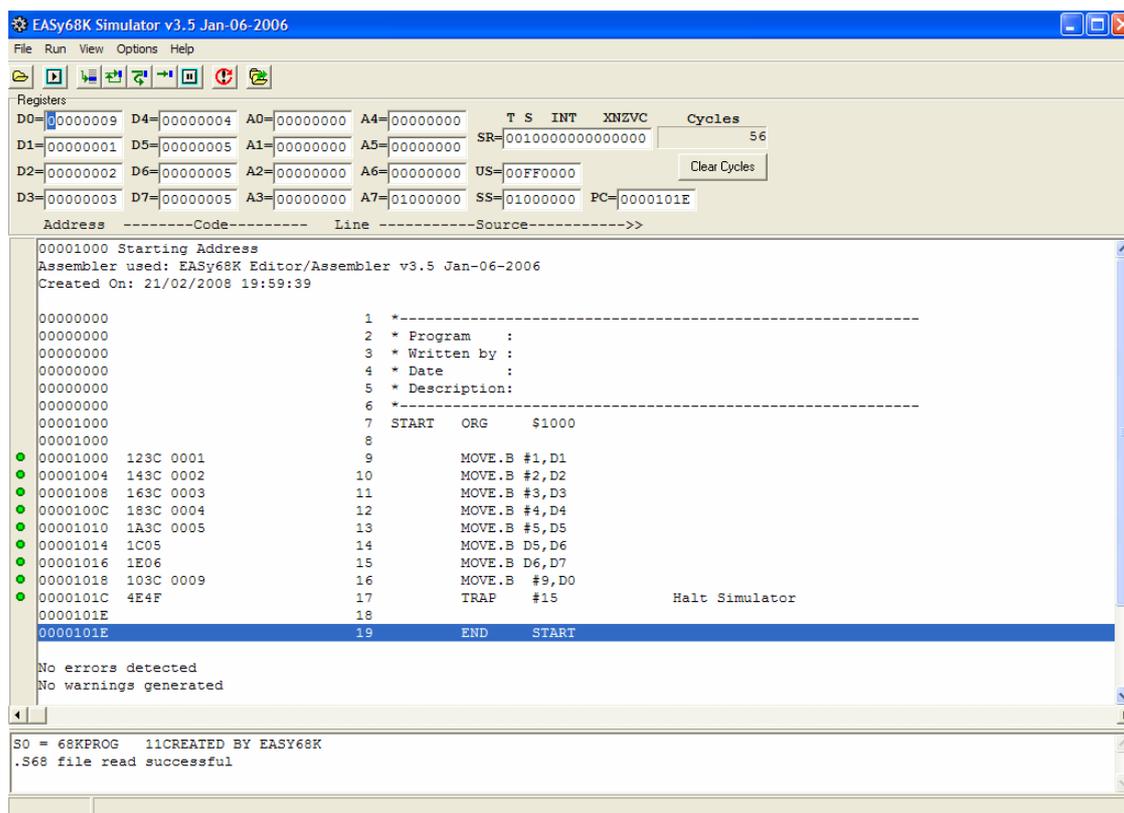


Figura 7: Interfaz gráfica del simulador EASY68k para el Microprocesador MC68000 utilizado en el curso de Arquitectura de Microprocesadores (DIPSEIL).

Tabla 1. Datos globales de la encuesta sobre la simulación del curso Arquitectura y Simulación de Microprocesadores desarrollado en la plataforma DIPSEIL.

Estudiantes que hicieron el curso de Arquitectura y simulación de Microprocesadores en el proyecto DIPSEIL y presentaron el trabajo final.	15
Estudiantes que realizaron el cuestionario de simulación de Microprocesadores.	10
Tiempo medio utilizado en la realización del cuestionario de Microprocesadores.	26,4 minutos
Tipos de Microprocesadores utilizados en el curso.	MC68000
Simulador utilizado en el curso.	Easy68k v3.5 y v3.6 Beta
Estudiantes que prefieren trabajar <i>off-line</i>	6
Estudiantes que prefieren trabajar <i>on-line</i>	4



Figura 8. Servicios del taller de simulación.

En general, a todos les pareció útil la herramienta para el aprendizaje de la arquitectura de computadores. El microprocesador elegido fue el popular MC68000 de 16 bits que dispone de una arquitectura muy completa para el aprendizaje pero lo suficientemente sencilla para no complicar el curso sin obtener ningún beneficio mayor.

En la Tabla 2, se muestran los datos por cada estudiante y los comentarios sobre los aspectos más relevantes del simulador utilizado, así como, se han añadido las diferentes sugerencias que los estudiantes proponen para mejorar el mismo.

Un aspecto importante a tener en cuenta es que los estudiantes suelen preferir trabajar con una herramienta *off-line* en lugar de la utilización de un simulador *on-line*. Según describen, de esta manera tienen la posibilidad de trabajar donde quieran que estén sin depender de una conexión a Internet. También son conscientes de que esto conlleva no poder utilizar las ventajas propias de la utilización de una herramienta *on-line* donde podrían tener acceso a recursos dinámicos como *chats* y foros. Sin embargo, cada vez es más extendida la conexión a Internet, en cualquier lugar con cobertura, por tanto, este problema es más irrelevante a medida que pasa el tiempo.

Como objetivo general se pretende desarrollar una aplicación software (Laboratorio Virtual) orientada a Internet para la Simulación de Microprocesadores, así como evaluar si finalmente la herramienta mejora la calidad de la enseñanza. Ésta se realizará desde un punto de vista educativo y no empresarial por lo que el diseño partirá desde esta premisa. Por otro lado, se pretende también como objetivo desarrollar y mejorar el estudio de la Simulación de Microprocesadores en general mediante la aplicación de Nuevas Tecnologías.

El laboratorio se creará mediante la implementación de algunos microprocesadores como el MC68000 de 16 bits y el Intel 8086, sin embargo, se pretende que el Laboratorio sea abierto a la implementación de futuros Microprocesadores y elementos hardware o para el desarrollo de otros pasados y que se comunique con los recursos de una plataforma educativa para fines docentes.

Tabla 2. Datos de la encuesta por estudiante.

Nombre	Tiempo (m-minutos)	Prefiere on-line / off-line	Comentarios
Estudiante 1	20 m	Off-line	Facilidad de uso. Poder ver directamente los registros y bits de estado
Estudiante 2	15 m	Off-line / On-line	Posibilidad de ejecutar paso a paso las instrucciones. Propone una interfaz multidocumento.
Estudiante 3	6 m	Off-line	Permite ver en todo momento los valores de los registros y el estado de la memoria y de la pila
Estudiante 4	30 m	On-line / Off-line	Sugiere que el Editor/Compilador y el Simulador estén en una aplicación Considera potente el Editor aunque sugiere diferentes colores según las instrucciones. Sugiere un depurador en tiempo real.
Estudiante 5	20 m	Off-line	Considera útil poder ejecutar el programa línea a línea y revisar los registros de estado, así como la ventana de entrada/salida. Sugiere mejorar el depurador indicando exactamente dónde está el fallo. Al ejecutar paso a paso sugiere que se iluminen los diferentes bits de memoria afectados. Considera útil trabajar con varias ventanas.
Estudiante 6	20 m	Off-line	Sugiere una ayuda en español y que se pudieran ver cronogramas de los procesos.
Estudiante 7	35 m 47 segundos	On-line/ Off-line	Herramienta de fácil manejo. Sugiere que funcione en cualquier Sistema Operativo. Incorporación de un diccionario de instrucciones en tiempo real para ayuda en la Edición de un programa. (Ver el texto de la Sugerencia general de un Simulador)
Estudiante 8	15 m	Off-line	Ha considerado útil la posibilidad de ejecutar instrucción a instrucción el código y analizar el estado de los registros y memoria. La interface gráfica la ve muy intuitiva. Sugiere que se indique el tipo de error cometido en la depuración, no sólo la localización.
Estudiante 9	45 m	On-line/ Off-line	Considera útil la información de los diferentes registros y posiciones de memoria junto con la posibilidad de ejecutar el programa instrucción a instrucción. En las prácticas no ha necesitado vistas Hardware ni Break Points pero las considera útiles
Estudiante 10	57 m	Off-line	Ha sido el único en usar EASy68k Editor/Assembler v 3.6 beta. La considera muy intuitiva de manejar. Sugiere disponer de ayuda en castellano. No le gusta la presentación de la memoria.

5. Conclusión

Las novedades tecnológicas que permiten avanzar en las diferentes disciplinas educativas llevan un trabajo paralelo de personalización por un lado y reutilización por otro.

La formación docente masiva y a distancia se debe apoyar en estas tecnologías si quieren avanzar y ofrecer una garantía de enseñanza acorde a la situación actual de conocimientos tecnológicos. La investigación que aquí se propone muestra un camino de universalización que implica aprovechar todos los recursos creados para su utilización personal sin el esfuerzo añadido de volver a crear lo ya existente.

Agradecimientos

Los autores quieren agradecer al Ministerio de Ciencia y Educación de España y al Plan Nacional Español de I+D+I 2004-2007 el apoyo en el proyecto TSI2005-08225-C07-03 "MOSAICLearning: Aprendizaje móvil y electrónico, de código abierto, basado en estándares, seguro, contextual, personalizado y colaborativo".

Referencias

- [1] dotLRN. Learn, Research, Network. <http://dotlrn.org/>. URL con último acceso el 21/02/2008.
- [2] Moodle. <http://moodle.org/>. URL con último acceso el 21/02/2008.
- [3] iLab. Internet Access to Real Lab (MIT). <http://icampus.mit.edu/ilabs/>. URL con ultimo acceso el 21/02/2008.
- [4] IMS. <http://www.imsglobal.org/question/>. URL con ultimo acceso el 21/02/2008.
- [5] Adlnet. <http://www.adlnet.gov/scorm/>. URL con ultimo acceso el 21/02/2008.
- [6] F. J. Ceballos. *Java™ 2. Interfaces gráficas y aplicaciones para Internet*. Ed. Ra-Ma (2006).
- [7] P. M. Anasagasti. *Fundamentos de los Computadores*. Ed. Thomson. (1998).
- [8] M. Castro y otros. *Estructura y Tecnología de Computadores I*. Ed. UNED (2002).
- [9] A. Froufe. *Java Server Pages*. Ed. Ra-Ma (2002).
- [10] Distributed Internet-based Performance Support Environment for Individualized Learning (DIPSEIL). <http://www.dipseil.net/>. URL con último acceso el 21/02/2008.
- [11] Easy68k. <http://www.monroeccc.edu/ckelly/EASy68K.htm>. URL con último acceso el 21/02/2008.